

9.4

IBM MQ 관리

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [539 페이지의 『주의사항』](#)에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ의 버전 9릴리스 4 및 모든 후속 릴리스와 수정에 적용됩니다.

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

관리	7
IBM MQ 큐 관리자 및 연관된 자원을 관리하는 방법.....	8
제어 명령을 사용하여 IBM MQ for Multiplatforms 관리.....	10
MQSC 명령을 사용하여 IBM MQ 관리.....	11
MQSC 명령 구문.....	12
MQSC 입력 파일 구문.....	14
runmqsc 에서 대화식으로 MQSC 명령 실행.....	16
runmqsc 의 텍스트 파일에서 MQSC 명령 실행.....	21
시동 시 MQSC 스크립트에서 자동 구성.....	22
PCF 명령을 사용하여 IBM MQ 관리 자동화.....	23
IBM MQ PCF(Programmable Command Format) 소개.....	24
MQAI를 사용하여 PCF의 사용 단순화.....	35
REST API를 사용하여 관리.....	70
administrative REST API 시작하기.....	70
REST API를 사용한 원격 관리.....	75
REST API 시간소인.....	79
REST API 오류 핸들링.....	79
REST API 검색.....	81
REST API 자국어 지원(NLS).....	83
REST API 버전.....	84
IBM MQ Console를 사용하여 관리.....	86
IBM MQ Console 시작하기.....	86
IBM MQ Console의 Quick Tour.....	87
IBM MQ Console 설정.....	112
IBM MQ Explorer를 사용하여 관리.....	112
IBM MQ Explorer로 수행 가능한 작업.....	113
IBM MQ Explorer 설정.....	114
IBM MQ 작업 표시줄 애플리케이션(Windows 전용) 사용.....	119
IBM MQ 경보 모니터 애플리케이션(Windows 전용).....	119
로컬 IBM MQ 오브젝트에 대한 작업.....	120
큐 관리자에 대한 작업.....	120
MQI 채널 중지.....	130
로컬 큐에 대한 작업.....	130
리모트 큐에 대한 작업.....	139
알리어스 큐에 대한 작업.....	142
모델 큐에 대한 작업.....	144
데드-레터 큐에 대한 작업.....	145
관리 토픽에 대한 작업.....	162
구독에 대한 작업.....	166
서비스에 대한 작업.....	170
트리거에 대한 오브젝트 관리.....	177
두 시스템 사이에 dmpmqmsg 유틸리티 사용.....	179
리모트 IBM MQ 오브젝트에 대한 작업.....	183
원격 관리를 위한 큐 관리자 구성.....	183
원격 관리를 위한 명령 서버 관리.....	187
리모트 큐 관리자에서 MQSC 명령 실행.....	188
코드화 문자 세트 간 데이터 변환.....	189
Managed File Transfer 관리.....	194
MFT 에이전트 시작.....	195
MFT 에이전트 나열.....	200
MFT 에이전트 중지.....	200

새 파일 전송 시작.....	202
스케줄된 파일 전송 작성.....	205
보류 중인 파일 전송에 대한 작업.....	206
파일 전송 트리거.....	207
진행 중인 파일 전송 모니터링.....	208
전송 로그에서 파일 전송 상태 보기.....	210
MFT 자원 모니터링.....	211
파일 전송 템플릿에 대한 작업.....	240
파일에서 메시지로 데이터 전송.....	242
메시지에서 파일로 데이터 전송.....	256
프로토콜 브릿지.....	266
Connect:Direct 브릿지.....	287
IBM Integration Bus 에서 MFT 에 대한 작업.....	302
MFT 복구 및 재시작.....	302
중단된 전송의 복구에 대한 제한시간 설정.....	303
MQ Telemetry 관리.....	308
Linux 및 AIX에서 텔레메트리에 대한 큐 관리자 구성.....	308
Windows에 텔레메트리용 큐 관리자 구성.....	310
메시지를 MQTT 클라이언트로 송신하기 위한 분산 큐잉 구성.....	312
MQTT 클라이언트 식별, 권한 부여, 인증.....	314
TLS를 사용하여 텔레메트리 채널 인증.....	320
텔레메트리 채널의 발행물 개인정보 보호.....	324
MQTT Java 클라이언트 및 텔레메트리 채널의 TLS 구성.....	324
텔레메트리 채널 JAAS 구성.....	329
AMQP 클라이언트 관리.....	331
AMQP 서비스가 큐 관리자 시동 시 자동으로 시작되지 않음.....	331
AMQP 클라이언트에서 사용 중인 IBM MQ 오브젝트 보기.....	331
AMQP 클라이언트 식별, 권한 부여 및 인증.....	333
채널에서 발행물 개인정보 보호.....	335
TLS로 AMQP 클라이언트 구성.....	335
큐 관리자에서 AMQP 클라이언트 연결 끊기.....	336
멀티캐스트 관리.....	336
멀티캐스트 시작하기.....	337
IBM MQ 멀티캐스트 토픽 토폴로지.....	338
멀티캐스트 메시지의 크기 조절.....	338
멀티캐스트 메시징에 대한 데이터 변환 사용 가능.....	340
멀티캐스트 애플리케이션 모니터링.....	341
멀티캐스트 메시지 신뢰성.....	341
고급 멀티캐스트 태스크.....	342
IBM MQ for IBM i 관리.....	345
CL 명령을 사용하여 IBM MQ for IBM i 관리.....	345
IBM MQ for IBM i 관리의 대체 방법.....	358
IBM i에 대한 작업 관리.....	364
가용성, 백업, 복구 및 재시작(IBM i).....	370
IBM MQ for IBM i 일시정지.....	410
Administering IBM MQ for z/OS.....	413
Issuing queue manager commands on z/OS.....	414
Using the operations and control panels on z/OS.....	428
Using the IBM MQ for z/OS utilities.....	436
Using the Command Facility on z/OS.....	439
Working with IBM MQ objects on z/OS.....	439
Implementing the system using multiple cluster transmission queues.....	442
Writing programs to administer IBM MQ for z/OS.....	444
Managing IBM MQ resources on z/OS.....	456
Recovery and restart on z/OS.....	494
IBM MQ and IMS.....	515
Operating Advanced Message Security on z/OS.....	527
IBM MQ Internet Pass-Thru 관리.....	528

MQIPT 시작 및 중지.....	528
명령행을 사용하여 MQIPT 관리.....	530
백업 작성.....	536
성능 조정.....	536
주의사항.....	539
프로그래밍 인터페이스 정보.....	540
상표.....	540

IBM MQ 관리

IBM MQ 큐 관리자 및 연관된 자원을 관리하려면 해당 자원을 활성화하고 관리하기 위해 사용할 수 있는 태스크 세트에서 선호하는 방법을 선택하십시오.

이 태스크 정보

다음과 같이 IBM MQ 오브젝트를 로컬로 또는 원격으로 관리할 수 있습니다.

로컬 관리

로컬 관리는 로컬 시스템에 정의된 큐 관리자에 대해 관리 태스크를 수행하는 것을 의미합니다. 기타 시스템 (예: TCP/IP 터미널 에뮬레이션 프로그램 **telnet**을 통해)에 액세스할 수 있고 거기에서 관리를 수행할 수 있습니다. IBM MQ에서 포함되는 채널이 없기 때문에 로컬 관리로서 이를 고려할 수 있습니다. 즉, 통신이 운영 체제에 의해 관리됩니다.

자세한 정보는 120 페이지의 『로컬 IBM MQ 오브젝트에 대한 작업』의 내용을 참조하십시오.

원격 관리


IBM MQ는 원격 관리를 통해 단일 접속 지점에서의 관리를 지원합니다. 원격 관리를 사용하면 로컬 시스템에서 명령을 발행하여 다른 시스템에서 처리할 수 있고 IBM MQ Explorer에도 적용됩니다. 예를 들어, 리모트 큐 관리자에서 큐 정의를 변경하기 위한 리모트 명령을 실행할 수 있습니다. 적절한 채널이 정의되도록 해야 하지만 해당 시스템에 로그인할 필요는 없습니다. 대상 시스템의 큐 관리자 및 명령 서버가 실행 중이어야 합니다.

일부 명령은 특히, 큐 관리자를 작성하거나 시작하고 명령 서버를 시작하는 방법으로 실행될 수 없습니다. 이 유형의 태스크를 수행하려면 원격 시스템에 로그인하여 거기서 명령을 발행하거나 명령을 자동으로 발행할 수 있는 프로세스를 작성해야 합니다. 이 제한은 IBM MQ Explorer에도 적용됩니다.




자세한 정보는 183 페이지의 『리모트 IBM MQ 오브젝트에 대한 작업』의 내용을 참조하십시오.


IBM MQ에서 큐 관리자 및 해당 관련 자원을 작성 및 관리하기 위해 사용할 수 있는 여러 가지 메소드가 있습니다. 이 메소드에는 명령행 인터페이스, 그래픽 사용자 인터페이스 및 관리 API가 포함됩니다.

플랫폼에 따라 IBM MQ를 관리하는 데 사용할 수 있는 여러 명령 세트가 있습니다.

- 8 페이지의 『IBM MQ 제어 명령』
- 8 페이지의 『MQSC(IBM MQ Script) 명령』
- 8 페이지의 『프로그래밍 가능 명령 형식(PCF)』
- The administrative REST API
-  9 페이지의 『IBM i 제어 언어(CL)』

또한 IBM MQ 오브젝트를 작성 및 관리하기 위한 다음과 같은 기타 옵션도 있습니다.

-   9 페이지의 『IBM MQ Explorer』
- 9 페이지의 『IBM MQ Console』
-  9 페이지의 『MSCS(Microsoft Cluster Service)』

 IBM MQ for z/OS®에서 관리 인터페이스 및 옵션에 대한 정보는 413 페이지의 『Administering IBM MQ for z/OS』의 내용을 참조하십시오.

PCF 명령을 사용하여 로컬 및 리모트 큐 관리자 모두에 대한 일부 관리 및 모니터링 태스크를 자동화할 수 있습니다. 이 명령은 일부 플랫폼에서 IBM MQ 관리 인터페이스(MQAI)를 사용하여 단순화될 수도 있습니다. 관리 태스크 자동화에 대한 자세한 정보는 23 페이지의 『PCF 명령을 사용하여 IBM MQ 관리 자동화』의 내용을 참조하십시오.

관련 개념

[IBM MQ 기술 개요](#)

관련 태스크

[계획](#)

[구성](#)

[관련 참조](#)

[명령 세트 비교](#)

IBM MQ 큐 관리자 및 연관된 자원을 관리하는 방법

IBM MQ 제어 명령, IBM MQ 스크립트 명령 (MQSC), 프로그래밍 가능 명령 형식 (PCF), administrative REST API, IBM MQ Console 및 IBM MQ Explorer를 사용하여 IBM MQ 큐 관리자 및 연관된 자원을 관리할 수 있습니다. IBM i의 경우 IBM i Control Language를 사용할 수도 있고, Windows의 경우 MSCS (Microsoft Cluster Service)를 사용할 수도 있습니다.

IBM MQ 제어 명령

Multi

제어 명령은 다수의 IBM MQ 관리 태스크를 수행하는 방법을 제공합니다. AIX®, Linux®, and Windows의 경우 시스템 명령행에서 다음 명령을 실행합니다. IBM i의 경우 Qshell내에서 이러한 명령을 실행합니다. [10 페이지의 『제어 명령을 사용하여 IBM MQ for Multiplatforms 관리』](#)의 내용을 참조하십시오.

MQSC(IBM MQ Script) 명령

큐 관리자 자체, 큐, 프로세스 정의, 이름 리스트, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트를 포함하여, 큐 관리자 오브젝트를 관리하기 위해 MQSC 명령을 사용하십시오.

ALW

AIX, Linux, and Windows에서는 **runmqsc** 명령 프롬프트를 열고 해당 프롬프트에서 로컬 또는 리모트 큐 관리자에 MQSC 명령을 실행합니다. 이를 대화식으로 수행하거나 ASCII 텍스트 파일에서 일련의 명령을 실행할 수 있습니다. 자세한 정보는 [16 페이지의 『runmqsc에서 대화식으로 MQSC 명령 실행』](#) 및 [21 페이지의 『runmqsc의 텍스트 파일에서 MQSC 명령 실행』](#)의 내용을 참조하십시오.

IBM i

IBM i에서는 스크립트 파일에 명령 목록을 작성한 후 **STRMQMMQSC** 명령을 사용하여 파일을 실행합니다. 자세한 정보는 [359 페이지의 『IBM i에서 MQSC 명령을 사용하여 관리』](#)의 내용을 참조하십시오.

z/OS

z/OS의 경우에는, 명령에 따라 여러 소스에서 MQSC 명령을 실행할 수 있습니다. 자세한 정보는 [414 페이지의 『Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS』](#)의 내용을 참조하십시오.

프로그래밍 가능 명령 형식(PCF)

프로그래밍 가능 명령 형식(PCF)는 네트워크의 큐 관리자(PCF 지원) 및 프로그램 사이에서 교환할 수 있는 명령 및 응답 메시지를 정의합니다. IBM MQ 오브젝트(인증 정보 오브젝트, 채널, 채널 리스너, 이름 목록, 프로세스 정의, 큐 관리자, 큐, 서비스 및 스토리지 클래스)의 관리를 위해 시스템 관리 애플리케이션 프로그램에서 PCF 명령을 사용할 수 있습니다. 애플리케이션은 네트워크의 단일 지점에서 작동하여 로컬 큐 관리자를 사용하여 로컬 또는 원격으로 모든 큐 관리자와 명령 및 응답 정보를 통신할 수 있습니다.

PCF에 대한 자세한 정보는 [24 페이지의 『IBM MQ PCF\(Programmable Command Format\) 소개』](#)의 내용을 참조하십시오.

PCF 정의 및 명령 및 응답의 구조는 [프로그래밍 가능 명령 형식\(PCF\) 참조서를 참조하십시오.](#)

administrative REST API

administrative REST API는 IBM MQ를 관리하는 데 사용할 수 있는 RESTful 인터페이스를 제공합니다. administrative REST API를 사용하는 경우 IBM MQ 오브젝트를 나타내는 URL에서 HTTP 메소드를 호출합니다. 예를 들어, 다음 URL에서 HTTP 메소드 GET을 사용하여 IBM MQ 설치에 대한 정보를 요청할 수 있습니다.

```
https://localhost:9443/ibmmq/rest/v1/admin/installation
```


administrative REST API를 프로그래밍 언어의 HTTP/REST 구현과 함께 사용하거나 cURL 등의 도구 또는 REST 클라이언트 브라우저 추가 기능을 사용하여 사용할 수 있습니다.

For more information, see [The administrative REST API](#)

IBM MQ Console

IBM MQ Console을 사용하여 웹 브라우저에서 IBM MQ를 관리할 수 있습니다.

자세한 정보는 [86 페이지의 『IBM MQ Console을 사용하여 관리』](#)의 내용을 참조하십시오.

IBM MQ Explorer



IBM MQ Explorer를 사용하여 다음 조치를 수행할 수 있습니다.

- 큐 관리자, 큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스, 클러스터 등의 다양한 자원을 정의 및 제어하십시오.
- 로컬 큐 관리자 및 해당 연관된 프로세스를 시작하거나 중지하십시오.
- 워크스테이션 또는 기타 워크스테이션에서 큐 관리자 및 연관된 오브젝트를 보십시오.
- 큐 관리자, 클러스터 및 채널의 상태를 검사하십시오.
- 큐 상태에서 어떤 애플리케이션, 사용자 또는 채널에 특정 큐가 열려 있는지를 보려면 검사하십시오.

Windows 및 Linux for x86-64 시스템에서는 시스템 메뉴 또는 MQExplorer 실행 파일을 사용하여 IBM MQ Explorer를 시작할 수 있습니다.

Linux Linux에서 IBM MQ Explorer를 성공적으로 시작하려면 홈 디렉토리가 존재하고 이 홈 디렉토리에 파일을 작성할 수 있어야 합니다.

자세한 정보는 [112 페이지의 『IBM MQ Explorer를 사용하여 관리』](#)의 내용을 참조하십시오.

IBM MQ Explorer를 사용하여 다른 플랫폼에서 리모트 큐 관리자를 관리할 수 있습니다 z/OS 포함.

IBM MQ 9.3.0에서 IBM MQ Explorer가 IBM MQ 설치 패키지에서 제거되었습니다. 별도의 다운로드로 사용할 수 있는 상태로 유지되며 Fix Central에서 사용할 수 있는 독립형 IBM MQ Explorer 다운로드에서 설치할 수 있습니다. 자세한 정보는 [Linux 및 Windows에서 IBM MQ Explorer를 독립형 애플리케이션으로 설치 및 설치 제거의 내용](#)을 참조하십시오.

IBM i 제어 언어(CL)



이는 IBM MQ for IBM i에 관리 명령을 실행하는 데 선호되는 방법입니다. 명령은 명령행에서 또는 CL 프로그램을 작성하여 실행할 수 있습니다. 이러한 명령은 PCF 명령과 유사한 기능을 수행하지만 형식은 다릅니다. CL 명령은 서버용으로 독점적으로 설계되어 있고 CL 응답은 사람이 읽을 수 있는 반면 PCF 명령은 플랫폼 독립적이며 명령 형식과 응답 형식 둘 다 프로그램용으로 제공됩니다.

IBM i 제어 언어 (CL)에 대한 전체 세부사항은 [345 페이지의 『CL 명령을 사용하여 IBM MQ for IBM i 관리』](#) 및 [IBM MQ for IBM i CL 명령을 참조하십시오.](#)

MSCS(Microsoft Cluster Service)



MSCS(Microsoft Cluster Service)는 데이터 및 애플리케이션의 더 높은 가용성을 제공하고 시스템을 관리하는 것을 보다 용이하게 하면서 클러스터에 서버를 연결시킬 수 있도록 합니다. MSCS는 서버 또는 애플리케이션 실패를 자동으로 감지하고 복구할 수 있습니다.

MSCS 센스의 클러스터를 IBM MQ 클러스터와 혼동하지 않는 것이 중요합니다. 차이는 다음과 같습니다.

IBM MQ 클러스터

하나 이상의 컴퓨터에 있는 두 개 이상의 큐 관리자 그룹이며 자동 상호연결을 제공하고 로드 밸런싱 및 중복성을 위해 큐를 서로 공유할 수 있게 합니다.

MSCS 클러스터

서로 연결되어 있으며 한 대의 컴퓨터에 장애가 발생하면 MSCS가 장애 복구를 수행하여 애플리케이션의 상태 데이터를 장애가 발생한 컴퓨터에서 클러스터의 다른 컴퓨터로 전송하고 해당 조작을 그 위치에서 다시 시작하는 방식으로 구성된 컴퓨터의 그룹입니다.

MSCS(Supporting the Microsoft Cluster Service)에서는 MSCS를 사용하기 위해 IBM MQ for Windows 시스템을 구성하는 방법에 대한 자세한 정보를 제공합니다.

관련 태스크

11 페이지의 『MQSC 명령을 사용하여 IBM MQ 관리』

큐 관리자 자체, 큐, 프로세스 정의, 채널, 클라이언트 연결 채널, 리스너, 서비스, 이름 목록, 클러스터 및 인증 정보 오브젝트를 포함하여 MQSC 명령을 사용하여 큐 관리자 오브젝트를 관리할 수 있습니다. MQSC 명령은 모든 플랫폼에서 사용 가능합니다.

관련 참조

관리 참조

Multi

제어 명령을 사용하여 IBM MQ for Multiplatforms 관리

제어 명령은 다수의 IBM MQ 관리 태스크를 수행하는 방법을 제공합니다. AIX, Linux 및 Windows의 경우 시스템 명령행에서 다음 명령을 실행합니다. IBM i의 경우 Qshell내에서 이러한 명령을 실행합니다.

시작하기 전에

큐 관리자에서 작동하는 제어 명령을 사용하는 경우 작업하는 큐 관리자와 연관된 설치의 명령을 사용해야 합니다.

CHKLOCL(REQUIRED)을 사용한 연결 인증을 사용하도록 구성된 큐 관리자에서 작동하는 제어 명령을 사용할 때 연결 실패가 관찰되는 경우 다음을 수행하십시오.

- 제어 명령이 허용하는 경우 사용자 ID와 비밀번호를 제공하십시오.
- 존재하는 제어 명령의 MQSC 동등 항목을 사용하십시오.
- 연결할 수 없는 제어 명령을 실행해야 하는 동안 -ns 옵션을 사용하여 큐 관리자를 시작하십시오.

참고: 다른 플랫폼은 다른 순서로 입력된 명령 인수를 승인할 수 있습니다. 특히, 이는 Linux 에서 작동하는 명령이 다른 플랫폼에서 작동하지 않을 수 있음을 의미합니다. 이러한 이유로 항상 구문 다이어그램에 지정된 대로 인수를 입력해야 합니다.

제어 명령의 전체 목록은 [IBM MQ 제어 명령 참조](#)를 참조하십시오.

프로시저

- 

AIX and Linux 시스템에서 제어 명령을 사용하십시오.

IBM MQ for AIX or Linux 시스템에서는 셸 창에 제어 명령을 입력합니다.

제어 명령을 실행하려면 사용자 ID가 대부분의 제어 명령에 대한 mqm 그룹의 멤버여야 합니다. 이에 대한 자세한 정보는 [IBM MQ on AIX, Linux, and Windows 관리 권한](#)을 참조하십시오. 또한 환경 관련 정보를 참고하십시오. 플랫폼의 경우 엔터프라이즈가 사용됩니다.

UNIX and Linux 환경에서는 제어 명령(명령 이름 자체, 플래그 및 인수 포함)이 대소문자를 구분합니다. 예를 들어, 다음 명령에서:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- 명령 이름은 CRTMQM이 아니라 crtmqm이어야 합니다.

- 플래그는 -U가 아니라 -u여야 합니다.
 - 데드-레터 큐는 SYSTEM.DEAD.LETTER.QUEUE라고 합니다.
 - 인수는 jupiter.queue.manager로 지정되며 이는 JUPITER.queue.manager와 다릅니다.
- 예에 표시되는 대로 정확하게 명령을 입력하도록 주의하십시오.

Windows

Windows 시스템에서 제어 명령을 사용하십시오.

IBM MQ for Windows에서는 명령 프롬프트에서 제어 명령을 입력합니다.

제어 명령을 실행하려면 사용자 ID가 대부분의 제어 명령에 대한 mqm 그룹의 멤버여야 합니다. 이에 대한 자세한 정보는 [IBM MQ on AIX, Linux, and Windows 관리 권한을 참조하십시오](#). 또한 환경 관련 정보를 참고하십시오. 플랫폼의 경우 엔터프라이즈가 사용됩니다.

제어 명령 및 해당 플래그는 대소문자를 구분하지 않지만 해당 명령에 대한 인수(예: 큐 이름 및 큐 관리자 이름)는 대소문자를 구분합니다.

예를 들어, 다음 명령에서:

```
crtmqm /u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- 명령 이름은 대문자로, 소문자로 또는 대소문자를 혼합하여 입력할 수 있습니다. crtmqm, CRTMQM 및 CRTmqm이 모두 올바릅니다.
- 플래그는 -u, -U, /u 또는 /U로 입력할 수 있습니다.
- SYSTEM.DEAD.LETTER.QUEUE 및 jupiter.queue.manager는 표시된 대로 정확하게 입력해야 합니다.

IBM i

IBM i 시스템에서 제어 명령을 사용하십시오.

IBM MQ for IBM i의 경우 Qshell 환경에서 제어 명령을 실행합니다. Qshell을 사용하려면 IBM i 명령행에 STRQSH 를 입력하십시오. 언제든지 F3을 눌러 종료하고 명령행으로 돌아갈 수 있습니다.

소수의 제어 명령은 IBM i에서 지원되지 않습니다. 예를 들어, IBM i 시스템에 둘 이상의 IBM MQ 사본이 있을 수 없으므로 다중 설치 명령은 지원되지 않습니다. IBM i에서 지원되지 않는 명령은 [IBM MQ 제어 명령 참조](#)에서 **ALW** 플래그 지정됩니다.

관련 참조

[IBM MQ 제어 명령 참조](#)

MQSC 명령을 사용하여 IBM MQ 관리

큐 관리자 자체, 큐, 프로세스 정의, 채널, 클라이언트 연결 채널, 리스너, 서비스, 이름 목록, 클러스터 및 인증 정보 오브젝트를 포함하여 MQSC 명령을 사용하여 큐 관리자 오브젝트를 관리할 수 있습니다. MQSC 명령은 모든 플랫폼에서 사용 가능합니다.

이 태스크 정보

사용 가능한 MQSC 명령은 [MQSC 명령 참조](#)에 자세히 설명되어 있습니다.

MQSC 명령을 실행하는 방법은 플랫폼에 따라 다릅니다.

- ALW** AIX, Linux, and Windows의 경우 **runmqsc** 명령 프롬프트에서 큐 관리자에 MQSC 명령을 실행합니다. 다음과 같은 여러 가지 방법으로 이 명령 프롬프트를 사용할 수 있습니다.
 - 키보드에서 MQSC 명령을 대화식으로 실행합니다. [16 페이지의 『runmqsc 에서 대화식으로 MQSC 명령 실행』](#)의 내용을 참조하십시오.
 - ASCII 텍스트 파일에서 MQSC 명령을 실행합니다. [21 페이지의 『runmqsc 의 텍스트 파일에서 MQSC 명령 실행』](#)의 내용을 참조하십시오.

- 리모트 큐 관리자에서 MQSC 명령 실행 [188 페이지](#)의 『리모트 큐 관리자에서 MQSC 명령 실행』의 내용을 참조하십시오.
- ▶ **IBM i** IBM i에서는 스크립트 파일에 명령 목록을 작성한 후 **STRMQMMQSC** 명령을 사용하여 파일을 실행합니다. 자세한 정보는 [359 페이지](#)의 『IBM i에서 MQSC 명령을 사용하여 관리』의 내용을 참조하십시오.
- ▶ **z/OS** z/OS의 경우에는, 명령에 따라 여러 소스에서 MQSC 명령을 실행할 수 있습니다. 자세한 정보는 [414 페이지](#)의 『Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS』의 내용을 참조하십시오.

프로시저

- [12 페이지](#)의 『MQSC 명령 구문』
- [14 페이지](#)의 『MQSC: 특수 문자 및 일반 값』
- [16 페이지](#)의 『runmqsc에서 대화식으로 MQSC 명령 실행』
- [21 페이지](#)의 『runmqsc의 텍스트 파일에서 MQSC 명령 실행』
- [22 페이지](#)의 『시동 시 MQSC 스크립트에서 자동 구성』

관련 태스크

[MQSC 명령으로 문제점 해결](#)

관련 참조

[runmqsc\(MQSC 명령 실행\)](#)

MQSC 명령 구문

MQSC 명령을 사용하여 큐 관리자 오브젝트를 관리할 수 있습니다. MQSC 명령은 모든 플랫폼에서 사용 가능합니다. 명령 구문의 일부 요소는 플랫폼에 따라 다릅니다.

매개변수 순서

각 명령은 1차 매개변수(동사)로 시작하고, 이 뒤에 2차 매개변수(명사)가 옵니다. 그런 다음 대부분의 명령에서 와 같이 오브젝트가 하나 있는 경우 오브젝트의 이름 또는 일반 이름(괄호 안에) 옵니다. 그 뒤에 매개변수가 임의의 순서로 발생할 수 있습니다. 매개변수에 해당 값이 있는 경우에는 관련된 매개변수 다음에 바로 값이 발생해야 합니다.

참고: ▶ **z/OS** z/OS에서 보조 매개변수가 2차가 될 필요는 없습니다.

공백 및 쉼표

키워드, 괄호 및 값을 공백 및 쉼표 수로 분리할 수 있습니다. 구문 다이어그램에 표시된 쉼표는 항상 하나 이상의 공백으로 대체될 수 있습니다. z/OS를 제외하고 각 매개변수의 바로 앞에 하나 이상의 공백이 있어야 합니다(기본 매개변수 뒤).

많은 공백이 명령의 시작 또는 끝에서 발생할 수 있고 매개변수, 구두점 및 값 사이에 발생할 수 있습니다. 예를 들어, 다음 명령이 유효합니다.

```
ALTER QLOCAL ('Account' ) TRIGDPTH ( 1)
```

한 쌍의 구두점 내에 있는 공백은 중요합니다.

공백이 허용되는 어디에서나 추가 쉼표가 나타날 수 있으며 공백인 것처럼 취급됩니다(물론 따옴표로 묶은 문자열 안에 있지 않은 경우).

반복 매개변수

반복되는 매개변수는 허용되지 않습니다. REPLACE NOREPLACE에서와 같이, 해당 "NO" 버전을 가진 매개변수를 반복하도록 허용되지 않습니다.

문자열 및 작은따옴표

공백, 소문자 또는 특수 문자를 포함하는 문자열은 다음 중 하나가 true가 아닌 경우 작은따옴표로 묶어야 합니다.

- 특수 문자는 다음 문자 중 하나 이상입니다.
 - 마침표(.)
 - 정방향 슬래시(/)
 - 밑줄(_)
 - 퍼센트 부호(%)
- **z/OS** 명령은 IBM MQ for z/OS 조작 및 제어판에서 실행됩니다.
- 문자열은 별표로 끝나는 일반 값입니다 (IBM i에서는 작은따옴표로 묶어야 함)
- 문자열이 단일 별표입니다(예: TRACE(*))(IBM i의 경우 작은따옴표로 묶어야 함).
- 문자열이 콜론이 포함된 범위 스펙입니다(예: CLASS(01:03)).

문자열 자체가 작은따옴표를 포함하는 경우 작은따옴표는 두 개의 작은따옴표로 표시됩니다.

Multi 멀티플랫폼에서, 문자가 포함되지 않는 문자열(즉, 사이에 공백이 없는 두 개의 작은따옴표)은 작은따옴표로 묶인 공백으로 해석됩니다(즉, (' ')와 동일하게 해석됨). 공백이 없는 두 개의 작은따옴표가 길이가 0인 문자열로 해석될 때 이에 대한 예외는 사용 중인 속성이 다음 속성 중 하나인 경우입니다.

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

z/OS z/OS에서, 공백을 작은따옴표로 묶으려는 경우 (' ')와 같이 입력해야 합니다. (') 문자가 포함되지 않은 문자열은 ()을 입력하는 것과 같습니다.

MQCHARV 유형(예: 하위 사용자 데이터인 SELECTOR)을 기반으로 하는 문자열 속성의 후미 문자 공백은 중요한 것으로 취급되고, 이는 'abc '가 'abc'와 같지 않다는 것을 의미합니다.

빈 소괄호

사이에 의미 있는 정보가 없고 뒤에 닫기 괄호가 오는 열기 괄호는 특별히 언급된 경우를 제외하고 올바르지 않습니다. 예를 들어, 다음 문자열은 올바르지 않습니다.

```
NAME ( )
```

소문자 및 대문자

키워드는 대소문자를 구분하지 않습니다. ALTER, 대체 및 ALTER 모두 허용 가능합니다.

인용부호 내에 포함되지 않은 것은 대문자로 변환됩니다.

동의어

동의어는 일부 매개변수에 대해 정의됩니다. 예를 들어, DEF는 항상 DEFINE의 동의어이므로, DEF QLOCAL은 유효합니다. 그러나, 동의어는 최소 문자열이 아닙니다. DEF1는 DEFINE에 대한 올바른 동의어가 아닙니다.

참고: DELETE 매개변수에 대한 동의어가 없습니다. DEFINE에 대한 동의어, DEF를 사용할 때 오브젝트가 삭제되는 읽을 방지하기 위한 것입니다.

특수 문자

MQSC 명령은 특정 의미를 나타내도록 특정 특수 문자를 사용합니다. 이러한 특수 문자 및 이를 사용하는 방법에 대한 자세한 정보는 14 페이지의 『MQSC: 특수 문자 및 일반 값』의 내용을 참조하십시오.

관련 태스크

[MQSC 명령으로 문제점 해결](#)

관련 참조

[runmqsc\(MQSC 명령 실행\)](#)

MQSC: 특수 문자 및 일반 값


일부 문자 (예: 백슬래시 (\) 및 큰따옴표 (")) 문자는 MQSC 명령과 함께 사용될 때 특별한 의미를 갖습니다. 매개 변수와 함께 사용할 수 있는 일부 특수 문자는 일반 값을 가질 수 있지만 올바르게 지정해야 합니다.

백슬래시 (\) 와 큰따옴표 (") 앞에 문자를 \로 바꾸십시오. 즉, 텍스트에 \ 또는 " 를 사용하려면 \\ 또는 \" 를 입력하십시오.

일반 값을 가질 수 있는 매개변수는 별표로 끝납니다(예: ABC*). 일반 값은 해당 문자로 시작되는 모든 값을 의미하므로, ABC*는 ABC로 시작하는 모든 값입니다. 따옴표가 필요한 문자를 값에 사용할 경우 따옴표 안에 별표를 배치해야 합니다(예: 'abc*'). 별표는 마지막에 있거나 값의 유일한 문자여야 합니다.

물음표(?)와 콜론(:)은 일반 값에서 허용되지 않습니다.

필드에서 이러한 특수 문자를 사용해야 할 때에는(예를 들어, 설명의 일부분으로) 전체 문자열을 작은 따옴표로 묶어야 합니다.

문자	설명
	공백은 분리문자로 사용됩니다. 어포스트로피(')로 묶은 문자열의 경우를 제외하고 여러 개의 공백은 하나의 공백과 동등합니다. MQCHARV 유형에 기반한 해당 문자열 속성의 후미 문자 공백은 중요한 것으로 처리됩니다.
,	쉼표는 분리문자로 사용됩니다. 어포스트로피(')로 묶은 문자열의 경우를 제외하고 여러 개의 쉼표는 하나의 쉼표와 동등합니다.
'	어포스트로피는 문자열의 시작이나 끝을 표시합니다. IBM MQ는 물음표로 묶은 모든 문자를 입력된 그대로 둡니다. 포함하는 어포스트로피는 문자열 길이를 계산할 때 포함되지 않습니다.
"	문자열 내의 작은따옴표는 IBM MQ에서 문자열의 길이를 계산할 때 한 문자로 취급되며 문자열이 종료되지 않습니다.
=	 z/OS에서 등호는 쉼표나 공백으로 끝나는 매개변수 값의 시작을 나타냅니다.
(열기 괄호는 값의 목록 또는 매개변수 값의 시작을 나타냅니다.
)	닫기 괄호는 값의 목록 또는 매개변수 값의 끝을 나타냅니다.
:	콜론은 표시된 수가 포함된 범위를 나타냅니다. 예를 들어 (1:5)는 (1,2,3,4,5)를 의미합니다. 이 표기법은 TRACE 명령에서만 사용할 수 있습니다.
*	별표는 모두를 의미합니다. 예를 들어, DISPLAY TRACE (*)는 모든 추적 표시를 의미하고, DISPLAY QUEUE (PAY*)는 이름이 PAY로 시작하는 모든 큐의 표시를 의미합니다.

MQSC 입력 파일 구문

긴 명령이 있거나 특정 순서의 명령을 반복적으로 사용하는 경우 입력 파일을 사용하여 MQSC 명령을 실행할 수 있습니다. 입력 파일의 콘텐츠는 이 주제에 설명된 구문을 따라야 합니다.

개요

MQSC 명령은 stdin라고도 하는 표준 입력 디바이스를 통해 입력됩니다. 일반적으로 키보드이지만 입력 파일에서 입력을 가져오도록 지정할 수 있습니다.

다음 플랫폼 특정 도구와 함께 이 입력 파일을 사용할 수 있습니다.

- ▶ **ALW** AIX, Linux, and Windows의 **runmqsc** 명령. :NONE. 21 페이지의 『[runmqsc의 텍스트 파일에서 MQSC 명령 실행](#)』
- ▶ **IBM i** IBM i의 **STRMQM** 명령. :NONE. 359 페이지의 『[IBM i에서 MQSC 명령을 사용하여 관리](#)』
- ▶ **z/OS** z/OS에서 CSQINP1, CSQINP2 및 CSQINPX 초기화 데이터 세트 또는 CSQUTIL 배치 유틸리티. :NONE. 414 페이지의 『[Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS](#)』

구문

MQSC 입력 파일 구문:

- IBM MQ 환경 사이의 이식성의 경우, MQSC 명령 파일의 행 길이를 72자로 제한하십시오.
- 각 명령은 새 행에서 시작되어야 합니다.
- 첫 번째 위치에서 별표(*)로 시작하는 행은 무시됩니다. 이 행은 파일에 주석을 삽입할 때 사용할 수 있습니다.
- 빈 줄은 무시됩니다.
- 더하기 부호(+)는 명령이 다음 행의 첫 번째 비공백 문자에서 계속됨을 표시합니다. +를 사용하여 명령을 계속할 경우 다음 매개변수 앞에 공백을 적어도 한 개 남겨야 합니다(z/OS의 경우 이 사항이 반드시 적용되지 않음). 주석 또는 빈 행은 명령이 단일 문자열로 리어셈블링될 때 제거됩니다.
- 빼기 부호(-)는 다음 행의 시작 부분에서 명령이 계속된다는 것을 나타냅니다. 주석 또는 빈 행은 명령이 단일 문자열로 리어셈블링될 때 제거됩니다.
- Escape PCF(Programmable Command Format) 명령에 포함된 MQSC 명령은 더하기 부호 또는 빼기 부호로 계속될 수 있습니다. 전체 명령은 하나의 Escape 명령 내에 포함되어야 합니다. PCF 명령에 대한 정보는 24 페이지의 『[IBM MQ PCF\(Programmable Command Format\) 소개](#)』의 내용을 참조하십시오.
- 멀티플랫폼 및 CSQUTIL 배치 유틸리티 프로그램에서 실행된 명령에 대해 z/OS에서 이전 행 끝에 더하기 기호(+)를 입력한 경우에도 세미콜론 문자(;)를 사용하여 명령을 종료할 수 있습니다.
- 명령이 키보드 제어 문자(예: 탭)로 끝나면 안 됩니다.
- 텍스트 파일에서 stdin 경로를 재지정하여 클라이언트 모드에서 **runmqsc** 명령을 실행하고 신임 정보를 제공하기 위해 **-u** 플래그를 제공하는 경우, **runmqsc** 명령은 비밀번호를 입력하도록 프롬프트를 표시하지 않으며 대신 stdin에서 비밀번호를 읽습니다. stdin 를 통해 제공된 데이터의 첫 번째 행이 비밀번호인지 확인해야 합니다. "echo" 또는 "cat" 과 같은 명령행 도구를 사용하고 뒤에 MQSC 스크립트가 오는 비밀번호를 **runmqsc** 명령 stdin에 전달하여 이를 수행할 수 있습니다.
- ▶ **Windows** Windows에서 파운드 기호(£) 및 논리적 NOT(~)과 같은 특수 문자가 명령 스크립트(예: 오브젝트 설명의 일부)에 사용된 경우 **DISPLAY QLOCAL** 등의 명령 출력에 다르게 표시됩니다.

12 페이지의 『[MQSC 명령 구문](#)』도 참조하십시오.

예

다음 예는 **DEFINE QLOCAL** 명령을 표시하는 MQSC 명령 파일에서 추출한 것입니다.

```

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

```

그림 1. MQSC 명령 파일에서 추출

runmqsc 명령이 완료되면 보고서가 리턴됩니다. 다음 예는 보고서에서 추출된 것입니다.

```

Starting MQSC for queue manager jupiter.queue.manager.
.
12:  DEFINE QLOCAL('ORANGE.LOCAL.QUEUE') REPLACE +
:    DESCR(' ') +
:    PUT(ENABLED) +
:    DEFPRTY(0) +
:    DEFPSIST(NO) +
:    GET(ENABLED) +
:    MAXDEPTH(5000) +
:    MAXMSGL(1024) +
:    DEFSOPT(SHARED) +
:    NOHARDENBO +
:    USAGE(NORMAL) +
:    NOTRIGGER;
AMQ8006: IBM MQ queue created.
.
.

```

그림 2. MQSC 명령 보고서 파일에서 추출

또한 예제 MQSC 명령 파일을 사용하여 텍스트 파일을 작성할 수 있습니다.

amqscos0.tst

샘플 프로그램에서 사용한 오브젝트의 정의.

amqscic0.tst

CICS® 트랜잭션에 대한 큐의 정의.

Linux AIX Linux and AIX에서 해당 파일은 `MQ_INSTALLATION_PATH/samp` 디렉토리에 있습니다. `MQ_INSTALLATION_PATH`는 IBM MQ가 설치되어 있는 상위 레벨 디렉토리를 나타냅니다.

Windows Windows에서 해당 파일은 `MQ_INSTALLATION_PATH\tools\mqsc\samples` 디렉토리에 있습니다. `MQ_INSTALLATION_PATH`는 IBM MQ가 설치되어 있는 상위 레벨 디렉토리를 나타냅니다.

ALW runmqsc 에서 대화식으로 MQSC 명령 실행

AIX, Linux, and Windows에서는 **runmqsc** 명령 프롬프트를 사용하여 큐 관리자에 MQSC 명령을 대화식으로 실행할 수 있습니다. 대화식 실행은 빠른 테스트에 특히 적합합니다.

시작하기 전에

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

MQPROMPT 환경 변수를 사용하여 선택한 프롬프트를 설정하면 MQSC 환경에 있음을 보다 쉽게 확인하고 현재 환경의 일부 세부사항을 볼 수 있습니다. 자세한 정보는 19 페이지의 『MQSC 명령 프롬프트 설정』의 내용을 참조하십시오.

Linux **AIX** AIX and Linux 플랫폼에서 MQSC 명령을 대화식으로 실행하는 경우 **runmqsc** 명령 프롬프트는 추가 명령행 편집기 기능도 지원합니다. 20 페이지의 『runmqsc에 대해 명령 재호출 및 완료, Emacs 명령 키 사용』의 내용을 참조하십시오.

이 태스크 정보

runmqsc 명령은 MQSC 명령을 실행할 수 있는 명령 프롬프트를 여는 데 사용됩니다. 이러한 명령 및 해당 구문은 [MQSC 명령 참조](#)에 설명되어 있습니다.

이 태스크에 설명된 대로 **runmqsc** 명령 프롬프트를 시작할 때 명령에 설정된 플래그에 따라 세 가지 모드 중 하나로 실행되도록 프롬프트를 설정합니다.

- 확인 모드, 이 모드에서는 MQSC 명령이 로컬 큐 관리자에서 확인되지만 실행되지는 않습니다.
- 직접 모드, 이 모드에서는 MQSC 명령이 로컬 큐 관리자에서 실행됩니다.
- 간접 모드, 이 모드에서는 MQSC 명령이 리모트 큐 관리자에서 실행됩니다.

아래 프로시저는 프롬프트가 직접 모드에서 실행되도록 설정합니다. 기타 옵션은 기본 단계를 따르는 예제에 설명되어 있습니다.

프로시저

1. 명령 창 또는 셸을 열고 다음 명령을 입력하십시오.

```
runmqsc QMgrName
```

여기서 *QMgrName*은 MQSC 명령을 처리할 큐 관리자 이름을 지정합니다. *QMgrName*을 공백으로 두어 기본 큐 관리자에서 MQSC 명령을 처리할 수 있습니다.

2. 필요에 따라 MQSC 명령을 입력하십시오. 예를 들어, 로컬 큐 ORANGE.LOCAL.QUEUE를 작성하려면 다음 명령을 입력하십시오.

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

매개변수가 너무 많아 한 행에 맞출 수 없는 명령의 경우, 연속 문자를 사용하여 명령이 다음 행에서 계속된다는 것을 표시하십시오.

- 빼기 부호(-)는 다음 행의 시작에서 명령이 계속된다는 것을 표시합니다.
- 더하기 부호(+)는 다음 행의 첫 번째 비공백 문자에서 명령이 계속됨을 표시합니다.

명령 입력은 연속 문자가 아닌 공백이 아닌 행의 마지막 문자로 종료됩니다. 세미콜론(;)을 입력하여 명령 입력을 명확하게 종료할 수도 있습니다.

3. 다음 명령을 입력하여 MQSC 명령에 대한 작업을 중지하십시오.

```
end
```

또는 운영 체제에 **exit** 명령, **quit** 명령 또는 EOF 문자를 사용할 수 있습니다.

결과

MQSC 명령을 실행할 때, 큐 관리자는 조치를 확인하거나 사용자가 만든 오류에 대해 알려주는 운영자 메시지를 리턴합니다. 예를 들어, 다음 메시지는 큐가 작성되었음을 확인합니다.

```
AMQ8006: IBM MQ queue created.
```

다음 메시지는 구문 오류가 발생했음을 표시합니다.

```
AMQ8405: Syntax error detected at or near end of command segment below:-
AMQ8426: Valid MQSC commands are:
```

```
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
4 : end
```

이 메시지는 표준 출력 장치로 전송되며, 기본적으로 표시장치입니다. 명령을 올바르게 입력하지 않은 경우 명령에 대한 참조 정보를 참조하여 올바른 구문을 찾으십시오. [MQSC 명령 참조](#)를 참조하십시오.

예

다음은 이전 단계에서 사용된 `runmqsc QMgrName` 명령의 변형입니다. 이러한 변형은 `runmqsc` 명령 프롬프트의 다른 구성을 작성합니다.

- 다음 명령은 명령 필터링을 사용하여 단일 MQSC 명령을 MQSC 해석기에 전달합니다.

Windows의 경우:

```
echo display chstatus(*) | runmqsc QMname
```

Linux의 경우:

```
echo "display chstatus(*)" | runmqsc QMname
```

- 다음 명령은 큐 관리자 이름을 지정하지 않으므로 MQSC 명령은 기본 큐 관리자에서 처리됩니다.

```
runmqsc
```

- 이 명령은 명령을 제출하기 위해 QMLocal을 사용하여 QMRemote 큐 관리자에 명령을 제출합니다.

```
runmqsc -w 30 -m QMLocal QMRemote
```

- 이 명령은 명령을 실행하지 않고 로컬 큐 관리자에서 명령 구문이 올바른지 확인합니다. 확인할 명령은 입력 파일 `myprog.in`에서 읽습니다.

```
runmqsc -f myprog.in -v QmgrName
```

입력 및 출력 파일 작업에 대한 자세한 정보는 21 페이지의 『[runmqsc의 텍스트 파일에서 MQSC 명령 실행](#)』의 내용을 참조하십시오.

다음에 수행할 작업

`runmqsc` 명령 구문, 선택적 매개변수 및 리턴 코드에 대한 전체 세부사항은 [runmqsc \(MQSC 명령 실행\)](#)를 참조하십시오.

관련 태스크

21 페이지의 『[runmqsc의 텍스트 파일에서 MQSC 명령 실행](#)』

긴 명령이 있거나 특정 순서의 명령을 반복적으로 사용하는 경우 텍스트 파일을 사용하여 MQSC 명령을 실행할 수 있습니다. 텍스트 파일에서 `stdin`의 경로를 재지정할 수 있습니다. 또한 출력을 파일로 경로 재지정할 수도 있습니다.

관련 참조

[MQSC 명령 참조](#)

ALW MQSC 명령 프롬프트 설정

AIX, Linux, and Windows에서 **MQPROMPT** 환경 변수를 사용하여 **runmqsc** 명령을 실행할 때 표시되는 프롬프트를 설정하십시오. 이렇게 하면 사용자가 MQSC 환경에 있는지 확인하고 현재 환경에 대한 일부 세부사항을 보기가 쉬워집니다.

이 태스크 정보

runmqsc 명령을 실행할 때 표시되는 프롬프트를 설정할 수 있습니다. 프롬프트는 **runmqsc** 명령이 대화식으로 실행될 때와 입력이 파일 또는 표준 입력 디바이스 (stdin) 에서 **runmqsc** 로 경로 재지정될 때 모두 삽입됩니다.

명령 프롬프트에 일반 텍스트를 포함할 수 있으며 IBM MQ 서비스 오브젝트 정의와 동일한 방식으로 +VARNAME+ 표기법을 사용하여 환경 변수를 삽입할 수도 있습니다. 자세한 정보는 174 페이지의 『서비스 정의에서 대체 가능한 삽입 사용』의 내용을 참조하십시오.

다음 표에 설명된 대로 IBM MQ에서 제공하는 대체 가능한 추가 삽입이 몇 개 있습니다.

대체 가능 삽입	설명
MQ_HOST_NAME	시스템의 호스트 이름
MQ_FILE_SEP	플랫폼별 파일 구분 기호: <ul style="list-style-type: none"> Linux AIX AIX and Linux 시스템에서 MQ_FILE_SEP 는 /입니다. Windows Windows 시스템에서 MQ_FILE_SEP 의 위치는 \입니다.
MQ_PATH_SEP	플랫폼별 경로 구분 기호: <ul style="list-style-type: none"> Linux AIX AIX and Linux 시스템에서 MQ_PATH_SEP 는 :입니다. Windows Windows 시스템에서 MQ_PATH_SEP 의 위치는 ;입니다.
MQ_DATE_TIME	고정된 YYYY-MM-DD hh:mm:ss.SSS 형식의 로컬 시스템 날짜 및 시간입니다. 예를 들어, 다음과 같습니다. 2020-12-25 17:41:37.408

참고:

- MQ 대체 가능 삽입 값은 **runmqsc** 명령이 연관된 IBM MQ 설치 및 호스트 시스템과 관련됩니다.
- MQPROMPT** 는 삽입이 확장될 때 최대 256자로 제한됩니다. **MQPROMPT** 이 값을 확장하면 확장 없이 전체 **MQPROMPT** 문자열이 잘립니다.

예

다음 예제는 프롬프트를 MQSC로 설정합니다.

- Linux AIX

```
export MQPROMPT="MQSC"
```

- Windows

```
set "MQPROMPT=MQSC"
```

AIX 다음 예제는 AIX 시스템에서 **MQPROMPT** 변수를 설정합니다. 프롬프트는 사용자 이름 (연관된 시스템 환경 변수에서 가져옴), 큐 관리자 이름 및 IBM MQ 호스트 이름 (IBM MQ 대체 가능 삽입에서 가져옴) 을 표시하도록 설정됩니다.

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
C:\ > runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

다음 예제는 MQ 대체 가능 삽입에서 가져온 시간소인을 위의 **MQPROMPT** 예제에 추가합니다.

```
sh> export MQPROMPT="+MQ_DATE_TIME+ +USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
2020-11-24 18:10:00.404 myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=MQ_DATE_TIME+ +USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
C:\ > runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
2020-11-24 18:10:01.007 myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Linux **AIX** **runmqsc** 에 대해 명령 재호출 및 완료, Emacs 명령 키 사용

AIX and Linux 에서 **runmqsc** 명령 프롬프트를 사용하여 명령 재호출, 명령 완료 및 Emacs 명령 키를 사용으로 설정하십시오.

이 태스크 정보

AIX and Linux 시스템에서는 **runmqsc** 명령 프롬프트에서 다음과 같은 추가 명령행 편집기 기능을 사용할 수 있습니다.

- 위로 화살표 키와 아래로 화살표 키를 사용하여 이전에 입력한 명령을 다시 불러옵니다.
- Tab 키 및 스페이스바 키를 사용하여 명령의 다음 키워드 자동 완성
- Emacs 명령 키 또는 유사한 명령 키 기능

이러한 함수를 사용하려면 curses 라이브러리를 설치해야 합니다. curses 라이브러리가 시스템에 설치되지 않은 경우, **runmqsc** 프롬프트에는 명령행 편집기 기능이 없으며 **runmqsc** 명령 프롬프트가 시작될 때 메시지가 표시됩니다. 설치할 curses 라이브러리의 이름은 UNIX 플랫폼에 따라 다릅니다.

- **AIX** AIX에서 curses를 설치하십시오.
- **Linux** Linux에서 ncurses를 설치하십시오.

프로시저

- ncurses 또는 curses를 설치하십시오.

참고: 다음 예제에서는 Linux 에 대한 지시사항을 사용합니다.

다음 명령을 실행하여 기존 ncurses 패키지를 찾으십시오.

```
rpm -qa | grep -i ncurses
```

필수 ncurses 패키지는 다음과 같습니다.

```
ncurses-term-6.1-7.20180224.el8.noarch
ncurses-6.1-7.20180224.el8.x86_64
ncurses-base-6.1-7.20180224.el8.noarch
ncurses-c++-libs-6.1-7.20180224.el8.x86_64
ncurses-libs-6.1-7.20180224.el8.x86_64
ncurses-compat-libs-6.1-7.20180224.el8.x86_64
ncurses-devel-6.1-7.20180224.el8.x86_64
```

다음 명령을 실행하여 이전 텍스트에 나열된 모든 필수 ncurses 패키지를 설치할 수 있습니다.

```
yum install ncurses*
```

- Emacs 키 바인딩을 사용자 정의하십시오.

명령에 바인드되는 키를 사용자 정의할 수 있습니다. 예를 들어, 기본 Emacs 키 바인딩 대신 vi 바인딩에 키를 바인드할 수 있습니다.

홈 디렉토리에 저장된 `.editrc` 파일을 편집하여 키를 사용자 정의합니다. 자세한 정보는 FreeBSD man 페이지에서 `editrc`를 참조하십시오.

- 명령 재호출, 명령 완료 및 Emacs 명령 키를 사용 불가능하게 하십시오.

이를 수행하려면 환경 변수 `MQ_OVERRIDE_LIBEDIT_LOAD` 를 TRUE로 설정하십시오.

이 환경 변수는 `runmqsc` 명령 프롬프트가 다음 정보 메시지를 표시할 때 임시 해결책으로 사용할 수 있습니다.

```
AMQ8521I: Command completion and history unavailable
```

ALW runmqsc 의 텍스트 파일에서 MQSC 명령 실행

긴 명령이 있거나 특정 순서의 명령을 반복적으로 사용하는 경우 텍스트 파일을 사용하여 MQSC 명령을 실행할 수 있습니다. 텍스트 파일에서 `stdin`의 경로를 재지정할 수 있습니다. 또한 출력을 파일로 경로 재지정할 수도 있습니다.

시작하기 전에

이 태스크는 실행하려는 MQSC 명령을 포함하는 텍스트 파일을 작성했다고 가정합니다. 이러한 파일의 자세한 구문 및 예제는 14 페이지의 『MQSC 입력 파일 구문』의 내용을 참조하십시오.

MQPROMPT 환경 변수를 사용하여 MQSC 명령 프롬프트를 원하는 프롬프트로 설정할 수 있습니다. 자세한 정보는 19 페이지의 『MQSC 명령 프롬프트 설정』의 내용을 참조하십시오.

이 태스크 정보

`runmqsc` 명령의 입력은 표준 입력 디바이스(`stdin`이라고도 함)에서 가져옵니다. 일반적으로 이는 키보드이지만 직렬 포트 또는 파일에서 입력을 가져오도록 지정할 수 있습니다.

`runmqsc` 명령의 출력은 표준 출력 디바이스(`stdout`이라고도 함)에 대한 출력입니다. 일반적으로 이는 표시장치이지만, 직렬 포트 또는 파일로 출력의 경로를 재지정할 수 있습니다.

프로시저

1. 로컬 큐 관리자에서 명령을 실행하지 않고 파일의 명령 구문이 올바른지 확인하십시오.

다음 옵션 중 하나와 함께 `runmqsc` 명령에 `-v` 플래그를 사용하십시오.

- **-f** 옵션을 사용하여 입력 텍스트 파일 이름을 식별하십시오. 예를 들면, 다음과 같습니다.

```
runmqsc -f myprog.in -v localQmgrName
```

명령을 확인할 때 리모트 큐 관리자를 지정할 수 없습니다. 즉, **-w** 플래그를 지정할 수 없습니다.

리턴된 보고서는 [16 페이지의 그림 2](#)에 표시된 것과 유사합니다.

2. 명령 구문이 올바르면 **-v** 플래그를 제거한 다음 **runmqsc** 명령을 다시 실행하십시오.

이제 리모트 큐 관리자를 지정할 수 있습니다.

- 예를 들어, 다음 명령을 실행하십시오.

```
runmqsc -f myprog.in QmgrName
```

[16 페이지의 그림 1](#)에서는 명령 파일(예: myprog.in)의 추출을 표시하고 [16 페이지의 그림 2](#)에서는 보고서 파일(예: results.out)의 해당 출력 추출을 표시합니다.

다음에 수행할 작업

runmqsc 명령 구문, 선택적 매개변수 및 리턴 코드에 대한 전체 세부사항은 [runmqsc \(MQSC 명령 실행\)](#)를 참조하십시오.

관련 태스크

[19 페이지의 『MQSC 명령 프롬프트 설정』](#)

AIX, Linux, and Windows에서 **MQPROMPT** 환경 변수를 사용하여 **runmqsc** 명령을 실행할 때 표시되는 프롬프트를 설정하십시오. 이렇게 하면 사용자가 MQSC 환경에 있는지 확인하고 현재 환경에 대한 일부 세부사항을 보기가 쉬워집니다.

[16 페이지의 『runmqsc 에서 대화식으로 MQSC 명령 실행』](#)

AIX, Linux, and Windows에서는 **runmqsc** 명령 프롬프트를 사용하여 큐 관리자에 MQSC 명령을 대화식으로 실행할 수 있습니다. 대화식 실행은 빠른 테스트에 특히 적합합니다.

관련 참조

[MQSC 명령 참조](#)

Multi 시동 시 MQSC 스크립트에서 자동 구성

모든 큐 관리자 시작 시 MQSC 스크립트 또는 MQSC 스크립트 세트의 콘텐츠를 자동으로 적용하도록 큐 관리자를 구성할 수 있습니다.

이 기능을 사용하면 수정되고 다음 큐 관리자 재시작 시 자동으로 재실행될 수 있는 구성을 보유할 수 있습니다. 예를 들어, 스크립트가 마운트된 드라이브에 있는 경우 시작 시 모든 큐 관리자에 최신 버전이 적용되는 중앙 집중식 구성을 보유할 수 있습니다.

이 기능이 유용할 수 있는 특정 시나리오는 모두 적용되는 단일 구성 세트를 보유하여 균등 클러스터가 클러스터 내 모든 큐 관리자에 대한 동일한 정의를 포함하도록 하는 것입니다. 예를 들어, [새 균일한 클러스터 작성](#)을 참조하십시오.

시작하기 전에

다음을 사용할 수 있습니다.

1. 단일 스크립트. 그리고 MQSC 명령을 사용하여 텍스트 파일을 작성하십시오.
2. MQSC 스크립트 세트:
 - 구성이 존재하는 디렉토리를 식별합니다.
 - 해당 디렉토리에 각각 확장자가 .mqsc인 파일(예: queues.mqsc)을 작성합니다.

이 스크립트는 큐 관리자가 시작될 때마다 다시 적용된다는 점을 고려하면 명령이 재실행될 수 있는 것이 중요합니다. 예를 들어, **DEFINE** 명령은 **REPLACE** 문자열을 포함해야 합니다. 그렇지 않으면 오브젝트가 이미 존재하기 때문에 두 번째 큐 관리자가 시작할 때 실패로 표시됩니다.

MQSC 스크립트에서 * 접두부가 있는 행은 주석으로 처리됨에 유의하십시오.

MQSC 스크립트의 자동 구성 사용

중요사항: MQTT 유형의 채널에 대한 명령은 시작 중에 자동 구성에 대해 지원되지 않으므로 실행하지 않아야 합니다.

crtmqm 명령에 **-ic** 플래그를 사용하고 특정 파일 또는 디렉토리를 지정하여 새 큐 관리자를 구성할 수 있습니다. 제공된 값은 AutoConfig 스탠자 아래 **qm.ini** 파일에 **MQSCConfig** 속성으로 저장됩니다.

AutoConfig 스탠자 속성인 **MQSCConfig**를 추가하고 올바른 파일 또는 디렉토리를 지정하여 자동 MQSC 구성을 사용하도록 기존 큐 관리자를 구성할 수 있습니다. 예를 들면, 다음과 같습니다.

```
AutoConfig:
MQSCConfig=C:\mq_configuration\uniclus.mqsc
```

자동 구성 작동 방식

큐 관리자 시동 중에 **AutoConfig** 스탠자 속성 **MQSCConfig**로 식별되는 구성은 올바른 구문을 확인하기 위해 **runmqsc** 유효성 검증을 통해 전달된 후 큐 관리자 데이터 트리에서 **autocfg** 디렉토리에 단일 파일 **cached.mqsc**로 저장됩니다.

디렉토리의 다중 파일이 처리될 때 알파벳순으로 처리되며 MQSC end 또는 quit 명령을 포함하는 경우 해당 파일의 나머지 콘텐츠를 건너뜁니다.

처음으로 큐 관리자를 시작하는 동안 파일 또는 디렉토리를 읽을 수 없거나 파일에 올바르지 않은 MQSC 구문이 있으면 큐 관리자가 시작되지 않고 콘솔 및 큐 관리자 오류 로그 모두에 적절한 오류 메시지가 기록됩니다.

후속 재시작 시 가리키는 파일 또는 디렉토리를 읽을 수 없거나 올바르지 않은 MQSC 구문이 포함된 경우 이전에 캐시된 파일이 사용되고 큐 관리자의 오류 로그에 기록된 메시지가 이를 강조표시합니다.

cached.mqsc의 콘텐츠가 큐 관리자에 적용되는 시점에서 모든 MQSC 명령이 적용되면 연결할 애플리케이션에 대해 큐 관리자를 사용할 수 있습니다. 적용되는 구성의 **runmqsc** 로그는 큐 관리자의 오류 디렉토리에 **autocfgmqsc.LOG**라고 하는 파일로 저장됩니다.

또한 완료되지 않은 MQSC 명령이 명령 실패 이유를 식별하여 큐 관리자 오류 로그에 기록됩니다.

PCF 명령을 사용하여 IBM MQ 관리 자동화

일부 관리 및 모니터링 태스크를 자동화하면 설치에 유익할 것이라고 결정할 수 있습니다. 프로그래밍 가능 명령 형식(PCF) 명령을 사용하여 로컬 및 리모트 큐 관리자 모두의 관리 태스크를 자동화할 수 있습니다. 이 절에서는 사용자가 IBM MQ 오브젝트를 관리해 본 적이 있다고 가정합니다.

PCF 명령

IBM MQ 프로그래밍 가능 명령 형식(PCF) 명령이 관리 프로그램에서 프로그램 관리 태스크에 사용될 수 있습니다. 이 방식으로 프로그램으로부터 큐 관리자 오브젝트(큐, 프로세스 정의, 이름 리스트, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트)를 조작하고, 심지어 큐 관리자 자체를 조작할 수도 있습니다.

PCF 명령은 MQSC 명령에서 제공하는 동일한 범위의 기능을 포함합니다. 단일 노드에서 네트워크의 임의의 큐 관리자에 PCF 명령을 발행하는 프로그램을 작성할 수 있습니다. 이러한 방법으로 관리 태스크를 집중시키고 자동화할 수 있습니다.

각 PCF 명령은 IBM MQ 메시지의 애플리케이션 데이터 부분에 임베드되는 데이터 구조입니다. 각 명령은 다른 메시지와 동일한 방식으로 MQI 기능인 MQPUT을 사용하여 대상 큐 관리자로 송신됩니다. 명령 서버가 메시지를 수신하는 큐 관리자에서 실행 중인 경우, 명령 서버는 메시지를 명령 메시지로 해석하고 명령을 실행합니다. 응답을 가져오기 위해 애플리케이션은 MQGET 호출을 실행하고 응답 데이터는 다른 데이터 구조로 리턴됩니다. 그런 다음, 애플리케이션은 응답을 처리하고 그에 따라 수행합니다.

참고: MQSC 명령과는 다르게 PCF 명령 및 그에 대한 응답은 사용자가 읽을 수 있는 텍스트 형식이 아닙니다.

다음은 PCF 명령 메시지를 작성하는 데 필요한 항목입니다.

메시지 디스크립터

이는 표준 IBM MQ 메시지 디스크립터입니다.

- 메시지 유형(*MsgType*)은 MQMT_REQUEST입니다.
- 메시지 형식(*Format*)은 MQFMT_ADMIN입니다.

애플리케이션 데이터

PCF 헤더를 포함하는 PCF 메시지가 들어 있습니다.

- PCF 메시지 유형(*Type*)은 MQCFT_COMMAND를 지정합니다.
- 명령 ID는 명령(예: *Change Queue*(MQCMD_CHANGE_Q))을 지정합니다.

PCF 데이터 구조와 구현 방법에 대한 자세한 설명은 24 페이지의 『IBM MQ PCF(Programmable Command Format) 소개』의 내용을 참조하십시오.

PCF 오브젝트 속성

PCF의 오브젝트 속성은 MQSC 명령에 대한 문자 수가 8자로 제한되지 않습니다. 이는 이 안내서에서 이탤릭체로 표시됩니다. 예를 들어, RQMNAME의 PCF 등가물은 *RemoteQMgrName*입니다.

PCF 나가기

PCF 나가는 메시지 텍스트 내에 MQSC 명령을 포함하는 PCF 명령입니다. PCF를 사용하여 리모트 큐 관리자로 명령을 송신할 수 있습니다. PCF 나가기에 대한 자세한 정보는 [나가기](#)를 참조하십시오.

IBM MQ PCF(Programmable Command Format) 소개

프로그래밍 가능 명령 형식(PCF)는 네트워크의 큐 관리자(PCF 지원) 및 프로그램 사이에서 교환할 수 있는 명령 및 응답 메시지를 정의합니다. PCF를 사용하면 큐 관리자 관리와 다른 네트워크 관리를 간단하게 수행할 수 있습니다. 특히 네트워크가 규모면에서 커진 것처럼 분산 네트워크 및 복잡도의 복합적 관리의 문제를 해결할 때 사용할 수 있습니다.

PCF (Programmable Command Format) 는 모든 IBM MQ 플랫폼에서 지원됩니다.

문제점 PCF 명령 해결

분배된 네트워크의 관리가 복잡해질 수 있습니다. 관리의 문제점은 네트워크의 크기 및 복잡도가 증가하기 때문에 계속 증가합니다.

메시징 및 큐잉에 특정한 관리 예에는 다음이 포함됩니다.

- 자원 관리.
예: 큐 작성 및 삭제.
- 성능 모니터링.
예를 들어, 최대 큐 용량 또는 메시지 비율.
- 제어
예를 들어, 최대 큐 용량, 최대 메시지 길이와 같은 큐 매개변수 성능 조정 및 큐 사용 및 사용 안함.
- 메시지 라우팅.
대체의 정의가 네트워크를 통해 라우팅됩니다.

IBM MQ PCF 명령은 큐 관리자 관리 및 기타 네트워크 관리를 단순화하기 위해 사용될 수 있습니다. PCF 명령을 사용하면 네트워크 내의 단일 큐 관리자에서 네트워크 관리를 수행하기 위해 단일 애플리케이션을 사용할 수 있습니다.

PCF의 개념

PCF는 네트워크에서 PCF를 지원하는 큐 관리자와 프로그램 사이에 교환될 수 있는 명령 및 응답 메시지를 정의합니다. IBM MQ 오브젝트(인증 정보 오브젝트, 채널, 채널 리스너, 이름 목록, 프로세스 정의, 큐 관리자, 큐, 서버

스 및 스토리지 클래스)의 관리를 위해 시스템 관리 애플리케이션 프로그램에서 PCF 명령을 사용할 수 있습니다. 애플리케이션은 네트워크의 단일 지점에서 작동하여 로컬 큐 관리자를 사용하여 로컬 또는 원격으로 모든 큐 관리자와 명령 및 응답 정보를 통신할 수 있습니다.

각 큐 관리자에는 표준 큐 이름을 가지는 관리 큐가 있고 애플리케이션은 해당 큐로 PCF 명령 메시지를 보낼 수 있습니다. 각 큐 관리자에는 관리 큐에서 명령 메시지를 제공할 명령 서버도 있습니다. 따라서, PCF 명령 메시지가 네트워크에서 큐 관리자에 의해 처리될 수 있고 응답 데이터는 지정된 응답 큐를 사용하여 애플리케이션으로 리턴될 수 있습니다. PCF 명령 및 응답 메시지가 정상 메시지 큐 인터페이스(MQI)를 사용하여 보내지고 수신됩니다.

해당 매개변수를 포함하여 사용 가능한 PCF 명령 목록의 경우 PCF(Programmable Command Format)의 정의를 참조하십시오.

IBM MQ PCF(Programmable Command Format) 사용

IBM MQ 원격 관리를 위해 시스템 관리 프로그램에서 pcf를 사용할 수 있습니다.

이 섹션에는 다음이 포함됩니다.

- 25 페이지의 『PCF 명령 메시지』
- 27 페이지의 『IBM MQ의 PCF 응답』
- **z/OS** 29 페이지의 『Extended responses』
- IBM MQ 오브젝트 이름 지정 규칙
- 31 페이지의 『IBM MQ의 PCF 명령에 대한 권한 검사』

PCF 명령 메시지

PCF 명령 메시지는 해당 헤더 및 사용자 정의 메시지 데이터에서 식별되는 PCF 헤더, 매개변수로 구성됩니다. 메시지는 메시지 큐 인터페이스 호출을 사용하여 실행됩니다.

각 명령 및 해당 매개변수는 수많은 매개변수 구조가 뒤에 오는 PCF 헤더를 포함하는 분리 명령 메시지로서 보내집니다. PCF 헤더의 세부사항의 경우, MQCFH - PCF 헤더를 참조하고 매개변수 구조의 예는 MQCFST - PCF 문자열 매개변수를 참조하십시오. PCF 헤더는 동일한 메시지에서 수행하는 매개변수 구조의 수와 명령을 식별합니다. 각 매개변수 구조에서는 명령에 매개변수를 제공합니다.

명령 서버에서 생성되는 명령에 대한 응답에 유사한 구조가 있습니다. 매개변수 구조의 수가 뒤에 오는 PCF 헤더가 있습니다. 응답은 둘 이상의 메시지로 구성되지만 명령은 항상 하나의 메시지로만 구성됩니다.

Multi 멀티플랫폼에서 PCF 명령이 송신되는 큐를 항상 SYSTEM.ADMIN.COMMAND.QUEUE입니다.

z/OS z/OS에서는 명령이 SYSTEM.COMMAND.INPUT, SYSTEM.ADMIN.COMMAND.QUEUE 는 별명일 수 있습니다. 이 큐에 서비스를 제공하는 명령 서버에서 명령 메시지의 메시지 디스크립터에 있는 ReplyToQ 및 ReplyToQMgr 필드를 통해 정의된 큐에 응답을 전송합니다.

PCF 명령 메시지 발행 방법

정상 메시지 큐 인터페이스(MQI) 호출, MQPUT, MQGET 등을 사용하여 해당 큐로 PCF 명령 및 응답 메시지를 배치시키고 검색하십시오.

참고:

명령 서버가 해당 큐 관리자에서 처리할 PCF 명령에 대한 대상 큐 관리자에서 실행 중인지 확인하십시오.

제공된 헤더 파일 목록의 경우, IBM MQ COPY, 헤더, 포함 및 모듈 파일을 참조하십시오.

PCF 명령에 대한 메시지 디스크립터

IBM MQ 메시지 디스크립터는 MQMD - 메시지 디스크립터에서 완전하게 문서화됩니다.

PCF 명령 메시지는 메시지 디스크립터에 다음 필드를 포함합니다.

보고서

필요한 경우, 임의의 올바른 값.

MsgType

이 필드는 응답을 요청하는 메시지를 표시하기 위한 MQMT_REQUEST여야 합니다.

만기

필요한 경우, 임의의 올바른 값.

Feedback

MQFB_NONE으로 설정하십시오.

Multi Encoding

IBM MQ for Multiplatforms 시스템으로 보내는 경우 이 필드를 메시지 데이터에 사용되는 인코딩으로 설정하십시오. 필요한 경우 변환이 수행됩니다.

Multi CodedCharSetId

IBM MQ for Multiplatforms 시스템으로 보내는 경우 이 필드를 메시지 데이터에 사용되는 코드화된 문자 세트 식별자로 설정합니다. 필요한 경우 변환이 수행됩니다.

형식

MQFMT_ADMIN으로 설정하십시오.

Priority

필요한 경우, 임의의 올바른 값.

Persistence

필요한 경우, 임의의 올바른 값.

MsgId

전송 애플리케이션은 값을 지정할 수 있거나 MQMI_NONE은 고유한 메시지 ID를 생성하기 위한 큐 관리자를 요청하기 위해 지정될 수 있습니다.

CorrelId

전송 애플리케이션은 값을 지정할 수 있거나 MQCI_NONE은 상관 ID를 표시하지 않도록 지정될 수 있습니다.

ReplyToQ

응답을 수신할 큐의 이름.

ReplyToQMGr

응답에 대한 큐 관리자의 이름(또는 공백).

메시지 컨텍스트 필드

필요한 경우, 이러한 필드는 임의의 올바른 값으로 설정될 수 있습니다. 일반적으로 Put 메시지 옵션 MQPMO_DEFAULT_CONTEXT는 기본값에 메시지 컨텍스트 필드를 설정하기 위해 사용됩니다.

버전-2 MQMD 구조를 사용 중인 경우, 다음 추가 필드를 설정해야 합니다.

GroupId

MQGI_NONE으로 설정

MsgSeqNumber

1로 설정

오프셋

0으로 설정

MsgFlags

MQMF_NONE으로 설정

OriginalLength

MQOL_UNDEFINED으로 설정

사용자 데이터 송신

PCF 구조는 사용자 정의되는 메시지 데이터를 보내기 위해 사용될 수도 있습니다. 이 경우, 메시지 디스크립터 *Format* 필드는 MQFMT_PCF로 설정되어야 합니다.

지정된 큐에서 PCF 메시지 송신 및 수신

PCF 메시지를 지정된 큐에 보내기

지정된 큐로 메시지를 송신하기 위해 mqPutBag 호출은 지정된 백의 내용을 PCF 메시지로 변환하며 메시지를 지정된 큐로 송신합니다. 호출 후에는 백의 콘텐츠가 변경되지 않습니다.

이 호출에 대한 입력으로 다음을 제공해야 합니다.

- MQI 연결 핸들.
- 메시지가 놓일 큐에 대한 오브젝트 핸들.
- 메시지 디스크립터. 메시지 디스크립터에 대한 자세한 정보는 [MQMD - 메시지 디스크립터](#)의 내용을 참조하십시오.
- MQPMO 구조를 사용하여 메시지 넣기 옵션. MQPMO 구조에 대한 자세한 정보는 [MQPMO - 메시지 넣기 옵션](#)의 내용을 참조하십시오.
- 메시지로 변환될 백의 핸들.

참고: 백에 관리 메시지가 있으며 mqAddInquiry 호출이 값을 백에 삽입하는 데 사용된 경우, MQIASY_COMMAND 데이터 항목의 값은 MQAI가 인식하는 INQUIRE 명령이어야 합니다.

mqPutBag 호출에 대한 전체 설명은 [mqPutBag](#)을 참조하십시오.

지정된 큐로부터 PCF 메시지를 수신

지정된 큐로부터 메시지를 수신하기 위해 mqGetBag 호출은 지정된 큐로부터 PCF 메시지를 가져와서 메시지 데이터를 데이터 백으로 변환합니다.

이 호출에 대한 입력으로 다음을 제공해야 합니다.

- MQI 연결 핸들.
- 메시지가 놓일 큐에 대한 오브젝트 핸들.
- 메시지 디스크립터. MQMD 구조 내에 **Format** 매개변수는 MQFMT_ADMIN, MQFMT_EVENT 또는 MQFMT_PCF여야 합니다.

참고: 메시지가 작업 단위 내에서 수신되고(즉, MQGMO_SYNCPOINT 옵션과 함께) 메시지에 지원되지 않은 형식이 있는 경우, 작업 단위가 백아웃될 수 있습니다. 그런 다음, 메시지는 큐에서 복원되고 mpGetBag 호출 대신 MQGET 호출을 사용하여 검색할 수 있습니다. 메시지 디스크립터에 대한 자세한 정보는 [MQGMO - 메시지 가져오기 옵션](#)의 내용을 참조하십시오.

- MQGMO 구조를 사용하여 메시지 가져오기 옵션. MQGMO 구조에 대한 자세한 정보는 [MQMD - 메시지 디스크립터](#)의 내용을 참조하십시오.
- 변환된 메시지가 포함될 백의 핸들.

mqGetBag 호출에 대한 전체 설명은 [mqGetBag](#)을 참조하십시오.

IBM MQ의 PCF 응답

각 명령에 따라, 명령 서버는 하나 이상의 응답 메시지를 생성합니다. 응답 메시지에는 명령 메시지와 유사한 형식을 가지고 있습니다.

PCF 헤더에는 응답인 명령과 동일한 명령 ID 값이 있습니다(자세한 내용은 [MQCFH - PCF 헤더](#) 참조). 메시지 ID 및 상관 ID는 요청의 보고서 옵션에 따라 설정합니다.

명령 메시지의 PCF 헤더 유형이 MQCFT_COMMAND인 경우, 표준 응답만 생성됩니다. 이러한 명령은 멀티플랫폼에서만 지원됩니다. 이전 애플리케이션은 z/OS 에서 PCF를 지원하지 않습니다. IBM MQ Windows Explorer는 이러한 애플리케이션 중 하나입니다. 그러나 IBM WebSphere® MQ 6.0 이상의 IBM MQ Explorer는 z/OS 에서 PCF를 지원합니다.

명령 메시지의 PCF 헤더 유형이 MQCFT_COMMAND_XR인 경우, 확장된 응답 또는 표준 응답이 생성됩니다. 이러한 명령은 z/OS 및 일부 멀티플랫폼에서 지원됩니다. z/OS 에서 실행된 명령은 확장된 응답만 생성합니다.

단일 명령이 일반 오브젝트 이름을 지정하면, 자체 메시지에 일치하는 각 오브젝트에 대한 분리 응답이 리턴됩니다. 응답 발생의 경우, 일반 이름을 가진 단일 명령은 다중 개인 명령으로 처리됩니다(제어 필드 MQCFC_LAST 또는 MQCFC_NOT_LAST 제외). 그렇지 않으면, 하나의 명령 메시지는 하나의 응답 메시지를 생성합니다.

특정 PCF 응답은 요청되지 않은 경우에도 구조를 리턴할 수 있습니다. 이 구조는 응답의 정의 (PCF (Programmable Command Format)의 정의)에 표시됩니다. 항상 리턴됨으로 표시됩니다. 그 이유는 이러한 응답의 경우에 데이터가 적용되는 오브젝트를 식별하려면 응답에 오브젝트의 이름을 지정해야 하기 때문입니다.

응답을 위한 메시지 디스크립터

응답 메시지는 메시지 디스크립터에 다음 필드를 가집니다.

MsgType

이 필드는 MQMT_REPLY입니다.

MsgId

이 필드는 큐 관리자에 의해 생성됩니다.

CorrelId

이 필드는 명령 메시지의 보고서 옵션에 따라 생성됩니다.

형식

이 필드는 MQFMT_ADMIN입니다.

Encoding

MQENC_NATIVE로 설정하십시오.

CodedCharSetId

MQCCSI_Q_MGR로 설정하십시오.

Persistence

명령 메시지의 경우와 동일합니다.

Priority

명령 메시지의 경우와 동일합니다.

응답은 MQPMO_PASS_IDENTITY_CONTEXT로 생성됩니다.

Multi 표준 응답

MQCFT_COMMAND, 표준 응답의 헤더 유형을 가진 명령 메시지가 생성됩니다. 이러한 명령은 멀티플랫폼에서만 지원됩니다.

세 가지 종류의 표준 응답이 있습니다.

- 확인 응답
- 오류 응답
- 데이터 응답

확인 응답

이 응답은 MQCC_OK 또는 MQCC_WARNING의 *CompCode* 필드로, 명령어 형식 헤더로 시작하는 메시지로 구성됩니다.

MQCC_OK의 경우, *Reason*은 MQRC_NONE입니다.

MQCC_WARNING의 경우, *Reason*은 경고의 성질을 식별합니다. 이 경우에 명령 형식 헤더는 이 이유 코드에 적절한 하나 이상의 경고 매개변수 구조가 이어질 수도 있습니다.

이 경우, 조회 명령의 경우 추가 매개변수 구조가 다음 섹션에서 설명되는 대로 뒤에 옵니다.

오류 응답

명령에 오류가 있으면 하나 이상의 오류 응답 메시지가 송신됩니다. 일반적으로 하나의 응답 메시지만 있는 명령에 대해서도 둘 이상의 응답 메시지가 송신될 수 있습니다. 이러한 오류 응답 메시지의 경우, MQCFC_LAST 또는 MQCFC_NOT_LAST를 적절히 설정해야 합니다.

그러한 각 메시지는 특정 오류를 식별하는 MQCC_FAILED 및 Reason 필드의 CompCode 값으로 응답 형식 헤더로 시작합니다. 일반적으로, 각 메시지는 다른 오류를 설명합니다. 또한 각 메시지에는 헤더 뒤에 0 또는 하나(그 이상은 불가능)의 오류 매개변수 구조가 있습니다. 이 매개변수 구조는(하나가 있는 경우) 다음 중 하나를 포함하는 Parameter 필드를 가지는 MQCFIN 구조입니다.

- MQIACF_PARAMETER_ID

구조의 Value 필드는 오류가 있는 매개변수의 매개변수 ID입니다(예: MQCA_Q_NAME).

- MQIACF_ERROR_ID

이 값은 MQRC_UNEXPECTED_ERROR의 Reason 값(명령 형식 헤더와 함께) 사용됩니다. MQCFIN 구조의 Value 필드는 명령 서버에서 수신한 예상치 못한 이유 코드입니다.

- MQIACF_SELECTOR

명령과 함께 송신된 목록 구조(MQCFIL)에 복제 선택자 또는 올바르지 않은 것이 포함되는 경우 이 값이 발생합니다. 명령 형식 헤더에서 Reason 필드는 오류를 식별하고 MQCFIN 구조에서 Value 필드가 오류가 있는 명령의 MQCFIL 구조의 매개변수 값입니다.

- MQIACF_ERROR_OFFSET

이 값은 Ping 채널 명령에 데이터 비교 오류가 있을 때 발생합니다. 구조의 Value 필드는 Ping 채널 비교 오류의 오프셋입니다.

- MQIA_CODED_CHAR_SET_ID

이 값은 수신 PCF 명령 메시지의 메시지 디스크립터에 있는 코드화 문자 세트 ID가 대상 큐 관리자의 코드화 문자 세트 ID와 일치하지 않을 때 발생합니다. 구조의 Value 필드는 큐 관리자의 코드화 문자 세트 ID입니다.

마지막(또는 유일한) 오류 응답 메시지는 MQCC_FAILED의 CompCode 필드 및 MQRCCF_COMMAND_FAILED의 Reason 필드를 가지는 요약 응답입니다. 이 메시지에는 헤더를 따르는 매개변수 구조가 없습니다.

데이터 응답

이 응답은 조회 명령에 대한(이전에 설명된 것처럼) 확인 응답으로 구성됩니다. 확인 응답은 PCF(Programmable Command Format)의 정의에 설명된 대로 요청된 데이터를 포함하는 추가 구조가 이어집니다.

애플리케이션은 특정 순서로 리턴되는 이러한 추가 매개변수 구조에 따라 다르지 않아야 합니다.

Extended responses

Commands issued on z/OS generate extended responses.

There are three types of extended response:

- Message response, with type MQCFT_XR_MSG
- Item response, with type MQCFT_XR_ITEM
- Summary response, with type MQCFT_XR_SUMMARY

Each command can generate one, or more, sets of responses. Each set of responses comprises one or more messages, numbered sequentially from 1 in the *MsgSeqNumber* field of the PCF header. The *Control* field of the last (or only) response in each set has the value MQCFC_LAST. For all other responses in the set, this value is MQCFC_NOT_LAST.

Any response can include one, or more, optional MQCFBS structures in which the *Parameter* field is set to MQBACF_RESPONSE_SET, the value being a response set identifier. Identifiers are unique and identify the set of responses which contain the response. For every set of responses, there is an MQCFBS structure that identifies it.

Extended responses have at least two parameter structures:

- An MQCFBS structure with the *Parameter* field set to MQBACF_RESPONSE_ID. The value in this field is the identifier of the set of responses to which the response belongs. The identifier in the first set is arbitrary. In subsequent sets, the identifier is one previously notified in an MQBACF_RESPONSE_SET structure.

- An MQCFST structure with the *Parameter* field set to MQCACF_RESPONSE_Q_MGR_NAME, the value being the name of the queue manager from which the set of responses come.

Many responses have additional parameter structures, and these structures are described in the following sections.

You cannot determine in advance how many responses there are in a set other than by getting responses until one with MQCFC_LAST is found. Neither can you determine in advance how many sets of responses there are as any set might include MQBACF_RESPONSE_SET structures to indicate that additional sets are generated.

Extended responses to Inquire commands

Inquire commands normally generate an item response (type MQCFT_XR_ITEM) for each item found that matches the specified search criteria. The item response has a *CompCode* field in the header with a value of MQCC_OK, and a *Reason* field with a value of MQRC_NONE. It also includes other parameter structures describing the item and its requested attributes, as described in [Definitions of the Programmable Command Formats](#).

If an item is in error, the *CompCode* field in the header has a value of MQCC_FAILED and the *Reason* field identifies the particular error. Additional parameter structures are included to identify the item.

Certain Inquire commands might return general (not name-specific) message responses in addition to the item responses. These responses are informational, or error, responses of the type MQCFT_XR_MSG.

If the Inquire command succeeds, there might, optionally, be a summary response (type MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_OK, and a *Reason* field value of MQRC_NONE.

If the Inquire command fails, item responses might be returned, and there might optionally be a summary response (type MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

Extended responses to commands other than Inquire

Successful commands generate message responses in which the *CompCode* field in the header has a value of MQCC_OK, and the *Reason* field has a value of MQRC_NONE. There is always at least one message; it might be informational (MQCFT_XR_MSG) or a summary (MQCFT_XR_SUMMARY). There might optionally be additional informational (type MQCFT_XR_MSG) messages. Each informational message might include a number of additional parameter structures with information about the command; see the individual command descriptions for the structures that can occur.

Commands that fail generate error message responses (type MQCFT_XR_MSG), in which the *CompCode* field in the header has a value of MQCC_FAILED and the *Reason* field identifies the particular error. Each message might include a number of additional parameter structures with information about the error: see the individual error descriptions for the structures that can occur. Informational message responses might be generated. There might, optionally, be a summary response (MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

Extended responses to commands using CommandScope

If a command uses the **CommandScope** parameter, or causes a command using the **CommandScope** parameter to be generated, there is an initial response set from the queue manager where the command was received. Then a separate set, or sets, of responses is generated for each queue manager to which the command is directed (as if multiple individual commands were issued). Finally, there is a response set from the receiving queue manager which includes an overall summary response (type MQCFT_XR_SUMMARY). The MQCACF_RESPONSE_Q_MGR_NAME parameter structure identifies the queue manager that generates each set.


The initial response set has the following additional parameter structures:

- MQIACF_COMMAND_INFO (MQCFIN). Possible values in this structure are MQCMDI_CMDSCOPE_ACCEPTED or MQCMDI_CMDSCOPE_GENERATED.
- MQIACF_CMDSCOPE_Q_MGR_COUNT (MQCFIN). This structure indicates the number of queue managers to which the command is sent.

IBM MQ의 PCF 명령에 대한 권한 검사

PCF 명령이 처리될 때, 명령 메시지에 있는 메시지 디스크립터의 *UserIdentifier*가 필수 IBM MQ 오브젝트 권한 검사에 사용됩니다. 권한 검사는 이 주제에서 설명된 대로 각 플랫폼에 다르게 구현됩니다.

검사는 명령이 처리되는 시스템에서 수행되므로 이 사용자 ID는 반드시 대상 시스템에 존재해야 하며 명령 처리에 필요한 권한을 갖고 있어야 합니다. 메시지가 원격 시스템에서 발생한 경우, 대상 시스템에 기존 ID를 달성하는 한 방법은 로컬 시스템과 원격 시스템 모두에 대해 일치하는 사용자 ID를 가지고 있는 것입니다.

참고:  z/OS에 대한 권한 검사와 관련된 정보는 [태스크 1: z/OS 시스템 매개변수 식별을 참조하십시오](#)를 참조하십시오.

IBM MQ for IBM i



PCF 명령을 처리하려면 사용자 ID에 대상 시스템의 IBM MQ 오브젝트에 대한 *dsp* 권한이 있어야 합니다.

또한 IBM MQ 오브젝트 권한 검사는 [32 페이지의 표 2](#)에 나타난 바와 같이, 특정 PCF 명령에 대하여 실행됩니다.

대부분의 경우에 이러한 검사는 로컬 시스템에 실행된 동등한 IBM MQ CL 명령에 의해 수행된 해당 검사와 같은 검사입니다. IBM MQ 권한에서 IBM i 시스템 권한으로의 �핑 및 IBM MQ CL 명령에 대한 권한 요구사항에 대한 자세한 정보는 IBM i에서 [보안 설정](#)을 참조하십시오. 종료와 관계가 있는 보안에 대한 세부사항은 [보안 엑시트](#)를 사용하는 링크 레벨 보안 문서에 제공됩니다.

다음 명령 중 하나를 처리하려면 사용자 ID가 그룹 프로파일 QMQMADM의 구성원이어야 합니다.

- 채널 ping
- 채널 변경
- 채널 복사
- 채널 작성
- 채널 삭제
- 채널 재설정
- 채널 분석
- 채널 시작
- 채널 중지
- 채널 시작기 시작
- 채널 리스너 시작

IBM MQ for UNIX, Linux, and Windows



PCF 명령을 처리하기 위해 사용자 ID에는 대상 시스템에 큐 관리자 오브젝트에 대한 *dsp* 권한이 있어야 합니다. 또한 IBM MQ 오브젝트 권한 검사는 [32 페이지의 표 2](#)에 나타난 바와 같이, 특정 PCF 명령에 대하여 실행됩니다.

다음 명령 중 하나를 처리하려면 사용자 ID가 그룹 *mqm*에 속해야 합니다.

참고: Windows 전용의 경우, 사용자 ID는 관리자 또는 그룹 *mqm*에 속할 수 있습니다.

- 채널 변경
- 채널 복사
- 채널 작성

- 채널 삭제
- 채널 ping
- 채널 재설정
- 채널 시작
- 채널 중지
- 채널 시작기 시작
- 채널 리스너 시작
- 채널 분석
- 클러스터 재설정
- 클러스터 새로 고치기
- 큐 관리자 일시중단
- 큐 관리자 재개

멀티플랫폼용 IBM MQ 오브젝트 권한

Multi

표 2. 오브젝트 권한		
명령	IBM MQ 오브젝트 권한	클래스 권한(오브젝트 유형의 경우)
인증 정보 변경	dsp 및 chg	해당사항 없음
채널 변경	dsp 및 chg	해당사항 없음
채널 리스너 변경	dsp 및 chg	해당사항 없음
클라이언트 연결 채널 변경	dsp 및 chg	해당사항 없음
이름 목록 변경	dsp 및 chg	해당사항 없음
프로세스 변경	dsp 및 chg	해당사항 없음
큐 변경	dsp 및 chg	해당사항 없음
큐 관리자 변경	chg 참고 3 및 참고 5 참조	해당사항 없음
서비스 변경	dsp 및 chg	해당사항 없음
큐 지우기	clr	해당사항 없음
인증 정보 복사	dsp	crt
인증 정보 복사(바꾸기) 참고 1 참조	dsp에서 chg까지	crt
채널 복사	dsp	crt
채널 복사(바꾸기) 참고 1 참조	dsp에서 chg까지	crt
채널 리스너 복사	dsp	crt
채널 리스너 복사(바꾸기) 참고 1 참조	dsp에서 chg까지	crt
클라이언트 연결 채널 복사	dsp	crt
클라이언트 연결 채널 복사(바꾸기) 참고 1 참조	dsp에서 chg까지	crt
이름 목록 복사	dsp	crt
이름 목록 복사(바꾸기) 참고 1 참조	dsp에서 dsp 및 chg까지	crt

표 2. 오브젝트 권한 (계속)

명령	IBM MQ 오브젝트 권한	클래스 권한(오브젝트 유형의 경우)
프로세스 복사	dsp	crt
프로세스 복사(바꾸기) 참고 1 참조	dsp에서 chg까지	crt
큐 복사	dsp	crt
큐 복사(바꾸기) 참고 1 참조	dsp에서 dsp 및 chg까지	crt
인증 정보 작성	(시스템 기본 인증 정보) dsp	crt
인증 정보 작성(바꾸기) 참고 1 참조	(시스템 기본 인증 정보) dsp에서 chg까지	crt
채널 작성	(시스템 기본 채널) dsp	crt
채널 작성(바꾸기) 참고 1 참조	(시스템 기본 채널) dsp에서 chg까지	crt
채널 리스너 작성	(시스템 기본 리스너) dsp	crt
채널 리스너 작성(바꾸기) 참고 1 참조	(시스템 기본 리스너) dsp에서 chg까지	crt
클라이언트 연결 채널 작성	(시스템 기본 채널) dsp	crt
클라이언트 연결 채널 작성(바꾸기) 참고 1 참조	(시스템 기본 채널) dsp에서 chg까지	crt
이름 목록 작성	(시스템 기본 이름 목록) dsp	crt
이름 목록 작성(바꾸기) 참고 1 참조	(시스템 기본 이름 목록) dsp에서 dsp 및 chg까지	crt
프로세스 작성	(시스템 기본 프로세스) dsp	crt
프로세스 작성(바꾸기) 참고 1 참조	(시스템 기본 프로세스) dsp에서 chg까지	crt
큐 작성	(시스템 기본 큐) dsp	crt
큐 작성(바꾸기) 참고 1 참조	(시스템 기본 큐) dsp에서 dsp 및 chg까지	crt
서비스 작성	(시스템 기본 큐) dsp	crt
서비스 작성(바꾸기) 참고 1 참조	(시스템 기본 큐) dsp에서 chg까지	crt
인증 정보 삭제	dsp 및 dlt	해당사항 없음
권한 레코드 삭제	(큐 관리자 오브젝트) chg 참고 4 참조	참고 4 참조
채널 삭제	dsp 및 dlt	해당사항 없음
채널 리스너 삭제	dsp 및 dlt	해당사항 없음
클라이언트 연결 채널 삭제	dsp 및 dlt	해당사항 없음
이름 목록 삭제	dsp 및 dlt	해당사항 없음
프로세스 삭제	dsp 및 dlt	해당사항 없음
큐 삭제	dsp 및 dlt	해당사항 없음
서비스 삭제	dsp 및 dlt	해당사항 없음
인증 정보 조회	dsp	해당사항 없음
권한 레코드 조회	참고 4 참조	참고 4 참조
채널 조회	dsp	해당사항 없음

표 2. 오브젝트 권한 (계속)

명령	IBM MQ 오브젝트 권한	클래스 권한(오브젝트 유형의 경우)
채널 리스너 조회	dsp	해당사항 없음
채널 상태 조회(ChannelType MQCHT_CLSSDR의 경우)	inq	해당사항 없음
클라이언트 연결 채널 조회	dsp	해당사항 없음
이름 목록 조회	dsp	해당사항 없음
프로세스 조회	dsp	해당사항 없음
큐 조회	dsp	해당사항 없음
큐 관리자 조회	참고 3 참조	해당사항 없음
큐 상태 조회	dsp	해당사항 없음
서비스 조회	dsp	해당사항 없음
채널 ping	ctrl	해당사항 없음
큐 관리자 ping	참고 3 참조	해당사항 없음
큐 관리자 새로 고치기	(큐 관리자 오브젝트) chg	해당사항 없음
보안 새로 고치기(SecurityType MQSECTYPE_SSL의 경우)	(큐 관리자 오브젝트) chg	해당사항 없음
채널 재설정	ctrlx	해당사항 없음
큐 관리자 재설정	(큐 관리자 오브젝트) chg	해당사항 없음
큐 통계 재설정	dsp 및 chg	해당사항 없음
채널 분석	ctrlx	해당사항 없음
권한 레코드 설정	(큐 관리자 오브젝트) chg 참고 4 참조	참고 4 참조
채널 시작	ctrl	해당사항 없음
채널 중지	ctrl	해당사항 없음
연결 중지	(큐 관리자 오브젝트) chg	해당사항 없음
리스너 시작	ctrl	해당사항 없음
리스너 중지	ctrl	해당사항 없음
서비스 시작	ctrl	해당사항 없음
서비스 중지	ctrl	해당사항 없음
나가기	참고 2 참조	참고 2 참조

참고:

1. 대체할 오브젝트가 존재하면 이 명령이 적용됩니다. 그렇지 않으면 권한 검사는 대체없는 작성 또는 복사에 대한 것입니다.
2. 필수 권한은 이스케이프 텍스트에 의해 정의되는 MQSC 명령에 의해 판별되고 이는 이전 명령 중 하나와 동등합니다.
3. PCF 명령을 처리하기 위해 사용자 ID에는 대상 시스템에 큐 관리자 오브젝트에 대한 dsp 권한이 있어야 합니다.

- 이 PCF 명령은 명령 서버가 -a 매개변수로 시작되지 않는 경우 권한이 부여됩니다. 기본적으로 명령 서버는 큐 관리자가 -a 매개변수 없이 시작될 때 시작됩니다. 자세한 정보는 [PCF\(Programmable Command Format\) 참조서](#)를 참조하십시오.
- 큐 관리자에 사용자 ID chg 권한을 부여하면 모든 그룹 및 사용자에게 대한 권한 레코드를 설정하기 위한 기능을 제공합니다. 일반 사용자 또는 애플리케이션에 이 권한을 부여하지 마십시오.

IBM MQ에서는 보안 검사에 사용자 엑시트 프로그램을 제공할 수 있도록 일부 채널 보안 엑시트 지점을 제공하기도 합니다. 자세한 정보는 [채널 표시](#)를 참조하십시오.

Multi MQAI를 사용하여 PCF의 사용 단순화

MQAI (IBM MQ Administration Interface) 는 AIX, IBM i, Linux, 및 Windows에서 사용 가능한 IBM MQ 에 대한 프로그래밍 인터페이스입니다. 이는 프로그래밍 가능 명령 형식(PCF)을 사용하는 것보다 더 용이한 방법으로 오브젝트의 특성(또는 매개변수)을 핸들링하기 위해 데이터 백을 사용하여 IBM MQ 큐 관리자에서 관리 태스크를 수행합니다.

MQAI는 데이터 백을 사용하여 큐 관리자에서 관리 태스크를 수행합니다. 데이터 백을 사용하면 PCF를 사용하는 것보다 더 용이한 방법으로 오브젝트의 특성(또는 매개변수)을 처리할 수 있습니다.

MQAI를 사용하는 장점은 다음과 같습니다.

PCF 메시지의 사용을 단순화합니다.

MQAI는 IBM MQ를 관리하기 위한 더 쉬운 방법입니다. MQAI를 사용하면, 사용자의 PCF 메시지를 작성할 필요가 없습니다. 이는 복잡한 데이터 구조와 연관되는 문제점을 방지합니다.

MQI 호출을 사용하여 작성된 프로그램에서 매개변수를 전달하기 위해, PCF 메시지는 명령 및 문자열 또는 정수 데이터에 대한 세부사항을 포함해야 합니다. 수동으로 이 구성을 작성하려면, 여러 명령문을 모든 구성을 위한 사용자의 프로그램에 추가하여야 하고 메모리 영역을 할당해야 합니다. 이는 길고 힘든 작업일 수 있습니다.

MQAI를 사용하여 작성된 프로그램은 매개변수를 적절한 데이터 백으로 전달하고 각 구조에 대해 하나의 명령문만 필요합니다. MQAI 데이터 백을 사용하면 배열을 처리하고 스토리지를 할당하기 위한 필요성을 제거하고 PCF의 세부사항으로부터 어느 정도의 격리를 제공합니다.

오류 조건을 보다 쉽게 핸들링합니다.

PCF 명령에서 리턴 코드를 다시 가져오는 것은 어렵습니다. MQAI는 오류 조건을 핸들링하기 위해 프로그램을 더 용이하게 합니다.

애플리케이션 간 데이터 교환

애플리케이션 데이터가 PCF 형식으로 보내지고 MQAI에 의해 패키지로 묶이거나 풀릴 수 있습니다. 메시지 데이터가 정수 및 문자열로 구성되는 경우, MQAI를 사용하여 PCF 데이터에 대한 IBM MQ 내장 데이터 변환을 사용할 수 있습니다. 이는 데이터 변환 엑시트를 작성할 필요가 없습니다.

데이터 백을 작성하고 채운 후, mqExecute 호출을 사용하여 관리 명령 메시지를 큐 관리자의 명령 서버에 보낼 수 있습니다. 이 호출은 응답 메시지도 기다립니다. mqExecute 호출은 명령 서버를 가진 교환을 핸들링하고 응답 백에서 응답을 리턴합니다.

MQAI 사용의 예

다음 샘플 프로그램은 MQAI를 사용한 다양한 태스크 수행을 보여줍니다.

- [amqsaicq.c](#): 로컬 큐를 작성합니다.
- [amqsaiem.c](#): 단순 이벤트 모니터를 사용하여 화면에 이벤트를 표시합니다.
- [amqsailq.c](#): 모든 로컬 큐 및 현재 용량의 목록을 인쇄합니다.
- [amqsaicl.c](#): 모든 채널 및 해당 유형의 목록을 인쇄합니다.

MQAI 애플리케이션 빌드

MQAI를 사용하여 애플리케이션을 빌드하려면 IBM MQ에 도움이 된 것과 동일한 라이브러리에 링크하십시오. IBM MQ 애플리케이션 빌드 방법에 대한 정보는 [절차적 애플리케이션 빌드](#)를 참조하십시오.

MQAI를 사용하여 IBM MQ를 구성하기 위한 힌트 및 팁

MQAI는 명령 서버 자체에서 직접 처리하지 않고 PCF 메시지를 사용하여 관리 명령을 명령 서버로 송신합니다. MQAI를 사용하여 IBM MQ를 구성하기 위한 팁은 36 페이지의 『MQAI를 사용하여 IBM MQ를 구성하기 위한 힌트 및 팁』에서 찾을 수 있습니다.

관련 참조

[IBM MQ 관리 인터페이스 참조](#)

Multi

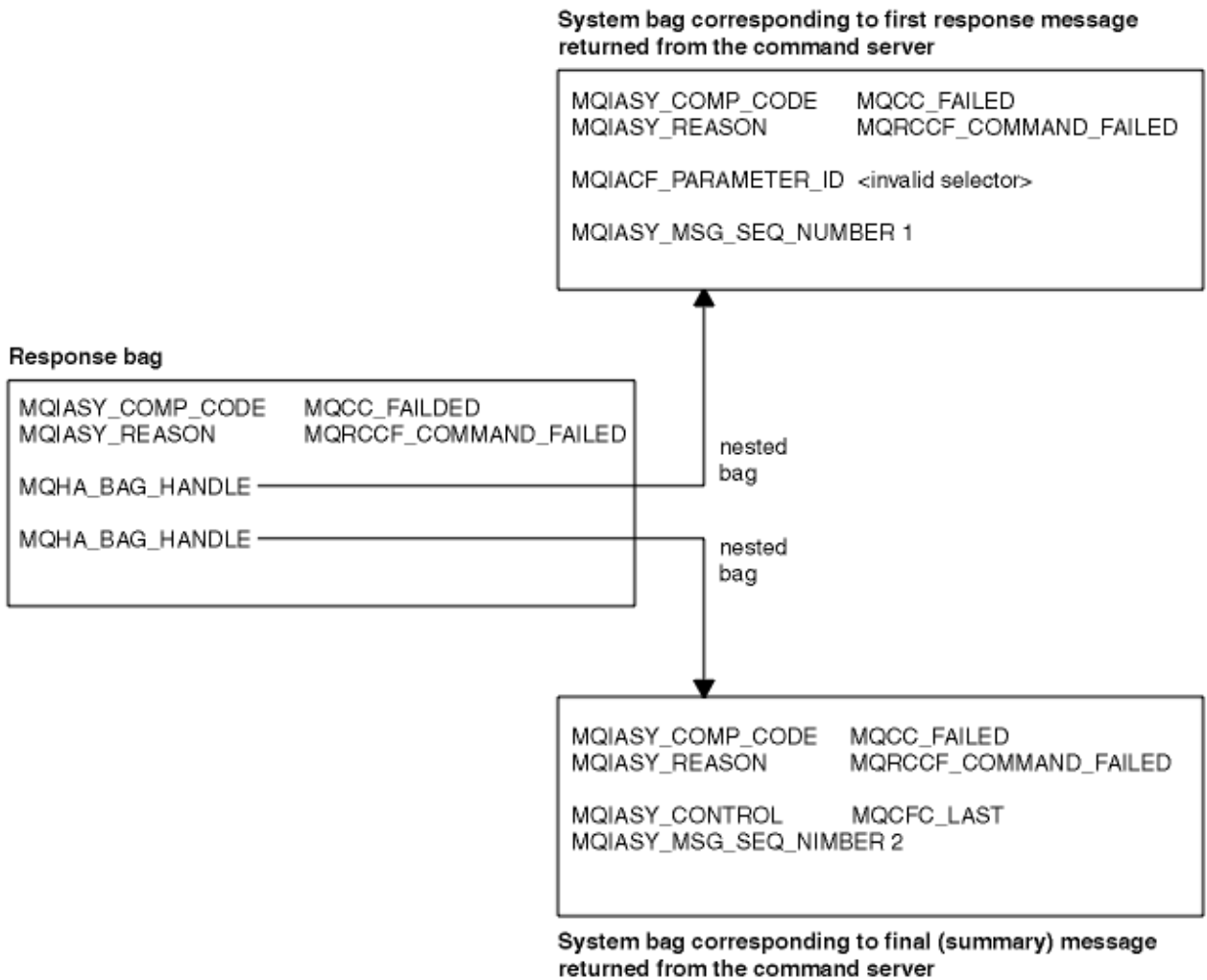
MQAI를 사용하여 IBM MQ를 구성하기 위한 힌트 및 팁

MQAI(IBM MQ Administration Interface)는 명령 서버 자체에서 직접 처리하지 않고 PCF 메시지를 사용하여 관리 명령을 명령 서버로 전송합니다. 다음은 MQAI를 사용하여 IBM MQ를 구성하기 위한 몇 가지 팁입니다.

- IBM MQ의 문자열은 고정된 길이로 채워진 공백입니다. C를 사용할 때, 널(Null) 종료 문자열은 일반적으로 IBM MQ 프로그래밍 인터페이스에 입력 매개변수로서 공급될 수 있습니다.
- 문자열 속성의 값을 지우려면 빈 문자열이 아닌 단일 공백 문자로 설정하십시오.
- 해당 속성에서 변경하고 조회하려는 속성을 미리 고려하십시오.
- 특정 속성은 변경될 수 없습니다(예: 큐 이름 또는 채널 유형). 수정될 수 있는 해당 속성만 변경하려고 시도하는지 확인하십시오. 특정 PCF 변경 오브젝트를 위한 필수 및 선택적 매개변수 목록을 참조하십시오. [PCF\(Programmable Command Format\)의 정의를 참조하십시오.](#)
- MQAI 호출에 실패하면 실패에 대한 일부 세부사항은 응답 백으로 리턴됩니다. 추가적인 세부사항은 선택자 MQHA_BAG_HANDLE이 액세스할 수 있는 중첩 백에서 찾을 수 있습니다. 예를 들어, mqExecute 호출이 MQRCCF_COMMAND_FAILED의 이유 코드와 함께 실패하는 경우, 이 정보는 응답 백에서 리턴됩니다. 이 이유 코드에 대한 가능한 이유는 지정된 선택자가 명령 메시지의 유형에 효과적이지 않았다는 것이고 정보에 대한 이 세부사항이 백 핸들에 의해 액세스될 수 있는 중첩 백에서 발견됩니다.

MQExecute에 대한 자세한 정보는 69 페이지의 『mqExecute 호출을 사용하여 qm 명령 서버에 관리 명령 전송』의 내용을 참조하십시오.

다음 다이어그램에 이 시나리오가 표시됩니다.



Multi 고급 MQAI 주제

색인 작성, 데이터 변환 및 메시지 디스크립터의 사용에 대한 정보입니다.

색인 작성

삽입 순서를 보존하기 위해 백으로부터 기존 데이터 항목을 대체하거나 제거할 때 색인이 사용됩니다.

데이터 변환

MQAI 데이터 백에 포함된 문자열은 다양한 코드화 문자 세트에 있을 수 있으며 `mqSetInteger` 호출을 사용하여 이러한 문자열을 변환할 수 있습니다.

메시지 디스크립터 사용

MQAI는 데이터 백이 작성될 때 초기값으로 설정되는 메시지 디스크립터를 작성합니다.

Multi MQAI의 색인 작성

색인은 백에서 기존 데이터 항목을 바꾸거나 제거할 때 사용됩니다. 세 가지 색인 작성 유형이 있으며, 색인 작성을 사용하면 데이터 항목을 쉽게 검색할 수 있습니다.

백의 데이터 항목 내에 있는 각 선택자 및 값에 세 개의 연관된 색인 번호가 있습니다.

- 동일한 선택자가 있는 다른 항목에 대한 색인
- 항목이 속하는 선택자의 카테고리(사용자 또는 시스템)에 대한 색인
- 백에 있는 모든 데이터 항목에 대한 색인(사용자 및 시스템)

이를 사용하면 38 페이지의 그림 3에 표시된 대로 사용자 선택자, 시스템 선택자 또는 둘 다에 의한 색인 작성이 가능합니다.

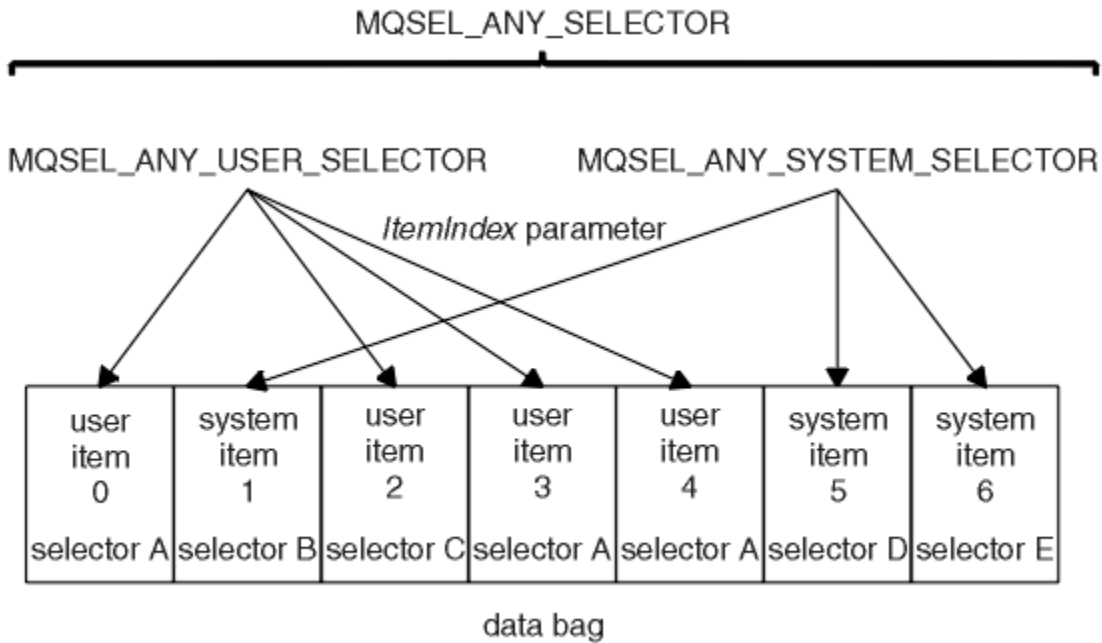


그림 3. 색인 작성

38 페이지의 그림 3에서 다음 색인 쌍으로 사용자 항목 3(선택자 A)을 참조할 수 있습니다.

- 선택자 A(ItemIndex 1)
- MQSEL_ANY_USER_SELECTOR(ItemIndex 2)
- MQSEL_ANY_SELECTOR(itemIndex 3)

색인은 C의 배열과 같이 0을 기반으로 합니다. 'n'개가 발생하는 경우 색인의 범위는 0에서 'n-1'까지이며 겹은 없습니다.

색인은 백에서 기존 데이터 항목을 바꾸거나 제거할 때 사용됩니다. 이러한 방식으로 사용될 때 삽입 순서는 그대로 유지되지만 다른 데이터 항목의 색인에 영향을 줄 수 있습니다. 이 예에 대해서는 66 페이지의 『백 내의 정보 변경』 및 68 페이지의 『데이터 항목 삭제』의 내용을 참조하십시오.

세 유형의 색인 작성을 사용하면 데이터 항목을 쉽게 검색할 수 있습니다. 예를 들어, 백에 특정 선택자에 대한 세 인스턴스가 있는 경우, mqCountItems 호출로 해당 선택자의 인스턴스 수를 셀 수 있고 mqInquire* 호출로 선택자와 색인을 모두 지정하여 해당 값만 조회할 수 있습니다. 이는 값 목록이 있는 속성에서 유용합니다(예: 채널에 있는 엑시트 중 일부).

Multi MQAI의 데이터 변환 처리

MQAI 데이터 백에 포함된 문자열은 다양한 코드화 문자 세트에 있을 수 있습니다. mqSetInteger 호출을 사용하여 이러한 문자열을 변환할 수 있습니다.

PCF 메시지와 같이, MQAI 데이터 백에 포함된 문자열은 다양한 코드화 문자 세트에 있습니다. 대개 PCF 메시지의 모든 문자열은 동일한 코드화 문자 세트, 즉 큐 관리자와 동일한 세트에 있습니다.

데이터 백의 각 문자열 항목에는 두 개의 값(문자열과 CCSID)이 있습니다. 백에 추가되는 문자열은 mqAddString 또는 mqSetString 호출의 Buffer 매개변수에서 가져옵니다. CCSID는 MQIASY_CODED_CHAR_SET_ID의 선택자를 포함한 시스템 항목에서 얻을 수 있습니다. 이를 백 CCSID라고 하며 mqSetInteger 호출을 사용하여 변경할 수 있습니다.

데이터 백에 포함된 문자열 값을 조회하는 경우, CCSID는 호출의 출력 매개변수입니다.

39 페이지의 표 3에서는 데이터 백을 메시지로 변환하거나 그 반대인 경우에 적용되는 규칙을 보여줍니다.

표 3. CCSID 처리

MQAI 호출	CCSID	호출할 입력	호출할 출력
mqBagToBuffer	Bag CCSID(1)	무시함	변경하지 않음
mqBagToBuffer	백에 있는 문자열 CCSID	사용함	변경하지 않음
mqBagToBuffer	버퍼에 있는 문자열 CCSID	적용할 수 없음	백에 있는 문자열 CCSID에서 복사함
mqBufferToBag	Bag CCSID(1)	무시함	변경하지 않음
mqBufferToBag	버퍼에 있는 문자열 CCSID	사용함	변경하지 않음
mqBufferToBagmqBufferToBag	백에 있는 문자열 CCSID	적용할 수 없음	버퍼에 있는 문자열 CCSID에서 복사함
mqPutBag	MQMD CCSID	사용함	변경하지 않음(2)
mqPutBag	Bag CCSID(1)	무시함	변경하지 않음
mqPutBag	백에 있는 문자열 CCSID	사용함	변경하지 않음
mqPutBag	송신한 메시지에 있는 문자열 CCSID	적용할 수 없음	백에 있는 문자열 CCSID에서 복사함
mqGetBag	MQMD CCSID	메시지의 데이터 변환에 사용함	리턴된 데이터의 CCSID로 설정됨(3)
mqGetBag	Bag CCSID(1)	무시함	변경하지 않음
mqGetBag	메시지에 있는 문자열 CCSID	사용함	변경하지 않음
mqGetBag	백에 있는 문자열 CCSID	적용할 수 없음	메시지에 있는 문자열 CCSID에서 복사함
mqExecute	요청-백 CCSID	요청 메시지의 MQMD에 사용됨(4)	변경하지 않음
mqExecute	응답-백 CCSID	응답 메시지의 데이터 변환에 사용함(4)	리턴된 데이터의 CCSID로 설정됨(3)
mqExecute	요청 백에 있는 문자열 CCSID	요청 메시지에 사용함	변경하지 않음
mqExecute	응답 백에 있는 문자열 CCSID	적용할 수 없음	응답 메시지에 있는 문자열 CCSID에서 복사함

참고:

1. 백 CCSID는 선택자 MQIASY_CODED_CHAR_SET_ID가 있는 시스템 항목입니다.
2. MQCCSI_Q_MGR은 실제 큐 관리자 CCSID로 변경됩니다.
3. 데이터 변환을 요청하는 경우, 리턴된 데이터의 CCSID가 출력 값과 동일합니다. 데이터 변환을 요청하지 않는 경우, 리턴된 데이터의 CCSID는 메시지 값과 동일합니다. 데이터 변환을 요청했지만 실패하는 경우 메시지가 리턴되지 않습니다.
4. CCSID가 MQCCSI_DEFAULT인 경우 큐 관리자의 CCSID가 사용됩니다.

관련 개념

189 페이지의 『코드화 문자 세트 간 데이터 변환』

두 문자 세트가 모두 단일 언어 또는 하나의 유사한 언어 그룹과 관련된 경우 IBM MQ 정의 형식(내장 형식이라고도 함)의 메시지 데이터는 큐 관리자에 의해 한 코드화 문자 세트에서 다른 문자 세트로 변환할 수 있습니다.

191 페이지의 『ccsid_part2.tbl 파일』

ccsid_part2.tbl 파일은 추가 CCSID 정보를 제공하는 데 사용됩니다. ccsid_part2.tbl 파일은 IBM MQ 9.0이전에 사용된 ccsid.tbl 파일을 대체합니다.

Multi MQAI에서 메시지 디스크립터 사용

MQAI가 생성하는 메시지 디스크립터는 데이터 백이 작성될 때 초기값으로 설정됩니다.

PCF 명령 유형은 선택자가 MQIASY_TYPE인 시스템 항목에서 얻을 수 있습니다. 데이터 백을 작성하는 경우, 이 항목의 초기값은 작성한 백의 유형에 따라 설정됩니다.

백 유형	MQIASY_TYPE 항목의 초기값
MQCBO_ADMIN_BAG	MQCFT_COMMAND
MQCBO_COMMAND_BAG	MQCFT_COMMAND
MQCBO_*	MQCFT_USER

MQAI가 메시지 디스크립터를 생성할 때 **Format** 및 **MsgType** 매개변수에 사용된 값은 40 페이지의 표 4에 표시된 대로 선택자가 MQIASY_TYPE인 시스템 항목 값에 따라 다릅니다.

PCF 명령 유형	형식	MsgType
MQCFT_COMMAND	MQFMT_ADMIN	MQMT_REQUEST
MQCFT_REPORT	MQFMT_ADMIN	MQMT_REPORT
MQCFT_RESPONSE	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQMT_DATAGRAM
MQCFT_EVENT	MQFMT_EVENT	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

40 페이지의 표 5에서는 관리 백 또는 명령 백을 작성하는 경우, 메시지 디스크립터의 *Format*이 MQFMT_ADMIN이고 *MsgType*이 MQMT_REQUEST임을 보여줍니다. 이는 응답이 되돌아올 것으로 예상되는 경우 명령 서버에 송신된 PCF 요청 메시지에 적합합니다.

메시지 디스크립터의 다른 매개변수는 40 페이지의 표 6에 표시된 값을 사용합니다.

매개변수	값
<i>StrucId</i>	MQMD_STRUC_ID
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	40 페이지의 표 5의 내용을 참조하십시오.
<i>Expiry</i>	30초(41 페이지의 『1』 참고)
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	CCSID 백에 따라 다름(41 페이지의 『2』 참고)
<i>Format</i>	40 페이지의 표 5의 내용을 참조하십시오.

표 6. 메시지 디스크립터 값 (계속)	
매개변수	값
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	MQMI_NONE
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	41 페이지의 『3』 참조
<i>ReplyToQMGr</i>	공백

참고:

1. 이 값은 mqExecute 호출에서 **OptionsBag** 매개변수를 사용하여 대체할 수 있습니다. 이에 대한 정보는 [mqExecute](#)의 내용을 참조하십시오.
2. 38 페이지의 『MQAI의 데이터 변환 처리』의 내용을 참조하십시오.
3. MQMT_REQUEST 유형의 메시지에 대한 MQAI 생성 임시 동적 큐 또는 사용자 지정 응답 큐의 이름입니다. 그렇지 않으면 공백입니다.

Multi 로컬 큐를 작성하기 위한 샘플 C 프로그램(amqsaicq.c)

샘플 C 프로그램 amqsaicq.c은(는) MQAI를 사용하여 로컬 큐를 작성합니다.

```

/*****
/*
/* Program name: AMQSAICQ.C
/*
/* Description: Sample C program to create a local queue using the
/* IBM MQ Administration Interface (MQAI).
/*
/* Statement: Licensed Materials - Property of IBM
/*
/* 84H2000, 5765-B73
/* 84H2001, 5639-B42
/* 84H2002, 5765-B74
/* 84H2003, 5765-B75
/* 84H2004, 5639-B43
/*
/* (C) Copyright IBM Corp. 1999, 2024
/*
/*****
/*
/* Function:
/* AMQSAICQ is a sample C program that creates a local queue and is an
/* example of the use of the mqExecute call.
/*
/* - The name of the queue to be created is a parameter to the program.
/*
/* - A PCF command is built by placing items into an MQAI bag.
/* These are:-
/* - The name of the queue
/* - The type of queue required, which, in this case, is local.
/*
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q.
/* The call generates the correct PCF structure.
/* The call receives the reply from the command server and formats into
/* the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server then the code returned by the
/* command server is retrieved from the system bag that is
/* embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*

```

```

/*                                                                 */
/*****
/* AMQSAICQ has 2 parameters - the name of the local queue to be created */
/* - the queue manager name (optional) */
/*                                                                 */
/*****
/* Includes */
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF          */
#include <cmqbc.h>        /* MQAI         */

void CheckCallResult(MQCHAR *, MQLONG , MQLONG );
void CreateLocalQueue(MQHCONN, MQCHAR *);

int main(int argc, char *argv[])
{
    MQHCONN hConn;          /* handle to IBM MQ connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG connReason;     /* MQCONN reason code */
    MQLONG compCode;       /* completion code */
    MQLONG reason;        /* reason code */

    /*****
    /* First check the required parameters */
    /*****
    printf("Sample Program to Create a Local Queue\n");
    if (argc < 2)
    {
        printf("Required parameter missing - local queue name\n");
        exit(99);
    }

    /*****
    /* Connect to the queue manager */
    /*****
    if (argc > 2)
        strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
        MQCONN(QMName, &hConn, &compCode, &connReason);

    /*****
    /* Report reason and stop if connection failed */
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Call the routine to create a local queue, passing the handle to the */
    /* queue manager and also passing the name of the queue to be created. */
    /*****
    CreateLocalQueue(hConn, argv[1]);

    /*****
    /* Disconnect from the queue manager if not already connected */
    /*****
    if (connReason != MQRC_ALREADY_CONNECTED)
    {
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
    }
    return 0;

}
/*****
/*
/* Function:      CreateLocalQueue
/* Description:  Create a local queue by sending a PCF command to the command
/*              server.
/*
/*
/*****

```

```

/*                                                                 */
/* Input Parameters:  Handle to the queue manager                  */
/*                   Name of the queue to be created              */
/*                                                                 */
/* Output Parameters: None                                        */
/*                                                                 */
/* Logic: The mqExecute call is executed with the command MQCMD_CREATE_Q. */
/* The call generates the correct PCF structure.                  */
/* The default options to the call are used so that the command is sent */
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.                            */
/* The reply from the command server is placed on a temporary dynamic */
/* queue.                                                         */
/* The reply is read from the temporary queue and formatted into the */
/* response bag.                                                 */
/*                                                                 */
/* The completion code from the mqExecute call is checked and if there */
/* is a failure from the command server then the code returned by the */
/* command server is retrieved from the system bag that is      */
/* embedded in the response bag to the mqExecute call.          */
/*                                                                 */
/*****
void CreateLocalQueue(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG reason;                /* reason code                */
    MQLONG compCode;              /* completion code            */
    MQHBAG commandBag = MQHB_UNUSABLE_HBAG; /* command bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG resultBag;            /* result bag from mqExecute */
    MQLONG mqExecuteCC;          /* mqExecute completion code */
    MQLONG mqExecuteRC;          /* mqExecute reason code     */

    printf("\nCreating Local Queue %s\n\n", qName);

    /*****
    /* Create a command Bag for the mqExecute call. Exit the function if the */
    /* create fails.                                                         */
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &commandBag, &compCode, &reason);
    CheckCallResult("Create the command bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Create a response Bag for the mqExecute call, exit the function if the */
    /* create fails.                                                         */
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
    CheckCallResult("Create the response bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Put the name of the queue to be created into the command bag. This will */
    /* be used by the mqExecute call.                                         */
    /*****
    mqAddString(commandBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, qName, &compCode,
                &reason);
    CheckCallResult("Add q name to command bag", compCode, reason);

    /*****
    /* Put queue type of local into the command bag. This will be used by the */
    /* mqExecute call.                                                         */
    /*****
    mqAddInteger(commandBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
    CheckCallResult("Add q type to command bag", compCode, reason);

    /*****
    /* Send the command to create the required local queue.                  */
    /* The mqExecute call will create the PCF structure required, send it to */
    /* the command server and receive the reply from the command server into */
    /* the response bag.                                                      */
    /*****
    mqExecute(hConn,                /* IBM MQ connection handle */
              MQCMD_CREATE_Q,       /* Command to be executed    */
              MQHB_NONE,           /* No options bag            */
              commandBag,          /* Handle to bag containing commands */
              responseBag,         /* Handle to bag to receive the response */
              MQHO_NONE,          /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
              MQHO_NONE,          /* Create a dynamic q for the response */
              &compCode,           /* Completion code from the mqExecute */
              &reason);           /* Reason code from mqExecute call */

```

```

if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n")
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call and find the error if it failed. */
*****/
if ( compCode == MQCC_OK )
    printf("Local queue %s successfully created\n", qName);
else
{
    printf("Creation of local queue %s failed: Completion Code = %d
           qName, compCode, reason);
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        /*****
        /* Get the system bag handle out of the mqExecute response bag. */
        /* This bag contains the reason from the command server why the */
        /* command failed. */
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &resultBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        *****/
        mqInquireInteger(resultBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                        &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(resultBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                        &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag", compCode,
                        reason);
        printf("Error returned by the command server: Completion code = %d :
              Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}
/*****
/* Delete the command bag if successfully created. */
*****/
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
    CheckCallResult("Delete the command bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
*****/
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}
} /* end of CreateLocalQueue */

/*****
/*
/* Function: CheckCallResult
/*
*****/
/*
/* Input Parameters:  Description of call
/*                   Completion code
/*                   Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
*****/

```

```

void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d :
              Reason = %d\n", callText, cc, rc);
}

```

Multi 이벤트 모니터를 사용하여 이벤트를 표시하는 샘플 C 프로그램(amqsaiem.c)

샘플 C 프로그램 amqsaiem.c은(는) MQAI를 사용하여 기본 이벤트 모니터를 보여줍니다.

```

*****/
/*
/* Program name: AMQSAIEM.C
/*
/* Description: Sample C program to demonstrate a basic event monitor
/*               using the IBM MQ Admin Interface (MQAI).
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/*
*****/
/*
/* Function:
/* AMQSAIEM is a sample C program that demonstrates how to write a simple
/* event monitor using the mqGetBag call and other MQAI calls.
/*
/* The name of the event queue to be monitored is passed as a parameter
/* to the program. This would usually be one of the system event queues:-
/*     SYSTEM.ADMIN.QMGR.EVENT      Queue Manager events
/*     SYSTEM.ADMIN.PERFM.EVENT     Performance events
/*     SYSTEM.ADMIN.CHANNEL.EVENT   Channel events
/*     SYSTEM.ADMIN.LOGGER.EVENT    Logger events
/*
/* To monitor the queue manager event queue or the performance event queue,
/* the attributes of the queue manager need to be changed to enable
/* these events. For more information about this, see Part 1 of the
/* Programmable System Management book. The queue manager attributes can
/* be changed using either MQSC commands or the MQAI interface.
/* Channel events are enabled by default.
/*
/* Program logic
/* Connect to the Queue Manager.
/* Open the requested event queue with a wait interval of 30 seconds.
/* Wait for a message, and when it arrives get the message from the queue
/* and format it into an MQAI bag using the mqGetBag call.
/* There are many types of event messages and it is beyond the scope of
/* this sample to program for all event messages. Instead the program
/* prints out the contents of the formatted bag.
/* Loop around to wait for another message until either there is an error
/* or the wait interval of 30 seconds is reached.
/*
*****/
/*
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored
/* - the queue manager name (optional)
/*
*****/
/* Includes
*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>           /* MQI
#include <cmqcfc.h>        /* PCF
#include <cmqbc.h>         /* MQAI

*****/
/* Macros

```

```

/*****
#if MQAT_DEFAULT == MQAT_WINDOWS_NT
#define Int64 "I64"
#elif defined(MQ_64_BIT)
#define Int64 "l"
#else
#define Int64 "ll"
#endif

/*****
/* Function prototypes */
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
void GetQEvents(MQHCONN, MQCHAR *);
int PrintBag(MQHBAG);
int PrintBagContents(MQHBAG, int);

/*****
/* Function: main */
/*****
int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QM name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */

    /*****
    /* First check the required parameters */
    /*****
    printf("Sample Event Monitor (times out after 30 secs)\n");
    if (argc < 2)
    {
        printf("Required parameter missing - event queue to be monitored\n");
        exit(99);
    }

    /*****
    /* Connect to the queue manager */
    /*****
    if (argc > 2)
        strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(QMName, &hConn, &compCode, &connReason);
    /*****
    /* Report the reason and stop if the connection failed */
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Call the routine to open the event queue and format any event messages */
    /* read from the queue. */
    /*****
    GetQEvents(hConn, argv[1]);

    /*****
    /* Disconnect from the queue manager if not already connected */
    /*****
    if (connReason != MQRC_ALREADY_CONNECTED)
    {
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
    }

    return 0;
}

/*****
/*
/* Function: CheckCallResult */
/*
/*
/*****
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*

```

```

/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the
/* reason code if the completion code is not successful */
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

/*****
/*
/* Function: GetQEvents */
/*
/*****
/*
/* Input Parameters: Handle to the queue manager */
/* Name of the event queue to be monitored */
/*
/* Output Parameters: None */
/*
/* Logic: Open the event queue. */
/* Get a message off the event queue and format the message into */
/* a bag. */
/* A real event monitor would need to be programmed to deal with */
/* each type of event that it receives from the queue. This is */
/* outside the scope of this sample, so instead, the contents of */
/* the bag are printed. */
/* The program waits for 30 seconds for an event message and then */
/* terminates if no more messages are available. */
/*****
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason; /* MQOPEN reason code */
    MQLONG reason; /* reason code */
    MQLONG compCode; /* completion code */
    MQHOBJ eventQueue; /* handle to event queue */

    MQHBAG eventBag = MQHB_UNUSABLE_HBAG; /* event bag to receive event msg */
    MQOD od = {MQOD_DEFAULT}; /* Object Descriptor */
    MQMD md = {MQMD_DEFAULT}; /* Message Descriptor */
    MQGMO gmo = {MQGMO_DEFAULT}; /* get message options */
    MQLONG bQueueOK = 1; /* keep reading msgs while true */

    /*****
    /* Create an Event Bag in which to receive the event. */
    /* Exit the function if the create fails. */
    /*****
    mqCreateBag(MQCBO_USER_BAG, &compCode, &reason);
    CheckCallResult("Create event bag", compCode, reason);
    if (compCode != MQCC_OK)
        return;

    /*****
    /* Open the event queue chosen by the user */
    /*****
    strncpy(od.ObjectName, qName, (size_t)MQ_Q_NAME_LENGTH);
    MQOPEN(hConn, &od, MQOO_INPUT_AS_Q_DEF+MQOO_FAIL_IF QUIESCING, &eventQueue,
        &compCode, &openReason);
    CheckCallResult("Open event queue", compCode, openReason);

    /*****
    /* Set the GMO options to control the action of the get message from the */
    /* queue. */
    /*****
    gmo.WaitInterval = 30000; /* 30 second wait for message */
    gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF QUIESCING + MQGMO_CONVERT;
    gmo.Version = MQGMO_VERSION_2; /* Avoid need to reset Message ID */
    gmo.MatchOptions = MQMO_NONE; /* and Correlation ID after every */
    /* mqGetBag

    /*****
    /* If open fails, we cannot access the queue and must stop the monitor. */
    /*****
    if (compCode != MQCC_OK)
        bQueueOK = 0;

    /*****

```



```

/* Main loop to get an event message when it arrives */
/*****
while (bQueueOK)
{
    printf("\nWaiting for an event\n");

    /*****
    /* Get the message from the event queue and convert it into the event */
    /* bag. */
    /*****
    mqGetBag(hConn, eventQueue, &md, &gmo, eventBag, &compCode, &reason);

    /*****
    /* If get fails, we cannot access the queue and must stop the monitor. */
    /*****
    if (compCode != MQCC_OK)
    {
        bQueueOK = 0;

        /*****
        /* If get fails because no message available then we have timed out, */
        /* so report this, otherwise report an error. */
        /*****
        if (reason == MQRC_NO_MSG_AVAILABLE)
        {
            printf("No more messages\n");
        }
        else
        {
            CheckCallResult("Get bag", compCode, reason);
        }
    }

    /*****
    /* Event message read - Print the contents of the event bag */
    /*****
    else
    {
        if ( PrintBag(eventBag) )
            printf("\nError found while printing bag contents\n");

        } /* end of msg found */
    } /* end of main loop */
    /*****
    /* Close the event queue if successfully opened */
    /*****
    if (openReason == MQRC_NONE)
    {
        MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
        CheckCallResult("Close event queue", compCode, reason);
    }

    /*****
    /* Delete the event bag if successfully created. */
    /*****
    if (eventBag != MQHB_UNUSABLE_HBAG)
    {
        mqDeleteBag(&eventBag, &compCode, &reason);
        CheckCallResult("Delete the event bag", compCode, reason);
    }

} /* end of GetQEvents */

/*****
/*
/* Function: PrintBag */
/*
/*
/*****
/*
/* Input Parameters: Bag Handle */
/*
/* Output Parameters: None */
/*
/* Returns: Number of errors found */
/*
/* Logic: Calls PrintBagContents to display the contents of the bag. */
/*****
int PrintBag(MQHBAG dataBag)
{
    int errors;

```

```

    printf("\n");
    errors = PrintBagContents(dataBag, 0);
    printf("\n");

    return errors;
}

/*****
/*
/* Function: PrintBagContents
/*
/*
*****/
/*
/* Input Parameters:  Bag Handle
/*                    Indentation level of bag
/*
/* Output Parameters: None
/*
/* Returns:          Number of errors found
/*
/* Logic: Count the number of items in the bag
/*          Obtain selector and item type for each item in the bag.
/*          Obtain the value of the item depending on item type and display the
/*          index of the item, the selector and the value.
/*          If the item is an embedded bag handle then call this function again
/*          to print the contents of the embedded bag increasing the
/*          indentation level.
*****/
int PrintBagContents(MQHBAG dataBag, int indent)
{
    /*****
    /* Definitions
    *****/
    #define LENGTH 500                /* Max length of string to be read*/
    #define INDENT 4                  /* Number of spaces to indent */
                                        /* embedded bag display */

    /*****
    /* Variables
    *****/
    MQLONG  itemCount;                /* Number of items in the bag */
    MQLONG  itemType;                /* Type of the item */
    int     i;                        /* Index of item in the bag */
    MQCHAR  stringVal[LENGTH+1];     /* Value if item is a string */
    MQBYTE  byteStringVal[LENGTH];   /* Value if item is a byte string */
    MQLONG  stringLength;            /* Length of string value */
    MQLONG  ccsid;                   /* CCSID of string value */
    MQINT32 iValue;                  /* Value if item is an integer */
    MQINT64 i64Value;                /* Value if item is a 64-bit
                                        /* integer */
    MQLONG  selector;                /* Selector of item */
    MQHBAG  bagHandle;               /* Value if item is a bag handle */
    MQLONG  reason;                  /* reason code */
    MQLONG  compCode;                /* completion code */
    MQLONG  trimLength;              /* Length of string to be trimmed */
    int     errors = 0;              /* Count of errors found */
    char    blanks[] = "            "; /* Blank string used to
                                        /* indent display */

    /*****
    /* Count the number of items in the bag
    *****/
    mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

    if (compCode != MQCC_OK)
        errors++;
    else
    {
        printf("
        printf("
        printf("
    }

    /*****
    /* If no errors found, display each item in the bag
    *****/
    if (!errors)
    {
        for (i = 0; i < itemCount; i++)

```

```

{
/*****
/* First inquire the type of the item for each item in the bag */
/*****
mqInquireItemInfo(dataBag, /* Bag handle */
                  MQSEL_ANY_SELECTOR, /* Item can have any selector*/
                  i, /* Index position in the bag */
                  &selector, /* Actual value of selector */
                  /* returned by call */
                  &itemType, /* Actual type of item */
                  /* returned by call */
                  &compCode, /* Completion code */
                  &reason); /* Reason Code */

if (compCode != MQCC_OK)
    errors++;

switch(itemType)
{
case MQITEM_INTEGER:
/*****
/* Item is an integer. Find its value and display its index, */
/* selector and value. */
/*****
mqInquireInteger(dataBag, /* Bag handle */
                 MQSEL_ANY_SELECTOR, /* Allow any selector */
                 i, /* Index position in the bag */
                 &iValue, /* Returned integer value */
                 &compCode, /* Completion code */
                 &reason); /* Reason Code */

if (compCode != MQCC_OK)
    errors++;
else
    printf("%.s %-2d %-4d (%d)\n",
           indent, blanks, i, selector, iValue);
break

case MQITEM_INTEGER64:
/*****
/* Item is a 64-bit integer. Find its value and display its */
/* index, selector and value. */
/*****
mqInquireInteger64(dataBag, /* Bag handle */
                   MQSEL_ANY_SELECTOR, /* Allow any selector */
                   i, /* Index position in the bag */
                   &i64Value, /* Returned integer value */
                   &compCode, /* Completion code */
                   &reason); /* Reason Code */

if (compCode != MQCC_OK)
    errors++;
else
    printf("%.s %-2d %-4d (%"Int64"d)\n",
           indent, blanks, i, selector, i64Value);
break;

case MQITEM_STRING:
/*****
/* Item is a string. Obtain the string in a buffer, prepare */
/* the string for displaying and display the index, selector, */
/* string and Character Set ID. */
/*****
mqInquireString(dataBag, /* Bag handle */
                MQSEL_ANY_SELECTOR, /* Allow any selector */
                i, /* Index position in the bag */
                LENGTH, /* Maximum length of buffer */
                stringVal, /* Buffer to receive string */
                &stringLength, /* Actual length of string */
                &ccsid, /* Coded character set ID */
                &compCode, /* Completion code */
                &reason); /* Reason Code */

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check */
/* explicitly for call failure. */
/*****
if (compCode == MQCC_FAILED)
    errors++;

```

```

else
{
    /******
    /* Remove trailing blanks from the string and terminate with*/
    /* a null. First check that the string should not have been */
    /* longer than the maximum buffer size allowed.          */
    /******
    if (stringLength > LENGTH)
        trimLength = LENGTH;
    else
        trimLength = stringLength;
    mqTrim(trimLength, stringVal, stringVal, &compCode, &reason);
    printf("%.s %-2d %-4d '%s' %d\n",
           indent, blanks, i, selector, stringVal, ccsid);
}
break;

case MQITEM_BYTE_STRING:
/******
/* Item is a byte string. Obtain the byte string in a buffer, */
/* prepare the byte string for displaying and display the    */
/* index, selector and string.                                */
/******
mqInquireByteString(dataBag, /* Bag handle */
                    MQSEL_ANY_SELECTOR, /* Allow any selector */
                    i, /* Index position in the bag */
                    LENGTH, /* Maximum length of buffer */
                    byteStringVal, /* Buffer to receive string */
                    &stringLength, /* Actual length of string */
                    &compCode, /* Completion code */
                    &reason); /* Reason Code

/******
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check    */
/* explicitly for call failure.                                */
/******
if (compCode == MQCC_FAILED)
    errors++;
else
{
    printf("%.s %-2d %-4d X'",
           indent, blanks, i, selector);

    for (i = 0 ; i < stringLength ; i++)
        printf("

    printf("\n");
}
break;

case MQITEM_BAG:
/******
/* Item is an embedded bag handle, so call the PrintBagContents*/
/* function again to display the contents.                    */
/******
mqInquireBag(dataBag, /* Bag handle */
             MQSEL_ANY_SELECTOR, /* Allow any selector */
             i, /* Index position in the bag */
             &bagHandle, /* Returned embedded bag handle */
             &compCode, /* Completion code */
             &reason); /* Reason Code

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("%.s %-2d %-4d (%d)\n", indent, blanks, i,
           selector, bagHandle);
    if (selector == MQHA_BAG_HANDLE)
        printf("
    else
        printf("
        PrintBagContents(bagHandle, indent+INDENT);
}
break;

default:
    printf("
}
}
}
}

```

```
    return errors;
}
```

Multi 채널 오브젝트를 조회하기 위한 샘플 C 프로그램(amqsaicl.c)

샘플 C 프로그램 amqsaicl.c은(는) MQAI를 사용하여 채널 오브젝트를 조회합니다.

```

/*****
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/*               using the IBM MQ Administration Interface (MQAI)
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/*****
/*
/* Function:
/* AMQSAICL is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires all channels and their types.
/*
/* - A PCF command is built from items placed into an MQAI administration
/*   bag.
/*   These are:-
/*     - The generic channel name "*"
/*     - The attributes to be inquired. In this sample we just want
/*       name and type attributes
/*
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed.
/*   The call generates the correct PCF structure.
/*   The default options to the call are used so that the command is sent
/*   to the SYSTEM.ADMIN.COMMAND.QUEUE.
/*   The reply from the command server is placed on a temporary dynamic
/*   queue.
/*   The reply from the MQCMD_INQUIRE_CHANNEL is read from the
/*   temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/*   is a failure from the command server, then the code returned by the
/*   command server is retrieved from the system bag that has been
/*   embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAICL has 2 parameter - the queue manager name (optional)
/*                          - output file (optional) default varies
/*****
/*
/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

#include <cmqc.h>          /* MQI
#include <cmqcfh.h>       /* PCF
#include <cmqbc.h>        /* MQAI
#include <cmqxc.h>        /* MQCD

/*****
/* Function prototypes
/*
```

```

/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* DataTypes */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFILE OUTFILEHDL;
#else
typedef FILE OUTFILEHDL;
#endif

/*****
/* Constants */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    " *SDR      ", /* MQCHT_SENDER */
    " *SVR      ", /* MQCHT_SERVER */
    " *RCVR     ", /* MQCHT_RECEIVER */
    " *RQSTR    ", /* MQCHT_REQUESTER */
    " *ALL      ", /* MQCHT_ALL */
    " *CLTCN    ", /* MQCHT_CLNTCONN */
    " *SVRCONN  ", /* MQCHT_SVRCONN */
    " *CLUSRCVR", /* MQCHT_CLUSRCVR */
    " *CLUSSDR  " /* MQCHT_CLUSSDR */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER */
    "svr      ", /* MQCHT_SERVER */
    "rcvr     ", /* MQCHT_RECEIVER */
    "rqstr    ", /* MQCHT_REQUESTER */
    "all      ", /* MQCHT_ALL */
    "cltconn  ", /* MQCHT_CLNTCONN */
    "svrcn    ", /* MQCHT_SVRCONN */
    "clusrcvr", /* MQCHT_CLUSRCVR */
    "clusldr  " /* MQCHT_CLUSSDR */
};
#endif

/*****
/* Macros */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = _Ropen((fname), "wr", rtncode=Y");
#define CLOSEOUTFILE(hdl) \
    _Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _Rwrite((hdl), (buf), (buflen));
#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#else
#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));
#endif

#define ChlType2String(t) ChlTypeMap[(t)-1].name

```

```

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    /*****
    MQHCONN hConn; /* handle to MQ connection
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name
    MQLONG reason; /* reason code
    MQLONG connReason; /* MQCONN reason code
    MQLONG compCode; /* completion code
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute
    MQHBAG cAttrsBag; /* bag containing chl attributes
    MQHBAG errorBag; /* bag containing cmd server error
    MQLONG mqExecuteCC; /* mqExecute completion code
    MQLONG mqExecuteRC; /* mqExecute reason code
    MQLONG chlNameLength; /* Actual length of chl name
    MQLONG chlType; /* Channel type
    MQLONG i; /* loop counter
    MQLONG numberOfBags; /* number of bags in response bag
    MQCHAR chlName[MQ_OBJECT_NAME_LENGTH+1]; /* name of chl extracted from bag
    MQCHAR OutputBuffer[100]; /* output data buffer
    OUTFILEHDL *outfp = NULL; /* output file handle

    /*****
    /* Connect to the queue manager
    /*****
    if (argc > 1)
        strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(qmName, &hConn;, &compCode;, &connReason;);

    /*****
    /* Report the reason and stop if the connection failed.
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("Queue Manager connection", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Open the output file
    /*****
    if (argc > 2)
    {
        OPENOUTFILE(outfp, argv[2]);
    }
    else
    {
        OPENOUTFILE(outfp, OUTFILE);
    }

    if(outfp == NULL)
    {
        printf("Could not open output file.\n");
        goto MOD_EXIT;
    }

    /*****
    /* Create an admin bag for the mqExecute call
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &adminBag;, &compCode;, &reason;);
    CheckCallResult("Create admin bag", compCode, reason);

    /*****
    /* Create a response bag for the mqExecute call
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag;, &compCode;, &reason;);
    CheckCallResult("Create response bag", compCode, reason);

    /*****
    /* Put the generic channel name into the admin bag
    /*****
    mqAddString(adminBag, MQCACH_CHANNEL_NAME, MQBL_NULL_TERMINATED, "*",
                &compCode;, &reason;);
    CheckCallResult("Add channel name", compCode, reason);

    /*****
    /* Put the channel type into the admin bag
    /*****

```



```

/*****
mqAddInteger(adminBag, MQIACH_CHANNEL_TYPE, MQCHT_ALL, &compCode;, &reason;);
CheckCallResult("Add channel type", compCode, reason);

/*****
/* Add an inquiry for various attributes */
/*****
mqAddInquiry(adminBag, MQIACH_CHANNEL_TYPE, &compCode;, &reason;);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the channel names and channel types. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* MQ connection handle */
          MQCMD_INQUIRE_CHANNEL, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode;, /* Completion code from the mqexecute */
          &reason;); /* Reason code from mqexecute call */

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName="">\n");
    goto MOD_EXIT;
}

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each channel are in separate bags. */
    /*****
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags;,
                &compCode;, &reason;);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfbags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the channel attributes */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &AttrsBag,
                    &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the channel name out of the channel attributes bag */
        /*****
        mqInquireString(cAttrsBag, MQCACH_CHANNEL_NAME, 0, MQ_OBJECT_NAME_LENGTH,
                       chlName, &chlNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get channel name", compCode, reason);

        /*****
        /* Get the channel type out of the channel attributes bag */
        /*****
        mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                        &compCode, &reason);
        CheckCallResult("Get type", compCode, reason);

        /*****
        /* Use mqTrim to prepare the channel name for printing. */
        /* Print the result. */
        /*****
        mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
        sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
        WRITEOUTFILE(outfp, OutputBuffer, 29)

```

```

    }
}

else                                     /* Failed mqExecute */
{
    printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
           compCode, reason);
    /******
    /* If the command fails get the system bag handle out of the mqexecute */
    /* response bag.This bag contains the reason from the command server */
    /* why the command failed. */
    /******
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
                    &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /******
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /******
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                        &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                        &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag",
                        compCode, reason);
        printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
               mqExecuteCC, mqExecuteRC);
    }
}

MOD_EXIT:
/******
/* Delete the admin bag if successfully created. */
/******
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/******
/* Delete the response bag if successfully created. */
/******
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/******
/* Disconnect from the queue manager if not already connected */
/******
if (connReason != MQRCA_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
}

/******
/* Close the output file if open */
/******
if(outfp != NULL)
    CLOSEOUTFILE(outfp);

return 0;
}

/******
/*
/* Function: CheckCallResult */
/*
/*
/******
/*
/* Input Parameters: Description of call */
/*                   Completion code */
/*                   Reason code */
/*

```

```

/*                                                                    */
/* Output Parameters: None                                           */
/*                                                                    */
/* Logic: Display the description of the call, the completion code and the */
/*       reason code if the completion code is not successful         */
/*                                                                    */
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %ld : Reason = %ld\n", callText,
              cc, rc);
}

```

Multi 큐 및 인쇄 정보를 조회하기 위한 샘플 C 프로그램(amqsailq.c)

샘플 C 프로그램 amqsailq.c은(는) MQAI를 사용하여 로컬 큐의 현재 용량을 조회합니다.

```

/*****
/*                                                                    */
/* Program name: AMQSAILQ.C                                           */
/*                                                                    */
/* Description: Sample C program to inquire the current depth of the local */
/*             queues using the IBM MQ Administration Interface (MQAI)   */
/*                                                                    */
/* Statement:   Licensed Materials - Property of IBM                 */
/*             84H2000, 5765-B73                                       */
/*             84H2001, 5639-B42                                       */
/*             84H2002, 5765-B74                                       */
/*             84H2003, 5765-B75                                       */
/*             84H2004, 5639-B43                                       */
/*                                                                    */
/*             (C) Copyright IBM Corp. 1999, 2024                    */
/*                                                                    */
/*****
/*
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/*   bag.
/*   These are:-
/*     - The generic queue name "*"
/*     - The type of queue required. In this sample we want to
/*       inquire local queues.
/*     - The attribute to be inquired. In this sample we want the
/*       current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/*   The call generates the correct PCF structure.
/*   The default options to the call are used so that the command is sent
/*   to the SYSTEM.ADMIN.COMMAND.QUEUE.
/*   The reply from the command server is placed on a temporary dynamic
/*   queue.
/*   The reply from the MQCMD_INQUIRE_Q command is read from the
/*   temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/*   is a failure from the command server, then the code returned by
/*   command server is retrieved from the system bag that has been
/*   embedded in the response bag to the mqExecute call.
/*
/* - If the call is successful, the depth of each local queue is placed
/*   in system bags embedded in the response bag of the mqExecute call.
/*   The name and depth of each queue is obtained from each of the bags
/*   and the result displayed on the screen.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
/*****

```

```

/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h> /* MQI */
#include <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */

/*****
/* Function prototypes
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    /*****
    MQHCONN hConn; /* handle to IBM MQ connection */
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG qAttrsBag; /* bag containing q attributes */
    MQHBAG errorBag; /* bag containing cmd server error */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */
    MQLONG qNameLength; /* Actual length of q name */
    MQLONG qDepth; /* depth of queue */
    MQLONG i; /* loop counter */
    MQLONG numberOfBags; /* number of bags in response bag */
    MQCHAR qName[MQ_Q_NAME_LENGTH+1]; /* name of queue extracted from bag*/

    printf("Display current depths of local queues\n\n");

    /*****
    /* Connect to the queue manager
    /*****
    if (argc > 1)
        strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQHCONN(hConn, &compCode, &connReason);

    /*****
    /* Report the reason and stop if the connection failed.
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("Queue Manager connection", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Create an admin bag for the mqExecute call
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
    CheckCallResult("Create admin bag", compCode, reason);
    /*****
    /* Create a response bag for the mqExecute call
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
    CheckCallResult("Create response bag", compCode, reason);

    /*****
    /* Put the generic queue name into the admin bag
    /*****
    mqAddString(adminBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, "*",
        &compCode, &reason);
    CheckCallResult("Add q name", compCode, reason);

    /*****
    /* Put the local queue type into the admin bag
    /*****

```

```

mqAddInteger(adminBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type", compCode, reason);

/*****
/* Add an inquiry for current queue depths */
*****/
mqAddInquiry(adminBag, MQIA_CURRENT_Q_DEPTH, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the local queue names and queue depths. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
*****/
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_INQUIRE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

/*****
/* Check the command server is started. If not exit. */
*****/
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call. If successful find the current */
/* depths of all the local queues. If failed find the error. */
*****/
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each queue are in a separate bag. */
    *****/
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
                 &reason);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the queue attributes */
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &qAttrsBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the queue name out of the queue attributes bag */
        *****/
        mqInquireString(qAttrsBag, MQCA_Q_NAME, 0, MQ_Q_NAME_LENGTH, qName,
                      &qNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get queue name", compCode, reason);

        /*****
        /* Get the depth out of the queue attributes bag */
        *****/
        mqInquireInteger(qAttrsBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
                       &compCode, &reason);
        CheckCallResult("Get depth", compCode, reason);

        /*****
        /* Use mqTrim to prepare the queue name for printing. */
        /* Print the result. */
        *****/
        mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason);
        printf("%4d %-48s\n", qDepth, qName);
    }
}

```

```

    }
}

else /* Failed mqExecute */
{
    printf("Call to get queue attributes failed: Completion Code = %d :
        Reason = %d\n", compCode, reason);

    /******
    /* If the command fails get the system bag handle out of the mqExecute */
    /* response bag. This bag contains the reason from the command server */
    /* why the command failed. */
    /******
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag, &compCode,
            &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /******
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /******
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
            &compCode, &reason );
        CheckCallResult("Get the completion code from the result bag",
            compCode, reason);
        mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
            &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag",
            compCode, reason);
        printf("Error returned by the command server: Completion Code = %d :
            Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}

/******
/* Delete the admin bag if successfully created. */
/******
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/******
/* Delete the response bag if successfully created. */
/******
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/******
/* Disconnect from the queue manager if not already connected */
/******
if (connReason != MQRCA_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from queue manager", compCode, reason);
}
return 0;
}

*****
*
* Function: CheckCallResult
*
*
*****
*
* Input Parameters: Description of call
*                   Completion code
*                   Reason code
*
* Output Parameters: None
*
* Logic: Display the description of the call, the completion code and the
*        reason code if the completion code is not successful
*
*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)

```

```

{
  if (cc != MQCC_OK)
    printf("%s failed: Completion Code = %d : Reason = %d\n",
          callText, cc, rc);
}

```

Multi 데이터 백 및 MQAI

데이터 백은 MQAI(IBM MQ Administration Interface)를 사용하여 오브젝트의 매개변수 또는 특성을 처리하는 방법입니다.

데이터 백

- 데이터 백에는 0개 이상의 데이터 항목이 포함됩니다. 이러한 데이터 항목은 백에 있을 때 백 내에 순서화됩니다. 이는 삽입 순서라고 합니다. 각 데이터 항목에는 정수, 64비트 정수, 정수 필터, 문자열, 문자열 필터, 바이트 문자열, 바이트 문자열 필터 또는 다른 백의 처리가 될 수 있는 해당 데이터 항목의 값 및 데이터 항목을 식별하는 선택자가 포함됩니다. 데이터 항목은 63 페이지의 『MQAI에서 사용 가능한 데이터 항목의 유형』에서 상세하게 설명됩니다.

두 가지 유형의 선택자(사용자 선택자 및 시스템 선택자)가 있습니다. 이는 MQAI 선택자에서 설명됩니다. 선택자는 일반적으로 고유하지만 동일한 선택자에 대해 여러 값을 가질 수 있습니다. 이 경우, 색인은 필수적인 선택자의 특정 발생을 식별합니다. 색인은 37 페이지의 『MQAI의 색인 작성』에 설명되어 있습니다.

이러한 개념의 계층이 그림 1에 표시되어 있습니다.

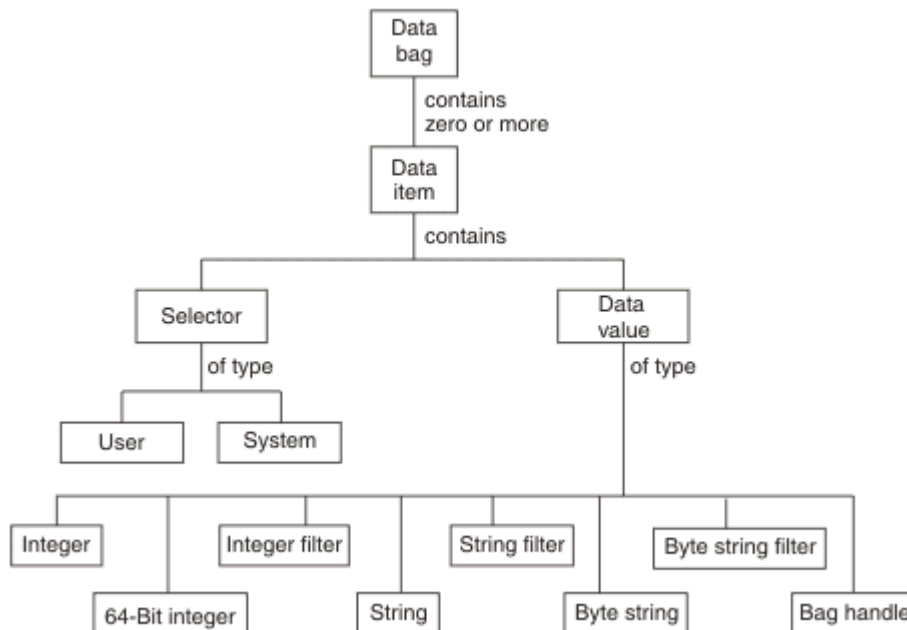


그림 4. MQAI 개념의 계층

계층은 이전 단락에서 설명되었습니다.

데이터 백의 유형

수행하려는 태스크에 따라 작성하려는 데이터 백의 유형을 선택할 수 있습니다.

사용자 백(user bag)

사용자 데이터에 사용되는 단순 백.

관리 백(administration bag)

관리 메시지를 명령 서버에 보냄으로써 IBM MQ 오브젝트를 관리하기 위해 사용되는 데이터를 위해 작성되는 백. 관리 백은 62 페이지의 『데이터 백 작성 및 삭제』에 설명된 대로 특정 옵션을 자동으로 구현합니다.

명령 백(command bag)

IBM MQ 오브젝트 관리를 위한 명령에도 작성된 백. 그러나, 관리 백과 달리, 명령 백은 이러한 옵션이 사용 가능할지라도 자동으로 특정 옵션을 의미하지는 않습니다. 옵션에 대한 자세한 정보는 62 페이지의 『데이터 백 작성 및 삭제』의 내용을 참조하십시오.

그룹 백

그룹화된 데이터 항목 세트에 사용되는 백. 그룹 백은 IBM MQ 오브젝트를 관리하기 위해 사용될 수 없습니다.

또한, **시스템 백**이 응답 메시지가 명령 서버에서 리턴되고 사용자의 출력 백에 배치될 때 MQAI에서 작성됩니다. 시스템 백은 사용자에게 의해 수정될 수 없습니다.

데이터 백 사용: 데이터 백을 사용하는 여러가지 방법이 이 토픽에 나열됩니다.

데이터 백 사용

데이터 백을 사용하는 여러가지 방법이 다음 목록에 표시됩니다.

- 데이터 백을 작성하고 삭제할 수 있습니다. 62 페이지의 『데이터 백 작성 및 삭제』
- 데이터 백을 사용하여 애플리케이션 사이에 데이터를 송신할 수 있습니다. 63 페이지의 『MQAI를 사용하는 데이터 백 넣기 및 수신』
- 데이터 백에 데이터 항목을 추가할 수 있습니다. 64 페이지의 『MQAI를 사용하여 백에 데이터 항목 추가』
- 데이터 백 내에서 조회 명령을 추가할 수 있습니다. 65 페이지의 『백에 조회 명령 추가』
- 데이터 백 내에서 조회할 수 있습니다. 65 페이지의 『데이터 백 내에서 조회』
- 데이터 백 내에서 데이터 항목을 셀 수 있습니다. 68 페이지의 『데이터 항목 계산』
- 데이터 백 내에서 정보를 변경할 수 있습니다. 66 페이지의 『백 내의 정보 변경』
- 데이터 백을 지울 수 있습니다. 67 페이지의 『mqClearBag 호출을 사용하여 백 지우기』
- 데이터 백을 자를 수 있습니다. 67 페이지의 『mqTruncateBag 호출을 사용하여 백 자르기』
- 백 및 버퍼를 변환할 수 있습니다. 67 페이지의 『백 및 버퍼 변환』

Multi

데이터 백 작성 및 삭제

데이터 백 작성

MQAI를 사용하려면 먼저 mqCreateBag 호출을 사용하여 데이터 백을 작성하십시오. 이 호출을 입력할 때에는 백의 작성을 제어하기 위해 하나 이상의 옵션을 제공합니다.

MQCreateBag 호출의 **Options** 매개변수는 사용자 백, 명령 백, 그룹 백 또는 관리 백을 작성할지를 선택하게 합니다.

사용자 백, 명령 백 또는 그룹 백을 작성하려면, 다음을 수행하기 위해 하나 이상의 추가 옵션을 선택할 수 있습니다.

- 백에 동일한 선택자가 두 번 이상 인접하여 나타나는 경우 목록 형식을 사용하십시오.
- 매개변수가 올바른 순서로 되어 있도록 하기 위해 PCF 메시지에 데이터 항목을 추가할 때 데이터 항목을 재정렬하십시오. 데이터 항목에 대한 자세한 정보는 63 페이지의 『MQAI에서 사용 가능한 데이터 항목의 유형』의 내용을 참조하십시오.
- 백에 추가할 항목에 대해 사용자 선택자 값을 검사하십시오.

관리 백은 자동으로 이러한 옵션을 내포합니다.

데이터 백은 해당 핸들로 식별됩니다. 백 핸들은 mqCreateBag으로부터 리턴되며 데이터 백을 사용하는 다른 모든 호출에 제공되어야 합니다.

mqCreateBag 호출에 대한 전체 설명은 [mqCreateBag](#)을 참조하십시오.

데이터 백 삭제

사용자가 작성한 데이터 백은 mpDeleteBag 호출을 사용하여 삭제되어야 합니다. 예를 들어, 백이 사용자 코드로 작성된 경우, 사용자 코드로 삭제되어야 합니다.

시스템 백은 MQAI가 자동으로 작성하고 삭제합니다. 이에 대한 자세한 정보는 69 페이지의 『mqExecute 호출을 사용하여 qm 명령 서버에 관리 명령 전송』의 내용을 참조하십시오. 사용자 코드는 시스템 백을 삭제할 수 없습니다.

mqDeleteBag 호출에 대한 전체 설명은 [mqDeleteBag](#)을 참조하십시오.

Multi MQAI를 사용하는 데이터 백 넣기 및 수신

mqPutBag 및 mqGetBag 호출을 사용하여 데이터 백을 넣고 가져오는 것으로 애플리케이션 사이에 데이터를 송신할 수도 있습니다. 이는 MQAI(IBM MQ Administration Interface)가 애플리케이션이 아닌 버퍼를 처리하게 합니다.

mqPutBag 호출은 지정된 백의 콘텐츠를 PCF 메시지로 변환하고 메시지를 지정된 큐로 보내고 mqGetBag 호출은 지정된 큐로부터 메시지를 제거하고 이를 데이터 백으로 변환합니다. 따라서, mqPutBag 호출은 MQPUT가 뒤에 오는 mqBagToBuffer 호출과 같고 mqGetBag은 mqBufferToBag이 뒤에 오는 MQGET 호출과 같습니다.

특정 큐에서 PCF 메시지를 보내고 수신하는 것에 대한 정보는 27 페이지의 『지정된 큐에서 PCF 메시지 송신 및 수신』의 내용을 참조하십시오.

참고: mqGetBag 호출을 사용하기 위해 선택하는 경우, 메시지 내의 PCF 세부사항은 정확해야 합니다. 그렇지 않으면, 적절한 오류 결과 및 PCF 메시지가 리턴되지 않습니다.

Multi MQAI에서 사용 가능한 데이터 항목의 유형

데이터 항목은 작성될 때 데이터 백을 채우기 위해 MQAI(IBM MQ Administration Interface)에서 사용됩니다. 이러한 데이터 항목은 사용자 또는 시스템 항목일 수 있습니다.

이러한 사용자 항목은 관리되고 있는 오브젝트의 속성과 같은 사용자 데이터를 포함합니다. 시스템 항목은 생성된 메시지를 통해 더 많은 제어를 위해 사용되어야 합니다(예: 메시지 헤더의 생성). 시스템 항목에 대한 자세한 정보는 64 페이지의 『시스템 항목 및 MQAI』의 내용을 참조하십시오.

데이터 항목의 유형

데이터 백을 작성할 때, 정수 또는 문자열 항목으로 채울 수 있습니다. 세 종류의 모든 항목에 대해 조회할 수 있습니다.

데이터 항목은 정수 또는 문자열 항목일 수 있습니다. MQAI에서 사용 가능한 데이터 항목의 유형은 다음과 같습니다.

- 정수
- 64비트 정수
- 정수 필터
- 문자열
- 문자열 필터
- 바이트 문자열
- 바이트 문자열 필터
- 백 핸들

데이터 항목 사용

데이터 항목을 사용하는 방법은 다음과 같습니다.

- 68 페이지의 『데이터 항목 계산』.

- 68 페이지의 『데이터 항목 삭제』.
- 64 페이지의 『MQAI를 사용하여 백에 데이터 항목 추가』.
- 65 페이지의 『데이터 항목 필터링 및 조회』.

Multi 시스템 항목 및 MQAI

MQAI(IBM MQ Administration Interface)에서 사용할 수 있는 시스템 항목은 다음과 같습니다.

- PCF 헤더의 생성. 시스템 항목은 PCF 명령 ID, 제어 옵션, 메시지 순서 매기기 및 명령 유형을 제어할 수 있습니다.
- 데이터 변환. 시스템 항목은 백에서 문자열 항목을 위한 문자 세트 ID를 핸들링합니다.

다른 모든 데이터 항목처럼, 시스템 항목은 선택자와 값으로 구성됩니다. 이러한 선택자 및 이의 용도에 대한 정보는 [MQAI 선택자를 참조하십시오](#).

시스템 항목은 고유합니다. 하나 이상의 시스템 항목을 시스템 선택자로 식별될 수 있습니다. 각 시스템 선택자가 하나만 발생합니다.

대부분의 시스템 항목은 수정될 수 있지만(66 페이지의 『백 내의 정보 변경』의 내용 참조), 백 작성 옵션이 사용자에 의해 변경될 수 없습니다. 시스템 항목은 삭제할 수 없습니다 (68 페이지의 『데이터 항목 삭제』의 내용 참조.)

Multi MQAI를 사용하여 백에 데이터 항목 추가

MQAI(IBM MQ Administration Interface)를 사용하여 데이터 백이 작성될 때, 데이터 항목으로 채울 수 있습니다. 이러한 데이터 항목은 사용자 또는 시스템 항목일 수 있습니다.

데이터 항목에 대한 자세한 정보는 63 페이지의 『MQAI에서 사용 가능한 데이터 항목의 유형』의 내용을 참조하십시오.

MQAI는 정수 항목과 64비트 정수 항목, 정수 필터 항목, 문자열 항목, 문자열 필터, 바이트 문자열 항목과 바이트 문자열 필터 항목을 백에 추가하게 하며 이것은 64 페이지의 그림 5에 표시되어 있습니다. 항목은 선택자로 식별됩니다. 일반적으로 한 선택자는 한 항목만을 식별하지만 항상 그런 것은 아닙니다. 지정된 선택자가 있는 데이터 항목이 이미 백에 있는 경우, 선택자의 추가 인스턴스가 백 끝에 추가됩니다.

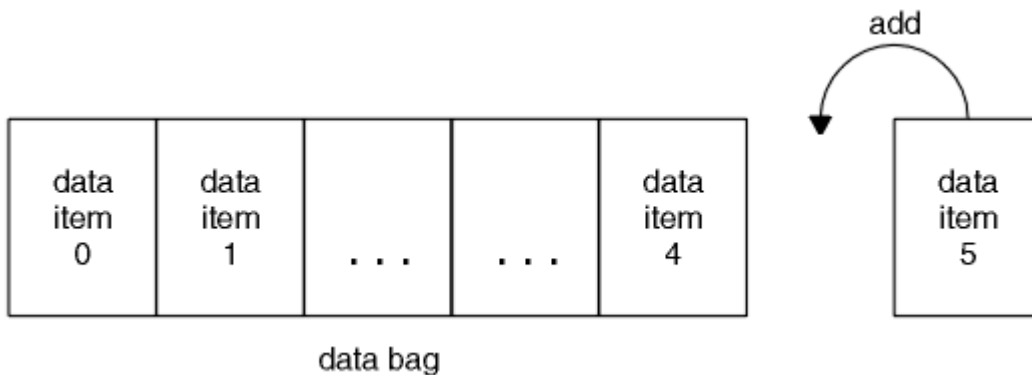


그림 5. 데이터 항목 추가

mqAdd* 호출을 사용하여 데이터 항목을 백에 추가하십시오.

- 정수 항목을 추가하려면 [mqAddInteger](#)에 설명된 대로 mqAddInteger 호출을 사용하십시오.
- 64비트 정수 항목을 추가하려면 [mqAddInteger64](#)에 설명된 대로 mqAddInteger64 호출을 사용하십시오.
- 정수 필터 항목을 추가하려면 [mqAddIntegerFilter](#)에 설명된 대로 mqAddIntegerFilter 호출을 사용하십시오.
- 문자열 항목을 추가하려면 [mqAddString](#)에 설명된 대로 mqAddString 호출을 사용하십시오.
- 문자열 필터 항목을 추가하려면 [mqAddStringFilter](#)에 설명된 대로 mqAddStringFilter 호출을 사용하십시오.
- 바이트 문자열 항목을 추가하려면 [mqAddByteString](#)에 설명된 대로 mqAddByteString 호출을 사용하십시오.
- 바이트 문자열 필터 항목을 추가하려면 [mqAddByteStringFilter](#)에 설명된 대로 mqAddByteStringFilter 호출을 사용하십시오.

데이터 항목을 백에 추가하는 것에 대한 자세한 정보는 64 페이지의 『시스템 항목 및 MQAI』의 내용을 참조하십시오.

Multi 백에 조회 명령 추가

mqAddInquiry 호출은 백에 조회 명령을 추가하는 데 사용됩니다. 호출은 특히 관리 목적이므로 관리 백과 함께 사용될 수 있습니다. 이는 IBM MQ에서 조회하려는 속성의 선택자를 지정하도록 합니다.

mqAddInquiry 호출에 대한 전체 설명은 [mqAddInquiry](#)를 참조하십시오.

Multi 데이터 항목 필터링 및 조회

IBM MQ 오브젝트의 속성에 대해 조회하기 위해 MQAI를 사용할 때, 두 가지 방법으로 프로그램으로 리턴되는 데이터를 제어할 수 있습니다.

- mqAddInteger 및 mqAddString 호출을 사용하여 리턴되는 데이터를 **필터링** 할 수 있습니다. 이 접근방법은 *Selector* 및 *ItemValue* 쌍을 지정하게 합니다. 예:

```
mqAddInteger(inputbag, MQIA_Q_TYPE, MQQT_LOCAL)
```

이 예는 큐 유형(*Selector*)이 로컬이어야 하고(*ItemValue*) 조회 중인 것에 대한 오브젝트의 속성(이 경우, 큐)과 이 스펙이 일치해야 한다는 것을 지정합니다.

필터링될 수 있는 기타 속성은 24 페이지의 『IBM MQ PCF(Programmable Command Format) 소개』에서 찾을 수 있는 PCF Inquire* 명령에 해당됩니다. 예를 들어, 채널의 속성에 대해 조회하려면 이 제품 문서에서 채널 조회 명령을 참조하십시오. 채널 조회 명령의 "필수 매개변수" 및 "선택적 매개변수"는 필터링에 사용할 수 있는 선택자를 식별합니다.

- mqAddInquiry 호출을 사용하는 오브젝트의 특정 속성을 **조회** 할 수 있습니다. 이는 관심있는 선택자를 지정합니다. 선택자를 지정하지 않으면, 오브젝트의 모든 속성이 리턴됩니다.

다음은 큐 속성의 필터링 및 조회의 예입니다.

```
/* Request information about all queues */
mqAddString(adminbag, MQCA_Q_NAME, "*")

/* Filter attributes so that local queues only are returned */
mqAddInteger(adminbag, MQIA_Q_TYPE, MQQT_LOCAL)

/* Query the names and current depths of the local queues */
mqAddInquiry(adminbag, MQCA_Q_NAME)
mqAddInquiry(adminbag, MQIA_CURRENT_Q_DEPTH)

/* Send inquiry to the command server and wait for reply */
mqExecute(MQCMD_INQUIRE_Q, ...)
```

Multi 데이터 백 내에서 조회

다음에 대해 조회할 수 있습니다.

- mqInquireInteger 호출을 사용하여 정수 항목 값. [mqInquireInteger](#)의 내용을 참조하십시오.
- mqInquireInteger64 호출을 사용하는 64비트 정수 항목의 값. [mqInquireInteger64](#)의 내용을 참조하십시오.
- mqInquireIntegerFilter 호출을 사용하는 정수 필터 항목의 값. [mqInquireIntegerFilter](#)의 내용을 참조하십시오.
- mqInquireString 호출을 사용하여 문자열 항목 값. [mqInquireString](#)의 내용을 참조하십시오.
- mqInquireStringFilter 호출을 사용하는 문자열 필터 항목의 값. [mqInquireStringFilter](#)의 내용을 참조하십시오.
- mqInquireByteString 호출을 사용하는 바이트 문자열 항목의 값. [mqInquireByteString](#)의 내용을 참조하십시오.
- mqInquireByteStringFilter 호출을 사용하는 바이트 문자열 필터 항목의 값. [mqInquireByteStringFilter](#)의 내용을 참조하십시오.

- mqInquireBag 호출을 사용하는 백 핸들 값. mqInquireBag의 내용을 참조하십시오.

또한 mqInquireItemInfo 호출을 사용하여 특정 항목의 유형(정수, 64비트 정수, 정수 필터, 문자열, 문자열 필터, 바이트 문자열, 바이트 문자열 필터 또는 백 핸들)에 대해 조사할 수 있습니다. mqInquireItemInfo의 내용을 참조하십시오.

Multi 백 내의 정보 변경

MQAI는 mqSet* 호출을 사용하여 백 내의 정보를 변경하도록 합니다. 다음을 수행할 수 있습니다.

1. 백 내의 데이터 항목을 수정합니다. 색인은 수정될 항목의 발생을 식별하여 매개변수의 개별 인스턴스가 대체될 수 있게 합니다(66 페이지의 그림 6의 내용 참조).

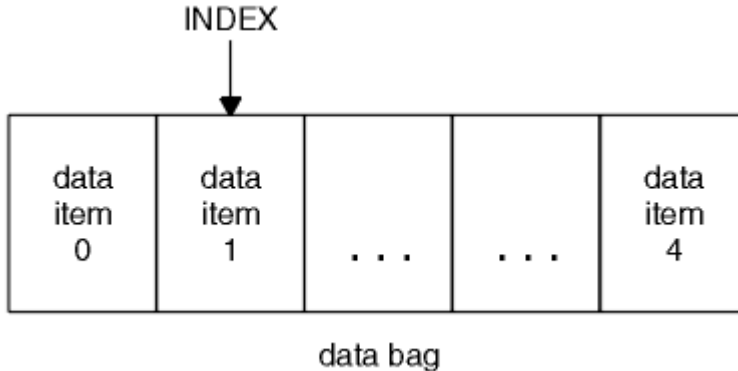


그림 6. 단일 데이터 항목 수정

2. 지정된 선택자의 기존 모든 발생을 삭제하고 백의 끝에 새로운 발생을 추가합니다. (66 페이지의 그림 7의 내용 참조.) 특수 색인 값을 사용하면 매개변수의 모든 인스턴스가 대체될 수 있습니다.

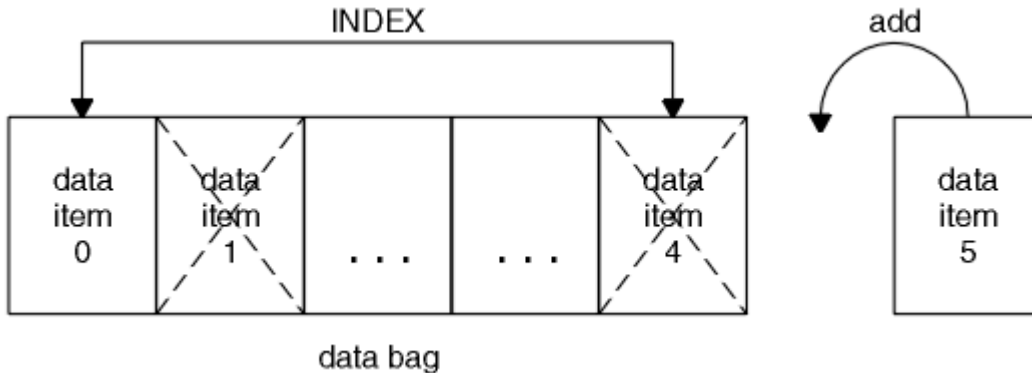


그림 7. 모든 데이터 항목 수정

참고: 색인은 백 내의 삽입 순서를 보존하지만, 기타 데이터 항목의 색인에 영향을 미칠 수 있습니다.

mqSetInteger 호출을 사용하여 백 내의 정수 항목을 수정하십시오. mqSetInteger64 호출을 사용하여 64비트 정수 항목을 수정하십시오. mqSetIntegerFilter 호출을 사용하여 정수 필터 항목을 수정하십시오. mqSetString 호출을 사용하여 문자열 항목을 수정하십시오. mqSetStringFilter 호출을 사용하여 문자열 필터 항목을 수정하십시오. mqSetByteString 호출을 사용하여 바이트 문자열 항목을 수정하십시오. mqSetByteStringFilter 호출을 사용하여 바이트 문자열 필터 항목을 수정하십시오. 또는 이러한 호출을 사용하여 지정된 선택자의 기존 모든 발생을 삭제하고 백의 끝에 새 발생을 추가할 수 있습니다. 데이터 항목은 사용자 항목 또는 시스템 항목입니다.

이러한 호출에 대한 전체 설명은 다음을 참조하십시오.

- [mqSetInteger](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetString](#)
- [mqSetStringFilter](#)

- [mqSetByteString](#)
- [mqSetByteStringFilter](#)

Multi [mqClearBag](#) 호출을 사용하여 백 지우기

[mqClearBag](#) 호출은 사용자 백에서 모든 사용자 항목을 제거하고 초기값으로 시스템 항목을 재설정합니다. 백에 포함된 시스템 백도 삭제됩니다.

[mqClearBag](#) 호출에 대한 전체 설명은 [mqClearBag](#)을 참조하십시오.

Multi [mqTruncateBag](#) 호출을 사용하여 백 자르기

[mqTruncateBag](#) 호출은 가장 최근 추가된 항목부터 백의 끝에서 항목을 삭제하여 사용자 백에 있는 사용자 항목의 수를 줄입니다. 예를 들어, 둘 이상의 메시지를 생성하기 위해 동일한 헤더 정보를 사용할 때 사용될 수 있습니다.

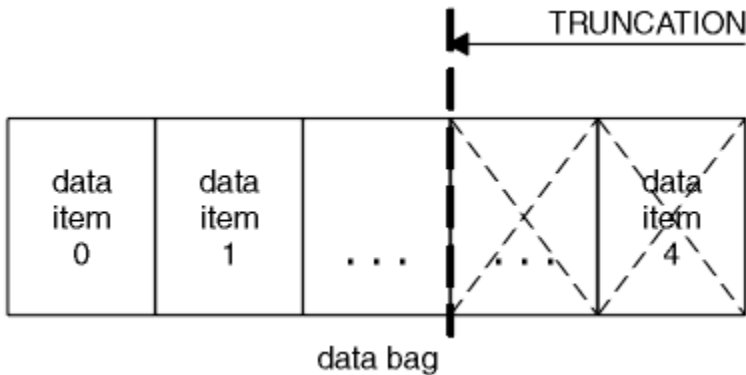


그림 8. 백 자르기

[mqTruncateBag](#) 호출에 대한 전체 설명은 [mqTruncateBag](#)을 참조하십시오.

Multi 백 및 버퍼 변환

애플리케이션 사이에 데이터를 송신하려면, 먼저 메시지 데이터를 백에 놓습니다. 그런 다음, [mqBagToBuffer](#) 호출을 사용하여 백에 있는 데이터를 PCF 메시지로 변환합니다. [MQPUT](#) 호출을 사용하여 PCF 메시지를 필수 큐로 송신합니다. 이 내용은 67 페이지의 그림 9에 표시되어 있습니다. [mqBagToBuffer](#) 호출에 대한 전체 설명은 [mqBagToBuffer](#)를 참조하십시오.

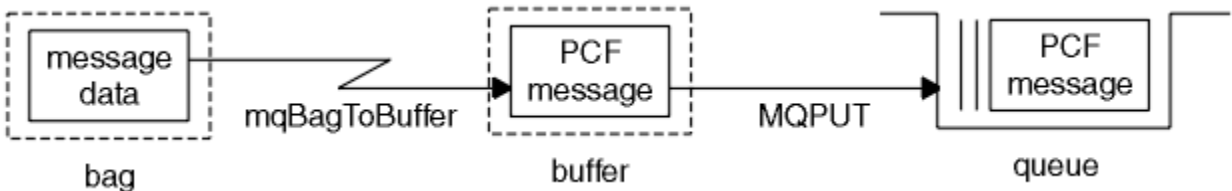


그림 9. PCF 메시지로 백 변환

데이터를 수신하려면, 먼저 메시지가 [MQGET](#) 호출을 사용하여 버퍼로 수신됩니다. 그런 다음, 버퍼에 올바른 PCF 메시지가 있다는 가정 하에 [mqBufferToBag](#) 호출을 사용하여 버퍼에 있는 데이터를 백으로 변환합니다. 이 내용은 67 페이지의 그림 10에 표시되어 있습니다. [mqBufferToBag](#) 호출에 대한 전체 설명은 [mqBufferToBag](#)을 참조하십시오.

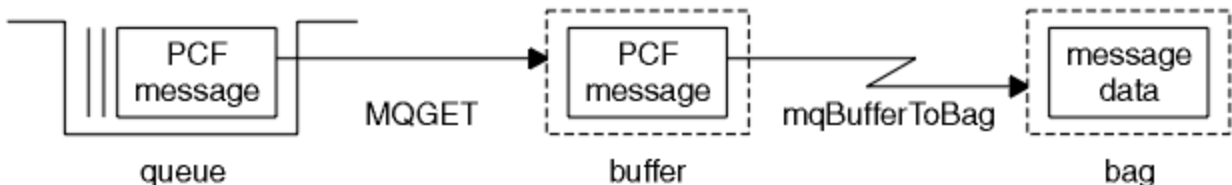


그림 10. 백 양식으로 PCF 메시지 변환

Multi 데이터 항목 계산

mqCountItems 호출로 데이터 백에 저장된 사용자 항목, 시스템 항목 또는 모두의 수를 계산하여 이 숫자를 리턴할 수 있습니다. 예를 들어, mqCountItems(Bag, 7, ...)는 선택자가 7인백의 항목 수를 리턴합니다. 개별 선택자, 사용자 선택자, 시스템 선택자 또는 모든 선택자별로 항목을 계수할 수 있습니다.

참고: 이 호출은 백에 있는 고유 선택자 수가 아닌 데이터 항목 수를 계수합니다. 선택자는 여러 번 발생할 수 있으므로 데이터 항목 수보다 백에 있는 고유 선택자 수가 더 적을 수 있습니다.

mqCountItems 호출에 대한 전체 설명은 [mqCountItems](#)를 참조하십시오.

Multi 데이터 항목 삭제

여러 방식으로 백에서 항목을 삭제할 수 있습니다. 다음을 수행할 수 있습니다.

- 백에서 하나 이상의 사용자 항목을 제거합니다. 자세한 정보는 68 페이지의 『mqDeleteItem 호출을 사용하여 백에서 데이터 항목 삭제』의 내용을 참조하십시오.
- 백에서 모든 사용자 항목을 삭제합니다(백 지우기). 자세한 정보는 67 페이지의 『mqClearBag 호출을 사용하여 백 지우기』의 내용을 참조하십시오.
- 백의 끝에서 사용자 항목을 삭제합니다(백 자르기). 자세한 정보는 67 페이지의 『mqTruncateBag 호출을 사용하여 백 자르기』의 내용을 참조하십시오.

Multi mqDeleteItem 호출을 사용하여 백에서 데이터 항목 삭제

mqDeleteItem 호출은 백에서 하나 이상의 사용자 항목을 제거합니다. 색인을 사용하여 다음 중 하나를 삭제합니다.

1. 지정된 선택자의 단일 발생. (68 페이지의 그림 11의 내용 참조.)

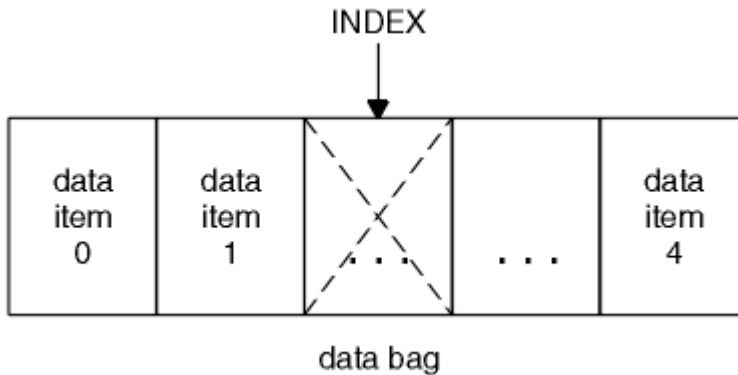


그림 11. 단일 데이터 항목 삭제

또는

2. 지정된 선택자의 모든 발생. (68 페이지의 그림 12의 내용 참조.)

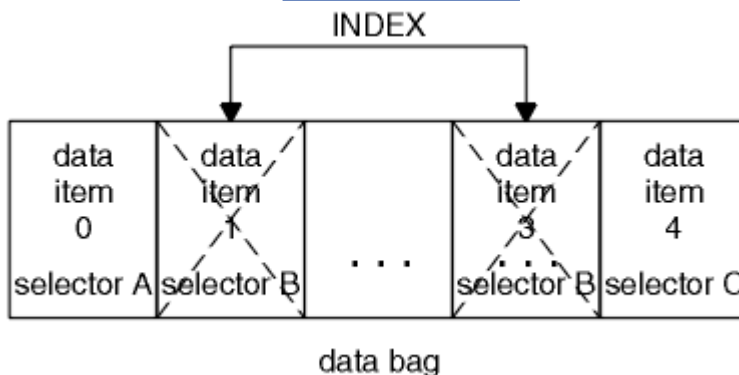


그림 12. 모든 데이터 항목 삭제

참고: 색인은 백 내의 삽입 순서를 보존하지만, 기타 데이터 항목의 색인에 영향을 미칠 수 있습니다. 예를 들어, mqDeleteItem 호출은 삭제된 항목 다음에 나오는 데이터 항목의 색인 값을 보존하지 않습니다. 삭제된 항목으로 인해 남은 공간을 채우기 위해 색인을 재구성하기 때문입니다.

mqDeleteItem 호출에 대한 전체 설명은 [mqDeleteItem](#)을 참조하십시오.

Multi mqExecute 호출을 사용하여 qm 명령 서버에 관리 명령 전송

데이터 백이 작성되고 채워지면, 관리 명령 메시지가 mqExecute 호출을 사용하여 큐 관리자의 명령 서버에 보내질 수 있습니다. 이는 명령 서버와의 교환을 처리하고 백에 응답을 리턴합니다.

데이터 백을 작성하고 채운 후, 관리 명령 메시지를 큐 관리자의 명령 서버에 보낼 수 있습니다. 이를 수행하는 가장 쉬운 방법은 mqExecute 호출을 사용하는 것입니다. mqExecute 호출은 관리 명령 메시지를 비지속 메시지로 보내고 응답을 기다립니다. 응답은 응답 백에 리턴됩니다. 이는 여러 IBM MQ 오브젝트 또는 일련의 PCF 오류 응답 메시지와 관련되는 속성에 대한 정보를 포함할 수 있습니다. 따라서, 응답 백은 리턴 코드만 포함할 수도 있고 중첩 백을 포함할 수도 있습니다.

응답 메시지는 시스템이 작성하는 시스템 백에 배치됩니다. 예를 들어, 오브젝트 이름에 대해 조회하기 위해 시스템 백이 해당 오브젝트 이름을 보유하도록 작성되고 해당 백이 사용자 백에 삽입됩니다. 그런 다음 이러한 백에 대한 핸들이 응답 백에 삽입되고 MQHA_BAG_HANDLE 선택자가 중첩 백에 액세스할 수 있습니다. 시스템 백이 삭제되지 않으면 응답 백이 삭제될 때까지 스토리지에 계속 있습니다.

중첩의 개념이 69 페이지의 [그림 13](#)에 표시됩니다.

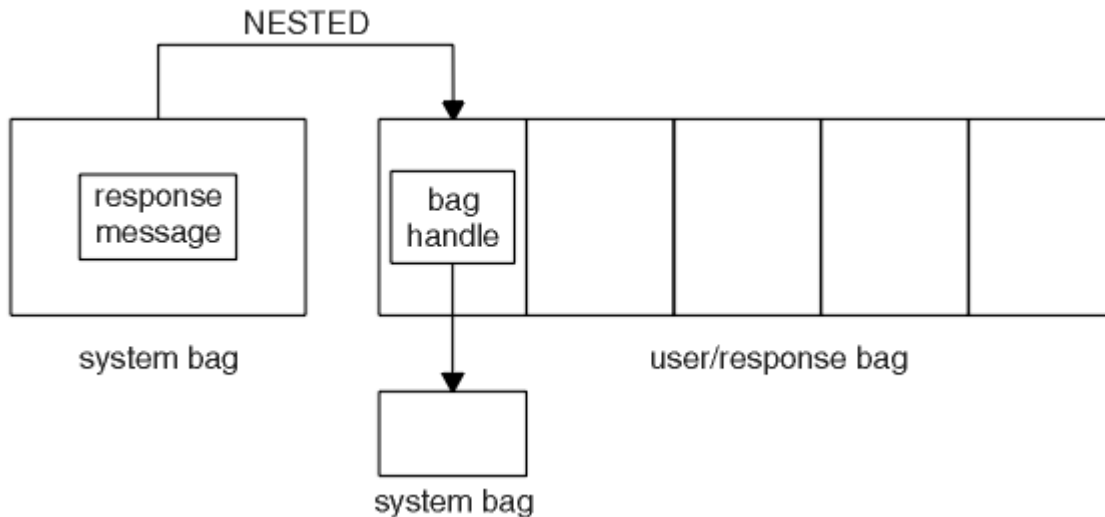


그림 13. 중첩

mqExecute 호출에 대한 입력으로서 다음을 제공해야 합니다.

- MQI 연결 핸들.
- 실행할 명령. MQCMD_* 값 중 하나여야 합니다.

참고: 이 값을 MQAI에서 인식하지 못하는 경우, 값은 여전히 허용됩니다. 그러나, 백에 값을 삽입하기 위해 mqAddInquiry 호출이 사용되는 경우, 이 매개변수는 MQAI에서 인식하는 INQUIRE 명령이어야 합니다. 즉, 매개변수는 MQCMD_INQUIRE_* 양식 중 하나여야 합니다.

- 선택적으로, 호출 처리를 제어하는 옵션을 포함하는 백의 핸들. 이는 또한 MQAI가 각 응답 메시지를 기다려야 하는 최대 시간(밀리초)을 지정할 수 있는 위치입니다.
- 실행할 관리 명령의 세부사항을 포함하는 관리 백의 핸들.
- 응답 메시지를 수신하는 응답 백의 핸들.

다음 핸들은 선택사항입니다.

- 관리 명령이 대체되는 큐의 오브젝트 핸들.

지정되는 오브젝트 핸들이 없는 경우, 관리 명령은 현재 연결된 큐 관리자에 속하는 SYSTEM.ADMIN.COMMAND.QUEUE에 배치됩니다. 기본값입니다.

- 응답 메시지가 대체되는 큐의 오브젝트 핸들.

MQAI에서 자동으로 작성하는 동적 큐에 응답 메시지를 배치하기 위해 선택할 수 있습니다. 작성된 큐는 호출 지속 기간에만 존재하며 mqExecute 호출 종료 시 MQAI에 의해 삭제됩니다.

mqExecute 호출의 사용 예제는 [예제 코드를 참조하십시오](#).

REST API를 사용하여 관리

administrative REST API 를 사용하여 큐 관리자 및 큐, Managed File Transfer 에이전트 및 전송과 같은 IBM MQ 오브젝트를 관리할 수 있습니다. 정보는 JSON 형식으로 administrative REST API에서 송수신됩니다. 이러한 RESTful API를 사용하면 IBM MQ 관리를 인기 있는 DevOps 및 자동화 도구에 임베드할 수 있습니다.

시작하기 전에

참고: **V 9.4.0** administrative REST API 는 독립형 IBM MQ Web Server 설치에서 사용할 수 없습니다. administrative REST API를 실행하는 IBM MQ 컴포넌트의 설치 옵션에 대한 자세한 정보는 [IBM MQ Console](#) 및 REST API의 내용을 참조하십시오.

사용 가능한 REST 자원에 대한 참조 정보는 [administrative REST API 참조](#)를 참조하십시오.

프로시저

- [70 페이지의 『administrative REST API 시작하기』](#)
- [73 페이지의 『administrative REST API 사용』](#)
- [75 페이지의 『REST API를 사용한 원격 관리』](#)
- [79 페이지의 『REST API 시간소인』](#)
- [79 페이지의 『REST API 오류 핸들링』](#)
- [81 페이지의 『REST API 검색』](#)
- [83 페이지의 『REST API 자국어 지원\(NLS\)』](#)

administrative REST API 시작하기

administrative REST API를 신속하게 시작하고, 큐를 작성하고 업데이트하고 보고 삭제하기 위해 cURL로 몇 가지 요청 예를 시도합니다.

시작하기 전에

administrative REST API 사용을 시작하기 위해 이 태스크의 예에는 다음 요구사항이 있습니다.

- 예에서는 시스템의 큐 관리자에 대한 정보를 표시하고 큐를 작성하고 업데이트하고 보고 삭제하기 위해 cURL 을 사용하여 REST 요청을 작성합니다. 따라서 이 태스크를 완료하려면 cURL이 시스템에 설치되어 있어야 합니다.
- 이 태스크를 완료하려면 **dspmqweb** 명령을 사용할 수 있도록 특정 권한이 있는 사용자여야 합니다.

- **z/OS** z/OS의 경우 **dspmqweb** 명령을 실행하는 권한 및 mqwebuser.xml 파일에 대한 쓰기 액세스 권한이 있어야 합니다.

- **Multi** 다른 모든 운영 체제에서는 [권한이 있는 사용자](#)여야 합니다.

- **IBM i** IBM i에서 명령은 QSHELL에서 실행되어야 합니다.

프로시저

1. administrative REST API, MFT용 administrative REST API , messaging REST API 또는 IBM MQ Console 에서 사용할 mqweb 서버를 구성했는지 확인하십시오.
기본 레지스트리를 사용하여 mqweb 서버를 구성하는 방법에 대한 자세한 정보는 [mqweb 서버의 기본 구성](#)을 참조하십시오.

2.

z/OS에서 **dspmweb** 명령을 사용할 수 있도록 **WLP_USER_DIR** 환경 변수를 설정하십시오. 다음 명령을 입력하여 mqweb 서버 구성을 가리키도록 변수를 설정하십시오.

```
export WLP_USER_DIR=WLP_user_directory
```

여기서 **WLP_user_directory** 은 **crtmqweb**에 전달되는 디렉토리의 이름입니다. 예:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

자세한 정보는 [mqweb 서버 작성](#)을 참조하십시오.

3. 결정하다 REST API 다음 명령을 입력하여 URL을 입력하세요.

```
dspmweb status
```

다음 단계의 예에서는 REST API URL은 기본 URL입니다. <https://localhost:9443/ibmmq/rest/v1/> . URL이 기본값과 다른 경우 다음 단계에서 URL을 대체하십시오.

4. mqadmin 사용자로 기본 인증을 사용하여 qmgr 자원에서 GET 요청을 시도하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/qmgr -X GET -u mqadmin:mqadmin
```

5. mqsc 자원을 사용하여 큐를 작성하고, 표시하고, 대체하고 삭제하십시오.

예에서는 큐 관리자 **QM1**을 사용합니다. 동일한 이름으로 큐 관리자를 작성하거나 시스템의 기존 큐 관리자를 대체하십시오.

a) 다음과 같이 mqsc 자원에서 POST 요청을 작성하여 로컬 큐를 작성하십시오.

요청 본문에서 새 큐의 이름이 **Q1**으로 설정됩니다. 기본 인증이 사용되며 임의의 값으로 **ibm-mq-rest-csrf-token** HTTP 헤더가 cURL REST 요청에서 설정됩니다. 이 추가 헤더는 POST, PATCH 및 DELETE 요청에 필요합니다.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "define", "qualifier": "qlocal", "name": "Q1"}'
```

b) 다음과 같이 mqsc 자원에서 POST 요청을 작성하여 71 페이지의 『5.a』 단계에서 작성된 로컬 큐를 보십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

c) 다음과 같이 mqsc 자원에서 OST 요청을 작성하여 큐에 대한 설명을 업데이트하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "alter", "qualifier": "qlocal", "name": "Q1", "parameters": {"descr": "new description"}'}
```

d) mqsc 자원에서 POST 요청을 작성하여 새 큐 설명을 보십시오. 응답에 설명 필드가 포함되도록 다음과 같이 요청 본문에 **responseParameters** 속성을 지정하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1", "responseParameters": [{"descr"}]}'
```

e) 다음과 같이 mqsc 자원에서 POST 요청을 작성하여 큐를 삭제하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --
```

```
data {"type": "runCommandJSON", "command": "delete", "qualifier": "qlocal",
"name": "Q1"}
```

f) 다음과 같이 mqsc 자원에서 POST 요청을 작성하여 큐가 삭제되었는지 증명하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u
mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --
data {"type": "runCommandJSON", "command": "display", "qualifier": "qlocal",
"name": "Q1"}
```



다음에 수행할 작업

- 예에서는 기본 인증을 사용하여 요청을 보냅니다. 토큰 기반 인증 또는 클라이언트 기반 인증을 대신 사용할 수 있습니다. 자세한 정보는 [REST API 및 IBM MQ Console로 클라이언트 인증서 인증 사용 및 REST API로 토큰 기반 인증 사용](#)을 참조하십시오.
- administrative REST API 사용과 조회 매개변수를 사용한 URL 구성에 대해 자세히 알아보십시오. [73 페이지의 『administrative REST API 사용』](#).
- 사용 가능한 administrative REST API 자원 및 사용 가능한 모든 선택적 조회 매개변수에 대한 참조 정보 찾아 보기: [administrative REST API 참조](#).
- administrative REST API를 사용하여 원격 시스템에서 IBM MQ 오브젝트를 관리하는 방법에 대해 학습: [75 페이지의 『REST API를 사용한 원격 관리』](#).
- MFT와 함께 administrative REST API 를 사용하는 방법을 학습하십시오. [72 페이지의 『MFT 용 REST API 시작하기』](#).
- messaging REST API, IBM MQ 메시지를 위한 RESTful 인터페이스 검색: [REST API를 사용한 메시징](#).
- IBM MQ Console, 브라우저 기반 GUI 검색: [86 페이지의 『IBM MQ Console를 사용하여 관리』](#).


MFT 용 REST API 시작하기

Managed File Transfer 용 administrative REST API 를 빠르게 시작하고 몇 가지 예제 요청을 시도하여 MFT 에 이진트 상태를 보고 전송 목록을 보십시오.

시작하기 전에

- 예에서는 전송 목록을 보고 MFT 에이전트 상태를 보기 위해 cURL을 사용하여 REST 요청을 송신합니다. 따라서 이 작업을 완료하려면 cURL이 시스템에 설치되어 있어야 합니다.
- 이 작업을 완료하려면 [dspmqweb](#) 명령을 사용할 수 있도록 특정 권한이 있는 사용자여야 합니다.
 -  z/OS의 경우 [dspmqweb](#) 명령을 실행하는 권한 및 mqwebuser.xml 파일에 대한 쓰기 액세스 권한이 있어야 합니다.
 -  다른 모든 운영 체제에서는 [권한이 있는 사용자](#)여야 합니다.

프로시저

1. mqweb 서버가 MFT용 administrative REST API 에 대해 구성되어 있는지 확인하십시오.
 - administrative REST API, MFT용 administrative REST API , messaging REST API 또는 IBM MQ Console에서 사용할 mqweb 서버를 구성했는지 확인하십시오. 기본 레지스트리를 사용하여 mqweb 서버를 구성하는 방법에 대한 자세한 정보는 [mqweb 서버의 기본 구성](#)을 참조하십시오.
 - mqweb 서버가 구성된 경우 mqweb 서버에 대한 [기본 구성](#) 의 8단계가 MFT에 대해 administrative REST API 를 사용으로 설정하도록 완료되었는지 확인하십시오.
2.  z/OS에서 [dspmqweb](#) 명령을 사용할 수 있도록 WLP_USER_DIR 환경 변수를 설정하십시오. 다음 명령을 입력하여 mqweb 서버 구성을 가리키도록 변수를 설정하십시오.

```
export WLP_USER_DIR=WLP_user_directory
```

여기서 `WLP_user_directory` 은 `crtmqweb`에 전달되는 디렉토리의 이름입니다. 예:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

자세한 정보는 [mqweb 서버 작성](#)을 참조하십시오.

3. 결정하다 REST API 다음 명령을 입력하여 URL을 입력하세요.

```
dspmqweb status
```

다음 단계의 예에서는 REST API URL은 기본 URL입니다. `https://localhost:9443/ibmmq/rest/v1/`. URL이 기본값과 다른 경우 다음 단계에서 URL을 대체하십시오.

4. `agent` 자원에 대한 GET 요청을 작성하여 이름, 유형 및 상태를 포함한 모든 에이전트에 대한 기본 세부사항을 리턴하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/agent/ -X GET -u mftadmin:mftadmin
```

5. **fteCreateTransfer** 명령을 사용하여 표시할 몇 가지 전송을 작성하십시오.

mqweb 서버는 전송에 대한 정보를 캐시하고 요청이 작성되면 이 정보를 리턴합니다. 이 캐시는 mqweb 서버가 재시작될 때 재설정됩니다. `console.log` 및 `messages.log` 파일을 보거나 z/OS에서 시작된 태스크의 출력을 보고 서버가 다시 시작되었는지 여부를 확인할 수 있습니다.

6. `transfer` 자원에 대한 GET 요청을 작성하여 mqweb 서버가 시작된 이후 이루어진 최대 4개의 전송에 대한 세부사항을 리턴하십시오.

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/transfer?limit=4 -X GET -u mftadmin:mftadmin
```

다음에 수행할 작업

- 예에서는 기본 인증을 사용하여 요청을 보냅니다. 토큰 기반 인증 또는 클라이언트 기반 인증을 대신 사용할 수 있습니다. 자세한 정보는 [REST API로 토큰 기반 인증 사용 및 REST API 및 IBM MQ Console로 클라이언트 인증서 인증 사용](#)을 참조하십시오.
- administrative REST API 사용과 조회 매개변수를 사용한 URL 구성에 대해 자세히 알아보십시오. [73 페이지의 『administrative REST API 사용』](#).
- 사용 가능한 administrative REST API for MFT 자원 및 사용 가능한 모든 선택적 조회 매개변수에 대한 참조 정보를 찾아보십시오 ([administrative REST API 참조](#)).
- messaging REST API, IBM MQ 메시징을 위한 RESTful 인터페이스 검색: [REST API를 사용한 메시징](#).
- IBM MQ Console, 브라우저 기반 GUI 검색: [86 페이지의 『IBM MQ Console를 사용하여 관리』](#).

administrative REST API 사용

administrative REST API를 사용하는 경우 큐 관리자 또는 큐 등의 다양한 IBM MQ 오브젝트를 나타내는 HTTP 메소드를 URL에서 호출합니다. HTTP 메소드(예: POST)는 URL이 나타내는 오브젝트에서 수행될 조치의 유형을 나타냅니다. 조치에 대한 자세한 정보가 HTTP 메소드 페이로드(payload)의 일부로 JSON 형식으로 제공되거나 조회 매개변수에서 인코딩됩니다. 조치 수행 결과에 대한 정보는 HTTP 응답의 본문으로 리턴될 수 있습니다.

시작하기 전에

administrative REST API를 사용하기 전에 다음 사항을 고려하십시오.

- administrative REST API를 사용하려면 mqweb 서버를 사용하여 인증해야 합니다. HTTP 기본 인증, 클라이언트 인증서 인증 또는 토큰 기반 인증을 사용하여 인증할 수 있습니다. 이러한 인증 방법을 사용하는 방법에 대한 자세한 정보는 [IBM MQ Console 및 REST API 보안](#)을 참조하십시오.
- REST API에서는 대소문자가 구분됩니다. 예를 들어, 다음 URL에 대한 HTTP GET은 큐 관리자가 `qmgr1`일 경우 정보를 표시하지 않습니다.

```
/ibmmq/rest/v1/admin/qmgr/QMGR1
```

- IBM MQ 오브젝트 이름에 사용할 수 있는 모든 문자를 URL로 직접 인코딩할 수 있는 것은 아닙니다. 이러한 문자를 올바르게 인코딩하려면 적절한 URL 인코딩을 사용해야 합니다.
 - 슬래시(/)는 %2F(으)로 인코드되어야 합니다.
 - 퍼센트 부호(%)는 %25(으)로 인코드되어야 합니다.
- 일부 브라우저의 작동에 문제가 생기므로, 마침표 또는 슬래시 문자만을 사용하여 오브젝트의 이름을 지정하지 마십시오.

이 태스크 정보

REST API를 사용하여 오브젝트에 대한 작업을 수행할 경우 먼저 해당 오브젝트를 나타내는 URL을 작성해야 합니다. 각 URL은 요청을 전송할 포트 또는 호스트 이름을 설명하는 접두부로 시작됩니다. URL의 나머지 부분은 특정 오브젝트 또는 오브젝트 세트(자원으로 알려져 있음)를 설명합니다.

자원에 대해 수행될 조치는 URL에 조회 매개변수가 필요한지 여부를 정의합니다. 이 조치는 사용되는 HTTP 메소드, 추가 정보가 JSON 양식으로 URL에 전송되는지 아니면 URL에서 리턴되는지도 정의합니다. 추가 정보가 HTTP 요청의 일부를 구성하거나 HTTP 응답의 일부로 리턴될 수 있습니다.

URL을 생성하고 HTTP 요청에서 전송할 선택적인 JSON 페이로드를 작성한 후 HTTP 요청을 IBM MQ에 전송할 수 있습니다. 선택한 프로그래밍 언어로 기본 제공되는 HTTP 구현을 사용하여 요청을 전송할 수 있습니다. cURL, 웹 브라우저 또는 웹 브라우저 추가 기능 등의 명령행 도구를 사용하여 요청을 전송할 수도 있습니다.

중요사항: 최소한 74 페이지의 『1.a』 및 74 페이지의 『1.b』 단계를 수행해야 합니다.

프로시저

1. URL을 구성하십시오.

a) 다음 명령을 입력하여 접두부 URL을 판별하십시오.

```
dspmweb status
```

사용할 URL에 `/ibmmq/rest/` 문구가 포함되어 있습니다.

b) 자원을 URL 경로에 추가하십시오.

다음 IBM MQ 자원을 사용할 수 있습니다.

- [/admin/installation](#)
- [/admin/qmgr](#)
- [/admin/큐](#)
- [/admin/등록](#)
- [/admin/채널](#)
- [/action/qmgr/{qmgrName}/mqsc](#)

다음 Managed File Transfer 자원을 사용할 수 있습니다.

- [/admin/에이전트](#)
- [/admin/전송](#)
- [/admin/모니터](#)

예를 들어, 큐 관리자와 상호 작용하려면 `/qmgr`를 접두부 URL에 추가하여 다음 URL을 작성하십시오.

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr
```

c) 옵션: 선택적인 추가 경로 세그먼트를 URL에 추가하십시오.

각 오브젝트 유형의 참조 정보에서 URL의 선택적인 세그먼트는 중괄호({})로 둘러싸서 식별할 수 있습니다.

예를 들어, 큐 관리자 이름 QM1을 URL에 추가하여 다음 URL을 작성하십시오.

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1
```

d) 옵션: 선택적 조회 매개변수를 URL에 추가하십시오.

물음표, ?, 변수 이름, 등호 =, URL에 대한 값 또는 값 목록.

예를 들어, 큐 관리자 QM1의 모든 속성을 요청하려면 다음 URL을 작성하십시오.

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1?attributes=*
```

e) 선택적 추가 조회 매개변수를 URL에 추가하십시오.

URL에 앰퍼샌드(&)를 추가한 후 d단계를 반복하십시오.

2. URL에서 관련 HTTP 메소드를 호출하십시오. 선택적 JSON 페이로드(payload)를 지정하고 인증할 적절한 보안 신임 정보를 제공하십시오. 예를 들면, 다음과 같습니다.

- 선택한 프로그래밍 언어의 HTTP/REST 구현을 사용하십시오.
- REST 클라이언트 브라우저 추가 기능 또는 cURL 같은 도구를 사용하십시오.

REST API를 사용한 원격 관리

REST API를 사용하여 리모트 큐 관리자만 해당 큐 관리자와 연관된 IBM MQ 오브젝트를 관리할 수 있습니다. 이 원격 관리에는 동일한 시스템에 있는 큐 관리자는 포함되지만, mqweb 서버와 동일한 IBM MQ 설치에 있는 큐 관리자는 포함되지 않습니다. 따라서 REST API를 사용하여 mqweb 서버를 실행하는 설치가 한 개뿐인 전체 IBM MQ 네트워크를 관리할 수 있습니다. 리모트 큐 관리자를 관리하려면 mqweb 서버와 동일한 설치에 있는 최소 하나의 큐 관리자만 게이트웨이 큐 관리자로 사용되도록 administrative REST API 게이트웨이를 구성해야 합니다. 그런 다음 지정된 관리 조치를 수행할 리모트 큐 관리자를 REST API 자원 URL에 지정할 수 있습니다.

시작하기 전에

administrative REST API 게이트웨이를 사용 안함으로 설정하면 원격 관리를 금지할 수 있습니다. 자세한 정보는 [administrative REST API 게이트웨이 구성](#)을 참조하십시오.

administrative REST API 게이트웨이를 사용하려면 다음 조건을 충족해야 합니다.

- mqweb 서버를 구성하고 시작해야 합니다. mqweb 서버 구성 및 시작에 대한 자세한 정보는 [70 페이지의 『administrative REST API 시작하기』](#)의 내용을 참조하십시오.
- 게이트웨이 큐 관리자로 구성할 큐 관리자는 mqweb 서버와 동일한 설치에 있어야 합니다.
- 관리할 리모트 큐 관리자는 IBM MQ 8.0 이상이어야 합니다.
- 요청에 지정된 속성이 요청을 송신하는 시스템에 대해 올바른지 확인해야 합니다. 예를 들어 게이트웨이 큐 관리자가 Windows에 있고 리모트 큐 관리자는 z/OS에 있는 경우, queue 자원의 HTTP GET 요청에 대해 dataCollection.statistics 속성이 리턴되도록 요청할 수 없습니다.
- 요청에 지정된 속성이 요청을 송신하는 IBM MQ의 레벨에 대해 올바른지 확인해야 합니다. 예를 들어 리모트 큐 관리자가 IBM MQ 8.0을 실행 중인 경우, queue 자원의 HTTP GET 요청에 대해 extended.enableMediaImageOperations 속성이 리턴되도록 요청할 수 없습니다.
- 지원되는 다음 REST 중 하나를 사용해야 합니다.

- /queue
- /subscription
- /channel
- /mqsc
- /qmgr

리모트 큐 관리자를 조회할 경우 /qmgr 자원은 name, status.started, status.channelInitiatorState, status.ldapConnectionState, status.connectionCount, status.publishSubscribeState 속성의 일부만 리턴합니다.

이 태스크 정보

administrative REST API 게이트웨이를 사용하여 리모트 큐 관리자를 관리하려면 원격 관리용 큐 관리자를 준비해야 합니다. 즉, 전송 큐, 리스너, 게이트웨이 큐 관리자와 리모트 큐 관리자 간 송신자 및 수신자 채널을 구성해야 합니다. 그런 다음 자원 URL에 큐 관리자를 지정하여 REST 요청을 리모트 큐 관리자로 송신할 수 있습니다. 게이트웨이 큐 관리자는 **setmqweb** 명령을 사용하여 mqRestGatewayQmgr 속성을 게이트웨이 큐 관리자의 이름으로 설정하거나 요청과 함께 송신된 헤더에 게이트웨이 큐 관리자의 이름을 송신하는 방법으로 지정됩니다. 요청은 게이트웨이 큐 관리자를 통해 리모트 큐 관리자로 송신됩니다. 그런 다음 게이트웨이 큐 관리자로 사용된 큐 관리자를 나타내는 헤더와 함께 응답이 리턴됩니다.

프로시저

1. 게이트웨이 큐 관리자와 관리할 리모트 큐 관리자 간 통신을 구성하십시오. 이 구성 단계는 runmqsc 및 PCF를 통해 원격 관리를 구성할 때 필요한 것과 동일한 단계입니다.
이 단계에 대한 자세한 정보는 [183 페이지의 『원격 관리를 위한 큐 관리자 구성』](#)의 내용을 참조하십시오.
2. 리모트 큐 관리자에 대한 보안을 구성하십시오.
 - a) 리모트 큐 관리자가 실행되는 시스템에 관련 사용자 ID가 있는지 확인하십시오. 원격 시스템에 있어야 하는 사용자 ID는 REST API 사용자의 역할에 따라 다릅니다.
 - REST API 사용자가 MQWebAdmin 또는 MQWebAdminRO 그룹에 있는 경우, mqweb 서버를 시작한 사용자 ID가 원격 시스템에 있어야 합니다. IBM MQ Appliance의 경우 mqweb 서버를 시작한 사용자는 mqsystem입니다.
 - REST API 사용자가 MQWebUser 그룹에 있는 경우, REST API 사용자 ID가 원격 시스템에 있어야 합니다.
 - b) 리모트 큐 관리자의 적합한 REST API 자원에 액세스하는 데 필요한 레벨의 권한이 관련 사용자 ID에 부여되었는지 확인하십시오.
 - SYSTEM.ADMIN.COMMAND.QUEUE에 메시지를 넣을 수 있는 권한
 - SYSTEM.REST.REPLY.QUEUE에 메시지를 넣을 수 있는 권한
 - 원격 관리용으로 정의된 전송 큐에 액세스할 수 있는 권한
 - 큐 관리자 속성을 표시할 수 있는 권한
 - REST 요청을 수행할 수 있는 권한. 자세한 정보는 [REST API 자원 참조 주제](#)의 보안 요구사항 절을 참조하십시오.
3. 게이트웨이로 사용할 로컬 큐 관리자를 구성하십시오. 기본 게이트웨이 큐 관리자를 구성하거나, HTTP 헤더에 게이트웨이 큐 관리자를 지정하거나, 두 가지 방법을 조합하여 사용할 수 있습니다.
 - 기본 게이트웨이 큐 관리자를 구성하려면 **setmqweb** 명령을 사용하십시오.

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

여기서 *qmgrName*은 게이트웨이 큐 관리자의 이름입니다.

이 게이트웨이 큐 관리자는 다음 두 가지 사항을 모두 충족할 때 사용됩니다.

- 큐 관리자가 REST 요청의 `ibm-mq-rest-gateway-qmgr` 헤더에 지정되어 있지 않습니다.
 - REST API 자원 URL에 지정된 큐 관리자가 로컬 큐 관리자가 아닙니다.
- 모든 REST 요청에 대해 게이트웨이 큐 관리자를 구성하려면 HTTP 헤더 `ibm-mq-rest-gateway-qmgr`을 게이트웨이 큐 관리자의 이름으로 설정하십시오.
4. 관리하려는 리모트 큐 관리자의 이름을 자원 URL에 포함시키십시오.
예를 들어 리모트 큐 관리자 `remoteQM`에서 큐 목록을 가져오려면 다음 URL을 사용하십시오.

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/remoteQM/queue
```


결과

`ibm-mq-rest-gateway-qmgr` 헤더가 REST 응답과 함께 리턴됩니다. 이 헤더는 게이트웨이 큐 관리자로 사용된 큐 관리자를 지정합니다.

administrative REST API를 사용하여 리모트 큐 관리자를 관리하는 데 문제가 있는 경우 다음을 수행하십시오.

- 리모트 큐 관리자가 실행되고 있는지 확인하십시오.
- 명령 서버가 원격 시스템에서 실행 중인지 확인하십시오.
- 채널 연결 끊기 간격이 만료되지 않았는지 확인하십시오. 예를 들어, 채널이 시작되었지만 잠시 후에 종료됩니다. 이는 채널을 수동으로 시작하는 경우 특히 중요합니다.

예

다음 예에서는 두 개의 시스템에 세 개의 IBM MQ 설치가 있습니다. Machine 1에는 Installation 1 및 Installation 2가 있습니다. Machine 2에는 Installation 3가 있습니다. mqweb 서버는 Installation 1용으로 구성되었습니다. 각 설치에는 하나의 큐 관리자가 있으며, 이러한 큐 관리자는 원격 관리용으로 구성되었습니다. 즉, 다음과 같은 리스너, 채널 및 큐가 구성되어 시작되었습니다.

- 큐 관리자 QM1에서 Machine 1의 Installation 1
 - 송신자 채널 QM1.to.QM2
 - 수신자 채널 QM2.to.QM1
 - 송신자 채널 QM1.to.QM3
 - 수신자 채널 QM3.to.QM1
 - 전송 큐 QM2
 - 전송 큐 QM3
 - 포트 1414에 구성된 리스너
- 큐 관리자 QM2에서 Machine 1의 Installation 2
 - 송신자 채널 QM2.to.QM1
 - 수신자 채널 QM1.to.QM2
 - 전송 큐 QM1
 - 포트 1415에 구성된 리스너
- 큐 관리자 QM3에서 Machine 2의 Installation 3
 - 송신자 채널 QM3.to.QM1
 - 수신자 채널 QM1.to.QM3
 - 전송 큐 QM1
 - 기본 리스너

큐 Qon2는 QM2에 정의되어 있고, 큐 Qon3는 QM3에 정의되어 있습니다.

사용자 `muser`는 두 시스템에 모두 정의되어 있고, 이 사용자에게는 REST API에서 MQWebAdmin 역할이 부여되었으며, 각 큐 관리자의 해당 큐에 액세스할 수 있는 권한이 부여되었습니다.

`setmqweb` 명령은 큐 관리자 QM1을 기본 게이트웨이 큐 관리자로 구성하는 데 사용됩니다.

다음 다이어그램은 이 구성을 보여줍니다.

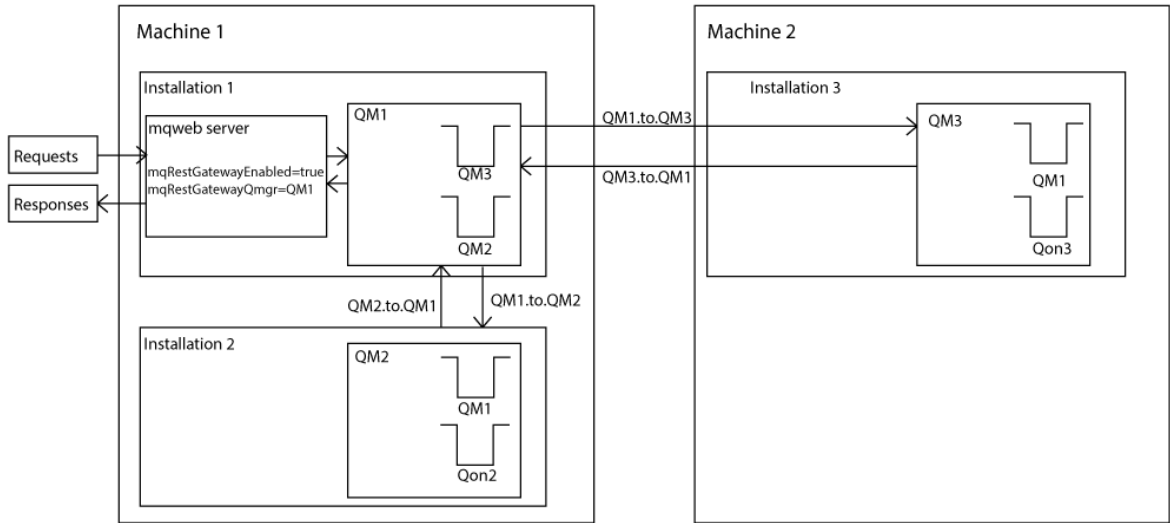


그림 14. REST API를 사용한 원격 관리 구성 예를 보여주는 다이어그램

다음 REST 요청이 mqweb 서버로 송신됩니다.

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM2/queue?
attributes=general.isTransmissionQueue
```

다음 응답이 수신됩니다.

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false
    },
    "name" : "Qon2",
    "type" : "local"
  }
  ]
}
```

다음 REST 요청이 mqweb 서버로 송신됩니다.

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM3/queue?
attributes=general.isTransmissionQueue,general.description
```

다음 응답이 수신됩니다.

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true,
      "description": "Transmission queue for remote admin."
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false,
      "description": "A queue on QM3."
    }
  }
  ]
}
```

```

    },
    "name" : "Qon3",
    "type" : "local"
  }
}

```

REST API 시간소인

administrative REST API에서 날짜 및 시간 정보를 리턴하는 경우 이 날짜 및 시간은 설정된 형식으로 협정 세계시(UTC)로 리턴됩니다.

날짜 및 시간은 다음과 같은 시간소인 형식으로 리턴됩니다.

```
YYYY-MM-DDTHH:mm:ss:sssZ
```

예를 들어, 2012-04-23T18:25:43.000Z입니다. 여기서 Z는 시간대가 협정 세계시(UTC)임을 나타냅니다.

이 시간소인의 정확도가 보장되지 않습니다. 예를 들어, mqweb 서버가 자원 URL에서 지정되는 큐 관리자와 동일한 시간대에 시작되지 않는 경우 시간소인이 정확하지 않습니다. 또한 일광 절약 시간 조정이 필요한 경우 시간소인이 정확하지 않습니다.

REST API 오류 핸들링

REST API는 적절한 HTTP 응답 코드(예: 404(찾을 수 없음)) 및 JSON 응답을 리턴하여 오류를 보고합니다. 200 - 299 범위에 있지 않은 HTTP 응답은 오류로 간주됩니다.

오류 응답 형식

응답은 UTF-8 인코딩의 JSON 형식입니다. 응답에는 중첩 JSON 오브젝트가 포함되어 있습니다.

- **error**라고 하는 단일 JSON 배열을 포함하는 외부 JSON 오브젝트입니다.
- 배열의 각 요소는 오류에 대한 정보를 나타내는 JSON 오브젝트입니다. 각 JSON 오브젝트에는 다음 특성이 포함되어 있습니다.

유형

문자열.

오류의 유형입니다.

messageId

문자열.

MQWBnnnnX 양식의 메시지에 대한 고유 ID입니다. 이 ID는 다음 요소를 포함합니다.

MQWB

메시지가 IBM MQ Rest API에서 발생했음을 표시하는 접두부입니다.

nnnn

메시지를 식별하는 고유한 번호입니다.

X

메시지의 심각도를 나타내는 단일 문자입니다.

- I - 메시지가 순전히 정보 제공 용도인 경우
- W - 메시지가 문제의 경고인 경우
- E - 메시지가 오류 발생을 나타내는 경우
- S - 메시지가 심각한 오류 발생을 나타내는 경우

메시지

문자열.

오류에 대한 설명입니다.

설명

문자열.


오류에 대한 설명입니다.

조치

문자열.

오류를 해결하기 위해 수행할 수 있는 단계에 대한 설명입니다.

qmgrName

 이 필드는 큐 관리자가 큐 공유 그룹의 멤버인 z/OS에 대해서만 사용할 수 있습니다. **commandScope** 선택적 조회 매개변수 또는 **queueSharingGroupDisposition** 속성을 지정해야 합니다.

문자열.

오류가 발생한 큐 관리자의 이름입니다.

이 필드는 messaging REST API에 적용할 수 없습니다.

completionCode

이 필드는 **type**이 pcf, java 또는 rest인 경우에만 사용할 수 있습니다.

숫자입니다.

실패와 연관된 MQ 완료 코드입니다.

reasonCode

이 필드는 **type**이 pcf, java 또는 rest인 경우에만 사용할 수 있습니다.

숫자입니다.

실패와 연관된 MQ 이유 코드입니다.

예외

이 필드는 **type**이 java인 경우에만 사용할 수 있습니다.

배열입니다.

체인 Java 또는 JMS 예외의 배열입니다. 예외 배열의 각 요소에는 **stackTrace** 문자열 배열이 포함되어 있습니다.

stackTrace 문자열 배열에는 행으로 분할된 각 예외의 세부사항이 포함되어 있습니다.

큐 공유 그룹에 대한 오류



큐 공유 그룹에서는 특정 명령에 대해 **commandScope**의 선택적 조회 매개변수를 지정할 수 있습니다. 이 매개변수를 사용하면 명령을 큐 공유 그룹의 다른 큐 관리자에 전파할 수 있습니다. 이 명령 중 하나가 독립적으로 실패하여 큐 공유 그룹에 대해 일부 명령은 성공하고 일부 명령은 실패할 수 있습니다.

명령이 부분적으로 실패하는 경우 HTTP 오류 코드 500이 리턴됩니다. 실패를 생성한 각각의 큐 관리자에 대해 해당 실패에 대한 정보가 요소로 **error** JSON 배열에 리턴됩니다. 성공적으로 명령을 실행한 각각의 큐 관리자에 대해 큐 관리자의 이름이 요소로 **success** JSON 배열에 리턴됩니다.

예

- 다음 예에서는 존재하지 않는 큐 관리자에 대한 정보를 가져오려는 시도에 대한 오류 응답을 보여줍니다.

```
"error": [
  {
    "type": "rest",
    "messageId": "MQWB0009E",
    "message": "MQWB0009E: Could not query the queue manager 'QM1'",
    "explanation": "The MQ REST API was invoked specifying a queue manager name which cannot be located.",
    "action": "Resubmit the request with a valid queue manager name or no queue manager name, to retrieve a list of queue managers. "
  }
]
```

- ▶ **z/OS** 다음 예에서는 일부 큐 관리자에 대해 존재하지 않는 큐 공유 그룹에서 큐를 삭제하려는 시도에 대한 오류 응답을 보여줍니다.

```
"error" : [
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name to retrieve a list of queues.",
    "qmgrName": "QM1"
  },
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name to retrieve a list of queues.",
    "qmgrName": "QM2"
  }
],
"success" : [{ "qmgrName": "QM3" }, { "qmgrName": "QM4" }]
```

MFT 요청의 오류

MFT REST API 서비스가 사용되지 않고 MFT REST API를 호출하는 경우, 다음 예외를 수신합니다.

```
{ "error": [ {
  "action": "Enable the Managed File Transfer REST API and resubmit the request.",
  "completionCode": 0,
  "explanation": "Managed File Transfer REST calls are not permitted as the service is disabled.",
  "message": "MQWB0400E: Managed File Transfer REST API is not enabled.",
  "msgId": "MQWB0400E",
  "reasonCode": 0,
  "type": "rest"
} ] }
```

MFT REST API 서비스가 사용되고 조정 큐 관리자가 mqwebuser.xml 파일에 설정되지 않은 경우 다음 예외를 수신합니다.

```
{ "error": [ {
  "action": "Set the coordination queue manager name and restart the mqweb server.",
  "completionCode": 0,
  "explanation": "Coordination queue manager name must be set before using Managed File Transfer REST services.",
  "message": "MQWB0402E: Coordination queue manager name is not set.",
  "msgId": "MQWB0402E",
  "reasonCode": 0,
  "type": "rest"
} ] }
```

REST API 검색

REST API에 대한 문서는 IBM Documentation에 Swagger 형식으로 제공됩니다. Swagger는 REST API를 문서화하기 위해 일반적으로 사용되는 방식입니다. mqweb 서버에서 API 발견 기능 (apiDiscovery) 을 사용하여 REST API 에 대한 Swagger 문서를 볼 수 있습니다.

시작하기 전에



▶ Stabilized

중요사항: apiDiscovery 기능이 안정화되었습니다. 이 기능을 계속 사용할 수 있습니다. 현재 IBM MQ 는 mpOpenAPI 기능의 사용을 지원하지 않습니다.

API 검색을 사용하여 mqweb 서버가 Swagger 문서를 볼 수 있도록 보안을 설정해야 합니다. 보안 사용에 필요한 단계에 대한 자세한 정보는 [IBM MQ Console](#) 및 [REST API 보안을 참조하십시오](#).

프로시저

1. 다음 디렉토리 중 하나에서 mqwebuser.xml 파일을 찾으십시오.

-  `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
-  `WLP_user_directory/servers/mqweb`

여기서 `WLP_user_directory`는 mqweb 서버 정의를 작성하기 위해 `crtmqweb` 스크립트를 실행할 때 지정한 디렉토리입니다.

2. mqwebuser.xml 파일에 적절한 XML을 추가하십시오.

- `<featureManager>` 태그가 mqwebuser.xml 파일에 있는 경우 `<featureManager>` 태그 내에서 다음 XML을 추가하십시오.
`<feature>apiDiscovery-1.0</feature>`
- `<featureManager>` 태그가 mqwebuser.xml 파일에 없는 경우 `<server>` 태그 내에서 다음 XML을 추가하십시오.

```
<featureManager>
  <feature>apiDiscovery-1.0</feature>
</featureManager>
```

3. 다음 방법 중 하나를 사용하여 Swagger 문서를 볼 수 있습니다.

- 검색할 수 있는 웹 페이지를 표시하고 브라우저에 다음 URL을 입력하여 REST API를 수행하십시오.

`https://host:port/ibm/api/explorer`

각 요청을 인증하는 것 외에도 각 POST, PATCH 또는 DELETE 요청에 대한 `ibm-mq-rest-csrf-token` 헤더를 포함해야 합니다. 이 헤더의 콘텐츠는 공백을 포함하여 임의의 문자열이 될 수 있습니다.

이 요청 헤더는 요청을 인증하는 데 사용 중인 신임 정보가 신임 정보의 소유자에 의해 사용 중인지 확인하는 데 사용됩니다. 즉, 이 토큰을 사용하여 사이트 간 요청 위조 공격을 방지합니다.

- 다음 URL에 대해 HTTP GET을 발행하여 전체 REST API를 설명하는 단일 Swagger 2 문서를 검색하십시오.

`https://host:port/ibm/api/docs`

프로그래밍 방식으로 사용 가능한 API를 탐색하고자 하는 애플리케이션에 대해 이 문서를 사용할 수 있습니다.

호스트

REST API를 사용할 수 있는 호스트 이름 또는 IP 주소를 지정합니다.

기본값은 localhost입니다.

포트

administrative REST API가 사용하는 HTTPS 포트 번호를 지정합니다.

기본값은 9443입니다.

호스트 이름 또는 포트 번호가 기본값에서 변경된 경우 REST API URL에서 올바른 값을 판별할 수 있습니다. `dspmqweb status` 명령을 사용하여 URL을 보십시오.

관련 정보

[dspmqweb 상태 \(mqweb 서버 상태 표시\)](#)

REST API 자국어 지원(NLS)

REST API는 특정한 자격 요건을 부여하여 HTTP 요청의 일부로 자국어를 지정하는 기능을 지원합니다.

백그라운드

HTTP 헤더를 사용하면 요청에 특정한 동작을 지정하고 추가 정보를 응답에 제공할 수 있습니다.

자국어로 정보를 리턴하도록 요청하는 기능이 HTTP 헤더에 포함되어 있습니다. REST API는 가능한 경우 이 헤더를 적용합니다.

자국어 지정

ACCEPT-LANGUAGE HTTP 헤더에, 하나 이상의 언어 태그가 제공될 수 있습니다. 선택적으로 태그에 랭크를 연관시켜, 목록의 스펙을 환경 설정에 따라 정렬시킬 수 있습니다. [이 페이지](#)의 토론이 원리 이해에 도움이 될 것입니다.

REST API는 이 헤더를 적용하여, ACCEPT-LANGUAGE 헤더에서 언어를 선택하고 해당 언어로 메시지를 리턴합니다. ACCEPT-LANGUAGE 헤더에 REST API가 지원할 수 있는 언어가 포함되지 않은 경우 메시지는 기본 언어로 리턴됩니다. 이 기본 언어는 REST API 웹 서버의 기본 로케일에 해당됩니다.

83 페이지의 [『변환되는 데이터』](#) 절에서 변환되는 데이터에 대해 설명합니다.

응답에 적용 가능한 언어 표시

REST API를 통한 응답의 CONTENT-LANGUAGE HTTP 헤더에 메시지가 리턴되는 언어가 표시됩니다.

변환되는 데이터

오류 또는 정보 메시지는 변환되지만 다른 텍스트는 변환되지 않습니다.

- 큐 관리자로부터 리턴되는 데이터는 변환되지 않습니다. 예를 들어, REST API를 통해 MOSC 명령을 실행하는 경우 큐 관리자의 응답은 큐 관리자의 로케일로 되어 있습니다.
- REST API에 대해 생성되는 (Swagger) 문서는 apiDiscovery 기능을 통해 노출되는 경우 영어입니다.

지원되는 언어

REST API 오류 및 정보 메시지는 영어뿐만 아니라 다음 언어로도 변환됩니다.

중국어

언어 태그 zh_CN으로 표시

대만어

언어 태그 zh_TW으로 표시

체코어

언어 태그 cs으로 표시

프랑스어

언어 태그 fr으로 표시

헝가리어

언어 태그 hu으로 표시

이탈리아어

언어 태그 it으로 표시

일본어

언어 태그 ja으로 표시

한국어

언어 태그 ko으로 표시

폴란드어

언어 태그 pl으로 표시

(브라질) 포르투갈어

언어 태그 pt_BR으로 표시

러시아어

언어 태그 ru으로 표시

스페인어

언어 태그 es으로 표시

예

이 예의 웹 서버는 영어가 기본 로케일입니다.

단일 지원 언어 지정

요청 헤더에 ACCEPT-LANGUAGE가 fr로 설정되어 있습니다. 이 설정은 변환 가능한 텍스트의 선호 언어가 프랑스어임을 지정합니다.

응답 헤더에 CONTENT-LANGUAGE가 fr로 설정되어 있습니다. 이 설정은 응답의 오류 및 정보 메시지가 프랑스어임을 표시합니다.

언어 목록 지정

요청 헤더에 ACCEPT-LANGUAGE가 am, fr로 설정되어 있습니다. 이 설정은 변환 가능한 텍스트에 대해 허용되는 언어는 0 암하라어와 프랑스어이며 암하라어가 변환 가능한 텍스트에 대한 선호 언어임을 지정합니다.

응답 헤더에 CONTENT-LANGUAGE가 fr로 설정되어 있습니다. 이 설정은 REST API가 암하라어를 지원하지 않으므로 응답의 오류 및 정보 메시지는 프랑스어임을 표시합니다.

지원되지 않는 단일 언어 지정

요청 헤더에 ACCEPT-LANGUAGE가 am로 설정되어 있습니다. 이 설정은 변환 가능한 텍스트의 선호 언어가 암하라어임을 지정합니다.

응답 헤더에 CONTENT-LANGUAGE가 en로 설정되어 있습니다. 이 설정은 REST API가 암하라어를 지원하지 않으므로 응답의 오류 및 정보 메시지는 영어임을 표시합니다.

REST API 버전

REST API 버전 번호는 REST 요청에 대한 기본 URL의 부분을 형성합니다. 예를 들어, `https://localhost:9443/ibmmq/rest/v2/admin/installation`입니다. 버전 번호는 향후 릴리스에 도입될 수도 있는 REST API의 변경에서 클라이언트를 분리하는 데 사용됩니다.

IBM MQ 9.2.0은 REST API의 버전 2를 소개합니다. 이 버전은 administrative REST API, messaging REST API 및 MFT REST API에 적용됩니다. 이 버전은 REST API에 사용된 자원 URL을 변경합니다. 버전 2에서의 자원 URL에 대한 URL 접두부는 다음 URL입니다.

```
https://host:port/ibmmq/rest/v2/
```

Stabilized REST API에 도입된 일부 변경으로 인해, REST API를 사용하는 클라이언트를 업데이트하도록 기존 REST API 기능이 변경될 수 있습니다. 이와 같은 변경으로 인해 클라이언트가 강제 업데이트되지 않도록 하기 위해 REST API 버전 번호는 증가하고 기존 기능은 이전 번호에서 안정화되었습니다. 기존 기능을 변경할 수 있는 새 기능은 새 버전 번호의 REST API에 추가됩니다. 따라서 클라이언트는 업데이트되지 않고 이전 버전에서 REST API를 계속 사용할 수 있습니다.

클라이언트 업데이트가 필요하게 되는 REST API 변경은 다음과 같습니다.

- REST API에서 리턴되거나 이 REST API로 송신되는 JSON의 기존 속성에 대한 지원 제거.
- URL, HTTP verb 또는 헤더 제거. 예를 들어, URL 또는 헤더의 이름이 바뀐 경우 또는 다른 verb가 사용되는 경우입니다.
- 기존 URL로 송신되는 데이터에 새 필수 JSON 속성 추가.
- 기존 URL로 송신되는 데이터에 새 필수 HTTP 헤더 추가.
- 기존 URL에 새 필수 조회 매개변수 추가.

이 유형의 변경이 Long Term Support (LTS) 릴리스에 존재하는 REST API 기능에 도입되면 이러한 변경 중 첫 번째에 대해 REST API의 버전 번호가 증가합니다. REST API를 사용하는 클라이언트에 대한 변경이 필요할 수 있는 Continuous Delivery (CD) 릴리스 내에서 작성되는 모든 후속 변경사항은 새 버전 번호를 사용합니다.

이 버전 번호는 다음 LTS 릴리스까지 후속 CD 릴리스 전체에서 동일하게 유지됩니다. 따라서 버전 번호는 LTS 릴리스 사이에서 한 번씩 증가합니다.

Stabilized 버전 번호가 증가하면 기존 REST API 기능은 이전 버전 번호에서 안정화됩니다. 즉, LTS 릴리스에서 사용 가능했던 기존 REST API 기능은 이전 버전 번호에서 사용 가능한 상태로 유지되지만 해당 버전에 대한 추가 변경사항은 작성되지 않습니다. REST API에 추가된 새 기능은 새 REST API 버전에 추가됩니다. 그러나 버전이 증가하기 전에 CD 릴리스의 REST API에 추가된 사항은 REST API의 이전 버전에 포함된다고 보장할 수 없습니다.

Deprecated 기존 클라이언트는 변경 없이 이전 버전 번호의 REST API를 계속 사용할 수 있습니다. 이전 버전의 REST API는 더 이상 사용되지 않아 결국 제거될 수 있습니다.

일부 변경에는 REST API를 사용하는 클라이언트에 대한 변경이 필요하지 않습니다. 이러한 변경으로 인해 버전 번호가 증가하지는 않습니다. 따라서 이러한 유형의 변경이 도입될 때 REST API를 사용하는 클라이언트를 업데이트하지 않아도 되는지 확인하십시오. 이러한 REST API 변경은 다음과 같습니다.

- REST API에서 리턴된 기존 데이터에 새 JSON 속성 추가.
- 새 URL 추가.
- 기존 URL에 새 HTTP verb 추가.
- 기존 URL에 새 상태 코드 추가.
- 기존 URL에 송신된 데이터에 새 선택적 JSON 속성 추가.
- 기존 URL에 새 조회 매개변수 추가.
- 기존 URL에 송신된 데이터에 새 헤더 추가.
- REST API에서 새 헤더 리턴.

새 Continuous Delivery REST API 기능 변경

CD 릴리스에 추가된 새 REST API 기능의 경우, REST API 클라이언트에 대한 변경이 필요할 수 있는 이 새 기능에 대한 변경사항은 버전 번호를 늘리지 않습니다. 즉, 버전 번호를 늘리지 않고 다음 LTS 릴리스 전에 새 기능이 변경될 수 있습니다. 기능이 LTS 릴리스에 포함되면 REST API 클라이언트 변경이 필요한 후속 변경으로 버전 번호가 증가합니다.

예

1. LTS 릴리스 X에서 REST API는 버전 1입니다.
2. CD 릴리스 X.0.1에서 새 URL에 대한 지원이 추가됩니다. 이 변경에는 REST API를 사용하는 클라이언트 변경이 필요하지 않습니다. 따라서 REST API는 버전 1에서 유지됩니다.
3. CD X.0.2에서 새 URL에 대한 지원이 추가됩니다. 이 변경에는 REST API를 사용하는 클라이언트 변경이 필요하지 않습니다. 따라서 REST API는 버전 1에서 유지됩니다.
4. LTS 릴리스 Y에서 REST API는 버전 1입니다.
5. CD 릴리스 Y.0.1에서는 기존 URL의 이름이 바뀝니다. 이 변경에는 REST API를 사용하는 클라이언트 변경이 필요합니다. 따라서 새 버전의 REST API는 버전 2로 작성됩니다. 이름이 바뀐 URL은 모든 기존 기능과 함께 REST API의 버전 2에 포함됩니다. REST API에 추가된 새 기능이 버전 2에 추가됩니다. 버전 1은 LTS 릴리스 Y의 레벨에서 안정화된 상태로 유지됩니다.
6. CD 릴리스 Y.0.2에서는 다른 기존 URL의 이름이 바뀝니다. CD 릴리스 Y에서 버전이 이미 증가되었으므로 REST API는 버전 2로 유지됩니다. 버전 1은 LTS 릴리스 Y의 레벨에서 안정화된 상태로 유지됩니다.
7. LTS 릴리스 Z에서 REST API는 버전 2로 유지됩니다. 버전 1은 LTS 릴리스 Y의 레벨에서 안정화된 상태로 유지됩니다.

IBM MQ Console를 사용하여 관리

IBM MQ Console를 사용하여 기본 관리 태스크를 수행할 수 있습니다.

참고: IBM MQ Console을 사용할 때는 큐 관리자에서 명령 서버를 사용 안함으로 설정하지 마십시오. 명령 서버가 큐 관리자에 대해 사용 불가능한 경우:

- IBM MQ Console 가 응답하지 않게 되고 명령 처리가 오래 지연됩니다.
- 큐 관리자에 실행되는 모든 명령이 제한시간을 초과합니다.

관련 태스크

[IBM MQ Console 추적](#)

IBM MQ Console 시작하기

mqweb 서버를 구성하고, IBM MQ Console의 URI를 판별하고, 콘솔에 연결하고, 콘솔에 로그인하십시오.

시작하기 전에

이 태스크를 완료하려면 **dspmweb** 명령을 사용할 수 있도록 특정 권한이 있는 사용자여야 합니다.

- **z/OS** z/OS의 경우 **dspmweb** 명령을 실행하는 권한 및 mqwebuser.xml 파일에 대한 쓰기 액세스 권한이 있어야 합니다.
- **Multi** 다른 모든 운영 체제에서는 [권한이 있는 사용자](#)여야 합니다.
- **IBM i** IBM i에서 명령은 QSHELL에서 실행되어야 합니다.

이 태스크 정보

다음과 같은 제한사항에 유의하십시오.

- **z/OS**
 - z/OS의 큐 관리자를 작성하거나 삭제하거나 시작하거나 중지할 수 없습니다.
 - z/OS의 채널 시작기를 시작하거나 중지할 수 없으며 채널 시작기 상태가 표시되지 않습니다.
 - 리스너를 표시 또는 관리할 수 없습니다.
 - 채널 시작, ping, 해결 및 재설정 명령은 CHLDISP(DEFAULT)를 사용해서만 실행될 수 있습니다.
 - QSGDISP(GROUP)를 사용하여 정의된 오브젝트는 표시되거나 관리될 수 없습니다.
 - 큐 관리자 보안을 관리할 수 없습니다.
 - 시스템 자원 사용량을 모니터링할 수 없습니다.
- **Multi**
 - IBM MQ Console을 사용하여 AMQP 채널을 작업할 수 없습니다.
 - IBM MQ Console 를 사용하여 MQTT 채널에 대해 작업할 수 없습니다.

프로시저

1. mqweb 서버가 아직 IBM MQ Console에서 사용하도록 구성되지 않은 경우, mqweb 서버를 구성하십시오. 기본 레지스트리를 사용하여 mqweb 서버를 구성하는 방법에 대한 자세한 정보는 [mqweb 서버의 기본 구성](#)을 참조하십시오.
2. **z/OS** z/OS에서 **dspmweb** 명령을 사용할 수 있도록 WLP_USER_DIR 환경 변수를 설정하십시오. 다음 명령을 입력하여 mqweb 서버 구성을 가리키도록 변수를 설정하십시오.

```
export WLP_USER_DIR=WLP_user_directory
```

여기서 `WLP_user_directory` 은 `crtmqweb`에 전달되는 디렉토리의 이름입니다. 예:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

자세한 정보는 [mqweb 서버 작성](#)을 참조하십시오.

3. 다음 명령을 입력하여 IBM MQ Console의 URL을 판별하십시오.

```
dspmqweb status
```

명령은 다음과 유사한 출력을 생성합니다.

```
MQWB1124I: Server 'mqweb' is running.
URLS:
https://localhost:9443/ibmmq/rest/v1/
https://localhost:9443/ibmmq/console/
```

IBM MQ Console의 URI는 접미부 `console/`로 끝납니다.

4. 브라우저에서 이전 단계의 URL을 입력하여 IBM MQ Console 에 연결하십시오.

`mqweb` 서버에 제공되는 기본 인증서가 신뢰할 수 있는 인증서가 아니므로 브라우저에서 보안 예외가 발생할 수 있습니다. IBM MQ Console로 진행하려면 선택하십시오.

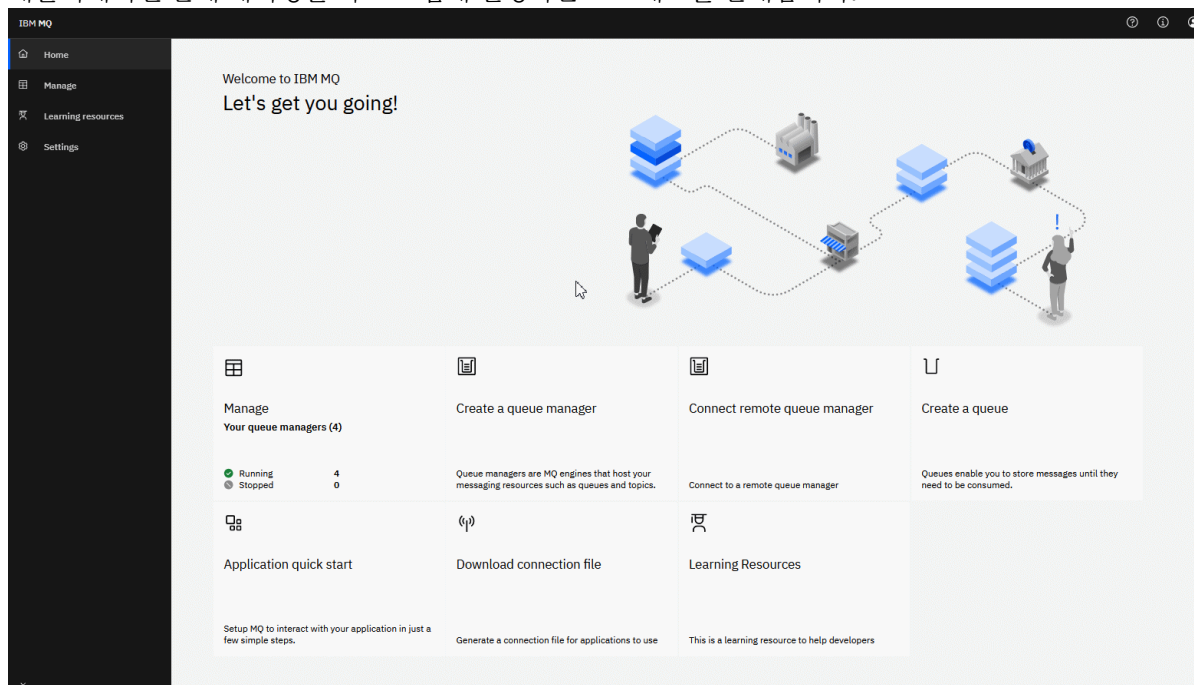
5. IBM MQ Console에 로그인하십시오. 사용자 이름 `mqadmin` 및 비밀번호 `mqadmin`을 사용하십시오.

다음에 수행할 작업

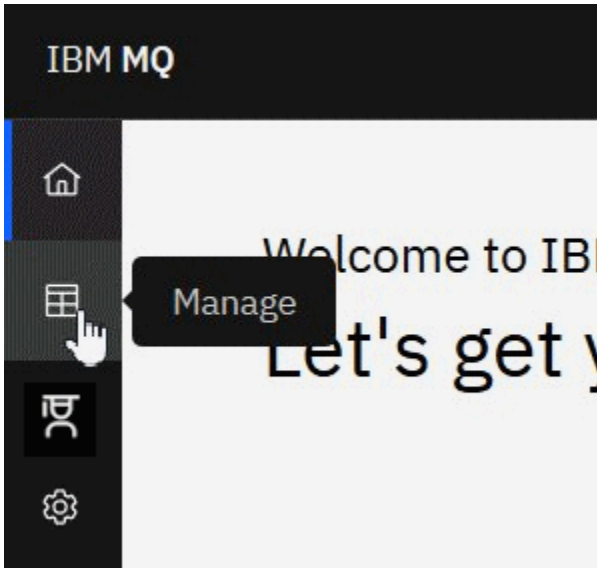
기본적으로 IBM MQ Console은 토큰 기반 인증을 사용하여 사용자를 인증합니다. 또한 클라이언트 인증서 인증을 사용할 수 있습니다. 자세한 정보는 [REST API 및 IBM MQ Console로 클라이언트 인증서 인증 사용](#)을 참조하십시오.

V 9.4.0 IBM MQ Console의 Quick Tour

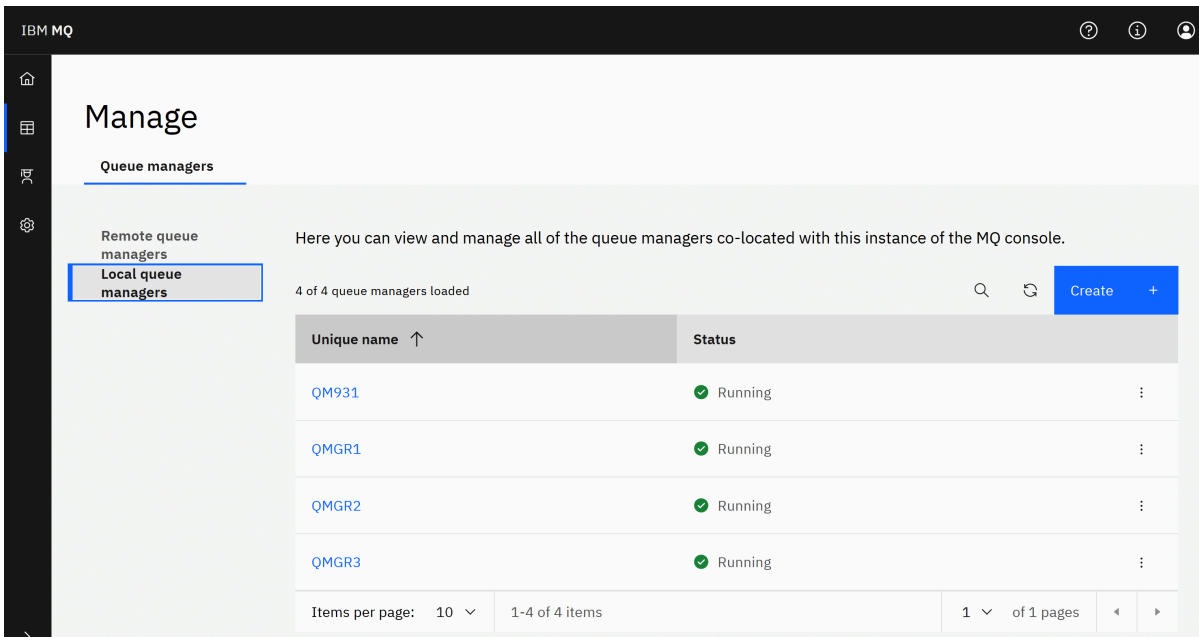
처음 IBM MQ Console에 로그인하면 랜딩 페이지로 이동합니다. 여기서 기존 큐 관리자를 관리하거나, 큐 관리자 또는 큐를 작성하거나, 일부 교육 주제를 탐색하거나, IBM Documentation에서 IBM MQ 제품 정보를 열도록 선택할 수 있습니다. 또한 애플리케이션 빠른 시작을 실행할 수 있으며, 이는 새 큐 관리자 또는 기존 큐 관리자와 애플리케이션 간에 메시지를 빠르고 쉽게 설정하는 프로세스를 안내합니다.



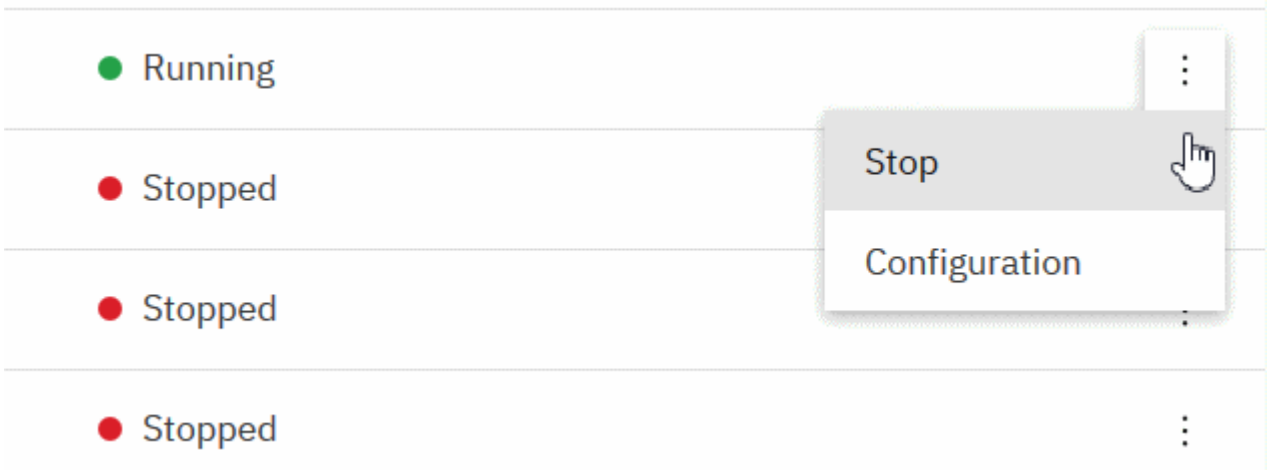
또는 관리 아이콘을 클릭하여 IBM MQ 오브젝트 관리를 바로 시작할 수 있습니다.



관리 보기는 처음에는 큐 관리자와 그 현재 상태를 보여줍니다. 또한 새 큐 관리자를 작성하고 리모트 큐 관리자를 연결할 수 있습니다.

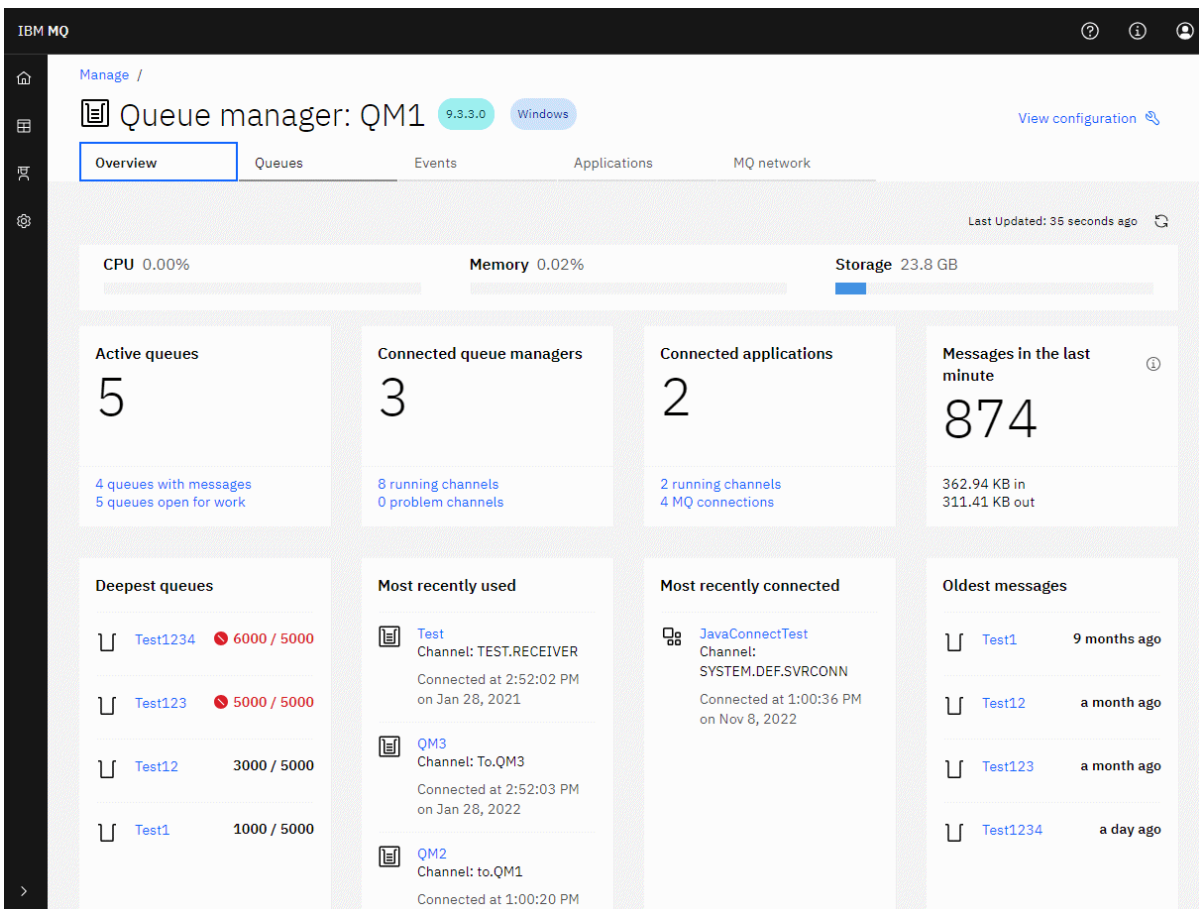


각 큐 관리자에는 실행 중인 큐 관리자를 중지하거나 구성하거나 중지된 큐 관리자를 시작하거나 삭제하는 메뉴가 있습니다.



큐 관리자의 권한 레코드, 인증 정보 오브젝트 및 채널 인증 레코드는 큐 관리자를 작성하고 새 큐 관리자를 추가할 수 있는 큐 관리자 구성 페이지의 **보안** 탭에서 찾을 수 있습니다.

해당 대시보드를 열 실행 중인 큐 관리자의 이름을 클릭하십시오.



큐 관리자 대시보드에서 다음 조치를 완료할 수 있습니다.

V 9.4.0 개요 탭에서 다음 정보를 보십시오.

CPU

큐 관리자의 예상 CPU 사용량 백분율입니다. (z/OS에는 적용되지 않습니다.)

메모리

큐 관리자에 대비합시다라고 예상되는 메모리 사용량의 백분율입니다. (z/OS 또는 Windows에는 적용되지 않습니다.)

스토리지

큐 관리자가 상주하는 디스크의 여유 공간에 대한 예상 백분율입니다. (z/OS에는 적용되지 않습니다.)

활성 큐

메시지가 있거나 입력 또는 출력을 위해 열려 있는 큐의 수입입니다.

연결된 큐 관리자

활성 채널에서 파생된 현재 연결된 큐 관리자의 수입입니다.

연결된 애플리케이션

현재 연결된 애플리케이션 수입입니다.

마지막 1분 동안의 메시지

10초마다 메시지 처리량을 표시하는 PUT/GET 시스템 주제의 요약을 표시합니다. (z/OS에는 적용되지 않습니다.)

구독

등록 수를 표시합니다. 시스템 토픽의 모니터링이 금지된 z/OS 및 기타 플랫폼에서만 표시됩니다 ([setmqweb](#) 특성참조).

가장 깊은 큐

큐를 깊이의 순서로 나열합니다. 현재 큐 용량 및 최대 큐 용량을 표시합니다.

가장 최근에 사용

현재 연결된 큐 관리자를 마지막 메시지 날짜별로 정렬하여 나열합니다.

가장 최근에 연결됨

현재 연결된 애플리케이션을 활성 서버 연결 채널에서 파생된 것으로 채널 시작 날짜 및 시간별로 정렬하여 나열합니다.

가장 오래된 메시지

가장 오래된 메시지 날짜 및 시간별로 정렬된 큐를 나열합니다.

개요 탭에 표시되는 정보는 시스템 주제의 모니터링에서 파생됩니다 ([시스템 주제에 공개된 지표참조](#)). z/OS 는 시스템 토픽 모니터링을 지원하지 않으며 콘솔 표시 용도의 모니터링은 다른 플랫폼에서 사용 안함으로 설정할 수 있습니다 ([setmqweb](#) 특성참조). 이 경우 개요 탭은 더 제한된 정보를 표시하며 해당 모양은 다음 예제와 유사합니다.

The screenshot shows the IBM MQ Queue Manager Overview page for remote-MQ53. The page is divided into several sections:

- Overview:** Shows 0 Active queues, 0 Connected queue managers, 0 Connected applications, and 0 Subscriptions. Below these are details: 0 queues with messages, 0 queues open for work; 0 running channels, 0 problem channels; 6 running channels, 0 MQ connections.
- Deepest queues:** No queues with messages found.
- Most recently used:** No connected queue managers. After you connect queue managers, their connections will display here.
- Most recently connected:** No connected applications. After you connect applications, their connections will display here.
- Oldest messages:** No queues with messages found.


큐 탭에서 다음을 수행하십시오.

- 새 큐 작성


- 기존 메시지를 보고 새 메시지를 작성하고 큐를 구성하려면 큐 이름을 클릭하십시오.

이벤트 탭에서 다음을 수행하십시오.

토픽

- 새 토픽 작성
- 기존 토픽 구성 
- 토픽 이름을 클릭하여 일치하는 구독을 보

구독

- 새 관리 또는 비관리 구독 작성
- 기존 구독 구성 

V 9.4.0 애플리케이션 탭에서 다음을 수행하십시오.

개요

다음 통계의 개요를 제공하는 타일을 포함합니다.

연결된 애플리케이션

연결된 애플리케이션의 수를 표시합니다. 다음 탭에 대한 링크를 제공합니다.

- 애플리케이션 인스턴스
- 연결

채널 인스턴스 실행

앱 채널 탭에서 정의되거나 중지된 SVRCONN 채널 인스턴스 및 해당 링크의 수를 표시합니다.

연결

연결 수를 표시합니다. 연결 탭에서 다음 정보에 대한 링크를 제공합니다.

- 로컬 연결 (채널 이름이 없는 연결)
- 원격 연결 (채널 이름이 있는 연결)

가장 일반적인 애플리케이션

자주 사용하는 애플리케이션의 목록을 사용 중인 연결 수에 따라 정렬하여 표시합니다.

가장 일반적인 채널

자주 사용되는 채널 목록을 활성 상태인 인스턴스 수별로 정렬하여 표시합니다.

가장 오래된 트랜잭션

애플리케이션 이름별로 가장 오래된 트랜잭션 목록을 표시합니다. 이러한 트랜잭션에는 열린 작업 단위 (UOW) 와의 연결이 있으며 UOW 시작 날짜 및 시간별로 정렬됩니다.

원격 연결 버전

공통으로 연결된 IBM MQ 버전, 즉 지정된 REMOTE_VERSION이 있는 채널 인스턴스의 목록을 표시합니다.

애플리케이션 채널 보안

공통으로 연결된 채널 보안 프로토콜 (즉, 지정된 SECURITY_PROTOCOL이 있는 채널 인스턴스) 의 목록을 표시합니다.

채널 전송률

메시지 및 바이트의 전송률별로 정렬된 공통 채널 목록을 표시합니다. 채널 시작 날짜 및 시간을 사용하여 지속 기간을 계산하고 MSGS 및 MQIACH_BYTES_SENT/ MQIACH_BYTES_RCVD를 사용하여 비율을 계산합니다.

애플리케이션

큐 관리자에 연결된 애플리케이션에 대한 정보를 봅니다.

채널

애플리케이션에 연결된 채널의 활동을 봅니다.

앱 채널

- 채널 시작, 중지, ping 및 구성 ⋮
- 새 채널 작성
- 채널 재설정

앱 채널 인스턴스

- 애플리케이션 채널 인스턴스의 상태 보기
- 채널의 인다우트 (in-doubt) 메시지 해석 ⋮

V 9.4.0 MQ 네트워크 탭에서 다음을 수행하십시오.

개요

다음 통계의 개요를 제공하는 타일을 포함합니다.

큐 관리자 채널 인스턴스 실행

비SVRCONN 채널 인스턴스 수를 표시합니다. **연결된 큐 관리자** 탭에서 다음 유형의 채널 인스턴스에 대한 링크를 표시합니다.

- 정의된 채널
- 중지된 채널

연결된 큐 관리자

MQCA_REMOTE_Q_MGR_NAME으로 연결된 큐 관리자의 수를 표시합니다. 또한 MQCMD_INQUIRE_CLUSTER_Q_MGR에서 리턴한 큐 관리자의 수를 제공합니다.

클러스터 멤버십

하나의 큐 관리자 클러스터만 있는 경우, 클러스터의 이름 및 큐 관리자가 전체 또는 부분 저장소인지 여부를 표시합니다. 클러스터에 표시되는 큐 관리자 수를 표시합니다. 둘 이상의 클러스터가 있는 경우 클러스터 수와 각 클러스터의 전체 및 부분 저장소 큐 관리자 수를 표시합니다.

실패한 큐 관리자 채널

재시도 상태 (비중지/실행) 에 있는 채널 목록을 표시합니다. 재시도 상태인 경우 남아 있는 재시도 횟수를 계산합니다. 목록에는 다음 상태 유형의 채널이 포함되어 있습니다.

- MQCHS_PAUSED
- MQCHS_RETRYING

가장 긴 메시지 지연

XMIT 시간 표시기 (긴 기간) 가 있는 채널 목록을 표시합니다.

무인 전송 큐

큐 용량이 0이 아니고 연관된 핸들이 없는 전송 큐의 목록을 표시합니다.

원격 연결 버전

공통으로 연결된 IBM MQ 버전, 즉 지정된 REMOTE_VERSION이 있는 채널 인스턴스의 목록을 표시합니다.

큐 관리자 채널 보안

공통으로 연결된 채널 보안 프로토콜 (즉, 지정된 SECURITY_PROTOCOL이 있는 채널 인스턴스) 의 목록을 표시합니다.

클러스터 상태

클러스터의 상태와 관련된 여러 독립 통계를 표시합니다. 상태에는 다음이 포함됩니다.

- 클러스터 오브젝트 (큐, 토픽, 큐 관리자) 의 수입입니다.
- 일시중단된 큐 관리자의 수입입니다 (MQIACF_SUSPEND가 YES로 설정됨).
- SYSTEM.CLUSTER.COMMAND.QUEUE 큐를 큐에 넣습니다.
- SYSTEM.TEMP.

이러한 모든 값이 0이면 이 타일이 표시되지 않고 **리스너** 타일이 대신 표시됩니다.

리스너

리스너 목록 및 리스너가 실행 중 상태인지 여부를 표시합니다. 클러스터 상태 타일이 표시되지 않는 경우에만 표시됩니다.

큐 관리자 연결됨

현재 이 큐 관리자에 연결된 큐 관리자의 세부사항을 봅니다.

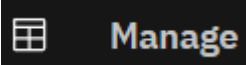
큐 관리자 채널

- 채널 시작, 중지, ping 및 구성 ⋮
- 새 채널 작성
- 채널 재설정


큐 관리자 채널 인스턴스


- 큐 관리자 채널 인스턴스의 상태 보기
- 채널의 인다우트 (in-doubt) 메시지 해석 ⋮

V9.4.0 IBM MQ Console: 로컬 큐 관리자에 대한 작업

관리 보기 의 최상위 레벨에서 로컬 큐 관리자를 작성, 구성 및 제어합니다.

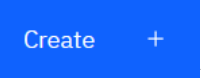

이 태스크 정보





 관리 보기에는 IBM MQ Console 가 실행 중인 IBM MQ 설치. 동일한 시스템에 있는 IBM MQ의 다른 설치와 연관된 큐 관리자는 나열되지 않습니다.

 z/OS에서, 관리 보기는 IBM MQ Console과 동일한 버전의 큐 관리자를 나열하며, IBM MQ Console이 실행 중인 시스템에서 정의됩니다. IBM MQ Console과 다른 버전의 큐 관리자는 나열되지 않습니다. 목록에서 작업할 개별 큐 관리자를 선택할 수 있습니다.

참고: IBM MQ Console 는 활성 상태인 경우 (즉, 기본 역할이 있는 경우) 로컬 RDQM 큐 관리자에 연결할 수 있지만 RDQM 특정 기능을 제공하지 않습니다.

프로시저

- 새 로컬 큐 관리자를 작성하려면 다음을 수행하십시오.
 - a) 큐 관리자 목록 보기에서 작성 단추  를 클릭하십시오.
 - b) 새 큐 관리자의 이름을 입력하십시오. 이름은 최대 48자입니다. 유효한 문자는 문자 및 숫자와 ".", "/", "_" 및 "%" 문자입니다.
 - c) 옵션: 큐 관리자가 대기할 사용 가능한 TCP/IP 포트를 입력하십시오. 포트 번호는 65535를 초과해서는 안 됩니다.
 - d) **작성**을 클릭하십시오. 새 큐 관리자가 작성되어 시작됩니다.
- 로컬 큐 관리자를 시작하려면 다음을 수행하십시오.
 - a) 목록에서 시작할 큐 관리자를 찾으십시오.
 - b) 메뉴  에서 **시작**을 선택하십시오.
- 로컬 큐 관리자를 중지하려면 다음을 수행하십시오.
 - a) 로컬 큐 관리자 위젯의 목록에서 중지할 큐 관리자를 선택하십시오.

- b) 메뉴  에서 **중지**를 선택하십시오.
- 로컬 큐 관리자를 삭제하려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중이면 이를 중지하십시오.
- b) 메뉴  에서 **구성 보기**를 선택하고 **큐 관리자 삭제**를 선택하십시오.
 - c) 확인 창에 이름을 입력하여 큐 관리자를 삭제할 것인지 확인하십시오. 큐 관리자 및 모든 연관된 오브젝트가 삭제됩니다.
- 로컬 큐 관리자의 특성을 보고 편집하려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중인지 확인한 후 큐 관리자 목록에서 찾으십시오.
- b) 메뉴  에서 **구성 보기**를 선택하십시오.
 - c) **특성** 탭이 선택되었는지 확인하십시오. 특성을 확인하고 필요한 경우 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령행에서만 특성을 편집할 수 있습니다. 특성에 대한 정보는 [큐 관리자 특성](#)에서 특성 정보를 볼 수 있습니다.
- 로컬 큐 관리자의 보안 설정에 대해 작업하려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중인지 확인하고 큐 관리자 목록에서 선택하십시오.
- b) 메뉴  에서 **구성 보기**를 선택하십시오.
 - c) **보안** 탭이 선택되었는지 확인하십시오.
 - d) 인증 오브젝트, 권한 부여 레코드 또는 채널 인증 오브젝트에 대해 작업할 수 있습니다. 자세한 정보는 다음 주제를 참조하십시오.
 - [94 페이지의 『IBM MQ Console: 인증 정보 오브젝트에 대한 작업』](#)
 - [95 페이지의 『IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업』](#)
 - [96 페이지의 『IBM MQ Console: 채널 인증 레코드에 대한 작업』](#)

V 9.4.0 IBM MQ Console: 인증 정보 오브젝트에 대한 작업


콘솔을 사용하여 큐 관리자에서 인증 정보 오브젝트를 추가 및 삭제할 수 있습니다. 특성을 보고 설정하거나 오브젝트에 대한 권한 레코드를 관리할 수도 있습니다.

이 태스크 정보

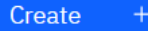


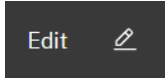



인증 정보 보기는 특정 큐 관리자에 대해 존재하는 인증 정보를 나열합니다. 목록에서 작업할 개별 인증 정보를 선택할 수 있습니다.

큐 관리자 인증 정보는 TLS(Transport Layer Security)에 대한 IBM MQ 지원의 일부입니다. 이러한 오브젝트에는 LDAP 서버에서 OCSP 또는 인증서 폐기 목록(CRL)을 사용하여 인증서 폐기 검사를 수행하는 데 필요한 정의와 사용자 ID 및 비밀번호 검사를 사용으로 설정하는 데 필요한 정의가 포함되어 있습니다.

프로시저

- 큐 관리자에 대한 인증 정보를 보려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중인지 확인하고 큐 관리자 목록에서 선택하십시오.
- b) 메뉴  에서 **구성 보기**를 선택하십시오.
 - c) **보안** 탭이 선택되었는지 확인하십시오.
 - d) 탐색 패널에서 **인증 정보**를 선택하십시오.
- 인증 정보 오브젝트를 추가하려면 다음을 수행하십시오.

Create +

- a) 인증 정보 목록 보기에서 작성 단추  를 클릭하십시오.
 - b) 인증 정보 오브젝트의 이름을 지정합니다. 유효한 문자는 문자 및 숫자와 ".", "/", "_ 및 %" 문자입니다.
 - c) 인증 정보 오브젝트의 유형을 지정하십시오.
 - d) 오브젝트 유형에 해당하는 추가 정보를 지정하십시오.
 - **CRL LDAP**의 경우 **LDAP 서버 이름**을 지정하십시오. 이 이름은 LDAP 서버가 실행 중인 호스트의 호스트 이름, IPv4 점분리 십진수 주소 또는 IPv6 16진 표기법이며 선택적 포트 번호가 포함됩니다. 선택적으로 LDAP 서버에 액세스하는 사용자의 사용자 이름과 비밀번호를 지정할 수 있습니다.
 - **OCSP**의 경우 **OCSP 응답자 URL**을 지정하십시오. 이 URL은 인증서 폐기를 확인하는 데 사용되는 응답자의 URL입니다. 이 값은 OCSP 응답자의 호스트 이름 및 포트 번호를 포함하는 HTTP URL이어야 합니다. OCSP 응답자가 HTTP의 디폴트인 포트 80을 사용하는 경우, 포트 번호를 생략할 수 있습니다. HTTP URL은 RFC 1738에 정의되어 있습니다.
 - **IDPW OS**의 경우 이 인증 유형에 대한 추가 옵션을 선택적으로 지정할 수 있지만 추가 요구사항은 없습니다.
 - **IDPW LDAP**의 경우 **LDAP 서버 이름** 및 **간단한 사용자** 이름을 지정하십시오. LDAP 서버 이름은 LDAP 서버가 실행 중인 호스트의 호스트 이름, IPv4 점분리 십진수 주소 또는 IPv6 16진 표기법이며 선택적 포트 번호가 포함됩니다. 간단한 사용자 이름은 연결에 대한 간단한 이름으로 사용되는 LDAP 사용자 레코드의 필드입니다. 이 인증 유형에 대한 추가 옵션을 선택적으로 지정할 수 있습니다.
 - e) **추가** 를 클릭하십시오.
- 인증 정보 오브젝트를 삭제하려면 다음을 수행하십시오.
 - a) 목록에서 삭제하려는 인증 정보 오브젝트에 대한 스페너 아이콘  을 선택하십시오.
 - b) 오브젝트 특성 보기에서 **인증 정보 오브젝트 삭제**를 클릭하십시오.
 - c) **삭제**를 클릭하여 인증 정보 오브젝트를 삭제할 것인지 확인하십시오. 오브젝트가 삭제됩니다.
 - 인증 정보 오브젝트의 특성을 보고 편집하려면 다음을 수행하십시오.
 - a) 목록에서 보려는 인증 정보 오브젝트에 대한 스페너 아이콘  을 선택하십시오.
 - b) 표시된 특성을 편집하려면 편집 단추  를 클릭하십시오.
 - c) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다.
 - d) 변경사항을 저장하려면 **저장**을 클릭하십시오.
 - 인증 정보 오브젝트의 권한 레코드를 보고 편집하려면 다음을 수행하십시오.
 - a) 목록에서 권한 레코드를 보려는 인증 정보 오브젝트에 대한 스페너 아이콘  을 선택하십시오.
 - b) **보안** 탭을 선택하십시오.
 - c) 기존 권한 레코드를 편집하거나 삭제하려면 메뉴  에서 **편집** 또는 **삭제**를 선택하십시오.
 - d) 새 권한 레코드를 추가하려면 **추가** 단추  를 클릭하고 새 권한 레코드의 세부사항을 제공한 후 **작성**을 클릭하십시오.


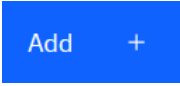
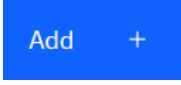
V9.4.0 IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업

해당 사용자 또는 그룹에 대한 권한 레코드를 지정하여 사용자 및 그룹이 큐 관리자에 대해 갖는 액세스 권한을 제어할 수 있습니다.

이 태스크 정보

권한 레코드를 사용하여 메시징 사용자 또는 메시징 사용자 그룹이 특정 큐 관리자에 대해 갖는 액세스 권한을 세부 조정할 수 있습니다. 두 가지 유형의 권한 레코드가 있습니다. 일반 권한을 제어하는 **큐 관리자 액세스** 레코드와 큐 관리자에 대한 오브젝트를 작성할 수 있는 사용자 및 그룹을 제어하는 **작성 권한** 레코드입니다.

프로시저

- 큐 관리자에 대한 권한 레코드를 보려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중인지 확인하고 큐 관리자 목록에서 선택하십시오.
 - b) 메뉴  에서 **구성 보기** 를 선택하십시오.
 - c) **보안** 탭이 선택되었는지 확인하십시오.
 - d) 탐색 패널에서 **권한 레코드**를 선택하십시오. 보기는 두 분할창에서 권한 레코드를 표시하여, 일반 권한 레코드 및 작성 권한 레코드로 작업할 수 있습니다.
- 일반 권한 레코드를 추가하려면 다음을 수행하십시오.
 - a) **큐 관리자 액세스** 목록 보기에서 추가 단추  를 클릭하십시오.
 - b) 사용자 또는 그룹에 대한 권한 레코드를 추가할지 여부를 선택하십시오.
 - c) 권한 레코드를 추가할 사용자 또는 그룹의 이름을 지정하십시오 (권한 레코드는 이를 이름으로 사용함).
 - d) 부여할 권한을 선택하십시오.
 - e) **작성**을 클릭하십시오.
- 작성 권한 레코드를 추가하려면 다음을 수행하십시오.
 - a) **작성할 권한** 목록 보기에서 추가 단추  를 클릭하십시오.
 - b) 사용자 또는 그룹에 대한 권한 레코드를 추가할지 여부를 선택하십시오.
 - c) 권한 레코드를 추가할 사용자 또는 그룹의 이름을 지정하십시오 (권한 레코드는 이를 이름으로 사용함).
 - d) 작성할 권한을 부여할 오브젝트의 유형을 선택하십시오.
 - e) **작성**을 클릭하십시오.
- 권한 레코드를 삭제하려면 다음을 수행하십시오.
 - a) 삭제하려는 권한 레코드를 선택하고 **삭제**를 선택하십시오.
 - b) **삭제**를 클릭하여 인증 정보 오브젝트를 삭제할 것인지 확인하십시오. 오브젝트가 삭제됩니다.
- 권한 레코드의 특성을 보고 편집하려면 다음을 수행하십시오.
 - a) 보려는 권한 레코드를 클릭하십시오.
 - b) 필요에 따라 설정을 변경하고 **저장**을 클릭하여 변경사항을 저장하십시오.

IBM MQ Console: 채널 인증 레코드에 대한 작업

IBM MQ Console을(를) 사용하여 큐 관리자에서 채널 인증 레코드를 추가 및 삭제할 수 있습니다. 채널 인증 레코드에 대한 특성을 보고 설정할 수도 있습니다.


이 태스크 정보

채널 레벨에서 연결 시스템에 부여된 액세스 권한에 대한 더 정밀한 제어를 실행하기 위해 채널 인증 레코드를 사용할 수 있습니다.

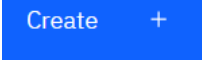
보안을 강화하기 위해 블로킹 채널 인증 레코드를 사용하여 채널에 대한 액세스를 차단할 수 있습니다. 주소 맵 채널 인증 레코드를 사용하여 지정된 사용자에 대한 액세스를 허용할 수도 있습니다. 채널 인증 레코드에 대해 자세히 알아보려면 [채널 인증 레코드](#)를 참조하십시오.

프로시저

- 큐 관리자에 대한 채널 인증 정보를 보려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중인지 확인하고 큐 관리자 목록에서 선택하십시오.


- b) 메뉴  에서 구성 보기 를 선택하십시오.
- c) 보안 탭이 선택되었는지 확인하십시오.
- d) 탐색 패널에서 채널 인증을 선택하십시오.

- 채널 인증 레코드를 추가하려면 다음을 수행하십시오.



- a) 채널 인증 정보 목록 보기에서 작성 단추  를 클릭하십시오.
- b) 사용할 규칙 유형을 선택하십시오. 허용, 차단 또는 경고를 선택하십시오.
- c) 채널 인증 규칙을 구성할 ID 유형을 선택하십시오. 선택한 규칙 유형에 따라 다른 ID 유형을 사용할 수 있습니다.
- d) 지정하는 ID에 대한 필수 정보를 제공하십시오. 기본적으로 값을 제공할 수 있도록 최소 권장 특성이 표시됩니다. 사용자 정의 작성을 선택하여 사용 가능한 모든 특성을 볼 수 있습니다.
- e) 작성 을 클릭하여 채널 인증 레코드를 작성하십시오.

채널 인증 레코드에 사용 가능한 설정에 대한 자세한 내용은 [채널 인증 레코드 및 SET CHLAUTH](#)를 참조하십시오.

- 채널 인증 레코드를 삭제하려면 다음을 수행하십시오.

- a) 삭제하려는 채널 인증 레코드 옆에 있는 스패너 아이콘  을 클릭하십시오.
- b) 채널 인증 편집 보기에서 채널 인증 오브젝트 삭제를 클릭하십시오.
- c) 삭제를 클릭하여 채널 인증 레코드를 삭제할 것인지 확인하십시오. 채널 인증 레코드가 삭제됩니다.

- 채널 인증 레코드의 특성을 보고 편집하려면 다음을 수행하십시오.

- a) 편집 또는 보려는 채널 인증 레코드 옆에 있는 스패너 아이콘  을 클릭하십시오. 특성이 표시됩니다.
- b) 편집 단추  를 클릭하십시오.
- c) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다.
- d) 변경사항을 저장하려면 저장 을 클릭하십시오.

Multi V 9.4.0 IBM MQ Console: 리스너에 대한 작업


IBM MQ Console을 사용하여 리스너 추가 및 삭제, 리스너 시작 및 중지, 리스너 특성 보기 및 설정, 리스너에 대한 권한 레코드 관리를 수행할 수 있습니다.

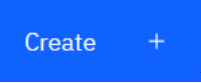



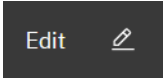


이 태스크 정보

리스너 보기는 특정 큐 관리자에 대해 존재하는 리스너를 표시합니다. 작업할 개별 리스너를 선택할 수 있습니다.

프로시저

- 큐 관리자의 리스너를 보려면 다음을 수행하십시오.
 - a) 큐 관리자가 실행 중인지 확인하고 큐 관리자 목록에서 선택하십시오.

- b) 메뉴  에서 구성 보기 를 선택하십시오.

- c) 리스너 탭을 선택하십시오.
- 리스너를 작성하려면 다음을 수행하십시오.
 - a) 작성 단추  를 클릭하십시오.
 - b) 작성 중인 리스너에 필요한 정보를 제공하십시오.
 - c) **작성**을 클릭하십시오. 새 리스너가 작성됩니다.
- 리스너를 시작하려면 다음을 수행하십시오.
 - a) 목록에서 시작할 리스너를 찾으십시오.
 - b) 메뉴  에서 **시작**을 선택하십시오.
- 리스너를 중지하려면 다음을 수행하십시오.
 - a) 목록에서 시작할 리스너를 찾으십시오.
 - b) 메뉴  에서 **중지**를 선택하십시오.
- 리스너의 특성을 보고 편집하려면 다음을 수행하십시오.
 - a) 목록에서 리스너를 찾으십시오.
 - b) 메뉴  에서 **구성 보기** 를 선택하십시오.
 - c) **특성** 탭이 선택되었는지 확인하십시오. 특성을 편집하려면 편집 단추  를 클릭하십시오.
 - d) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다. 특성에 대한 자세한 정보는 MQ 탐색기 문서에서 [리스너 특성을 참조](#)하십시오.
 - e) 변경사항을 저장하려면 **저장**을 클릭하십시오.
- 리스너의 권한 레코드를 보고 편집하려면 다음을 수행하십시오.
 - a) 목록에서 리스너를 찾으십시오.
 - b) 메뉴  에서 **구성 보기** 를 선택하십시오.
 - c) **보안** 탭을 클릭하십시오.
 - d) 큐 관리자 권한 레코드에 대해 설명된 대로 권한 레코드에 대해 작업하십시오. [95 페이지의 『IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업』](#)의 내용을 참조하십시오.
- 리스너를 삭제하려면 다음을 수행하십시오.
 - a) 목록에서 리스너를 찾으십시오.
 - b) 메뉴  에서 **구성 보기** 를 선택하십시오.
 - c) **리스너 삭제**를 클릭하십시오.

IBM MQ Console: 리모트 큐 관리자 추가

IBM MQ Console을(를) 사용하여 원격 시스템에서 실행 중인 큐 관리자를 관리할 수 있습니다.

시작하기 전에

- 원격으로 관리할 수 있도록 원격 시스템에서 큐 관리자를 준비해야 합니다. [100 페이지의 『명령행을 사용하여 IBM MQ Console에 리모트 큐 관리자 추가』](#)의 [100 페이지의 『1』](#), [100 페이지의 『2』](#), [101 페이지의 『3』](#) 및 [101 페이지의 『4』](#) 단계를 참조하십시오.

- 또한 IBM MQ Console에서 원격 연결을 사용으로 설정해야 합니다. 자세한 정보는 [리모트 큐 관리자 연결 동작 구성을 참조하십시오.](#)

이 태스크 정보

JSON 형식의 CCDT(Client Connection Definition Table)를 사용하여 원격 연결 세부사항을 지정합니다. 텍스트 편집기 (100 페이지의 [『명령행을 사용하여 IBM MQ Console 에 리모트 큐 관리자 추가』](#)의 101 페이지의 [『5』](#) 단계 참조) 를 사용하여 JSON CCDT를 작성하거나 IBM MQ Console를 사용하여 JSON CCDT를 작성할 수 있습니다.

또는 리모트 큐 관리자를 추가할 때 연결 세부사항을 직접 지정하여 IBM MQ Console에서 CCDT를 작성할 수 있습니다.

필요한 모든 태스크(리모트 큐 관리자 준비 및 CCDT 작성 외에)에 대해 명령행을 사용하여 리모트 큐 관리자를 IBM MQ Console에 연결할 수도 있습니다 [100 페이지의 『명령행을 사용하여 IBM MQ Console 에 리모트 큐 관리자 추가』](#) 을 참조하십시오.



주의: 다음 메시지를 수신하는 경우:

```
MQWB2026E: The request to connect to the remote queue manager 'rqmgr-qmgr_name' failed
with the error message:
'JMSSC0051: The property 'JMS_IBM_MQMD_AccountingToken' should be set using type '[B',
not 'java.lang.Object'.'
```

Java 오브젝트 유형 `byte[]` 이 예상되는 경우 계정 토큰에 `java.lang.Object` 을 (를) 전달하려고 합니다.

프로시저

- 기존 CCDT를 지정하여 리모트 큐 관리자를 추가하려면 다음을 수행하십시오.
 - a) 홈 페이지에서 [리모트 큐 관리자 연결](#)을 클릭하십시오.
 - b) 리모트 큐 관리자의 이름을 지정하십시오.
 - c) 선택적으로 큐 관리자의 고유 이름을 지정하십시오. 고유한 이름을 지정하지 않으면 "remote-" 접두부가 추가된 실제 이름이 사용됩니다.
 - d) **JSON CCDT를 사용하여 연결**이 선택되었는지 확인하십시오.
 - e) **찾아보기**를 클릭하고 사용하려는 JSON CCDT가 포함된 파일을 선택하십시오.
 - f) 다음을 클릭하여 사용자 페이지로 이동하고 선택적으로 리모트 큐 관리자에 연결할 사용자 이름과 비밀번호를 지정하십시오. 이 정보를 지정하지 않으면 원격 연결 구성 파일에서 인증 정보를 가져옵니다.
 - g) 다음을 클릭하여 인증서 페이지로 이동하십시오. CCDT가 "transmissionSecurity" 정보를 지정하는 경우 이 정보가 사용됩니다. 선택적으로 인증서(base64로 인코딩된 공개 키)를 붙여넣을 수 있으며 이는 글로벌 신뢰 저장소에 추가됩니다.

인증서는 신뢰 저장소에 추가되기 전에 `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt`에 임시로 저장됩니다. 연결이 성공적으로 추가되면 이 위치에서 인증서가 삭제됩니다.
- 리모트 큐 관리자를 추가하고 연결 정보를 수동으로 지정하려면 다음을 수행하십시오.
 - a) 홈 페이지에서 [리모트 큐 관리자 연결](#)을 클릭하십시오.
 - b) 리모트 큐 관리자의 이름을 지정하십시오.
 - c) 선택적으로 큐 관리자의 고유 이름을 지정하십시오. 고유한 이름을 지정하지 않으면 "remote-" 접두부가 추가된 실제 이름이 사용됩니다.
 - d) **수동 입력**을 선택하십시오.
 - e) 연결에서 사용할 클라이언트 연결 채널의 이름을 입력하십시오.
 - f) 리모트 큐 관리자가 실행 중인 호스트의 이름을 지정하십시오. 원격 MQ 설치가 감지되면 호스트 이름이 표시되고 연결하려는 리모트 큐 관리자의 호스트를 선택할 수 있습니다. 일부 네트워크 구성에서는 원격 MQ 인스턴스를 감지할 수 없습니다. 이 경우 호스트 이름과 포트를 수동으로 추가하십시오.

- g) 다음을 클릭하여 사용자 페이지로 이동하고 선택적으로 리모트 큐 관리자에 연결할 사용자 이름과 비밀번호를 지정하십시오. 이 정보를 지정하지 않으면 원격 연결 구성 파일에서 인증 정보를 가져옵니다.
- h) 다음을 클릭하여 인증서 페이지로 이동하십시오. 드롭다운 목록에서 SSL CipherSpec을 선택할 수 있습니다. 선택적으로 인증서(base64로 인코딩된 공개 키)를 붙여넣을 수 있으며 이는 글로벌 신뢰 저장소에 추가됩니다.

인증서는 신뢰 저장소에 추가되기 전에 `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt`에 임시로 저장됩니다. 연결이 성공적으로 추가되면 이 위치에서 인증서가 삭제됩니다.

- i) 다음을 클릭하여 요약 페이지를 보십시오. 뒤로 단추를 사용하여 이전 페이지를 다시 방문하고 수정할 수 있습니다. 정보가 만족스러우면 연결을 클릭하여 리모트 큐 관리자에 연결하십시오.
- 지정한 연결 정보는 웹 디렉토리의 CCDT 파일에 기록됩니다. 경로는 `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`입니다.

결과

리모트 큐 관리자는 IBM MQ Console의 리모트 큐 관리자 목록에 나타납니다. 연결에 성공하면 로컬 큐 관리자의 오브젝트로 작업하는 것과 동일한 방식으로 리모트 큐 관리자의 오브젝트를 관리할 수 있습니다.

V 9.4.0 명령행을 사용하여 IBM MQ Console 에 리모트 큐 관리자 추가

명령행에서 `setmqweb remote` 명령을 사용하여 IBM MQ Console 에 리모트 큐 관리자를 추가할 수 있습니다. 리모트 큐 관리자는 IBM MQ Console과(와) 동일한 시스템의 다른 설치에서 실행 중인 큐 관리자이거나 다른 시스템에서 실행 중인 큐 관리자일 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다.

- **ALW** ~에 AIX, Linux, and Windows , 다음에서 MQSC 명령을 실행합니다. `runmqsc` 명령 프롬프트. 보다 MQSC 명령을 대화식으로 실행 `runmqsc` 그리고 아래의 텍스트 파일에서 MQSC 명령 실행 `runmqsc` . 이 태스크의 경우 AIX, Linux, and Windows에서 실행 중이면 QM1:

```
runmqsc QM1
```

- **IBM i** ~에 IBM i , 스크립트 파일에 명령 목록을 만든 다음 `STRMQMMQSC` 명령. 보다 MQSC 명령을 사용하여 관리 IBM i .
- **z/OS** ~에 z/OS , MQSC 명령은 명령에 따라 여러 소스에서 실행될 수 있습니다. 보다 MQSC 및 PCF 명령을 실행할 수 있는 소스 IBM MQ for z/OS .

mqweb 서버가 IBM MQ Console에 대한 리모트 큐 관리자 연결을 허용하도록 구성되었는지 확인하십시오. 자세한 정보는 [리모트 큐 관리자 연결 동작 구성](#)을 참조하십시오.

프로시저

1. 리모트 큐 관리자에서 **DEFINE CHANNEL** MQSC 명령을 사용하여 큐 관리자의 리모트 관리를 허용하도록 서버 연결 채널을 작성하십시오.
예를 들어, 큐 관리자 QM1에 대한 서버 연결 채널 QM1.SVRCONN 를 작성하려면 다음 MQSC 명령을 입력하십시오.

```
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

DEFINE CHANNEL 및 사용 가능한 옵션에 대한 자세한 내용은 [DEFINE CHANNEL](#)을 참조하십시오.

2. 적절한 사용자에게 큐 관리자 및 큐 관리자와 연관된 MQ 오브젝트를 관리할 수 있는 권한이 부여되었는지 확인하십시오.
- **ALW** AIX, Linux, and Windows 의 경우 표준 명령행에서 `setmqaut` 제어 명령을 사용하십시오.

- ▶ **z/OS** z/OS에서 RACF 프로파일을 정의하여 권한 부여된 사용자에게 큐 관리자에 대한 액세스 권한을 부여하십시오.

예를 들어, AIX, Linux, and Windows에서 exampleUser 사용자에게 큐 관리자 QM1에 액세스할 수 있는 권한을 부여하려면 다음 제어 명령을 입력하십시오.

```
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

이 권한 부여된 사용자는 다음 사용자 중 하나일 수 있습니다.

- 이 큐 관리자를 원격으로 관리하려는 시스템에서 IBM MQ Console 를 실행하는 mqweb 서버를 시작하는 사용자 ID와 동일한 사용자 ID입니다.
- 102 페이지의 『7』 단계에서 **setmqweb remote** 명령에 포함된 사용자 ID 및 비밀번호와 일치하는 사용자 ID입니다. **setmqweb remote** 명령에 사용자 ID 및 비밀번호를 포함하면 IBM MQ Console 가 큐 관리자에 연결할 때 이 사용자 ID 및 비밀번호가 인증에 사용됩니다.
- 채널 보안 규칙으로 판별되는 사용자 ID입니다. 예를 들어, 원격 관리를 위해 IBM MQ Console 를 사용하는 IP 주소에서 연결을 허용하도록 서버 연결 채널에 채널 인증 규칙을 설정하고 이러한 모든 연결을 큐 관리자를 사용하도록 권한 부여된 특정 사용자 ID에 맵핑할 수 있습니다. 자세한 정보는 [채널에 대한 새 CHLAUTH 규칙 작성](#)을 참조하십시오.

3. **ALW**

리모트 큐 관리자에서 실행 중인 리스너가 없는 경우 **DEFINE LISTENER** MQSC 명령을 사용하여 수신 네트워크 연결을 승인하도록 리스너를 작성하십시오.

예를 들어, 리모트 큐 관리자 QM1에 대해 포트 1414에서 리스너 REMOTE.LISTENER 를 작성하려면 다음 MQSC 명령을 입력하십시오.

```
runmqsc QM1
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
end
```

4. **START LISTENER** MQSC 명령을 사용하여 리스너가 실행 중인지 확인하십시오.

▶ **ALW** 예를 들어, AIX, Linux, and Windows 에서 큐 관리자 QM1에 대한 리스너 REMOTE.LISTENER 를 시작하려면 다음 MQSC 명령을 입력하십시오.

```
runmqsc QM1
START LISTENER(REMOTE.LISTENER)
end
```

▶ **z/OS** 예를 들어, z/OS에서 리스너를 시작하려면 다음 MQSC 명령을 입력하십시오.

```
/cpf START LISTENER TRPTYPE(TCP) PORT(1414)
```

z/OS에서 리스너를 시작하려면 먼저 채널 시작기 주소 공간을 시작해야 합니다.

5. 리모트 큐 관리자 연결 정보가 포함된 JSON CCDT 파일을 작성하십시오.

- 원격으로 연결하려는 큐 관리자와 동일한 설치와 연관된 IBM MQ Console 를 사용하여 CCDT 파일을 생성하십시오.

홈 패널에서 [연결 파일 다운로드](#) 타일을 클릭하십시오.

- 연결을 정의하는 JSON 형식 CCDT 파일을 작성하십시오. JSON 형식 CCDT 작성에 대한 자세한 내용은 [JSON 형식 CCDT 구성](#)을 참조하십시오.

CCDT 파일에는 name, clientConnection 및 type 정보가 포함되어야 합니다. 선택적으로 transmissionSecurity 정보와 같은 추가 정보를 포함할 수 있습니다. 모든 CCDT 채널 속성 정의에 대한 자세한 정보는 [CCDT 채널 속성 정의의 전체 목록](#)을 참조하십시오.

다음 예는 리모트 큐 관리자 연결을 위한 기본 JSON CCDT 파일을 보여줍니다. 100 페이지의 『1』 단계에서 작성한 예제 서버 연결 채널과 동일한 이름으로 채널 이름을 설정하고, 리스너가 사용하는 포트와 동

일한 값으로 연결 포트를 설정합니다. 연결 호스트는 예제 리모트 큐 관리자 QM1이 실행 중인 시스템의 호스트 이름으로 설정됩니다.

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    }
  ],
  "type": "clientConnection"
}
```

6. JSON CCDT 파일을 IBM MQ Console 가 실행 중인 시스템에 복사하십시오.

7. IBM MQ Console를 실행 중인 설치에서 **setmqweb remote** 명령을 사용하여 IBM MQ Console 구성에 리모트 큐 관리자 정보를 추가하십시오.

최소한 리모트 큐 관리자를 IBM MQ Console에 추가하려면 큐 관리자 이름, 큐 관리자의 고유 이름(같은 큐 관리자 이름을 가질 수 있는 다른 리모트 큐 관리자를 구별하기 위해) 및 해당 큐 관리자의 CCDT URL을 제공해야 합니다. 고유 이름은 IBM MQ Console의 표시 이름이므로 리모트 큐 관리자임을 명확하게 하는 이름을 지정하십시오(예: "remote-QM2"). 리모트 큐 관리자 연결에 사용할 사용자 이름 및 비밀번호 또는 신뢰 저장소 및 키 저장소의 세부사항과 같이 지정할 수 있는 몇 가지 추가 옵션이 있습니다. **setmqweb remote** 명령으로 지정할 수 있는 매개변수의 전체 목록은 **setmqweb remote**를 참조하십시오.

예를 들어, 예제 CCDT 파일을 사용하여 예제 리모트 큐 관리자 QM1을 추가하려면 다음 명령을 입력하십시오.

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL "c:\myccdots\ccdt.json"
```

결과

리모트 연결 목록이 다음에 새로 고쳐질 때 리모트 큐 관리자가 IBM MQ Console 의 리모트 큐 관리자 목록에 표시됩니다. 연결에 성공하면 로컬 큐 관리자의 오브젝트로 작업하는 것과 동일한 방식으로 리모트 큐 관리자의 오브젝트를 관리할 수 있습니다.

예

다음 예는 큐 관리자 QM1에 대한 리모트 큐 관리자 연결을 설정합니다. IBM MQ Console 에는 사용자 exampleUser에 부여된 권한을 기반으로 큐 관리자를 관리할 수 있는 권한이 부여됩니다. 이 사용자의 신임 정보는 **setmqweb remote** 명령을 사용하여 리모트 큐 관리자 연결 정보를 구성할 때 IBM MQ Console 에 제공됩니다.

1. 리모트 큐 관리자 QM1 이 있는 시스템에서 서버 연결 채널 및 리스너가 작성됩니다. 리스너가 시작되고 exampleUser 사용자가 큐 관리자를 관리할 수 있도록 권한이 부여됩니다. 예를 들어, AIX, Linux, and Windows에서 다음 명령을 실행하십시오.

```
runmqsc QM1
#Define the server connection channel that will accept connections from the Console
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
# Define the listener to use for the connection from the Console
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
# Start the listener
START LISTENER(REMOTE.LISTENER)
end

#Set mq authorization for exampleUser to access the queue manager
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

2. IBM MQ Console 가 실행 중인 시스템에서 다음 연결 정보를 사용하여 QM1_ccdt.json 파일이 작성됩니다.

```
{
  "channel": [{
```

```

    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }
}

```

3. IBM MQ Console 가 실행 중인 시스템에서 큐 관리자 QM1 에 대한 리모트 큐 관리자 연결 정보가 mqweb 서버에 추가됩니다. exampleUser 의 신임 정보는 연결 정보에 포함되어 있습니다.

```

setmqweb remote add -uniqueName "remote-QM1" -qmgrName "QM1" -ccdtURL
"c:\myccdt\QM1_ccdt.json" -username "exampleUser" -password "password"

```

4. IBM MQ Console 에서는 리모트 큐 관리자 QM1을 보여줍니다.

V 9.4.0 IBM MQ Console: 오브젝트에 대한 작업

각 IBM MQ 큐 관리자는 연관된 여러 가지 서로 다른 유형의 오브젝트를 갖습니다.

이 태스크 정보

콘솔을 사용하여 다음 유형의 IBM MQ 오브젝트에 대해 작업할 수 있습니다.

- 큐
- 이벤트 오브젝트:
 - 토픽
 - 구독
- 애플리케이션 오브젝트:
 - 연결
 - 앱 채널
 - 앱 채널 인스턴스
- MQ 네트워크 오브젝트:
 - 연결된 큐 관리자
 - 큐 관리자 채널
 - 큐 관리자 채널 인스턴스

프로시저

IBM MQ 오브젝트와 작업하려면 다음을 수행하십시오.


1. 큐 관리자 목록 보기에서 작업하려는 오브젝트를 소유하는 큐 관리자를 클릭하십시오.
2. 큐, 이벤트, 애플리케이션 또는 MQ 네트워크 탭을 클릭하여 작업할 오브젝트 유형을 선택하십시오.
3. 오브젝트 작업에 대한 자세한 지시사항은 다음 주제 중 하나를 참조하십시오.

V 9.4.0 IBM MQ Console: 큐에 대한 작업

큐 탭에서 특정 큐 관리자에 대해 존재하는 큐를 볼 수 있습니다. 큐 추가 및 삭제, 큐에 메시지 추가 및 지우기, 메시지 찾아보기, 큐의 특성 보기 및 설정, 큐의 권한 레코드 관리를 수행할 수 있습니다.

이 태스크 정보

큐 보기는 특정 큐 관리자에 대해 존재하는 큐를 나열합니다. 큐 관리자를 클릭하고 큐 탭을 선택하여 큐 목록에 액세스합니다. 목록에서 작업할 개별 큐를 선택할 수 있습니다.

 z/OS에서는 큐에 대한 권한 레코드를 보거나 편집할 수 없습니다.

프로시저

- 큐를 추가하려면 다음을 수행하십시오.

a) 큐 탭에서 작성 단추  를 클릭하십시오.

b) 작성할 큐의 유형을 선택하십시오.

- 로컬 큐 - 큐가 속한 큐 관리자 내 메시지를 저장합니다.
- 알리어스 큐 - 동일한 큐 관리자의 다른 큐에 대한 포인터입니다.
- 리모트 큐 - 다른 큐 관리자에서 다른 큐에 대한 포인터입니다.
- 모델 큐 - 동적 큐 관리자가 작성될 때 사용되는 큐에 대한 템플릿입니다.

c) 작성 중인 큐 유형에 대한 필수 정보를 제공하십시오. 기본적으로 값을 제공할 수 있도록 최소 권장 특성이 표시됩니다. **사용자 정의 작성**을 선택하여 사용 가능한 모든 특성을 볼 수 있습니다.

d) **작성**을 클릭하십시오. 새 큐가 작성됩니다.

- 메시지를 큐에 넣으려면 다음을 수행하십시오.

a) 큐 목록 보기의 목록에서 메시지를 추가할 큐를 클릭하십시오. 모델 큐는 선택할 수 없습니다.

b) 작성 단추  를 클릭하십시오.

c) 큐에 넣을 메시지를 입력하십시오.


d) **작성**을 클릭하십시오.

- 큐에서 메시지를 지우려면 다음을 수행하십시오.


a) 큐 목록에서 메시지를 지울 로컬 큐를 클릭하십시오.

b) 큐 지우기 아이콘 () 을 클릭하십시오.

c) **큐 지우기**를 클릭하여 큐를 지울 것인지 확인하십시오.

-  큐에서 개별 메시지를 삭제하려면 다음을 수행하십시오.

a) 삭제할 메시지를 찾으십시오.

b)  메시지 옆에 있는 삭제 아이콘을 클릭하십시오.

c) **삭제**를 클릭하여 메시지를 지우려는지 확인하십시오.

- 큐에서 메시지를 찾아보려면 큐 목록 보기에서 큐를 클릭하십시오. 해당 큐에 있는 메시지의 목록이 표시됩니다.


- 큐를 삭제하려면 다음을 수행하십시오.


a) 큐 목록에서 삭제하려는 로컬 큐를 누르십시오.


b) 조치 단추  를 클릭하고 **큐 삭제**를 선택하십시오.


c) **삭제**를 클릭하여 큐를 삭제할 것인지 확인하십시오. 큐가 삭제됩니다.


- 큐의 특성을 보고 편집하려면 다음을 수행하십시오.

a) 편집할 큐 옆에 있는 메뉴  에서 **구성 보기**를 선택하십시오.

- b) 편집 단추  를 클릭하십시오.
 - c) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다. 특성에 대한 정보는 IBM MQ Explorer 문서의 [큐 특성](#)의 내용을 참조하십시오.
 - d) 변경사항을 저장하려면 **저장**을 클릭하십시오.
- 큐의 권한 레코드를 보고 편집하려면 다음을 수행하십시오.

- a) 편집할 큐 옆에 있는 메뉴  에서 **구성 보기**를 선택하십시오.
- b) **보안** 탭을 클릭하십시오.
- c) 큐 관리자 권한 레코드에 대해 설명된 대로 권한 레코드에 대해 작업하십시오. [95 페이지의 『IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업』](#)의 내용을 참조하십시오.

-  **V 9.4.0**
큐와 연관된 IBM MQ 오브젝트를 보려면 다음을 수행하십시오.

- a) 보려는 큐 옆에 있는 메뉴  에서 **연관된 오브젝트 보기**를 선택하십시오.
- b) 표시되는 패널에서 오브젝트를 보십시오. 나열된 각 오브젝트에 대한 세부사항을 보려면 링크를 클릭하십시오.


이 패널을 사용하여 큐에 메시지를 넣는 애플리케이션을 보고 서로 다른 큐 간의 관계를 볼 수 있습니다. 이는 문제를 식별하고 해결하는 데 도움이 될 수 있습니다.

IBM MQ Console: 주제에 대한 작업

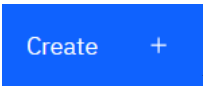


IBM MQ Console을 사용하여 토픽 추가 및 삭제하고 토픽의 특성을 설정할 수 있습니다.


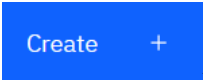
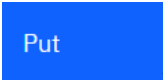

이 태스크 정보

토픽 보기는 특정 큐 관리자에 대해 존재하는 토픽을 나열합니다. 큐 관리자 **이벤트** 탭에서 토픽에 액세스합니다. 목록에서 작업할 개별 토픽을 선택할 수 있습니다.

 **z/OS** z/OS에서는 토픽에 대한 권한 레코드를 보거나 편집할 수 없습니다.

프로시저

- 토픽을 추가하려면 다음을 수행하십시오.
 - a) 큐 관리자 보기에서 **이벤트** 탭을 열고 **토픽**을 클릭하십시오.
 - b) 작성 단추  를 클릭하십시오.
 - c) 작성하려는 토픽에 대한 필수 정보를 제공하십시오. 기본적으로 값을 제공할 수 있도록 최소 권장 특성이 표시됩니다. **사용자 정의 작성**을 선택하여 사용 가능한 모든 특성을 볼 수 있습니다.
 - d) **작성**을 클릭하십시오. 새 토픽이 작성됩니다.
- 토픽을 삭제하려면 다음을 수행하십시오.
 - a) 삭제할 토픽 옆 스페너 아이콘  을 클릭하십시오.
 - b) 편집 큐 보기에서 **토픽 삭제**를 클릭하십시오.
 - c) **삭제**를 클릭하여 토픽을 삭제할 것인지 확인하십시오. 토픽이 삭제됩니다.
- 토픽의 특성을 보고 편집하려면 다음을 수행하십시오.
 - a) 편집할 토픽 옆 스페너 아이콘  을 클릭하십시오.

- b) 편집 단추  를 클릭하십시오.
- c) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다. 특성에 대한 정보는 MQ 탐색기 문서에서 [토픽 특성을 참조하십시오](#).
- d) 변경사항을 저장하려면 **저장**을 클릭하십시오.
- 토픽에 메시지를 발행하려면 하나 이상의 일치하는 구독이 있어야 합니다.
 - a) 토픽 목록에서 발행할 토픽을 클릭하십시오.
 - b) 일치하는 구독 이름을 클릭하십시오.
- c) 작성 단추  를 클릭하십시오.
- d) 발행하려는 메시지를 입력하십시오.
- e) 넣기 단추  를 클릭하십시오. 메시지는 일치하는 모든 구독에 기록됩니다.
- 토픽을 구독하려면 106 페이지의 『IBM MQ Console: 구독에 대한 작업』의 내용을 참조하십시오.
- 토픽의 권한 레코드를 보고 편집하려면 다음을 수행하십시오.
 - a) 권한 레코드를 편집할 토픽 옆 스페너 아이콘  을 클릭하십시오.
 - b) **보안** 탭을 클릭하십시오.
 - c) 큐 관리자 권한 레코드에 대해 설명된 대로 권한 레코드에 대한 작업은 95 페이지의 『IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업』의 내용을 참조하십시오.


IBM MQ Console: 구독에 대한 작업

IBM MQ Console을 사용하여 구독을 추가 및 삭제하고 구독의 특성을 보고 설정할 수 있습니다.


이 태스크 정보



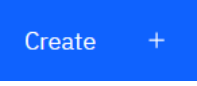

구독 보기는 특정 큐 관리자에 대해 존재하는 구독을 나열합니다. 큐 관리자 **이벤트** 탭에서 구독에 액세스합니다. 목록에서 작업할 개별 토픽을 선택할 수 있습니다. 목록에서 작업할 개별 구독을 선택할 수 있습니다.

구독에 대한 자세한 정보는 [구독자 및 구독과 DEFINE SUB](#)를 참조하십시오.

 z/OS에서 구독에 대한 권한 레코드를 보거나 편집할 수 없습니다.

프로시저

- 구독을 추가하려면 다음을 수행하십시오.
 - a) 큐 관리자 보기에서 **이벤트** 탭을 열고 **등록**을 클릭하십시오.
 - b) 관리 또는 비관리 구독을 작성할지 여부를 선택하십시오.
 - c) 작성 중인 구독에 대한 필수 정보를 제공하십시오. 기본적으로 값을 제공할 수 있도록 최소 권장 특성이 표시됩니다. **사용자 정의 작성**을 선택하여 사용 가능한 모든 특성을 볼 수 있습니다.
 - d) **작성**을 클릭하십시오. 새 구독이 작성됩니다.
- 구독을 삭제하려면 다음을 수행하십시오.
 - a) 구독 옆에 있는 삭제할 스페너 아이콘  을 클릭하십시오.
 - b) 편집 큐 보기에서 **구독 삭제**를 클릭하십시오.
 - c) **삭제**를 클릭하여 구독을 삭제할 것인지 확인하십시오. 구독이 삭제됩니다.
- 구독의 특성을 보고 편집하려면 다음을 수행하십시오.


- a) 구독 옆에 있는 편집할 스페너 아이콘  을 클릭하십시오.
- b) 편집 단추  를 클릭하십시오.
- c) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다.
- d) 변경사항을 저장하려면 **저장**을 클릭하십시오.
- 구독이 구독되는 토픽에 대한 메시지를 발행하려면 다음을 수행하십시오.
 - a) 구독 목록에서 토픽을 발행할 주제의 구독을 클릭하십시오.
 - b) 작성 단추  를 클릭하십시오.
 - c) 발행하려는 메시지를 입력하십시오.
 - d) 넣기 단추  를 클릭하십시오. 메시지는 사용자가 발행한 토픽과 일치하는 모든 구독에 기록됩니다.

IBM MQ Console: 큐 관리자 채널에 대한 작업

큐 관리자 채널로 IBM MQ Console 작업을 사용할 수 있습니다. 큐 관리자 채널을 추가 및 삭제하고, 채널을 시작 및 중지하고, 채널을 재설정 및 해결하며, 채널에 ping할 수 있습니다. 또한 큐 관리자 채널의 특성을 보고 설정하거나 채널에 대한 권한 레코드를 관리할 수 있습니다.

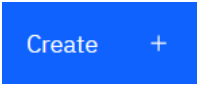


이 태스크 정보



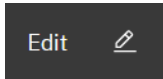





큐 관리자 채널은 네트워크에서 큐 관리자 사이에 메시지를 전송하는 논리 통신 링크입니다. 큐 관리자 채널 보기에는 실행 중인 채널 수, 재시도 수 및 중지 수에 대한 빠른 보기를 표시하는 패널이 포함되어 있습니다.

 z/OS에서는 채널에 대한 권한 레코드를 보거나 편집할 수 없습니다.

프로시저

- 큐 관리자 채널을 추가하려면 다음을 수행하십시오.

- a) 큐 관리자 보기에서 **MQ 네트워크** 탭을 열고 **큐 관리자 채널** 을 클릭한 후 작성 단추 
 - b) 작성하려는 큐 관리자 채널의 유형을 선택하고 다음 단추  를 클릭하십시오.
 - c) 작성 중인 채널에 대한 필수 정보를 제공하십시오. 기본적으로 값을 제공할 수 있도록 최소 권장 특성이 표시됩니다. **사용자 정의 작성**을 선택하여 사용 가능한 모든 특성을 볼 수 있습니다.
 - d) **작성**을 클릭하십시오. 새 채널은 **비활성** 상태로 작성됩니다.
- 큐 관리자 채널을 시작하려면 다음을 수행하십시오.
 - a) 목록에서 시작할 채널을 찾으십시오.
 - b) 메뉴  에서 **시작**을 선택하십시오.
- 큐 관리자 채널을 중지하려면 다음을 수행하십시오.
 - a) 목록에서 중지할 채널을 찾으십시오.

- b) 메뉴  에서 **중지**를 선택하십시오.
- 큐 관리자 채널의 특성을 보려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **구성 보기** 를 선택하십시오.
 - c) **특성** 탭이 선택되었는지 확인하십시오. 특성을 편집하려면 편집 단추  를 클릭하십시오.
 - d) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다. 특성에 대한 자세한 정보는 MQ 탐색기 문서에서 [채널 특성](#)을 참조하십시오.
 - e) 변경사항을 저장하려면 **저장**을 클릭하십시오.
- 큐 관리자 채널을 재설정하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **고급**을 선택하십시오.
 - c) **재설정** 섹션에서 메시지 순서 번호를 지정하십시오.
양쪽 끝에서 송신할 다음 메시지의 순서 번호가 일치하지 않으므로 시작되지 않는 경우 채널을 재설정해야 합니다. 메시지 순서 번호가 해당 번호를 지정합니다.
 - d) **채널 재설정**을 클릭하십시오.
- 송신자 또는 서버 채널을 해결하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **고급**을 선택하십시오.
 - c) **해결** 섹션에서, **메시지를 전송 큐로 복원** 또는 **메시지 제거**를 클릭하여 현재 메시지의 배치를 커밋하거나 백아웃할 지 여부를 선택하십시오.
- 큐 관리자 채널을 ping하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **Ping**을 선택하십시오.
- 큐 관리자 채널에 대한 권한 레코드를 보고 편집하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **구성 보기** 를 선택하십시오.
 - c) **보안** 탭을 클릭하십시오.
 - d) 큐 관리자 권한 레코드에 대해 설명된 대로 권한 레코드에 대한 작업은 95 페이지의 『IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업』의 내용을 참조하십시오.
- 큐 관리자 채널을 삭제하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **구성**을 선택하십시오.
 - c) **채널 삭제**를 클릭하십시오.

V 9.4.0 IBM MQ Console: 애플리케이션 채널에 대한 작업

IBM MQ Console을 사용하여 애플리케이션 채널을 작업할 수 있습니다. 채널 추가 및 삭제, 채널 시작 및 중지, 채널 재설정 및 해결, 채널 ping을 수행할 수 있습니다. 또한 애플리케이션의 특성을 보고 설정하고 채널에 대한 권한 레코드를 관리할 수도 있습니다.

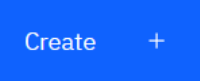
이 태스크 정보

애플리케이션 채널은 네트워크에서 큐 관리자에 연결하기 위해 애플리케이션이 사용하는 논리 통신 링크입니다. 애플리케이션 채널 보기에는 실행 중인 채널 수, 재시도 수 및 중지 수에 대한 빠른 보기를 표시하는 패널이 포함되어 있습니다.

z/OS z/OS에서는 채널에 대한 권한 레코드를 보거나 편집할 수 없습니다.

프로시저

- 애플리케이션 채널을 추가하려면 다음을 수행하십시오.

a) 큐 관리자 보기에서 **애플리케이션** 탭을 열고 **앱 채널** 을 클릭한 후 작성 단추  를 클릭하십시오.


b) 다음 단추  를 클릭하십시오.

c) 작성 중인 채널에 대한 필수 정보를 제공하십시오. 기본적으로 값을 제공할 수 있도록 최소 권장 특성이 표시됩니다. **사용자 정의 작성**을 선택하여 사용 가능한 모든 특성을 볼 수 있습니다.

d) **작성**을 클릭하십시오. 새 채널은 **비활성** 상태로 작성됩니다.


- 애플리케이션 채널을 시작하려면 다음을 수행하십시오.

a) 목록에서 시작할 채널을 찾으십시오.

b) 메뉴  에서 **시작**을 선택하십시오.


- 애플리케이션 채널을 중지하려면 다음을 수행하십시오.

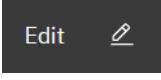
a) 목록에서 중지할 채널을 찾으십시오.

b) 메뉴  에서 **중지**를 선택하십시오.

- 애플리케이션 채널의 특성을 보려면 다음을 수행하십시오.

a) 목록에서 채널을 찾으십시오.

b) 메뉴  에서 **구성 보기** 를 선택하십시오.


c) **특성** 탭이 선택되었는지 확인하십시오. 특성을 편집하려면 편집 단추  를 클릭하십시오.

d) 필요에 따라 특성을 편집하십시오. 특성 텍스트 상자가 사용 불가능한 경우 특성이 읽기 전용이거나 명령 행에서만 특성을 편집할 수 있습니다. 특성에 대한 자세한 정보는 MQ 탐색기 문서에서 **채널 특성**을 참조하십시오.

e) 변경사항을 저장하려면 **저장**을 클릭하십시오.





- 애플리케이션 채널을 재설정하려면 다음을 수행하십시오.

a) 목록에서 채널을 찾으십시오.

b) 메뉴  에서 **고급**을 선택하십시오.

c) **재설정** 섹션에서 메시지 순서 번호를 지정하십시오.

양쪽 끝에서 송신할 다음 메시지의 순서 번호가 일치하지 않으므로 시작되지 않는 경우 채널을 재설정해야 합니다. 메시지 순서 번호가 해당 번호를 지정합니다.

- d) **채널 재설정**을 클릭하십시오.
- 송신자 또는 서버 채널을 해결하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **고급**을 선택하십시오.
 - c) **해결** 섹션에서, **메시지를 전송 큐로 복원** 또는 **메시지 제거**를 클릭하여 현재 메시지의 배치를 커밋하거나 백아울할 지 여부를 선택하십시오.
- 채널을 ping하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **Ping**을 선택하십시오.
- 애플리케이션 채널에 대한 권한 레코드를 보고 편집하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **구성**을 선택하십시오.
 - c) **보안** 탭을 클릭하십시오.
 - d) 큐 관리자 권한 레코드에 대해 설명된 대로 권한 레코드에 대한 작업은 [95 페이지의 『IBM MQ Console: 큐 관리자 권한 레코드에 대한 작업』](#)의 내용을 참조하십시오.
- 애플리케이션 채널을 삭제하려면 다음을 수행하십시오.
 - a) 목록에서 채널을 찾으십시오.
 - b) 메뉴  에서 **구성**을 선택하십시오.
 - c) **채널 삭제**를 클릭하십시오.

IBM MQ Console: 애플리케이션에 대한 작업

IBM MQ Console 를 사용하여 큐 관리자에 연결된 애플리케이션에 대한 정보를 볼 수 있습니다.

이 태스크 정보

애플리케이션은 서버 연결 채널을 사용하여 네트워크를 통해 큐 관리자에 연결됩니다. 애플리케이션 보기에는 큐 관리자에 연결된 애플리케이션 수의 빠른 보기를 표시하는 패널이 포함되어 있습니다.

프로시저

- 애플리케이션 정보를 보려면 다음을 수행하십시오.
 - a) 큐 관리자 보기에서 **애플리케이션** 탭을 여십시오.
 - b) **연결된 애플리케이션** 을 클릭하여 애플리케이션 보기를 여십시오.
 - c) 애플리케이션의 여러 인스턴스가 있는 경우 아래로 화살표를 클릭하여 각 인스턴스의 세부사항을 보십시오.
 - d) 자세한 정보를 보려면 보기에서 오브젝트를 클릭하십시오.

IBM MQ Console: Working with storage classes

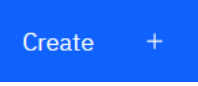
You can use the IBM MQ Console to add, view, delete and update storage classes on z/OS queue managers.

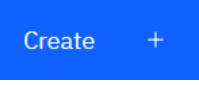
About this task

The storage classes view lists the storage classes that exist for a specific queue manager. You access **Storage classes** from the sidebar on the queue manager **Queues** tab.


See [Storage classes for IBM MQ for z/OS](#) and [DEFINE STGLASS](#) for more information about storage classes.

Procedure


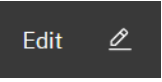
- To add a storage class:
 - From the queue manager view, open the **Queues** tab, and click **Storage classes**.
 - On the **Storage classes** screen, click the **Create**  button.
 - Provide the required information for the storage class you are creating.
By default, the minimum recommended properties you need to provide values for are displayed. You can view all of the available properties by selecting **Custom create**.

- Click the **Create**  button.
The new storage class is created.
- To delete a storage class:



- Click the spanner button  next to the storage class that you want to delete.
 - In the Edit storage class view, click **Delete storage class**.
 - Confirm that you want to delete the queue by clicking **Delete**. The storage class is deleted.
- To view and edit the properties of a storage class:



- Click the spanner button  next to the storage class that you want to edit.
- Click the Edit button .
- Edit the properties as required. If the property text box is disabled, the property is read-only, or can be set only at the time of creation.
- Click **Save** to save your changes.

IBM MQ Console: Working with page sets and buffer pools

You can use the IBM MQ Console to view page sets and buffer pools on z/OS queue managers.

About this task

The page sets and buffer pools views list the page sets and buffer pools that exist for a specific queue manager. You access the **Page sets** and **Buffer pools** views from the sidebar of the queue manager **Queues** tab

See [Page sets for IBM MQ for z/OS](#) for more information about page sets, and [Buffers and buffer pools for IBM MQ for z/OS](#) for more information about buffer pools.

Procedure

- To view the properties of a page set



Click the spanner button next to the page set that you want to view.

- To view the properties of a buffer pool



Click the spanner button next to the buffer pool that you want to view.

IBM MQ Console 설정

IBM MQ Console에 대한 일부 일반 설정을 지정할 수 있습니다.

설정 아이콘  **Settings** 을 클릭하여 IBM MQ Console 설정 보기로 전환하십시오.

설정을 사용하여 다음 기능을 제어하십시오.

- 10초마다 큐 관리자의 자동 새로 고치기. 이 기능은 켜거나 끌 수 있습니다.
- 시스템 오브젝트가 표시되는지 여부. 모든 오브젝트 유형에 대해 이를 지정하거나 개별적으로 오브젝트 유형을 선택할 수 있습니다.
- 추적 정보가 수집되는지 여부.

Windows

Linux

IBM MQ Explorer를 사용하여 관리

IBM MQ Explorer를 사용하면 Windows 또는 Linux x86-64에서만 실행 중인 컴퓨터에서 네트워크의 로컬 또는 원격 관리를 수행할 수 있습니다.

Linux x86-64 용 IBM MQ for Windows 및 IBM MQ 는 제어 또는 MQSC 명령을 사용하는 대신 관리 태스크를 수행하기 위해 IBM MQ Explorer 라는 관리 인터페이스를 제공합니다. [명령 세트 비교](#)에서는 IBM MQ Explorer를 사용하여 수행할 수 있는 것을 표시합니다.

IBM MQ Explorer 를 사용하면 관심있는 큐 관리자 및 클러스터에서 IBM MQ Explorer 를 지정하여 Windows 또는 Linux x86-64를 실행 중인 컴퓨터에서 네트워크의 로컬 또는 원격 관리를 수행할 수 있습니다. z/OS 등의 지원되는 플랫폼에서 실행되는 큐 관리자에 원격으로 연결하여 콘솔에서 전체 메시징 백본을 확인, 탐색 및 대체할 수 있습니다.

IBM MQ Explorer 가 관리할 수 있도록 리모트 IBM MQ 큐 관리자를 구성하려면 [114 페이지의 『IBM MQ Explorer의 필수 소프트웨어 및 정의』](#)의 내용을 참조하십시오.

이를 사용하여 Windows 또는 Linux x86-64 시스템 도메인 내에서 로컬 또는 원격으로 IBM MQ에 대한 작업 환경을 설정하고 미세 조정하는 것과 일반적으로 연관된 태스크를 수행할 수 있습니다.

Linux에서 IBM MQ Explorer는 Eclipse가 둘 이상 설치된 경우 시작되지 못할 수 있습니다. 이것이 발생하면 기타 Eclipse 설치에 사용하는 것에 다른 사용자 ID를 사용하여 IBM MQ Explorer를 시작하십시오.

Linux에서 IBM MQ Explorer를 성공적으로 시작하려면 홈 디렉토리가 존재하고 이 홈 디렉토리에 파일을 작성할 수 있어야 합니다.

IBM MQ Explorer 는 Fix Central에서 사용 가능한 독립형 IBM MQ Explorer 다운로드에서 설치할 수 있습니다. 자세한 정보는 Linux 및 Windows 에서 IBM MQ Explorer 를 독립형 애플리케이션으로 설치 및 설치 제거의 내용을 참조하십시오.

IBM MQ Explorer를 사용하면 일련의 콘텐츠 보기 및 특성 대화 상자를 사용하여 관리 태스크를 수행할 수 있습니다. 하나 이상의 Eclipse 플러그인을 작성하여 IBM MQ Explorer를 확장할 수도 있습니다.

IBM MQ Explorer 태스크

IBM MQ Explorer에서 다음 태스크를 수행할 수 있습니다.

- 큐 관리자를 작성하고 [삭제](#)하십시오(로컬 시스템에만 해당됨).
- [큐 관리자를 시작](#)하고 [중지](#)하십시오(로컬 시스템에만 해당됨).
- 큐 및 채널과 같은 [IBM MQ 오브젝트를 정의 및 표시](#)하고 대체하십시오.
- 큐에서 [메시지 찾아보기](#)를 수행하십시오.
- [채널을 시작 및 중지](#)하십시오.
- 채널, 리스너, 큐 또는 서비스 오브젝트에 대한 [상태 정보 보기](#)를 수행하십시오.
- 클러스터에서 큐 관리자를 보십시오.
- 특정 큐가 열려 있는 애플리케이션, 사용자 또는 채널을 확인하십시오.
- 새 클러스터 작성 마법사를 사용하여 [새 큐 관리자 클러스터](#)를 작성하십시오.
- 클러스터에 큐 관리자 추가 마법사를 사용하여 클러스터에 [큐 관리자를 추가](#)하십시오.
- TLS(Transport Layer Security) 채널 보안에 사용되는 [인증 정보 오브젝트](#)를 관리하십시오.
- 채널 시작기, 트리거 모니터 및 리스너를 작성하고 삭제하십시오.
- 명령 서버, 채널 시작기, 트리거 모니터 및 리스너를 시작하거나 중지하십시오.
- [큐 관리자가 시작될 때 자동으로 시작하도록 특정 서비스를 설정](#)하십시오.
- 큐 관리자의 특성을 수정하십시오.
- 로컬 기본 큐 관리자를 변경하십시오.
- [IBM MQ 오브젝트에서 JMS 오브젝트 작성 및 JMS 오브젝트의 IBM MQ 오브젝트](#).
- 현재 지원되는 유형 중 임의의 유형에 대해 [JMS 연결 팩토리](#)를 작성하십시오.
- 서비스에 대한 매개변수(예: 리스너의 경우 TCP 포트 번호) 또는 채널 시작기 큐 이름을 수정하십시오.
- [서비스 추적을 시작](#)하거나 [중지](#)하십시오.

콘텐츠 보기 및 특성 대화 상자

일련의 콘텐츠 보기 및 특성 대화 상자를 사용하여 관리 태스크를 수행합니다.

콘텐츠 보기

콘텐츠 보기는 다음을 표시할 수 있는 패널입니다.

- IBM MQ 자체에 관한 속성 및 관리 옵션.
- 하나 이상의 관련된 오브젝트에 관한 속성 및 관리 옵션.
- 클러스터에 대한 속성 및 관리 옵션.

특성 대화 상자

특성 대화 상자는 오브젝트에 관한 속성을 일련의 필드에 표시하는 패널로, 필드 중 일부를 편집할 수 있습니다.

네비게이터 보기를 사용하여 IBM MQ Explorer를 탐색합니다. 네비게이터를 사용하면 요청하는 콘텐츠 보기를 선택할 수 있습니다.

IBM MQ Explorer 확장

IBM MQ Explorer는 Eclipse가 지원하는 Eclipse 프레임워크와 기타 플러그인 애플리케이션의 플랫폼과 일치하는 스타일로 정보를 제공합니다.

시스템 관리자는 IBM MQ Explorer 확장을 통해 IBM MQ Explorer 를 사용자 정의하여 IBM MQ를 관리하는 방법을 개선할 수 있습니다.

자세한 정보는 [MQ Explorer 확장](#)을 참조하십시오.

Windows Linux IBM MQ Explorer 사용 여부 결정

설치 시 IBM MQ Explorer를 사용할지 여부를 결정할 때 이 주제에 나열된 정보를 고려하십시오.

다음과 같은 점을 알고 있어야 합니다.

오브젝트 이름

MQSC 명령을 사용하여 오브젝트에 대해 작업할 때 IBM MQ Explorer에서 큐 관리자 및 기타 오브젝트에 소문자 이름을 사용하는 경우, 오브젝트 이름을 작은따옴표로 묶어야 합니다. 그렇지 않으면 IBM MQ가 이를 인식하지 못합니다.

큰 큐 관리자

IBM MQ Explorer는 작은 큐 관리자를 사용할 때 가장 잘 작동됩니다. 단일 큐 관리자에 많은 수의 오브젝트가 있는 경우, IBM MQ Explorer가 보기에 표시할 필수 정보를 추출하는 동안 지연이 발생할 수 있습니다.

클러스터

IBM MQ 클러스터는 잠재적으로 수백 또는 수천 개의 큐 관리자를 포함할 수 있습니다. IBM MQ Explorer는 트리 구조를 사용하여 클러스터에서 큐 관리자를 표시합니다. 클러스터에서 큐 관리자를 선택할 때까지 IBM MQ Explorer가 큐 관리자에 연결되지 않으므로 클러스터의 실제 크기는 IBM MQ Explorer의 속도에 큰 영향을 주지 않습니다.

IBM MQ Explorer 설정

다음 절에서는 IBM MQ Explorer를 설정하기 위해 수행해야 하는 단계를 간략하게 설명합니다.

- [114 페이지의 『IBM MQ Explorer의 필수 소프트웨어 및 정의』](#)
- [115 페이지의 『IBM MQ Explorer 보안』](#)
- [118 페이지의 『IBM MQ Explorer에서 큐 관리자 및 클러스터 표시 및 숨기기』](#)
- [118 페이지의 『클러스터 멤버십 및 IBM MQ Explorer』](#)
- [119 페이지의 『IBM MQ Explorer에 대한 데이터 변환』](#)

IBM MQ Explorer의 필수 소프트웨어 및 정의

IBM MQ Explorer 사용을 시도하기 전에 다음 요구사항을 충족하는지 확인하십시오.

IBM MQ Explorer는 TCP/IP 통신 프로토콜을 사용하여 리모트 큐 관리자에 연결될 수 있습니다.

다음은 확인하십시오.

1. 명령 서버는 원격으로 관리되는 모든 큐 관리자에서 실행되고 있습니다.
2. 적절한 TCP/IP 리스너 오브젝트는 모든 리모트 큐 관리자에서 실행되어야 합니다. 이 오브젝트는 IBM MQ 리스너 또는 AIX and Linux 시스템에 inetd 디먼일 수 있습니다.
3. 기본 이름 지정된 SYSTEM.ADMIN.SVRCONN에 의해 서버 연결 채널은 모든 리모트 큐 관리자에 존재합니다.

다음 MQSC 명령을 사용하여 채널을 작성할 수 있습니다.

```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

이 명령은 기본 채널 정의를 작성합니다. 더 복잡한 정의(예: 보안을 설정하기 위해)를 원하는 경우, 추가 매개 변수가 필요합니다. 자세한 정보는 [채널 정의](#)를 참조하십시오.

4. 시스템 큐(SYSTEM.MQEXPLORER.REPLY.MODEL)가 존재해야 합니다.

IBM MQ Explorer 보안

특정 오브젝트에 대한 사용자 액세스 제어가 중요한 환경에서 IBM MQ를 사용하는 경우 IBM MQ Explorer 사용의 보안 측면을 고려해야 합니다.

IBM MQ Explorer를 사용하기 위한 권한 부여

사용자는 IBM MQ Explorer를 사용할 수 있지만, 큐 관리자에 연결 및 액세스하고 이를 관리하기 위해서는 특정 권한이 필요합니다.

IBM MQ Explorer를 사용하여 로컬 관리 태스크를 수행하려면, 사용자는 관리 태스크를 수행하기 위한 필수 권한을 가지고 있어야 합니다. 사용자가 mqm 그룹의 구성원인 경우, 사용자에게 모든 로컬 관리 태스크를 수행하기 위한 권한이 있습니다.

IBM MQ Explorer를 사용하여 리모트 큐 관리자에 연결하고 원격 관리 태스크를 수행하기 위해 IBM MQ Explorer를 실행하는 사용자는 다음 권한을 가지고 있어야 합니다.

- 대상 큐 관리자 오브젝트의 CONNECT 권한
- 대상 큐 관리자 오브젝트의 INQUIRE 권한
- 대상 큐 관리자 오브젝트의 DISPLAY 권한
- 큐 SYSTEM.MQEXPLORER.REPLY.MODEL에 대한 INQUIRE 권한
- 큐 SYSTEM.MQEXPLORER.REPLY.MODEL에 대한 DISPLAY 권한
- 큐 SYSTEM.MQEXPLORER.REPLY.MODEL에 대한 INPUT(get) 권한
- 큐 SYSTEM.MQEXPLORER.REPLY.MODEL에 대한 OUTPUT(put) 권한
- 큐 SYSTEM.ADMIN.COMMAND.QUEUE에 대한 OUTPUT(put) 권한
- 큐 SYSTEM.ADMIN.COMMAND.QUEUE의 INQUIRE 권한
- 선택된 조치를 수행하기 위한 권한

참고: INPUT 권한은 큐로부터 사용자로의 입력(get 조작)과 관련됩니다. OUTPUT 권한은 사용자에서 큐로의 출력에 관련됩니다(Put 조작).

IBM MQ for z/OS에서 리모트 큐 관리자에 연결하고 IBM MQ Explorer를 사용하여 원격 관리 태스크를 수행하려면 다음이 제공되어야 합니다.

- 시스템 큐 SYSTEM.MQEXPLORER.REPLY.MODEL에 대한 RACF® 프로파일
- 큐 AMQ.MQEXPLORER.*에 대한 RACF 프로파일

또한 IBM MQ Explorer를 실행하는 사용자는 다음 권한을 가지고 있어야 합니다.

- RACF 시스템 큐에 대한 UPDATE 권한, SYSTEM.MQEXPLORER.REPLY.MODEL
- 큐 AMQ.MQEXPLORER.*에 대한 RACF UPDATE 권한
- 대상 큐 관리자 오브젝트의 CONNECT 권한
- 선택된 조치를 수행하기 위한 권한
- MQCMD5 클래스에서 모든 hlq.DISPLAY.object 프로파일에 대한 READ 권한

IBM MQ 오브젝트에 권한을 부여하는 방법에 대한 정보는 [AIX, Linux, and Windows 시스템에서 IBM MQ 오브젝트에 대한 액세스 부여를 참조하십시오](#).

사용자가 수행하기 위해 권한을 부여 받지 않는 조작을 수행하려고 시도하면, 대상 큐 관리자는 인증 실패 프로시저를 호출하고 작동에 실패합니다.

IBM MQ Explorer의 기본 필터는 모든 IBM MQ 오브젝트를 표시하는 것입니다. 사용자에게 DISPLAY 권한이 없는 IBM MQ 오브젝트가 있는 경우, 권한 실패가 생성됩니다. 권한 이벤트가 기록되는 경우, 사용자에게 DISPLAY 권한이 있는 해당 오브젝트에 표시되는 오브젝트 범위를 제한하십시오.

IBM MQ Explorer에서 리모트 큐 관리자에 연결하기 위한 보안

IBM MQ Explorer 및 각 리모트 큐 관리자 간의 채널을 보안 설정해야 합니다.

IBM MQ Explorer는 리모트 큐 관리자에 MQI 클라이언트 애플리케이션으로 연결합니다. 이는 각 리모트 큐 관리자에 서버 연결 채널의 정의 및 적절한 TCP/IP 리스너가 있어야 한다는 것을 의미합니다. 서버 연결 채널을 보

안 설정하지 않으면 악의적인 애플리케이션이 동일한 서버 연결 채널에 연결하고 무제한 권한으로 큐 관리자 오브젝트에 액세스할 수 있습니다. 서버 연결 채널을 보안 설정하려면 채널의 MCAUSER 속성에 공백이 아닌 값을 지정하거나 채널 인증 레코드를 사용하거나 보안 엑시트를 사용하십시오.

MCAUSER 속성의 기본값은 로컬 사용자 ID입니다. 서버 연결 채널의 MCAUSER 속성으로 비공백 사용자 이름을 지정하는 경우, 이 채널을 사용하여 큐 관리자에 연결되는 모든 프로그램은 이름 지정된 사용자의 ID로 실행되고 동일한 권한 레벨을 가집니다. 이는 채널 인증 레코드를 사용하는 경우에는 발생하지 않습니다.

IBM MQ Explorer와 함께 보안 엑시트 사용

IBM MQ Explorer를 사용하여 기본 보안 엑시트 및 큐 관리자 특정 보안 엑시트를 지정할 수 있습니다.

IBM MQ Explorer의 모든 새 클라이언트 연결에 사용할 수 있는 기본 보안 엑시트를 정의할 수 있습니다. 연결이 작성될 때 이 기본 엑시트는 대체될 수 있습니다. 또한 단일 큐 관리자 또는 일련의 큐 관리자에 대한 보안 엑시트를 정의할 수 있으며, 연결이 작성될 때 적용됩니다. IBM MQ Explorer를 사용하여 엑시트를 지정합니다. 자세한 정보는 IBM MQ Explorer 도움말을 참조하십시오.

TLS 사용 MQI 채널을 사용하여 리모트 큐 관리자에 연결하기 위해 IBM MQ Explorer 사용




IBM MQ Explorer는 MQI 채널을 사용하여 리모트 큐 관리자에 연결됩니다. TLS 보안을 사용하여 MQI 채널 보안을 설정하려면 클라이언트 채널 정의 테이블을 사용하여 채널을 설정해야 합니다.

클라이언트 채널 정의 테이블을 사용하여 MQI 채널을 설정하는 방법에 대한 정보는 [IBM MQ MQI clients](#)의 내용을 참조하십시오.

클라이언트 채널 정의 테이블을 사용하여 채널을 설정한 경우, 116 페이지의 『리모트 큐 관리자를 호스트하는 시스템의 태스크』 및 116 페이지의 『IBM MQ Explorer를 호스트하는 시스템의 태스크』에 설명된 대로 IBM MQ Explorer를 사용하여 TLS 사용 MQI 채널을 사용하는 리모트 큐 관리자에 연결할 수 있습니다.

리모트 큐 관리자를 호스트하는 시스템의 태스크

리모트 큐 관리자를 호스트하는 시스템에서 다음 태스크를 수행하십시오.

1. 채널의 서버 연결 및 클라이언트 연결 쌍을 정의하고 두 채널의 서버 연결에서 `SSLCIPH` 속성에 적합한 값을 지정하십시오. `SSLCIPH` 속성에 대한 자세한 정보는 [TLS로 채널 보호](#)를 참조하십시오.
 2. 큐 관리자의 `@ipcc` 디렉토리에 있는 채널 정의 테이블 `AMQCLCHL.TAB`을(를) IBM MQ Explorer을(를) 호스팅하는 시스템으로 전송하십시오.
 3. 지정된 포트에서 TCP/IP 리스너를 시작하십시오.
 4. CA 및 개인 TLS 인증서를 모두 큐 관리자의 SSL 디렉토리에 배치하십시오.
 -   AIX and Linux 시스템의 경우, `/var/mqm/qmgrs/+QMNAME+/SSL`
 -  Windows 시스템의 경우, `C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL`
- 여기서, `+QMNAME+`은(는) 큐 관리자의 이름을 나타내는 토큰입니다.
5. 이름이 `key.kdb`인 CMS 유형의 키 데이터베이스 파일을 작성하십시오. 키 데이터베이스를 작성하는 데 사용되는 `runmqakm` 명령에 `-stash` 매개변수를 지정하여 파일에 키 데이터베이스 비밀번호를 숨기십시오.
 6. 이전 단계에서 작성된 키 데이터베이스에 CA 인증서를 추가하십시오.
 7. 큐 관리자에 대한 개인 인증서를 키 데이터베이스로 가져오십시오.

Windows 시스템에서 TLS에 대해 작업하는 방법에 대한 자세한 정보는 [AIX, Linux, and Windows에서 TLS에 대한 작업](#)을 참조하십시오.

IBM MQ Explorer를 호스트하는 시스템의 태스크

IBM MQ Explorer를 호스트하는 시스템에서 다음 태스크를 수행하십시오.

1. 이름이 `key.jks`인 JKS 유형의 키 데이터베이스 파일을 작성하십시오. 이 키 데이터베이스 파일에 대한 비밀번호를 설정하십시오.

IBM MQ Explorer가 TLS 보안을 위해 사용하는 키 저장소는 Java 키 저장소 (JKS) 파일이어야 합니다.
2. 이전 단계에서 작성된 키 데이터베이스에 CA 인증서를 추가하십시오.

3. 큐 관리자에 대한 개인 인증서를 키 데이터베이스로 가져오십시오.
4. Windows 및 Linux 시스템에서는 시스템 메뉴, MQExplorer 실행 파일 또는 **strmqcfcg** 명령을 사용하여 IBM MQ Explorer 를 시작하십시오.
5. IBM MQ Explorer 도구 모음에서 **창 -> 환경 설정**을 클릭한 다음 **IBM MQ 탐색기**를 펼치고 **SSL 클라이언트 인증서 저장소**를 클릭하십시오. 신뢰성있는 인증서 저장소 및 개인 인증서 저장소 모두에 116 페이지의 『IBM MQ Explorer를 호스트하는 시스템의 태스크』의 1단계에서 작성된 JKS 파일의 이름 및 비밀번호를 입력한 다음, **확인**을 클릭하십시오.
6. **환경 설정** 창을 닫고 **큐 관리자**를 마우스 오른쪽 단추로 클릭하십시오. **큐 관리자 표시/숨기기**를 클릭한 다음 **큐 관리자 표시/숨기기** 화면에서 **추가**를 클릭하십시오.
7. 큐 관리자의 이름을 입력하고 **직접 연결** 옵션을 선택하십시오. 다음을 클릭하십시오.
8. **클라이언트 채널 정의 테이블(CCDT) 사용**을 선택하고 리모트 큐 관리자를 호스트하는 시스템에서 116 페이지의 『리모트 큐 관리자를 호스트하는 시스템의 태스크』의 2단계에서 리모트 큐 관리자로부터 전송한 채널 테이블 파일의 위치를 지정하십시오.
9. **완료**를 누르십시오. IBM MQ Explorer에서 리모트 큐 관리자에 액세스할 수 있습니다.

IBM MQ Explorer로 다른 큐 관리자를 통해 연결

IBM MQ Explorer를 사용하면 IBM MQ Explorer가 이미 연결되어 있는 큐 관리자에 중간 큐 관리자를 통해 연결할 수 있습니다.

이 경우 IBM MQ Explorer는 다음을 지정하여 중간 큐 관리자에 PCF 명령 메시지를 넣습니다.

- 대상 큐 관리자의 이름으로서 오브젝트 디스크립터(MQOD)의 *ObjectQMGrName* 매개변수. 큐 이름 해석에 대한 자세한 정보는 [이름 해석](#)을 참조하십시오.
- 로컬 사용자 ID로서 메시지 디스크립터(MQMD)의 *UserIdentifier* 매개변수.

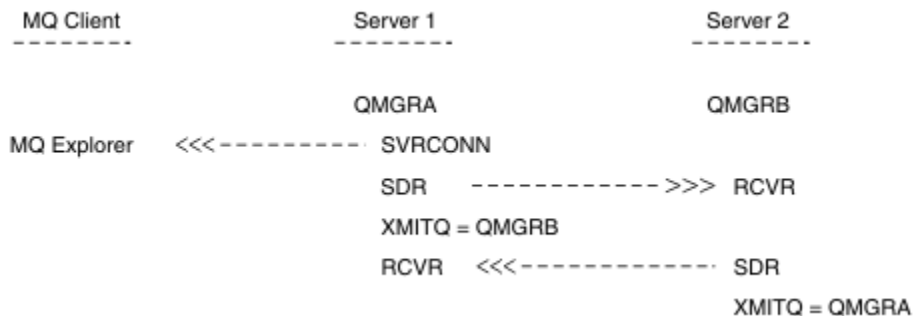
연결이 중간 큐 관리자를 통해 대상 큐 관리자에 연결하기 위해 사용되는 경우, 사용자 ID가 메시지 디스크립터(MQMD)의 *UserIdentifier* 매개변수에 다시 플로우됩니다. 대상 큐 관리자의 MCA 리스너가 이 메시지를 승인하기 위해서는 MCAUSER 속성이 설정되어야 하거나 put 권한이 있는 사용자 ID가 이미 존재해야 합니다.

대상 큐 관리자의 명령 서버는 메시지 디스크립터(MQMD)에서 *UserIdentifier* 매개변수에 사용자 ID를 지정하여 메시지를 송신 큐에 넣습니다. 이러한 넣기가 성공하기 위해서는 대상 큐 관리자에 넣기 권한이 있는 사용자 ID가 이미 존재해야 합니다.

다음 예제에서는 중간 큐 관리자를 통해 큐 관리자를 IBM MQ Explorer에 연결하는 방법을 보여줍니다.

큐 관리자에 대한 원격 관리 연결을 설정하십시오. 다음을 확인하십시오.

- 서버의 큐 관리자가 활성화고 서버 연결 채널(SVRCONN)이 정의되어 있습니다.
- 리스너가 활성화입니다.
- 명령 서버가 활성화입니다.
- SYSTEM.MQ EXPLORER.REPLY.MODEL 큐가 작성되었고 사용자에게 충분한 권한이 있습니다.
- 큐 관리자 리스너, 명령 서버 및 송신자 채널이 시작됩니다.



이 예제에서

- IBM MQ Explorer는 클라이언트 연결을 사용하여 큐 관리자 QMGRA(서버1에서 실행)에 연결됩니다.

- Server2의 큐 관리자 QMGRB는 이제 중간 큐 관리자(QMGRA)를 통해 IBM MQ Explorer에 연결될 수 있습니다.
- IBM MQ Explorer를 사용하여 QMGRB에 연결할 때 중간 큐 관리자로 QMGRA를 선택하십시오.

이 상황에서는 IBM MQ Explorer에서 QMGRB로의 직접 연결이 없습니다. QMGRB에 대한 연결은 QMGRA를 통해 이루어집니다.

서버2의 큐 관리자 QMGRB가 송신자-수신자 채널을 사용하여 서버1의 QMGRA에 연결됩니다. QMGRA와 QMGRB 사이의 채널은 원격 관리가 사용 가능한 그러한 방식으로 설정되어야 합니다. 183 페이지의 『원격 관리를 위한 큐 관리자 구성』의 내용을 참조하십시오.

IBM MQ Explorer에서 큐 관리자 및 클러스터 표시 및 숨기기

IBM MQ Explorer는 한 번에 둘 이상의 큐 관리자를 표시할 수 있습니다. 큐 관리자 표시/숨기기 패널(큐 관리자 트리 노드에 대한 메뉴에서 선택 가능)에서 다른(원격) 시스템에 대한 정보를 표시할지 여부를 선택할 수 있습니다. 로컬 큐 관리자는 자동으로 감지됩니다.

리모트 큐 관리자를 표시하려면 다음을 수행하십시오.

1. 큐 관리자 트리 노드를 마우스의 오른쪽 단추로 클릭한 후 **큐 관리자 표시/숨기기**를 선택하십시오.
2. 추가를 클릭하십시오. 큐 관리자 표시/숨기기 패널이 표시됩니다.
3. 제공된 필드에서 리모트 큐 관리자와 호스트 이름 또는 IP 주소의 이름을 입력하십시오.

호스트 이름 또는 IP 주소는 기본 서버 연결 채널(SYSTEM.ADMIN.SVRCONN) 또는 사용자 정의된 서버 연결 채널을 사용하여 리모트 큐 관리자에 클라이언트 연결을 설정하는 데 사용됩니다.

4. **완료**를 누르십시오.

큐 관리자 표시/숨기기 패널은 모든 가시적 큐 관리자의 목록을 표시합니다. 이 패널을 사용하여 탐색 보기에서 큐 관리자를 숨길 수 있습니다.

IBM MQ Explorer가 클러스터의 멤버인 큐 관리자를 표시하는 경우, 클러스터가 감지되고 자동으로 표시됩니다.

이 패널에서 리모트 큐 관리자의 목록을 내보내려면 다음을 수행하십시오.

1. 큐 관리자 표시/숨기기 패널을 닫으십시오.
2. IBM MQ Explorer의 탐색 분할창에서 가장 높은 **IBM MQ** 트리 노드를 마우스 오른쪽 단추로 클릭한 후 **내보내기 IBM MQ Explorer 설정**을 선택하십시오.
3. **IBM MQ Explorer > IBM MQ Explorer 설정**을 클릭하십시오.
4. **연결 정보 > 리모트 큐 관리자**를 선택하십시오.
5. 내보낸 설정을 저장하기 위한 파일을 선택하십시오.
6. 마지막으로, **마침**을 클릭하여 리모트 큐 관리자 연결 정보를 지정된 파일로 내보내십시오.

리모트 큐 관리자의 목록을 가져오려면 다음을 수행하십시오.

1. IBM MQ Explorer의 탐색 분할창에서 가장 높은 **IBM MQ** 트리 노드를 마우스 오른쪽 단추로 클릭한 후 **IBM MQ Explorer 설정 가져오기**를 선택하십시오.
2. **IBM MQ Explorer > IBM MQ Explorer 설정**을 클릭하십시오.
3. **찾아보기**를 클릭하고 리모트 큐 관리자 연결 정보를 포함하는 파일의 경로로 이동하십시오.
4. **열기**를 클릭하십시오. 리모트 큐 관리자의 목록이 파일에 포함되는 경우 **연결 정보 > 리모트 큐 관리자** 상태가 선택됩니다.
5. 마지막으로 **마침**을 클릭하여 리모트 큐 관리자 연결 정보를 IBM MQ Explorer로 가져오십시오.

클러스터 멤버십 및 IBM MQ Explorer

IBM MQ Explorer에는 클러스터의 멤버인 큐 관리자에 대한 정보가 필요합니다.

큐 관리자가 클러스터의 멤버인 경우, 클러스터 트리 노드가 자동적으로 채워집니다.

IBM MQ Explorer가 실행되는 동안 큐 관리자가 클러스터의 멤버가 될 경우, 효율적으로 통신할 수 있고 요청될 때 올바른 클러스터 정보를 표시할 수 있도록 클러스터에 대한 최신 관리 데이터로 IBM MQ Explorer를 유지보수해야 합니다. 이를 수행하려면 IBM MQ Explorer에 다음 정보가 필요합니다.

- 저장소 큐 관리자의 이름
- 리모트 큐 관리자에 있는 경우 저장소 큐 관리자의 연결 이름

이 정보를 사용하여 IBM MQ Explorer는 다음을 수행할 수 있습니다.

- 저장소 큐 관리자를 사용하여 클러스터에 있는 큐 관리자 목록을 확보합니다.
- 지원되는 플랫폼 및 명령 레벨에 있는 클러스터의 멤버인 큐 관리자를 관리합니다.

다음의 경우 관리가 가능하지 않습니다.

- 선택된 저장소가 사용 불가능하게 됩니다. IBM MQ Explorer는 대체 저장소로 자동으로 전환되지 않습니다.
- 선택된 저장소가 TCP/IP를 통해 접속될 수 없습니다.
- 선택된 저장소가 IBM MQ Explorer에서 지원하지 않는 플랫폼과 명령 레벨에서 실행되고 있는 큐 관리자에서 실행되고 있습니다.

관리될 수 있는 클러스터 멤버는 로컬일 수 있으며 TCP/IP를 사용하여 접속할 수 있는 경우에는 원격일 수 있습니다. IBM MQ Explorer는 클라이언트 연결을 사용하지 않고 클러스터의 멤버인 로컬 큐 관리자에 직접 연결합니다.

IBM MQ Explorer에 대한 데이터 변환

IBM MQ Explorer는 CCSID 1208(UTF-8)에서 작동합니다. 이 때문에 IBM MQ Explorer가 리모트 큐 관리자의 데이터를 올바르게 표시할 수 있습니다. 큐 관리자에 직접 연결하거나 중간 큐 관리자를 사용하는지 여부에 따라, IBM MQ Explorer는 모든 수신 메시지가 CCSID 1208(UTF-8)로 변환되도록 요구합니다.

IBM MQ Explorer가 인식하지 못하는 CCSID를 가진 큐 관리자와 IBM MQ Explorer 사이에 연결을 설정할 경우 오류 메시지가 발행됩니다.

지원되는 변환이 [코드 페이지 변환](#)에서 설명됩니다.

Windows IBM MQ 작업 표시줄 애플리케이션(Windows 전용) 사용

IBM MQ 작업 표시줄 애플리케이션은 서버의 Windows 시스템 트레이에 아이콘을 표시합니다. 아이콘은 일부 단순한 조치를 수행할 수 있는 메뉴 및 IBM MQ의 현재 상태를 제공합니다.

Windows에서 IBM MQ 아이콘은 서버의 시스템 트레이에 있고 색상 코드화 상태 기호로 오버레이됩니다. 이는 다음 의미 중 하나를 가질 수 있습니다.

초록색

올바르게 작업 중이며 현재 경보가 작동하지 않음

파랑색

불확정; IBM MQ이 시작 중이거나 시스템 종료 중임

노랑색

경보; 하나 이상의 서비스가 실패 중이거나 이미 실패했음

메뉴를 표시하려면 IBM MQ 아이콘을 마우스 오른쪽 단추로 클릭하십시오. 메뉴에서 다음 조치를 수행할 수 있습니다.

- 열기를 클릭하여 IBM MQ 경보 모니터를 여십시오.
- 종료를 클릭하여 IBM MQ 작업 표시줄 애플리케이션을 종료하십시오.
- **IBM MQ Explorer**를 클릭하여 IBM MQ Explorer를 시작하십시오.
- 중지**IBM MQ**를 클릭하여 IBM MQ를 중지하십시오.
- 정보**IBM MQ**를 클릭하여 IBM MQ 경보 모니터에 대한 정보를 표시하십시오.

Windows IBM MQ 경보 모니터 애플리케이션(Windows 전용)

IBM MQ 경보 모니터는 로컬 시스템에서 IBM MQ로 문제를 식별하고 기록하는 오류 검출 도구입니다.

경보 모니터는 IBM MQ 서버의 로컬 설치의 현재 상태에 대한 정보를 표시합니다. 이는 Windows ACPI(Advanced Configuration and Power Interface)를 모니터링하고 ACPI 설정이 실시되는지 확인합니다.

IBM MQ 경보 모니터에서 다음을 수행할 수 있습니다.

- IBM MQ Explorer에 대한 직접적인 액세스
- 모든 미해결 경로와 관련하는 정보 보기
- 로컬 시스템에서 IBM MQ 서비스 시스템 종료
- 구성 가능한 사용자 계정 또는 Windows Workstation 또는 서버로 네트워크를 통한 경보 메시지 라우팅

로컬 IBM MQ 오브젝트에 대한 작업

MQI(Message Queue Interface)를 사용하는 애플리케이션 프로그램을 지원하기 위해 로컬 IBM MQ 오브젝트를 관리할 수 있습니다.

이 태스크 정보

이 컨텍스트에서 로컬 관리는 IBM MQ 오브젝트 작성, 표시, 변경, 복사 및 삭제를 의미합니다.

이 절에 설명된 접근 방법 이외에도 IBM MQ Explorer를 사용하여 로컬 IBM MQ 오브젝트를 관리할 수 있습니다. 자세한 정보는 [112 페이지의 『IBM MQ Explorer를 사용하여 관리』](#)의 내용을 참조하십시오.

프로시저

- 다음 주제의 정보를 활용하면 로컬 IBM MQ 오브젝트를 관리하는 데 도움이 됩니다.
 - MQI를 사용하는 애플리케이션 프로그램
 - [11 페이지의 『MQSC 명령을 사용하여 IBM MQ 관리』](#)
 - [127 페이지의 『큐 관리자 속성 표시 및 대체』](#)
 - [130 페이지의 『로컬 큐에 대한 작업』](#)
 - [142 페이지의 『알리어스 큐에 대한 작업』](#)
 - [144 페이지의 『모델 큐에 대한 작업』](#)
 - [170 페이지의 『서비스에 대한 작업』](#)
 - [177 페이지의 『트리거에 대한 오브젝트 관리』](#)

큐 관리자에 대한 작업

제어 명령을 사용하여 큐 관리자를 시작하고 중지할 수 있습니다. MQSC 명령을 사용하여 큐 관리자 속성을 표시하거나 대체할 수 있습니다.

관련 태스크

[멀티플랫폼에서 큐 관리자 작성](#)

Multi 큐 관리자 시작

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

이 태스크 정보

`strmqm` 명령을 사용하여 큐 관리자를 시작할 수 있습니다. `strmqm` 명령 및 해당 옵션에 대한 설명은 [strmqm](#)을 참조하십시오.

Windows → **Linux** 또는 Windows 및 Linux(x86 및 x86-64 플랫폼) 시스템의 경우 IBM MQ Explorer를 사용하여 큐 관리자를 시작할 수 있습니다.

Windows Windows의 경우 IBM MQ Explorer를 사용하여 시스템이 시작될 때 자동으로 큐 관리자를 시작할 수 있습니다. 자세한 정보는 [112 페이지의 『IBM MQ Explorer를 사용하여 관리』](#)의 내용을 참조하십시오.


프로시저

- **strmqm** 명령을 사용하여 큐 관리자를 시작하려면 시작하려는 큐 관리자의 이름을 이 명령 뒤에 함께 입력하십시오.
예를 들어, QMB라는 큐 관리자를 시작하려면 다음 명령을 입력하십시오.

```
strmqm QMB
```

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

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

-  IBM MQ Explorer를 사용하여 큐 관리자를 시작하려면 다음의 단계를 완료하십시오.
 - a) IBM MQ Explorer을(를) 여십시오.
 - b) 네비게이터 보기에서 큐 관리자를 선택하십시오.
 - c) 시작을 클릭하십시오.

결과

큐 관리자가 시작됩니다.

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

Multi 큐 관리자 중지

endmqm 명령을 사용하여 큐 관리자를 중지할 수 있습니다. 이 명령은 네 가지 방법(제어된 종료 또는 일시정지된 종료, 종료, 즉시 종료, 강제 종료(preemptive shutdown) 및 대기 종료)을 제공합니다. 또는 Windows 및 Linux에서 IBM MQ Explorer를 사용하여 큐 관리자를 중지할 수 있습니다.

이 태스크 정보

endmqm 명령으로 단일 인스턴스 큐 관리자를 중지하는 방법에는 다음 네 가지가 있습니다.

제어된(또는 정상) 종료

기본적으로 **endmqm** 명령은 지정된 큐 관리자의 정상 종료(Quiesced shutdown)를 수행합니다. 정상 종료는 연결된 모든 애플리케이션이 끊어질 때까지 대기합니다. 따라서 완료하는 데 다소 시간이 걸릴 수 있습니다.

즉시 종료

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

강제 종료(preemptive shutdown)

큐 관리자가 즉시 중지됩니다. 큐 관리자가 정상적인 **endmqm** 명령으로 중지되지 않는 경우와 같이 예외적인 경우에만 이 유형의 종료를 사용하십시오.

대기 종료

이 유형의 종료는 큐 관리자가 중지된 후에만 사용자에게 제어가 리턴되는 점을 제외하고 제어된 종료와 같습니다.

endmqm 명령을 실행하면 다중 인스턴스 큐 관리자의 모든 인스턴스가 단일 인스턴스 큐 관리자가 중지될 때와 동일한 방식으로 중지됩니다. **endmqm** 명령은 다중 인스턴스 큐 관리자의 대기 인스턴스 중 하나 또는 활성 인스턴스에서 실행할 수 있습니다. 그러나 큐 관리자를 종료하려면 활성 인스턴스에서 **endmqm**을 실행해야 합니다.

필수가 아닌 큐 관리자 유지보수 태스크를 인터럽트하거나 인터럽트하지 않고 지정한 시간(초)의 대상 시간 내에 큐 관리자를 종료하는 옵션이 있습니다. [123 페이지의 『대상 시간 내에 큐 관리자 종료』](#)의 내용을 참조하십시오.



주의:

- 지속 메시지는 사용된 종료 유형 (수동으로 종료하는 IBM MQ 프로세스 포함) 에 관계없이 지속되는 반면, 비지속 메시지는 모든 유형의 종료에서 생존하도록 보장할 수 없습니다.

큐 특성 NPMCLASS (HIGH) 를 지정하면 최상의 수행을 기반으로 비지속 메시지가 저장됩니다. **endmqm -t, endmqm -tp, endmqm -p**를 사용하거나 IBM MQ 프로세스를 수동으로 종료하면 **endmqm -w** 또는 **endmqm -i** 와 비교하여 IBM MQ 종료 또는 재시작 주기를 유지하는 NPMCLASS (HIGH) 메시지의 가능성이 줄어듭니다.

- 큐 관리자를 종료하고 다시 시작하는 데 결합된 시간은 특히 **-p** 및 **-tp** 옵션을 사용하는 경우 더 갑작스런 시스템 종료 메소드를 사용한 결과로 인해 더 길어질 수 있습니다.

큐 관리자가 큐 관리자를 종료하기 위해 IBM MQ 프로세스 종료에 의존해야 하는 경우 큐 관리자가 재시작될 때 큐 관리자 상태의 추가 조정이 필요할 수 있습니다.

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

팁: 큐 관리자를 종료하는 중에 발생하는 문제점은 대개 애플리케이션 때문입니다. 예를 들어, 애플리케이션이 다음과 같은 경우,

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

큐 관리자를 중지하려고 할 때 문제점이 발생하면 Ctrl-C를 사용하여 **endmqm** 명령을 중단할 수 있습니다. 그런 다음 다른 **endmqm** 명령을 실행할 수 있지만, 이번에는 필요한 종료를 유형의 지정하는 매개변수를 사용하여 실행할 수 있습니다.

Windows **Linux** **endmqm** 명령을 사용하는 대신, Windows 및 Linux에서 IBM MQ Explorer를 통해 제어된 종료 또는 즉시 종료를 수행하여 큐 관리자를 중지할 수 있습니다.

프로시저

- **endmqm** 명령을 사용하여 큐 관리자를 중지하려면 적절한 매개변수(필요한 경우)와 시작하려는 큐 관리자의 이름을 이 명령 뒤에 함께 입력하십시오.

참고: 작업 중인 큐 관리자와 연관된 설치에서 **endmqm** 명령을 사용해야 합니다. 큐 관리자와 연관된 설치를 찾으려면 **dspmq** 명령을 사용하십시오.

```
dspmq -o installation
```

- 제어된(정상) 종료를 수행하려면 다음 예에 표시된 것과 같이, **endmqm** 명령을 입력하십시오. 그러면 QMB 라는 큐 관리자가 중지됩니다.

```
endmqm QMB
```

또는 다음 예에 표시된 것과 같이, **endmqm** 명령을 **-c** 매개변수와 함께 입력하는 것은 **endmqm QMB** 명령과 같습니다.

```
endmqm -c QMB
```

두 경우 모두, 제어 권한이 즉시 사용자에게 리턴되므로, 큐 관리자가 중지될 때 이에 대한 알림이 사용자에게 통보되지 않습니다. 제어를 리턴하기 전에 모든 애플리케이션이 중지되고 큐 관리자가 종료될 때까지 명령이 대기하도록 하려면 다음 예에 표시된 대로 **-w** 매개변수를 대신 사용하십시오.

```
endmqm -w QMB
```

- 즉시 종료를 수행하려면 다음 예에 표시된 것과 같이 **endmqm** 명령을 **-i** 매개변수와 함께 입력하십시오.

```
endmqm -i QMB
```

- 강제 종료(preemptive shutdown)를 수행하려면 다음 예에 표시된 것과 같이 **endmqm** 명령을 **-p** 매개변수와 함께 입력하십시오.

```
endmqm -p QMB
```



주의: 강제 종료를 수행할 경우 연결된 애플리케이션에서 예측할 수 없는 결과가 초래될 수 있습니다. 정상적인 **endmqm** 명령을 사용하여 큐 관리자를 중지하려는 다른 모든 시도가 실패할 때까지는 이 옵션을 사용하지 마십시오. **ALW** 강제 종료(preemptive shutdown)가 작동하지 않는 경우 대신 124 페이지의 『수동으로 큐 관리자 중지』을 시도하십시오.

- 자동 클라이언트 다시 연결을 요청하려면 **endmqm** 명령을 **-r** 매개변수와 함께 입력하십시오. 이 매개변수는 클라이언트를 큐 관리자 그룹의 다른 큐 관리자에 다시 연결하는 데 영향을 미칩니다.

참고: 기본 **endmqm** 명령을 사용하여 큐 관리자를 종료하면 자동 클라이언트 다시 연결을 트리거하지 않습니다.

- 활성 인스턴스를 종료한 후 다중 인스턴스 큐 관리자의 대기 인스턴스로 전송하려면 다중 인스턴스 큐 관리자의 활성 인스턴스에서 **endmqm** 명령을 **-s** 매개변수와 함께 입력하십시오.
- 다중 인스턴스 큐 관리자의 대기 인스턴스는 종료하고 활성 인스턴스는 실행 중인 상태로 유지하려면 다중 인스턴스 큐 관리자의 대기 인스턴스에서 **endmqm** 명령을 **-x** 매개변수와 함께 입력하십시오.

Windows Linux

Windows 및 Linux에서 IBM MQ Explorer를 사용하여 큐 관리자를 중지하려면 다음 단계를 완료하십시오.

- IBM MQ Explorer을(를) 여십시오.
- 네비게이터 보기에서 큐 관리자를 선택하십시오.
- 중지를 누르십시오.
큐 관리자 종료 패널이 표시됩니다.
- 제어됨 또는 즉시를 선택하십시오.
- 확인을 클릭하십시오.
큐 관리자가 중지됩니다.

관련 태스크

[유지보수 레벨 업데이트를 AIX의 다중 인스턴스 큐 관리자에 적용](#)

[유지보수 레벨 업데이트를 Linux의 다중 인스턴스 큐 관리자에 적용](#)

[유지보수 레벨 업데이트를 Windows의 다중 인스턴스 큐 관리자에 적용](#)

관련 참조

[endmqm\(큐 관리자 종료\)](#)

대상 시간 내에 큐 관리자 종료

필수가 아닌 큐 관리자 유지보수 태스크를 인터럽트하거나 인터럽트하지 않고 지정한 시간 (초) 내에 큐 관리자를 종료할 수 있습니다.

endmqm 명령을 사용할 때 대상 시간을 지정하는 두 가지 방법이 있습니다. **-t** 옵션을 사용하면 모든 큐 관리자 유지보수 태스크를 완료할 수 있으며, 이는 큐 관리자 종료 단계를 연장할 수 있습니다. **-tp** 옵션은 지정된 대상 시간을 준수해야 하는 경우 필수가 아닌 큐 관리자 유지보수 태스크를 인터럽트합니다.

필수가 아닌 유지보수 태스크에는 큐 파일 압축 및 지속적인 NPMCLASS (HIGH) 메시지가 포함됩니다. 이 페이지의 나머지 부분에서는 'housekeeping'이라는 단어가 사용됩니다.

애플리케이션 사용 패턴에 따라 큐 파일 압축에 시간이 오래 걸릴 수 있으므로 기본 목표가 큐 관리자를 빠르게 종료하는 것인 경우 **-tp** 옵션을 사용하십시오.

대상 시간을 지정할 때 **-w**, **-i** 또는 **-p**의 종료 유형은 시작 종료 유형을 표시합니다.

참고: `immediate` 종료는 임의의 실행 중인 애플리케이션이 일시정지되는 점에서 `controlled` 종료와는 기본적으로 다르지만 순서대로 진행됩니다. `immediate` 시스템 종료는 여전히 하우스키핑을 수행합니다. 시간 제한 종료는 대상 시간을 충족하는 데 방해가 될 때 이러한 조치를 종료합니다.

큐 관리자는 대상 시간을 충족시키기 위해 필요에 따라 종료를 에스컬레이션합니다. 예를 들면, 다음과 같습니다.

- `-w`부터 시작되는 10초 `-t` 대상은 7초 정지 중, 하우스키핑을 포함한 큐 관리자의 2초 즉시 종료, 추가적인 하우스키핑이 없는 즉시 종료일 수 있습니다.

```
endmqm -w -t 10 queue_manager
```

- 10초 `-tp` 대상은 7초 정지 중, 하우스키핑을 포함한 큐 관리자의 2초 즉시 종료, 추가적인 하우스키핑이 없는 1초 즉시 종료 후 IBM MQ 프로세스 종료 시작일 수 있습니다.

```
endmqm -c -tp 10 queue_manager
```

- `-i` 에서 두 번째 2초 `-tp` 대상은 하우스키핑을 포함한 1초 즉시 종료, 추가적인 하우스키핑이 없는 1초 즉시 종료 후 IBM MQ 프로세스 종료 시작일 수 있습니다.

```
endmqm -i -tp 2 queue_manager
```

- `-w`에서의 1초 대상은 `wait`에서의 0.1초(예를 들어, IBM MQ 리턴 코드를 연결된 애플리케이션에 전송하기에 충분한 시간), 하우스키핑을 포함한 큐 관리자의 0.9초 즉시 종료, 추가적인 하우스키핑 없이 즉시 종료 후 IBM MQ 프로세스 종료 시작일 수 있습니다.

관련 참조

[endmqm\(큐 관리자 종료\)](#)

ALW 수동으로 큐 관리자 중지

큐 관리자를 중지하고 제거하기 위한 표준 방법이 실패할 경우 큐 관리자를 수동으로 중지할 수 있습니다.

이 태스크 정보

큐 관리자를 중지하는 표준 방식은 121 페이지의 『큐 관리자 중지』에 설명된 대로 `endmqm` 명령을 사용하는 것입니다. 큐 관리자를 표준 방법으로 중지할 수 없는 경우에는 큐 관리자를 수동으로 중지할 수 있습니다. 이 작업을 수행하는 방법은 사용하는 플랫폼에 따라 다릅니다.

프로시저

- **Windows**
Windows에서 큐 관리자를 중지하려면 124 페이지의 『Windows에서 수동으로 큐 관리자 중지』의 내용을 참조하십시오.
- **Linux** **AIX**
AIX 또는 Linux에서 큐 관리자를 중지하려면 125 페이지의 『AIX and Linux에서 수동으로 큐 관리자 중지』의 내용을 참조하십시오.

관련 태스크

[Multiplatforms에서 큐 관리자 작성 및 관리](#)

관련 참조

[endmqm](#)

Windows Windows에서 수동으로 큐 관리자 중지

Windows에서 `endmqm` 명령을 사용하여 큐 관리자를 중지할 수 없는 경우에는 IBM MQ 서비스를 실행 및 중지하는 프로세스를 종료하여 큐 관리자를 수동으로 중지할 수 있습니다.

이 태스크 정보

팁: Windows 태스크 관리자 및 **tasklist** 명령은 태스크에 관한 제한된 정보를 제공합니다. 특정 큐 관리자와 관련된 프로세스를 판별하는 데 도움이 되는 자세한 정보를 얻으려면 프로세스 탐색기(procexp.exe)와 같은 도구를 사용해 보십시오. 이 도구는 Microsoft 웹 사이트(<http://www.microsoft.com>)에서 다운로드할 수 있습니다.

Windows에서 큐 관리자를 중지하려면 다음 단계를 완료하십시오.

프로시저

1. Windows 태스크 관리자를 사용하여 실행 중인 프로세스의 이름(ID)을 나열하십시오.
2. 실행 중인 경우, 다음 순서로 Windows 태스크 관리자 또는 **taskkill** 명령을 사용하여 프로세스를 종료하십시오.

표 7. 중지할 Windows 프로세스(실행 중인 경우)	
프로세스 이름	설명
AMQZMUC0	중요한 프로세스 관리자
AMQZXMA0	실행 제어기
AMQZFUMA	OAM 프로세스
AMQZLAA0	LQM 에이전트
AMQZLSA0	LQM 에이전트
AMQZMUFO	유틸리티 관리자
AMQZMGRO	프로세스 제어기
AMQZMUR0	재시작 가능한 프로세스 관리자
AMQFQPUB	발행/구독 프로세스
AMQFCXBA	브로커 작업 프로그램 프로세스
AMQRMPPA	프로세스 폴링 프로세스
AMQCRSTA	스레드되지 않은 응답자 작업 프로세스
AMQCRS6B	LU62 수신자 채널 및 클라이언트 연결
AMQRRMFA	저장소 프로세스(클러스터의 경우)
AMQPCSEA	명령 서버
RUNMQTRM	서버의 트리거 모니터 호출
RUNMQDLQ	데드-레터 큐 핸들러 호출
RUNMQCHI	채널 시작기 프로세스
RUNMQLSR	채널 리스너 프로세스
AMQXSSVN	공유 메모리 서버

3. Windows 제어판의 **관리 도구 > 서비스**에서 IBM MQ 서비스를 중지하십시오.
4. 모든 방법을 시도했지만 큐 관리자가 중지되지 않으면 시스템을 다시 시작하십시오.

Linux → AIX **AIX and Linux에서 수동으로 큐 관리자 중지**

endmqm 명령을 사용하여 AIX 또는 Linux 에서 큐 관리자를 중지할 수 없는 경우 실행 중인 프로세스를 종료하고 IBM MQ 서비스를 중지하여 큐 관리자를 수동으로 중지할 수 있습니다.

이 태스크 정보

AIX 또는 Linux에서 큐 관리자를 중지하려면 다음 단계를 완료하십시오.

큐 관리자를 수동으로 중지하는 경우, FFST가 사용되고 FDC 파일이 /var/mqm/errors에 있을 수 있습니다. 이는 큐 관리자에서 결함으로 간주되지 않아야 합니다.

큐 관리자는 수동 중지 방법으로 중지된 이후에도 정상적으로 재시작됩니다.

프로시저

1. **ps** 명령을 사용하여 여전히 실행 중인 큐 관리자 프로그램의 프로세스 ID (PID) 를 찾으십시오.

예를 들어, 큐 관리자가 QMNAME인 경우, 다음 명령을 사용하십시오.

```
ps -ef | grep QMNAME
```

2. **ps** 명령을 사용하여 발견된 PID를 지정하고 **kill** 명령을 사용하여 여전히 실행 중인 큐 관리자 프로세스를 종료하십시오.

프로세스를 종료하려면 **kill -KILL <pid>** 또는 동등한 **kill -9 <pid>** 명령을 사용하십시오.

매번 해당 명령을 실행하여 강제 종료하려는 PID를 하나씩 작업해야 합니다.

중요사항: 9(SIGKILL) 이외의 신호를 사용하는 경우 프로세스가 중지되지 않으며 예측할 수 없는 결과가 발생할 수 있습니다.

다음 순서로 프로세스를 종료하십시오.

표 8. 중지할 AIX and Linux 프로세스(실행 중인 경우)	
프로세스 이름	설명
amqzmuc0	중요한 프로세스 관리자
amqzxma0	실행 제어기
amqzfuma	OAM 프로세스
amqzlaa0	LQM 에이전트
amqzlsa0	LQM 에이전트
amqzmuf0	유틸리티 관리자
amqzmur0	재시작 가능한 프로세스 관리자
amqzmgr0	프로세스 제어기
amqfqpub	발행/구독 프로세스
amqfcxba	브로커 작업 프로그램 프로세스
amqrmppa	프로세스 폴링 프로세스
amqcrsta	스레드되지 않은 응답자 작업 프로세스
amqcrs6b	LU62 수신자 채널 및 클라이언트 연결
amqrrmfa	저장소 프로세스(클러스터의 경우)
amqpcsea	명령 서버
runmqtrm	서버의 트리거 모니터 호출
runmqdlq	데드-레터 큐 핸들러 호출
runmqchi	채널 시작기 프로세스
runmqlsr	채널 리스너 프로세스

관련 태스크

121 페이지의 『큐 관리자 중지』

endmqm 명령을 사용하여 큐 관리자를 중지할 수 있습니다. 이 명령은 네 가지 방법(제어된 종료 또는 일시정지된 종료, 종료, 즉시 종료, 강제 종료(preemptive shutdown) 및 대기 종료)을 제공합니다. 또는 Windows 및 Linux 에서 IBM MQ Explorer를 사용하여 큐 관리자를 중지할 수 있습니다.

Multi 큐 관리자 재시작

strmqm 명령을 사용하여 큐 관리자를 재시작할 수 있으며 Windows 및 Linux x86-64 시스템의 경우 IBM MQ Explorer에서 큐 관리자를 재시작할 수 있습니다.

이 태스크 정보

strmqm 명령을 사용하여 큐 관리자를 재시작할 수 있습니다. **strmqm** 명령 및 해당 옵션에 대한 설명은 [strmqm](#) 을 참조하십시오.

Windows **Linux** Windows 및 Linux x86-64 시스템의 경우 큐 관리자를 시작할 때와 동일한 방식으로 IBM MQ Explorer를 사용하여 큐 관리자를 재시작할 수 있습니다.

프로시저

- **strmqm** 명령을 사용하여 큐 관리자를 재시작하려면 재시작하려는 큐 관리자의 이름을 이 명령 뒤에 함께 입력하십시오.

예를 들어, **strmqm saturn.queue.manager**라는 큐 관리자를 시작하려면 다음 명령을 입력하십시오.

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

- **Windows** **Linux** IBM MQ Explorer를 사용하여 큐 관리자를 시작하려면 다음의 단계를 완료하십시오.
 - a) IBM MQ Explorer을(를) 여십시오.
 - b) 네비게이터 보기에서 큐 관리자를 선택하십시오.
 - c) 시작을 클릭하십시오.

결과

큐 관리자가 재시작됩니다.

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

큐 관리자 속성 표시 및 대체

DISPLAY QMGR MQSC 명령을 사용하면 큐 관리자의 큐 관리자 매개변수를 표시할 수 있습니다. **ALTER QMGR** MQSC 명령을 사용하여 로컬 큐 관리자에 대한 큐 관리자 매개변수를 대체할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#) .

프로시저

- **runmqsc** 명령에 지정된 큐 관리자의 속성을 표시하려면 **DISPLAY QMGR** MQSC 명령을 사용하십시오.

```
DISPLAY QMGR
```

다음 예에서는 이 명령의 일반 출력을 보여줍니다.

```
DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408I: Display Queue Manager details.
QMNAME(QM1)
ACCTINT(1800)
ACCTQ(OFF)
ACTVCONO(DISABLED)
ADVCAP(DISABLED)
ALTTIME(14.24.34)
AUTHOREV(DISABLED)
CERTLABL(ibmwebsphereqm1)
CHAD(DISABLED)
CHAEXIT( )
CHLAUTH(ENABLED)
CLWLEXIT( )
CLWLMRUC(999999999)
CMDEV(DISABLED)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CRDATE(2020-12-22)
CUSTOM( )
DEFCLXQ(SCTQ)
DESCR( )
IMGINTVL(60)
IMGRCOVO(YES)
IMGSCHEM(MANUAL)
INITKEY( )
LOCALEV(DISABLED)
MARKINT(5000)
MAXMSGL(4194304)
MAXPRTY(9)
MONACLS(QMGR)
MONQ(OFF)
PERFMEV(DISABLED)
PSMODE(ENABLED)
PSNPMSG(DISCARD)
PSRTCNT(5)
QMID(QM1_2020-12-22_15.42.49)
REPOS( )
REVDNS(ENABLED)
SCHINIT(QMGR)
SPLCAP(DISABLED)
SSLCRYP( )
SSLFIPS(NO)
SSLKEYR(C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key)
SSLRKEYC(32767)
STATCHL(OFF)
STATMQI(OFF)
STRSTPEV(ENABLED)
SYNCPT
TRIGINT(999999999)
XRCAP(NO)
ACCTCONO(DISABLED)
ACCTMQI(OFF)
ACTIVREC(MSG)
ACTVTRC(OFF)
ALTDATE(2022-05-05)
AMQPCAP(NO)
CCSID(437)
CERTVPOL(ANY)
CHADEV(DISABLED)
CHLEV(DISABLED)
CLWLDATA( )
CLWLLEN(100)
CLWLUSEQ(LOCAL)
CMDLEVEL(930)
CONFIGEV(DISABLED)
CRTIME(15.42.49)
DEADQ( )
DEFXMITQ( )
DISTL(YES)
IMGLOGLN(OFF)
IMGRCOVQ(YES)
INHIBTEV(DISABLED)
IPADDRV(IPV4)
LOGGEREV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWS10)
PSCLUS(ENABLED)
PSNPRES(NORMAL)
PSSYNCP(IFPER)
REMOOTEV(DISABLED)
REPOSNL( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
KEYRPWD( )
STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SUITEB(NONE)
TREELIFE(1800)
VERSION(09030000)
```

참고: SYNCPT는 읽기 전용 큐 관리자 속성입니다.

ALL 매개변수는 **DISPLAY QMGR** 명령에 대한 기본값입니다. 이는 모든 큐 관리자 속성을 표시합니다. 특히, 출력은 기본 큐 관리자 이름, 데드-레터 큐 이름 및 명령 큐 이름을 알려줍니다.

이러한 큐가 명령을 입력하여 존재하는지 확인할 수 있습니다.

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

스텝 **SYSTEM.***과 일치하는 큐의 목록이 표시됩니다. 괄호가 필요합니다.

- **runmqsc** 명령에서 지정된 큐 관리자의 속성을 대체하려면 변경하려는 속성 및 값을 지정하는 **MQSC** 명령 **ALTER QMGR**를 사용하십시오.

예를 들어, 다음 명령을 사용하여 **jupiter.queue.manager**의 속성을 대체하십시오.

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

ALTER QMGR 명령은 사용된 데드-레터 큐를 변경하고 이벤트가 억제되도록 합니다.

ALTER QMGR 명령에 지정되지 않은 매개변수의 기존 값은 변경되지 않은 상태로 남아 있게 됩니다.

관련 태스크

[멀티플랫폼에서 큐 관리자 작성](#)

관련 참조

[큐 관리자의 속성](#)

[runmqsc\(MQSC 명령 실행\)](#)

[DISPLAY QMGR](#)

[ALTER QMGR](#)

Multi 큐 관리자 삭제

dltmqm 제어 명령을 사용하여 큐 관리자를 삭제할 수 있습니다. 또는 Windows 및 Linux 시스템의 경우 IBM MQ Explorer를 사용하여 큐 관리자를 삭제할 수 있습니다.

시작하기 전에



주의:

- 큐 관리자 삭제는 극단적인 조치입니다. 모든 큐와 해당 메시지 및 모든 오브젝트 정의를 포함하여 큐 관리자와 연관된 모든 자원이 삭제되기 때문입니다. **dltmqm** 제어 명령을 사용하는 경우, 마음을 바꿀 수 있는 프롬프트가 표시되지 않습니다. Enter키를 누르면 연관된 모든 자원이 유실됩니다.
- **Windows** Windows의 경우 큐 관리자를 삭제하면 큐 관리자가 자동 시작 목록에서도 제거됩니다 (120 페이지의 『큐 관리자 시작』에 설명되어 있음). 명령이 완료되면 IBM MQ queue manager ending 메시지가 표시됩니다. 큐 관리자가 삭제되었다는 메시지는 표시되지 않습니다.
- 클러스터 큐 관리자를 삭제해도 클러스터에서는 제거되지 않습니다. 자세한 정보는 **dltmqm**에 있는 사용 참고사항을 참조하십시오.

이 태스크 정보

dltmqm 제어 명령을 사용하여 큐 관리자를 삭제할 수 있습니다. **dltmqm** 명령과 해당 옵션에 대한 설명은 **dltmqm**을 참조하십시오. 이 명령은 신뢰할 수 있는 관리자만 사용할 수 있습니다. (보안에 대한 정보는 **AIX, Linux, and Windows**에서 보안 설정을 참조하십시오.)

Windows **Linux** 또는 Windows 및 Linux(x86 및 x86-64 플랫폼) 시스템의 경우 IBM MQ Explorer를 사용하여 큐 관리자를 삭제할 수 있습니다.

프로시저

- **dltmqm** 명령을 사용하여 큐 관리자를 삭제하려면 다음의 단계를 완료하십시오.
 - a) 큐 관리자를 중지합니다.
 - b) 다음 명령을 실행하십시오.

```
dltmqm QMB
```

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

- **Windows** **Linux**
 - a) IBM MQ Explorer을(를) 여십시오.
 - b) 네비게이터 보기에서 큐 관리자를 선택하십시오.
 - c) 큐 관리자가 중지되지 않은 경우 중지하십시오.

- 큐 관리자를 중지하려면 큐 관리자를 마우스 오른쪽 단추로 클릭한 후 **중지**를 클릭하십시오.
- d) 큐 관리자를 삭제합니다.
- 큐 관리자를 삭제하려면 큐 관리자를 마우스 오른쪽 단추로 클릭한 후 **삭제**를 클릭하십시오.

결과

큐 관리자가 삭제됩니다.

MQI 채널 중지

서버 연결 채널에 대한 STOP CHANNEL 명령을 실행할 때, 클라이언트 연결 채널을 중지하기 위해 사용할 메소드를 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.

대화를 공유하는 것이 사용으로 설정되면, IBM MQ MQI client는 적절한 시간에 중지 요청을 알게 됩니다. 이번은 네트워크의 속도에 의존합니다. 클라이언트 애플리케이션은 IBM MQ에 대한 후속 호출을 발행한 결과로서 중지 요청을 인식하게 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 단단계 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 스테이지에서 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.

관련 개념

[채널](#)

로컬 큐에 대한 작업

이 절에는 로컬, 모델 및 알리어스 큐를 관리하기 위해 사용할 수 있는 MQSC 명령 예가 포함됩니다.

이러한 명령에 대한 자세한 정보는 [MQSC 명령](#)을 참조하십시오.

관련 참조

[큐에 대한 이름 지정 제한](#)

[기타 오브젝트에 대한 이름 지정 제한](#)

DEFINE QLOCAL을 사용하여 로컬 큐 정의

애플리케이션의 경우, 로컬 큐 관리자는 애플리케이션이 연결되는 큐 관리자입니다. 로컬 큐 관리자에 의해 관리되는 큐는 해당 큐 관리자에 대해 로컬이라고 합니다. MQSC 명령 **DEFINE QLOCAL**을 사용하여 로컬 큐를 작성합니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

MQSC 명령 **DEFINE QLOCAL**을 사용하여 로컬 큐를 작성합니다. 또한 기본 로컬 큐 정의에 정의된 기본값을 사용하거나 기본 로컬 큐 정의에서 큐 특성을 수정할 수 있습니다.

참고: 기본 로컬 큐의 이름은 SYSTEM.DEFAULT.LOCAL.QUEUE로 지정되고 시스템 설치 시 작성됩니다.

프로시저

- 로컬 큐를 작성하려면 다음 예에 표시된 대로 **DEFINE QLOCAL** 명령을 입력하십시오.
이 예에서 **DEFINE QLOCAL** 명령은 다음 특성을 가진 ORANGE.LOCAL.QUEUE라는 큐를 정의합니다.
 - 가져오기(GET)에 사용되고, 넣기(PUT)에 사용되고 우선순위 순서 기본에서 실행됩니다.
 - 이는 정상 큐입니다. 이는 이니시에이션 큐 또는 전송 큐가 아니고 트리거 메시지를 생성하지 않습니다.
 - 최대 큐 용량은 5000개의 메시지입니다. 최대 메시지 길이는 4194304 바이트입니다.

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT (ENABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (PRIORITY) +
  MAXDEPTH (5000) +
  MAXMSGL (4194304) +
  USAGE (NORMAL)
```

참고:

1. 설명의 값을 제외하고는, 예제에 표시된 모든 속성 값은 기본값입니다. 이러한 예는 설명 용도로 포함된 것입니다. 기본값이 원하는 것이거나 변경되지 않는 것이라고 확인하는 경우 생략할 수 있습니다. [131 페이지의 『DISPLAY QUEUE를 사용하여 기본 오브젝트 속성 표시』](#)도 참조하십시오.
2. **USAGE (NORMAL)**는 이 큐가 전송 큐가 아님을 표시합니다.
3. 동일한 큐 관리자에 ORANGE.LOCAL.QUEUE라는 이름의 로컬 큐가 이미 있는 경우에는 이 명령이 실패합니다. 큐의 기존 정의를 덮어쓰려는 경우 **REPLACE** 속성을 사용하지만, [133 페이지의 『ALTER QLOCAL 또는 DEFINE QLOCAL을 사용하여 로컬 큐 속성 변경』](#)의 내용도 참조하십시오.

관련 참조

[DEFINE QLOCAL](#)

DISPLAY QUEUE를 사용하여 기본 오브젝트 속성 표시

DISPLAY QUEUE MQSC 명령을 사용하여 IBM MQ 오브젝트가 정의될 때 기본 오브젝트에서 가져온 속성을 표시할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

IBM MQ 오브젝트를 정의할 때, 기본 오브젝트에서 지정하지 않는 속성을 사용합니다. 예를 들어, 로컬 큐를 정의할 때, 큐는 정의에서 생략하는 속성을 기본 로컬 큐에서 상속하고, 이를 SYSTEM.DEFAULT.LOCAL.QUEUE라고 합니다. **DISPLAY QUEUE** 명령을 사용하여 이러한 속성이 정확히 어떤 속성인지 확인할 수 있습니다.

프로시저

- 로컬 큐의 기본 오브젝트 속성을 표시하려면 다음 명령을 사용하십시오.

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

DISPLAY 명령의 구문은 이에 상응하는 **DEFINE** 명령의 구문과 다릅니다. **DISPLAY** 명령에는 큐 이름만 지정할 수 있지만, **DEFINE** 명령에는 큐 유형(QLOCAL, QALIAS, QMODEL, QREMOTE)을 지정해야 합니다.

개별적으로 지정하여 선택적으로 속성을 표시할 수 있습니다. 예를 들면, 다음과 같습니다.

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

이 명령은 다음과 같이 세 개의 지정된 속성을 표시합니다.

```
AMQ8409: Display Queue details.  
QUEUE(ORANGE.LOCAL.QUEUE) TYPE(QLOCAL)  
CURDEPTH(0) MAXDEPTH(5000)  
MAXMSGL(4194304)
```

CURDEPTH는 현재 큐 깊이 즉, 큐에 있는 메시지의 수입니다. 이는 큐 용량을 모니터링하여 큐가 채워지지 않는다는 것을 확인할 수 있으므로 표시할 유용한 속성입니다.

관련 참조

[DISPLAY QUEUE](#)

[DEFINE QUEUES](#)

DEFINE QLOCAL을 사용하여 로컬 큐 정의 복사

DEFINE QLOCAL MQSC 명령에서 **LIKE** 속성을 사용하여 큐 정의를 복사할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

이 태스크 정보

DEFINE 명령을 **LIKE** 속성과 함께 사용하여 시스템 기본 로컬 큐의 속성이 아닌 지정된 큐와 같은 속성을 사용하여 큐를 작성할 수 있습니다. 이 양식의 **DEFINE** 명령을 사용하여 큐 정의를 복사할 수도 있지만, 원래 속성에 대한 하나 이상의 변경사항이 대체될 수 있습니다.

참고:

- DEFINE** 명령에 **LIKE** 속성을 사용하면, 큐 속성만 복사됩니다. 큐에서 메시지를 복사하지 않습니다.
- LIKE**를 지정하지 않고 로컬 큐를 정의하는 경우 이는 다음과 같습니다.

```
DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)
```

프로시저

- 시스템 기본 로컬 큐의 속성이 아닌 지정된 큐와 같은 속성을 사용하여 큐를 작성하려면 다음 예에 표시된 대로 **DEFINE** 명령을 입력하십시오.

큐를 작성할 때 입력된 것과 동일하게 복사되도록 큐 이름을 입력하십시오. 이름이 소문자를 포함하면, 이름을 작은따옴표로 묶으십시오.

이 예는 시스템 기본 로컬 큐의 속성이 아닌 ORANGE.LOCAL.QUEUE 큐와 같은 속성을 사용하여 큐를 작성합니다.

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

- 큐 정의는 복하고 원본의 속성에 대해 하나 이상의 변경사항을 대체하려는 경우 다음 예에 표시된 대로 **DEFINE** 명령을 입력하십시오.

이 명령은 ORANGE.LOCAL.QUEUE 큐의 속성을 THIRD.QUEUE 큐에 복사하지만 새 큐의 최대 메시지 길이는 4194304가 아닌 1024바이트가 되도록 지정합니다.

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

관련 참조

[DEFINE QUEUES](#)

ALTER QLOCAL 또는 DEFINE QLOCAL을 사용하여 로컬 큐 속성 변경

ALTER QLOCAL 또는 **DEFINE QLOCAL** MQSC 명령을 **REPLACE** 속성과 함께 사용하여 두 가지 방법으로 큐 속성을 변경할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

이 태스크 정보

ALTER 및 **DEFINE** 명령의 **REPLACE** 속성을 사용하여 기존 정의를 지정된 새 정의로 바꿀 수 있습니다. **ALTER** 사용과 **DEFINE** 사용의 차이점은 **REPLACE**와 함께 **ALTER**를 사용할 경우 지정되지 않은 매개변수를 변경하지 않지만 **REPLACE**와 함께 **DEFINE**을 사용하면 모든 매개변수를 설정합니다.

프로시저

- 큐 속성을 변경하려면 다음 예에 표시된 대로 **ALTER** 명령 또는 **DEFINE** 명령을 사용하십시오. 이들 예에서 ORANGE.LOCAL.QUEUE 큐의 최대 메시지 길이는 10,000바이트로 감소합니다.
 - **ALTER** 명령 사용:

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

이 명령은 단일 속성, 즉 최대 메시지 길이의 속성을 변경합니다. 기타 모든 속성이 동일하게 유지됩니다.

- **DEFINE** 명령을 **REPLACE** 옵션과 함께 사용:

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

이 명령은 최대 메시지 길이를 변경할 뿐만 아니라, 기본값으로 제공된 기타 모든 속성들도 변경합니다. 따라서 예를 들어, 큐가 이전에 put 금지된 상태였던 경우에는 put 사용 가능으로 변경됩니다. SYSTEM.DEFAULT.LOCAL.QUEUE 큐에서 지정된 대로 put 사용 가능이 기본값이기 때문입니다.

기존 큐의 최대 메시지 길이를 줄여도 기존 메시지는 영향을 받지 않습니다. 그러나, 새 메시지는 새 기준을 충족해야 합니다.

관련 참조

[ALTER 큐](#)

[ALTER QLOCAL](#)
[DEFINE QUEUES](#)
[DEFINE QLOCAL](#)

CLEAR QLOCAL을 사용하여 로컬 큐 지우기

CLEAR QLOCAL MQSC 명령을 사용하여 로컬 큐를 지울 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

다음의 경우 큐를 지울 수 없습니다.

- 동기점 아래에 큐에 걸린 미확약된 메시지가 있습니다.
- 애플리케이션이 현재 큐를 열어 두었습니다.

이 태스크 정보

CLEAR QLOCAL 명령을 사용하여 로컬 큐를 지우려는 경우 큐의 이름을 로컬 큐 관리자에 정의해야 합니다.

참고: 이러한 선택을 변경할 수 있게 하는 어떤 프롬프트도 표시되지 않습니다. Enter 키를 누르면 메시지가 유실됩니다.

프로시저

로컬 큐에서 메시지를 지우려면 다음 예에 표시된 대로 **CLEAR QLOCAL**을 사용하십시오.

이 예에서는 MAGENTA.QUEUE라고 하는 로컬 큐에서 모든 메시지가 삭제됩니다.

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

관련 참조

[CLEAR QLOCAL](#)

DELETE QLOCAL을 사용하여 로컬 큐 삭제

DELETE QLOCAL MQSC 명령을 사용하여 로컬 큐를 삭제할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

미확약된 메시지가 있는 경우 큐를 삭제할 수 없습니다.

큐에 하나 이상의 커밋된 메시지가 있고 커밋되지 않은 메시지는 없는 경우 **PURGE** 옵션을 지정하는 경우에만 이 큐를 삭제할 수 있습니다. 그러면 이름 지정된 큐에 커밋된 메시지가 있더라도 삭제가 진행되고 이들 메시지도 영구 제거됩니다.

PURGE 대신에 **NOPURGE**를 지정하면 큐에 확약된 메시지가 있을 경우 큐가 삭제되지 않습니다.

프로시저

- 로컬 큐를 삭제하려면 다음 예에 표시된 대로 **DELETE QLOCAL** 명령을 사용하십시오. 이 예는 큐에 커밋된 메시지가 없는 경우에 PINK.QUEUE 큐를 삭제합니다.

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

이 예는 큐에 커밋된 메시지가 있더라도 PINK.QUEUE 큐를 삭제합니다.

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

관련 참조

[DELETE QLOCAL](#)

샘플 프로그램을 사용하여 큐 찾아보기

IBM MQ에서는 큐에서 메시지의 콘텐츠를 찾아보기 위해 사용할 수 있는 샘플 큐 브라우저를 제공합니다.

이 태스크 정보

이 브라우저는 다음 위치에서 소스 및 실행 파일 둘 다의 형식으로 제공되며 여기서 *MQ_INSTALLATION_PATH*는 IBM MQ가 설치되는 상위 레벨 디렉토리를 나타냅니다.

Windows Windows에서 샘플 큐 브라우저의 파일 이름과 경로는 다음과 같습니다.

소스

```
MQ_INSTALLATION_PATH\tools\c\samples\
```

실행 파일

```
MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe
```

Linux

AIX

AIX and Linux에서 파일 이름과 경로는 다음과 같습니다.

소스

```
MQ_INSTALLATION_PATH/samp/amqsbcg0.c
```

실행 파일

```
MQ_INSTALLATION_PATH/samp/bin/amqsbcg
```

프로시저

- 샘플 프로그램을 실행하려면 다음 예에 표시된 대로 명령을 입력하십시오.
샘플 프로그램에는 두 개의 입력 매개변수 즉, 메시지를 찾아볼 큐의 이름과 해당 큐를 소유하는 큐 관리자가 필요합니다. 예를 들면, 다음과 같습니다.

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

결과

이 명령의 일반적인 결과는 다음 예에 표시되어 있습니다.

```
AMQSBCG0 - starts here
*****

MQOPEN - 'SYSTEM.ADMIN.QMGR.EVENT'

MQGET of message number 1
****Message descriptor****

StrucId : 'MD ' Version : 2
Report : 0 MsgType : 8
Expiry : -1 Feedback : 0
Encoding : 546 CodedCharSetId : 850
Format : 'MQEVENT '
Priority : 0 Persistence : 0
MsgId : X'414D512073617475726E2E71756575650005D30033563DB8'
CorrelId : X'000000000000000000000000000000000000000000000000'
BackoutCount : 0
ReplyToQ : '
ReplyToQMgr : 'saturn.queue.manager'
** Identity Context
```


이 속성과 동등한 PCF는 **MQIA_MAX_Q_FILE_SIZE**입니다. [큐 변경, 복사 및 작성](#)을 참조하십시오.

큐 상태에 대한 두 가지 속성은 다음과 같습니다.

CURFSIZE

메가바이트 단위로 큐 파일의 현재 크기를 표시하며 가장 근사치의 메가바이트로 반올림됩니다.

MQSC 명령, IBM MQ Explorer 또는 administrative REST API를 사용하여 이 속성의 값을 설정하거나 표시할 수 있습니다.

자세한 정보는 [CURFSIZE](#)를 참조하십시오.

이 속성과 동등한 PCF는 **MQIA_CUR_Q_FILE_SIZE**입니다. [큐 조회](#) 및 [큐 조회 \(응답\)](#)를 참조하십시오.

CURMAXFS

큐에서 사용 중인 현재 블록 크기가 지정된 경우 가장 가까운 메가바이트로 증가할 수 있는 큐 파일의 현재 최대 크기를 표시합니다.

MQSC 명령, IBM MQ Explorer 또는 administrative REST API를 사용하여 이 속성의 값을 설정하거나 표시할 수 있습니다.

자세한 정보는 [CURMAXFS](#)를 참조하십시오.

이 속성과 동등한 PCF는 **MQIA_CUR_MAX_FILE_SIZE**입니다. [큐 조회](#) 및 [큐 조회 \(응답\)](#)를 참조하십시오.

블록 크기 및 세분화

큐 파일은 블록이라고 하는 세그먼트로 나뉘어집니다. 큐 파일의 최대 크기를 증가시키기 위해 큐 관리자가 큐의 블록 크기 또는 세분화를 변경해야 할 수도 있습니다.

새로 정의된 큐가 큰 **MAXFSIZE** 값으로 작성되는 경우 큐는 적합한 블록 크기로 작성됩니다. 그러나 기존 큐의 **MAXFSIZE** 값이 증가된 경우 (예: **ALTER QLOCAL** MQSC 명령 사용), 큐 관리자가 큐를 재구성하기 위해 큐를 비울 수 있도록 허용해야 할 수 있습니다.

자세한 정보는 138 페이지의 『[저장할 수 있는 IBM MQ 큐 파일의 데이터 계산](#)』의 내용을 참조하십시오.



주의: 일부 파일 시스템과 운영 체제는 전체 파일 시스템의 크기 또는 개별 파일의 크기에 제한이 있습니다. 엔터프라이즈가 사용하는 시스템의 한계를 확인해야 합니다.

관련 참조

[ALTER QUEUES](#)

[DISPLAY QUEUE](#)

[DISPLAY QSTATUS](#)

Multi IBM MQ 큐 파일의 크기 변경

큐 파일의 최대 크기를 늘리거나 줄일 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다.

- ▶ **ALW** ~에 AIX, Linux, and Windows, 다음에서 MQSC 명령을 실행합니다. **runmqsc** 명령 프롬프트. 보다 MQSC 명령을 대화식으로 실행 **runmqsc** 그리고 아래의 텍스트 파일에서 MQSC 명령 실행 **runmqsc**.
- ▶ **IBM i** ~에 IBM i, 스크립트 파일에 명령 목록을 만든 다음 **STRMQMQSC** 명령. 보다 [MQSC 명령을 사용하여 관리 IBM i](#).

큐 파일의 새 크기를 설정하기 전에 [DISPLAY QLOCAL](#) MQSC 명령을 사용하여 변경할 큐 파일의 크기를 확인하십시오. 예를 들어, 다음 명령을 실행하십시오.

```
DISPLAY QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) MAXFSIZE
```

다음 출력이 수신됩니다.

```
AMQ8409I: Display queue details
```

QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE) TYPE (QLOCAL)
MAXFSIZE (DEFAULT)

큐 파일의 최대 크기가 기본값인 2,088,960MB임을 표시합니다.

이 태스크 정보

다음 프로시저는 다음을 수행하는 방법을 보여줍니다.

- 큐 파일이 확장될 수 있는 최대 크기를 줄입니다.
- 큐 파일이 확장될 수 있는 최대 크기를 늘립니다.



주의: 애플리케이션이 작성된 방식과 성능에 미칠 수 있는 영향에 대한 고려 없이 큐 파일의 크기를 늘리는 것은 주의해야 합니다. 매우 큰 큐 파일에서 무작위로 메시지에 액세스하는 것은 매우 느릴 수 있습니다.

큐 파일의 최대 크기를 기본값 이상으로 늘리려는 경우 상관 ID 및 IBM MQ classes for JMS

JM 3.0 또는 IBM MQ classes for Jakarta Messaging 선택자 문자열과 같은 메시지 선택자를 사용하는 것에 주의해야 합니다. 큐 파일이 클수록 큐에 대한 선입선출(FIFO) 액세스에 더 적합해집니다.

개별 큐 파일에 매우 많은 양의 데이터를 보유하는 것은 순환 로깅을 위해 구성된 큐 관리자에서만 수행되거나 개별 큐에 대해 매체 이미지가 사용되지 않는 경우에만 수행해야 합니다.

큐 관리자의 조작에 영향을 줄 수 있으므로 SYSTEM 큐의 크기를 제한해서는 안 됩니다.

프로시저

1. 최대 큐 파일 크기 줄이기

- a) 다음 MQSC 명령을 실행하여 크기가 500GB인 SMALLQUEUE라는 로컬 파일을 작성하십시오.

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
  2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

그러면 다음 메시지를 수신합니다. AMQ8006I:

참고: 파일에 이미 있는 데이터 양보다 작은 값으로 큐를 구성하는 경우 새 메시지를 큐에 넣을 수 없습니다.

애플리케이션이 충분한 공간이 없는 큐 파일에 메시지를 넣으려고 시도하는 경우 이 애플리케이션은 리턴 코드 MQRC_Q_SPACE_NOT_AVAILABLE을 수신합니다. 큐에서 충분한 메시지를 파괴적으로 읽으면 애플리케이션이 새 메시지를 큐에 넣기 시작할 수 있습니다.

2. 최대 큐 파일 크기를 늘리십시오.

- a) 다음 MQSC 명령을 실행하여 크기가 5TB인 LARGEQUEUE이라는 로컬 파일을 작성하십시오.

```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
  3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```



Multi 저장할 수 있는 IBM MQ 큐 파일의 데이터 계산

큐에 저장할 수 있는 데이터의 양은 큐가 나뉘어지는 개별 블록의 크기로 제한됩니다. MQSC 명령을 사용하여 블록 크기 및 세분성을 확인하고 큐 파일의 크기를 확인할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다.

- **ALW** ~에 AIX, Linux, and Windows, 다음에서 MQSC 명령을 실행합니다. **runmqsc** 명령 프롬프트. 보다 MQSC 명령을 대화식으로 실행 **runmqsc** 그리고 아래의 텍스트 파일에서 MQSC 명령 실행 **runmqsc**.
- **IBM i** ~에 IBM i, 스크립트 파일에 명령 목록을 만든 다음 **STRMQMMQSC** 명령. 보다 MQSC 명령을 사용하여 관리 IBM i.

프로시저

- 블록 크기 및 단위를 확인하십시오.

기본 블록 크기는 512바이트입니다. 2TB보다 큰 큐 파일을 지원하려면 큐 관리자는 블록 크기를 증가시켜야 합니다.

큐에 대해 **MAXFSIZE**를 구성할 때 블록 크기가 자동으로 계산되지만 큐에 이미 그에 대한 메시지가 있는 경우 수정된 블록 크기를 큐에 적용할 수 없습니다. 큐가 비어 있으면 큐 관리자가 구성된 **MAXFSIZE**를 지원하도록 블록 크기를 자동으로 수정합니다.

DISPLAY QSTATUS 명령에는 큐가 새 블록 크기를 사용하도록 수정되었음을 확인할 수 있는 새 속성 **CURMAXFS**가 있습니다.

다음 예제에서 **CURMAXFS** 값 4177920은 큐 파일이 현재 크기가 약 4TB까지 증가할 수 있음을 확인합니다. 큐에 구성된 **MAXFSIZE**의 값이 **CURMAXFS**의 값보다 큰 경우, 큐 관리자는 큐 파일의 블록 크기를 재구성하기 전에 큐가 비워질 때까지 계속 대기합니다.

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
QUEUE(LARGEQUEUE) TYPE(QUEUE)
CURMAXFS(4177920) CURDEPTH(100000)
```

- 큐 파일의 크기를 확인하십시오.

DISPLAY QSTATUS 명령에서 **CURFSIZE** 속성을 사용하여 디스크에 있는 큐 파일의 현재 크기 (MB)를 표시할 수 있습니다. 이 속성은 파일 시스템에 직접 액세스할 수 없는 IBM MQ Appliance와 같은 플랫폼에서 유용할 수 있습니다.

```
DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
1 : DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
AMQ8450I: Display queue status details
QUEUE(SMALLQUEUE) TYPE(QUEUE)
CURDEPTH(4024) CURFSIZE(10)
```

참고: 큐에서 제거된 메시지가 큐에 있는 경우 **CURFSIZE** 속성은 즉시 감소하지 않습니다.

일반적으로 큐 파일의 공간은 큐가 열려 있는 애플리케이션이 없고 큐에 인다우트 (in-doubt) 메시지가 저장되어 있지 않은 경우에만 해제됩니다. 큐 관리자에 의해 로드되는 큐 파일의 필수 자르기 또는 압축은 [체크포인트](#), 큐 관리자 종료 또는 큐의 매체 이미지를 기록하는 동안 발생합니다.

관련 참조

[ALTER QUEUES](#)
[DISPLAY QSTATUS](#)

리모트 큐에 대한 작업

리모트 큐는 리모트 큐 관리자의 큐를 참조하는 로컬 큐 관리자의 정의입니다. 로컬 위치에서 리모트 큐를 정의할 필요는 없지만 이를 수행하는 경우 리모트 큐가 있는 큐 관리자의 ID로 규정되는 이름을 지정하지 않고 애플리케이션이 로컬로 정의된 이름으로 리모트 큐를 참조할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

애플리케이션이 로컬 큐 관리자에 연결되고 MQOPEN 호출을 실행합니다. 열린 호출에서 지정된 큐 이름은 로컬 큐 관리자에서 리모트 큐 정의의 이름입니다. 리모트 큐 정의에서는 대상 큐, 대상 큐 관리자 및 선택적으로 전송 큐의 이름을 제공합니다. 리모트 큐에 메시지를 배치하기 위해 애플리케이션은 MQOPEN 호출에서 리턴되는 핸들을 지정하면서 MQPUT 호출을 실행합니다. 큐 관리자는 메시지의 초기에 전송 헤더에서 리모트 큐 이름 및 리모트 큐 관리자 이름을 사용합니다. 이 정보는 네트워크에서 메시지를 올바른 정의로 라우팅하기 위해 사용됩니다.

관리자로서 리모트 큐 정의를 대체하여 메시지의 정의를 제어할 수 있습니다.

프로시저

- 리모트 큐 관리자가 소유한 큐에 메시지를 넣습니다.

애플리케이션이 큐 관리자에 연결됩니다(예: saturn.queue.manager). 대상 큐는 다른 큐 관리자에 의해 소유됩니다.

MQOPEN 호출에서 애플리케이션은 이러한 필드를 지정합니다.

필드 값	설명
<i>ObjectName</i> CYAN.REMOTE.QUEUE	리모트 큐 오브젝트의 로컬 이름을 지정합니다. 이는 대상 큐 및 대상 큐 관리자를 정의합니다.
<i>ObjectType</i> (큐)	이 오브젝트를 큐로 식별합니다.
<i>ObjectQmgrName</i> Blank 또는 saturn.queue.manager	이 필드는 선택적입니다. 공백이면, 로컬 큐 관리자 이름은 추측됩니다. (이는 리모트 큐 정의가 있는 큐 관리자입니다.)

이후, 애플리케이션은 이 큐에 메시지를 배치하기 위해 MQPUT 호출을 실행합니다.

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

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

설명:

QREMOTE (CYAN.REMOTE.QUEUE)

리모트 큐 오브젝트의 로컬 이름을 지정합니다. 이는 이 큐 관리자에 연결된 애플리케이션이 리모트 큐 관리자 jupiter.queue.manager에서 큐 AUTOMOBILE.INSURANCE.QUOTE.QUEUE를 열기 위해 MQOPEN 호출에 지정해야 하는 이름입니다.

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

큐의 사용을 설명하는 추가 텍스트를 제공합니다.

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

리모트 큐 관리자에서 대상 큐의 이름을 지정합니다. 이는 큐 이름 CYAN.REMOTE.QUEUE를 지정하는 애플리케이션에서 보내지는 메시지에 대한 실제 대상 큐입니다. 큐 AUTOMOBILE.INSURANCE.QUOTE.QUEUE는 리모트 큐 관리자에서 로컬 큐로 정의되어야 합니다.

RQMNAME (jupiter.queue.manager)

대상 큐 AUTOMOBILE.INSURANCE.QUOTE.QUEUE를 소유하는 리모트 큐 관리자의 이름을 지정합니다.

XMITQ (INQUOTE.XMIT.QUEUE)

전송 큐의 이름을 지정합니다. 이는 선택적입니다. 전송 큐의 이름이 지정되지 않으면, 리모트 큐 관리자와 동일한 이름을 가지는 큐가 사용됩니다.

두 가지 경우 모두 적절한 전송 큐가 전송 큐임을 지정하는 **Usage** 속성이 있는 로컬 큐로 정의되어야 합니다(MQSC 명령에서 USAGE(XMITQ)).

- 리모트 큐에 메시지를 넣습니다 (대체 방법).

리모트 큐의 로컬 정의를 사용하는 것이 리모트 큐에 메시지를 배치하는 유일한 방법은 아닙니다. 애플리케이션은 MQOPEN 호출의 일부로서 리모트 큐 관리자 이름을 포함하여 전체 큐 이름을 지정할 수 있습니다. 이 경우, 리모트 큐의 로컬 정의는 필요하지 않습니다. 그러나, 이는 애플리케이션이 런타임 시 리모트 큐 관리자의 이름에 대해 알거나 이에 대한 액세스 권한을 가지고 있어야 한다는 것을 의미합니다.

- 리모트 큐와 함께 다른 명령을 사용하십시오.

MQSC 명령을 사용하여 리모트 큐 오브젝트의 속성을 표시하거나 대체할 수 있고, 또는 리모트 큐 오브젝트를 삭제할 수 있습니다. 예를 들면, 다음과 같습니다.

- 리모트 큐의 속성을 표시하려면 다음을 수행하십시오.

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- 배치를 사용 가능하게 하기 위해 리모트 큐를 변경하려면 다음을 수행하십시오. 이는 대상 큐에 영향을 주는 것이 아니라, 이 리모트 큐를 지정하는 애플리케이션에만 영향을 줍니다.

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

- 이 리모트 큐를 삭제하려면 다음을 수행하십시오. 이는 대상 큐에 영향을 주지 않고, 해당 로컬 정의에만 영향을 줍니다.

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

참고: 리모트 큐를 삭제할 때, 리모트 큐의 로컬 표현만 삭제합니다. 리모트 큐 자체 또는 메시지를 삭제하지 않습니다.

리모트 큐 정의를 알리어스로 사용할 수 있습니다.

다른 큐 관리자에서 큐를 찾는 것 외에도 큐 관리자 알리어스 및 응답 대상 큐 알리어스에 대한 리모트 큐의 로컬 정의를 사용할 수도 있습니다. 두 유형의 알리어스 모두 리모트 큐의 로컬 정의를 통해 해결됩니다. 메시지가 목적지에 도착하기에 적절한 채널을 설정해야 합니다.

큐 관리자 알리어스

알리어스는 메시지에 지정된 것처럼 대상 큐 관리자의 이름이 메시지 라우트의 큐 관리자에 의해 수정된다는 프로세스입니다. 큐 관리자의 네트워크 내에서 메시지의 목적지를 제어하기 위해 사용할 수 있으므로 큐 관리자 알리어스는 중요합니다.

제어 시점에서 큐 관리자의 리모트 큐 정의를 대체하여 이를 수행하십시오. 송신 애플리케이션은 지정된 큐 관리자 이름이 알리어스라는 것을 알지 못합니다.

큐 관리자 알리어스에 대한 자세한 정보는 [알리어스의 개념](#) 을 참조하십시오.

응답 대상 큐 알리어스

선택적으로, 애플리케이션은 큐에 요청 메시지를 배치할 때 응답 대상 큐의 이름을 지정할 수 있습니다.

메시지를 처리하는 애플리케이션이 응답 대상 큐의 이름을 추출하는 경우, 필요하면 응답 메시지를 송신할 위치를 압니다.

응답 대상 큐 알리어스는 요청 메시지에 지정된 것처럼 응답 대상 큐가 메시지 라우트에서 큐 관리자에 의해 대체되는 프로세스입니다. 송신 애플리케이션은 지정된 응답 대상 큐 이름이 알리어스임을 알지 못합니다.

응답 대상 큐 알리어스를 사용하면 응답 대상 큐의 이름과 선택적으로 큐 관리자를 대체할 수 있습니다. 이것은 응답 메시지에 사용되는 라우트를 제어할 수 있습니다.

요청 메시지, 응답 메시지 및 응답 대상 큐에 대한 자세한 정보는 [메시지의 유형 및 응답 대상 큐 및 큐 관리자를 참조하십시오](#).

응답 대상 큐 알리어스에 대한 자세한 정보는 [응답 대상 큐 알리어스 및 클러스터를 참조하십시오](#).

알리어스 큐에 대한 작업

다른 큐 또는 토픽을 간접적으로 참조하기 위한 알리어스 큐를 정의할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 관리 IBM MQ MQSC 명령 사용.



주의: 분배 목록에서는 토픽 오브젝트를 가리키는 알리어스 큐 사용을 지원하지 않습니다. 알리어스 큐가 분배 목록의 토픽 오브젝트를 가리키는 경우 IBM MQ에서 MQRC_ALIAS_BASE_Q_TYPE_ERROR를 리턴합니다.

이 태스크 정보

알리어스 큐가 참조하는 큐는 다음 중 하나일 수 있습니다.

- 로컬 큐(130 페이지의 『[DEFINE QLOCAL을 사용하여 로컬 큐 정의](#)』 참조).
- 리모트 큐의 로컬 정의(139 페이지의 『[리모트 큐에 대한 작업](#)』 참조).
- 토픽.

알리어스 큐는 실제 큐는 아니지만, 런타임 시 실제(또는 대상) 큐로 해석되는 정의입니다. 알리어스 큐 정의는 대상 큐를 지정합니다. 애플리케이션이 알리어스 큐에 대해 MQOPEN 호출을 할 때, 큐 관리자는 대상 큐 이름에 대한 알리어스를 해석합니다.

알리어스 큐는 로컬로 정의된 알리어스 큐로 해석될 수 없습니다. 그러나, 알리어스 큐는 로컬 큐 관리자가 멤버인 클러스터의 어딘가에서 정의되는 알리어스 큐로 해석될 수 있습니다. 추가 정보는 [이름 해석](#)을 참조하십시오.

알리어스 큐는 다음에 유용합니다.

- 다른 애플리케이션에 대상 큐에 대한 다른 레벨의 액세스 권한 제공.
- 다른 애플리케이션에서 다른 방법으로 동일한 큐에 대해 작업할 수 있도록 허용. (아마도 다른 기본 특성 또는 다른 기본 지속 값을 지정하려고 합니다.)
- 유지보수, 마이그레이션 및 워크로드 밸런싱 단순화. (아마도 애플리케이션을 변경하지 않고 대상 큐 이름을 변경하려고 합니다. 이는 알리어스를 계속 사용합니다.)

예를 들어, 애플리케이션이 MY.ALIAS.QUEUE라는 큐에 메시지를 배치하기 위해 개발된다고 가정하십시오. MQOPEN 요청을 작성할 때와 간접적으로 이 큐에 메시지를 배치하는 경우 이 큐의 이름을 지정합니다. 애플리케이션은 큐가 알리어스 큐라는 것을 인지하지 않습니다. 이 알리어스를 사용하는 각 MQI 호출의 경우, 큐 관리자는 실제 큐 이름을 해석합니다. 이는 이 큐 관리자에 정의되는 로컬 큐 또는 리모트 큐가 될 수 있습니다.

TARGET 속성의 값을 변경하여 MQI 호출을 다른 큐 관리자의 다른 큐로 경로 재지정할 수 있습니다. 이는 유지보수, 마이그레이션 및 로드 밸런싱에 유용합니다.

프로시저

- 알리어스 큐를 정의하십시오.

다음 MQSC 명령은 알리어스 큐를 작성합니다.

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

이 명령은 MY.ALIAS.QUEUE를 큐 YELLOW.QUEUE로 지정하는 MQI 호출의 경로를 재지정합니다. 명령은 대상 큐를 작성하지 않습니다. MQI 호출은 YELLOW.QUEUE가 런타임 시 존재하지 않는 경우 실패합니다.

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

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

이 명령은 MQI 호출의 경로를 다른 큐 MAGENTA.QUEUE로 재지정합니다.

알리어스 큐를 사용하여 단일 큐(대상 큐)가 다른 애플리케이션에 대해 다른 속성을 갖는 것처럼 보이도록 할 수 있습니다. 각 애플리케이션에 하나씩 두 개의 알리어스를 정의하여 이를 수행합니다. 두 개의 애플리케이션이 있다고 가정하십시오.

- 애플리케이션 ALPHA는 YELLOW.QUEUE에 메시지를 배치할 수 있지만, 여기에서 메시지를 가져올 수 없습니다.
- 애플리케이션 BETA는 YELLOW.QUEUE에서 메시지를 가져올 수 있지만, 여기에 메시지를 배치할 수 없습니다.

다음 MQSC 명령은 애플리케이션 ALPHA에 대해 넣기가 사용 가능하고 가져오기가 사용 불가능한 알리어스를 정의합니다.

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

다음 명령은 애플리케이션 BETA에 사용 가능한 get 및 사용 불가능한 put인 알리어스를 정의합니다.

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

ALPHA는 MQI 호출에서 큐 이름 ALPHAS.ALIAS.QUEUE를 사용합니다. BETA는 큐 이름 BETAS.ALIAS.QUEUE를 사용합니다. 이는 모두 동일한 큐에 액세스하지만, 다른 방법으로 액세스합니다.

로컬 큐와 함께 해당 속성을 사용하는 동일한 방법으로 큐 알리어스를 정의할 때 LIKE 및 REPLACE 속성을 사용할 수 있습니다.

- 알리어스 큐와 함께 다른 명령을 사용하십시오.

적절한 MQSC 명령을 사용하여 알리어스 큐 속성을 표시 또는 대체하거나 알리어스 큐 오브젝트를 삭제할 수 있습니다. 예를 들면, 다음과 같습니다.

DISPLAY QALIAS 명령을 사용하여 알리어스 큐의 속성을 표시하십시오.

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

ALTER QALIAS 명령을 사용하여 알리어스가 해석하는 기본 큐 이름을 대체하십시오. 여기서 **force** 옵션은 큐가 열려 있더라도 강제로 변경합니다.

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

DELETE QALIAS 명령을 사용하여 이 큐 알리어스를 삭제하십시오.

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

애플리케이션에서 현재 큐가 열려 있는 경우 알리어스 큐를 삭제할 수 없습니다.

관련 개념
[분배 목록](#)

관련 참조

[ALTER QALIAS](#)

[DEFINE QALIAS](#)

[DELETE QALIAS](#)

모델 큐에 대한 작업

모델 큐는 애플리케이션에게 필요한 대로 큐를 작성할 수 있는 편리한 방법을 제공합니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

큐 관리자는 모델 큐로서 정의되는 큐 이름을 지정하여 애플리케이션에서 MQI 호출을 수신하는 경우 동적 큐를 작성합니다. 새 동적 큐의 이름은 큐가 작성될 때 큐 관리자에서 생성됩니다. 모델 큐는 작성되는 동적 큐의 속성을 지정하는 템플릿입니다.

프로시저

- 모델 큐를 정의하십시오.

DEFINE QMODEL MQSC 명령을 사용하여 로컬 큐를 정의하는 것과 동일한 방법으로 속성 세트가 있는 모델 큐를 정의합니다. 모델 큐 및 로컬 큐에는 작성되는 동적 큐가 임시 또는 영구적인지 여부를 지정할 수 있는 모델 큐의 속성을 제외하고 동일한 속성 세트가 있습니다. (영구적 큐는 큐 관리자 재시작을 통해 유지보수되고 임시 큐는 그렇지 않습니다). 예를 들면, 다음과 같습니다.

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

이 명령은 모델 큐 정의를 작성합니다. **DEFTYPE** 속성을 보면 이 템플릿에서 작성된 실제 큐가 영구적 동적 큐임을 확인할 수 있습니다. 지정되지 않은 속성은 **SYSYSTEM.DEFAULT.MODEL.QUEUE** 기본 큐에서 자동적으로 복사됩니다.

모델 큐를 정의할 때 로컬 큐에서 사용하는 것과 동일한 방법으로 **LIKE** 및 **REPLACE** 속성을 사용할 수 있습니다.

- 모델 큐와 함께 다른 명령을 사용하십시오.

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

모델 큐의 속성을 표시하려면 **DISPLAY QUEUE** 명령을 사용하십시오.

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

이 모델에서 작성된 동적 큐에 배치할 수 있도록 모델을 대체하려면 **ALTER QMODEL** 명령을 사용하십시오.

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

이 모델 큐를 삭제하려면 **DELETE QMODEL** 명령을 사용하십시오.

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

관련 참조

[ALTER QMODEL](#)

[DEFINE QMODEL](#)

[DELETE QMODEL](#)

[DISPLAY QUEUE](#)

데드-레터 큐에 대한 작업

올바른 대상으로 전달될 수 없는 해당 메시지가 이후 검색을 위해 저장될 수 있도록 각 큐 관리자에는 일반적으로 데드-레터 큐로서 사용할 로컬 큐가 있습니다. 데드-레터 큐에 대한 큐 관리자에 대해 알려주고 데드-레터 큐에서 발견된 메시지가 처리되는 방법을 지정합니다. 데드-레터 큐를 사용하는 것은 메시지가 전달되는 순서에 영향을 줄 수 있습니다. 따라서 이를 사용하지 않도록 선택할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

SYSTEM.DEAD.LETTER.QUEUE라는 샘플 데드-레터 큐는 제품과 함께 사용할 수 있습니다. 이 큐는 큐 관리자를 작성할 때 자동으로 작성됩니다. 필요한 경우 이 정의를 수정하고 이름을 바꿀 수 있습니다.

데드-레터 큐에는 다음을 제외하고 특별한 요구사항이 없습니다.

- 이는 로컬 큐여야 합니다
- MAXMSGL(최대 메시지 길이) 속성은 큐가 데드-레터 헤더(MQDLH)의 크기 **뿐만 아니라** 큐 관리자가 핸들링해야 하는 가장 큰 메시지를 수용할 수 있도록 할 수 있어야 합니다

데드-레터 큐를 사용하는 것은 메시지가 전달되는 순서에 영향을 줄 수 있습니다. 따라서 이를 사용하지 않도록 선택할 수 있습니다.

프로시저

- 데드-레터 큐에 대해 큐 관리자에게 알리십시오.

이를 수행하려면 **crtmqm** 명령 (예: **crtmqm -u DEAD.LETTER.QUEUE**) 에서 데드-레터 큐 이름을 지정하거나 **ALTER QMGR** 명령에서 **DEADQ** 속성을 사용하여 나중에 지정하십시오. 이를 사용하기 전에 데드-레터 큐를 정의해야 합니다.

- 데드-레터 큐에서 발견된 메시지가 처리되는 방법을 지정하십시오.

메시지가 전달될 수 없을 때 데드-레터 큐가 사용되는지 여부를 판별하기 위해 **USEDLQ** 채널 속성을 설정합니다. 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.

IBM MQ 데드-레터 큐 핸들러를 사용하여 데드-레터 큐에서 찾은 메시지를 처리하거나 제거하는 방법을 지정합니다. 146 페이지의 『IBM MQ 데드-레터 큐에서 메시지 처리』을 참조하십시오.

관련 개념

[데드-레터 큐](#)

관련 태스크

[미배달 메시지 문제점 해결](#)

관련 참조

[ALTER QMGR](#)

IBM MQ 데드-레터 큐에서 메시지 처리

데드-레터 큐 (DLQ) 에서 메시지를 처리하려면 IBM MQ에서 제공하는 기본 DLQ 핸들러를 사용하십시오. 핸들러는 정의하는 규칙 테이블의 입력 항목에 대응하는 DLQ에 대한 메시지와 일치합니다.

이 태스크 정보

큐 관리자, 메시지 채널 에이전트(MCA) 및 애플리케이션에 의해 DLQ에 메시지가 기록될 수 있습니다. DLQ에 대한 모든 메시지는 데드 레터 헤더 구조, MQDLH를 접두부로 사용합니다. 큐 관리자 또는 메시지 채널 에이전트에 의해 DLQ에 놓여지는 메시지는 항상 다음 헤더가 있습니다. DLQ에 메시지를 넣는 애플리케이션은 이 헤더를 제공해야 합니다. MQDLH 구조의 *Reason* 필드에는 메시지가 DLQ에 있는 이유를 식별하는 이유 코드가 포함됩니다.

모든 IBM MQ 환경에는 정기적으로 DLQ에서 메시지를 처리하기 위한 루틴이 필요합니다. IBM MQ는 **runmqdlq** MQSC 명령을 사용하여 호출하는 데드-레터 큐 핸들러 (DLQ 핸들러) 라는 기본 루틴을 제공합니다.

DLQ에 있는 메시지를 처리하기 위한 지시사항은 사용자 작성 규칙 테이블 방식으로 DLQ 핸들러에 제공됩니다. 즉, DLQ 핸들러는 규칙 테이블의 입력 항목에 대응하여 DLQ에 있는 메시지와 일치합니다. DLQ 메시지가 규칙 테이블에 있는 입력 항목과 일치하면 DLQ 핸들러가 해당 입력 항목과 연관된 조치를 수행합니다.

관련 태스크

[미배달 메시지 문제점 해결](#)

관련 참조

[데드-레터 큐](#)

데드-레터 큐 핸들러 호출

runmqdlq 제어 명령을 사용하여 데드-레터 큐 (DLQ) 핸들러를 호출하십시오. 처리하려는 DLQ 및 두 가지 방법으로 사용하려는 큐 관리자의 이름을 지정할 수 있습니다.

시작하기 전에

DLQ 핸들러를 실행하려면 DLQ의 메시지가 전달되는 임의의 메시지 큐 및 DLQ 자체 모두에 액세스하기 위한 권한이 부여되어야 합니다. 메시지 컨텍스트에서 사용자 ID의 권한을 가지는 큐에 메시지를 넣기 위한 DLQ 핸들러의 경우, 기타 사용자의 ID를 가정하기 위한 권한이 부여되어야 합니다.

이 태스크 정보

다음 예는 큐 관리자 ABC1.QUEUE.MANAGER가 소유하는 ABC1.DEAD.LETTER.QUEUE라는 DLQ에 적용됩니다.

표시된 대로 DLQ 또는 큐 관리자를 지정하지 않으면, 설치를 위한 기본 큐 관리자가 해당 큐 관리자에 속하는 DLQ와 함께 사용됩니다.

runmqdlq 명령은 stdin에서 해당 입력사항을 가져옵니다. 규칙 테이블에서 stdin 를 경로 재지정하여 규칙 테이블을 **runmqdlq** 와 연관시킵니다.

runmqdlq 명령에 대한 자세한 정보는 [runmqdlq](#)를 참조하십시오.

프로시저

- **runmqdlq** 명령에 대한 매개변수로 DLQ 및 큐 관리자의 이름을 지정할 수 있습니다. 예를 들어, 명령 프롬프트에서 다음을 수행하십시오.

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

- 규칙 테이블에서 DLQ 및 큐 관리자의 이름을 지정할 수 있습니다.

예를 들면, 다음과 같습니다.

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

관련 개념

[데드-레터 큐](#)

관련 태스크

[미배달 메시지 문제점 해결](#)

샘플 DLQ 핸들러 **amqsdlq**

runmqdlq 명령을 사용하여 호출된 데드-레터 큐 핸들러 외에, IBM MQ 는 샘플 DLQ 핸들러 **amqsdlq** 의 소스에 **runmqdlq**에서 제공하는 것과 유사한 함수를 제공합니다.

사용자의 요구사항을 충족하는 DLQ 핸들러를 제공하도록 **amqsdlq**을(를) 사용자 정의할 수 있습니다. 예를 들어, 데드-레터 헤더 없이 메시지를 처리할 수 있는 DLQ 핸들러를 원한다고 결정할 수 있습니다. (기본 DLQ 핸들러와 샘플 **amqsdlq** 모두 데드-레터 헤더인 MQDLH로 시작하는 DLQ의 메시지만 처리합니다. MQDLH로 시작하지 않는 메시지는 정기적으로 오류가 있는 것으로 보고되며, 무기한 DLQ에 남게 됩니다.)

MQ_INSTALLATION_PATH은(는) IBM MQ이(가) 설치된 상위 레벨 디렉토리를 나타냅니다.

IBM MQ for Windows에서는 **amqsdlq**의 소스가 디렉토리에 제공됩니다.

```
MQ_INSTALLATION_PATH\tools\c\samples\d1q
```

그리고 컴파일된 버전이 디렉토리에 제공됩니다.

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

IBM MQ for UNIX 및 Linux 시스템에서 **amqsdlq**의 소스는 다음 디렉토리에 제공됩니다.

```
MQ_INSTALLATION_PATH/samp/d1q
```

그리고 컴파일된 버전이 디렉토리에 제공됩니다.

```
MQ_INSTALLATION_PATH/samp/bin
```

amqsdlqc(이)라는 샘플 프로그램의 빌드된 버전이 포함됩니다. 이를 사용하여 클라이언트 모드에서 리모트 큐 관리자에 연결할 수 있습니다. **amqsdlqc**를 사용하려면 환경 변수 **MQSERVER**, **MQCHLLIB** 또는 **MQCHLTAB** 중 하나를 설정하여 큐 관리자에 연결하는 방법을 식별해야 합니다. 예를 들면, 다음과 같습니다.

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414) "
```

DLQ 핸들러 규칙 테이블

데드-레터 큐 핸들러 규칙 테이블은 DLQ 핸들러가 DLQ에 도착하는 메시지를 처리하는 방법을 정의합니다.

규칙 테이블에는 두 가지 유형의 입력 항목이 있습니다.

- 표의 첫 번째 항목(선택적)에는 제어 데이터가 포함됩니다.
- 테이블의 기타 모든 입력 항목은 DLQ 핸들러가 따라야 할 규칙입니다. 각 규칙은 메시지가 일치되는 패턴(메시지 특성 세트)과, DLQ의 메시지가 지정된 패턴과 일치할 때 수행되는 조치로 구성됩니다. 규칙 테이블에는 최소 하나의 규칙이 있어야 합니다.

규칙 테이블의 각 입력 항목은 하나 이상의 키워드로 구성됩니다.

관련 개념

[데드-레터 큐](#)

관련 태스크

[미배달 메시지 문제점 해결](#)

데드-레터 큐 핸들러 규칙 테이블의 제어 데이터 입력 항목에 키워드를 포함할 수 있습니다.

참고:

- 수직 행()은 대안을 분리하며 이들 중 하나만 지정할 수 있습니다.
- 모든 키워드는 선택사항입니다.

INPUTQ (QueueName| ' (기본값)

처리하려는 DLQ의 이름입니다.

1. runmqdlq 명령에 매개변수로서 제공하는 INPUTQ 값이 규칙 테이블에서 INPUTQ 값을 대체합니다.
2. runmqdlq 명령에 매개변수로서 INPUTQ 값을 지정하지 않지만 규칙 테이블에서 값을 지정하는 경우, 규칙 테이블의 INPUTQ 값이 사용됩니다.
3. DLQ가 지정되지 않거나 규칙 테이블에서 INPUTQ(' ')를 지정하는 경우, runmqdlq 명령에 매개변수로서 제공되는 이름을 가지는 큐 관리자에 속하는 DLQ의 이름이 사용됩니다.
4. runmqdlq 명령에 매개변수로서 INPUTQ 값을 지정하지 않거나 규칙 테이블에서 값으로 INPUTQ 값을 지정하지 않는 경우, 규칙 테이블의 INPUTQ 키워드에서 이름 지정된 큐 관리자에 속하는 DLQ가 사용됩니다.

INPUTQM (QueueManager이름| ' (기본값)

INPUTQ 키워드에서 이름이 지정된 DLQ를 소유하는 큐 관리자의 이름입니다.

1. runmqdlq 명령에 매개변수로서 제공하는 INPUTQM 값이 규칙 테이블에서 INPUTQM 값을 대체합니다.
2. runmqdlq 명령에 매개변수로서 INPUTQM 값을 지정하지 않는 경우, 규칙 테이블에서 INPUTQM 값이 사용됩니다.
3. 큐 관리자가 지정되지 않거나 규칙 테이블에 INPUTQM(' ')를 지정하는 경우, 설치를 위한 기본 큐 관리자가 사용됩니다.

RETRYINT (간격| 60 (기본값))

DLQ 핸들러가 첫 번째 시도에서 처리될 수 없는 DLQ의 메시지를 처리해야 하고 반복되는 시도가 요청되는 간격(초)입니다. 기본적으로, 재시도 간격은 60초입니다.

WAIT (YES (기본값) |NO|nnn)

DLQ 핸들러가 처리할 수 있는 추가 메시지가 없다는 것을 감지할 때 DLQ에 도착할 추가 메시지를 기다려야 하는지 여부입니다.

YES

DLQ 핸들러가 무기한 대기합니다.

NO

DLQ 핸들러는 DLQ가 비어 있거나 처리할 수 있는 메시지가 포함되어 있다는 것을 감지할 때 종료됩니다.

nnn

DLQ 핸들러는 큐가 비어 있거나 처리할 수 있는 메시지를 포함하지 않는다는 것을 감지한 이후 종료하기 전에 도착할 새 작업에 대해 nnn초 동안 대기합니다.

사용이 많은 DLQ의 경우 WAIT(YES)를 지정하고, 활동 레벨이 낮은 DLQ의 경우 WAIT(NO) 또는 WAIT(nnn)를 지정하십시오. DLQ 핸들러가 종료하도록 허용되는 경우, 트리거를 사용하여 다시 호출하십시오. 트리거에 대한 자세한 정보는 트리거를 사용하여 IBM MQ 애플리케이션 시작을 참조하십시오.

규칙 테이블에 제어 데이터를 포함하는 것에 대한 대체는 runmqdlq 명령에 입력 매개변수로서 해당 큐 관리자 및 DLQ의 이름을 제공하는 것입니다. 규칙 테이블에 값을 지정하고 runmqdlq 명령에 입력으로 값을 지정하는 것이 모두 가능한 경우, runmqdlq 명령에 지정되는 값이 우위에 섭니다.

규칙 테이블에 제어-데이터 입력 항목을 포함하는 경우, 테이블에 첫 번째 입력 항목일 수 있습니다.

데드-레터 큐의 메시지가 일치되는 패턴 일치 키워드 및 DLQ 핸들러가 일치 메시지를 처리하는 방법을 판별하는 조치 키워드에 대한 설명입니다. 예시 규칙도 제공됩니다.

패턴 일치 키워드

DLQ의 메시지가 일치되는 값을 지정하려면 사용하는 패턴 일치 키워드는 다음과 같습니다. (모든 패턴 일치 키워드는 선택사항입니다):

APPLIDAT (*ApplIdentity*데이터|* (기본값))

DLQ의 메시지에 대한 메시지 디스크립터(MQMD)에 지정된 *ApplIdentityData* 값입니다.

APPLNAME (*PutAppl*이름|* (기본값))

DLQ에 있는 메시지의 메시지 디스크립터 MQMD의 *PutApplName* 필드에 지정된 대로 MQPUT 또는 MQPUT1 호출을 발행한 애플리케이션의 이름입니다.

APPLTYPE (*PutAppl*유형|* (기본값))

DLQ에 메시지의 메시지 디스크립터, MQMD에 지정되는 *PutApplType* 값입니다.

DESTQ (*QueueName*|* (기본값))

메시지의 목적지로 지정된 메시지 큐의 이름입니다.

DESTQM (*QueueManager*이름|* (기본값))

메시지가 향하는 메시지 큐의 큐 관리자의 이름입니다.

FEEDBACK (*Feedback*|* (기본값))

MsgType 값이 MQFB_REPORT일 때, *Feedback*은 보고서의 성질을 설명합니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQFB_COA를 사용하여 목적지 큐에서 도착 확인이 필요한 DLQ에 대한 해당 메시지를 식별할 수 있습니다.

FORMAT (형식|* (기본값))

메시지 송신자가 메시지 데이터 형식을 설명하는 데 사용하는 이름입니다.

MSGTYPE (*MsgType*|* (기본값))

DLQ에 있는 메시지의 메시지 유형입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQMT_REQUEST를 사용하여 응답이 필요한 DLQ에서 해당 메시지를 식별할 수 있습니다.

PERSIST (*Persistence*|* (기본값))

메시지의 지속성 값입니다. (메시지의 지속성은 큐 관리자 재시작 후에도 지속되는지 여부를 판별합니다.)

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQPER_PERSISTENT를 사용하여 지속적인 DLQ에서 메시지를 식별할 수 있습니다.

REASON (*ReasonCode*|* (기본값))

메시지가 DLQ에 놓여진 이유에 대해 설명하는 이유 코드입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQRC_Q_FULL을 사용하여 목적지 큐가 가득 찼기 때문에 DLQ에 배치된 메시지를 식별할 수 있습니다.

REPLYQ (*QueueName*|* (기본값))

DLQ에서 메시지의 메시지 디스크립터(MQMD)에 지정된 응답 대상 큐의 이름입니다.

REPLYQM (*QueueManager*이름|* (기본값))

DLQ에서 메시지의 메시지 디스크립터(MQMD)에 지정된 것으로 응답 대상 큐의 큐 관리자 이름입니다.

USERID (*UserIdentifier*|* (기본값))

DLQ에서 메시지의 메시지 디스크립터(MQMD)에 지정된 것으로 DLQ에서 메시지를 생성한 사용자의 사용자 ID입니다.

조치 키워드

일치 메시지가 처리되는 방법에 대해 설명하기 위해 사용되는 조치 키워드는 다음과 같습니다.

ACTION (DISCARD|IGNORE|RETRY|FWD)

이 규칙에 정의되는 패턴과 일치하는 DLQ에서 메시지에 대해 취하는 조치입니다.

DISCARD

DLQ에서 메시지를 삭제하십시오.

IGNORE

DLQ에 메시지를 남기십시오.

RETRY

해당 목적지 큐에 메시지를 넣으려는 첫 번째 시도가 실패하면 다시 시도하십시오. RETRY 키워드는 조치를 구현하기 위한 시도 횟수를 설정합니다. 제어 데이터의 RETRYINT 키워드는 시도 사이의 간격을 제어합니다.

FWD

FWDQ 키워드에 이름이 지정된 큐로 메시지를 전달합니다.

ACTION 키워드를 지정해야 합니다.

FWDQ (QueueName|&DESTQ|&REPLYQ)

ACTION(FWD)이 요청될 때 메시지가 전달되는 메시지 큐의 이름입니다.

QueueName

메시지 큐의 이름입니다. FWDQ(' ')가 올바르지 않습니다.

&DESTQ

MQDLH 구조의 DestQName 필드에서 큐 이름을 가져옵니다.

&REPLYQ

메시지 디스크립터(MQMD)의 ReplyToQ 필드에서 큐 이름을 가져옵니다.

FWDQ (&REPLYQ)를 지정하는 규칙이 비어 있는 ReplyToQ 필드와 메시지를 일치시킬 때 오류 메시지를 방지하려면 메시지 패턴에서 REPLYQ (?*)를 지정하십시오.

FWDQM (QueueManager이름 | & DESTQM | & REPLYQM | '' (기본값))

메시지를 전달할 큐의 큐 관리자입니다.

QueueManagerName

ACTION(FWD)이 요청될 때 메시지가 전달되는 큐의 큐 관리자의 이름입니다.

&DESTQM

MQDLH 구조의 DestQMGrName 필드에서 큐 관리자 이름을 가져옵니다.

&REPLYQM

메시지 디스크립터(MQMD)의 ReplyToQMGr 필드에서 큐 관리자 이름을 가져옵니다.

''

기본값인 FWDQM(' ')은 로컬 큐 관리자를 식별합니다.

HEADER (YES (기본값) |NO)

MQDLH가 ACTION(FWD)이 요청된 메시지에 남아 있어야 하는지 여부를 지정합니다. 기본적으로 MQDLH는 메시지에 남아 있습니다. HEADER 키워드는 FWD 이외의 다른 조치에는 유효하지 않습니다.

PUTAUT (DEF (기본값) | CTX)

DLQ 핸들러가 메시지를 넣을 때 필요한 권한입니다.

DEF

DLQ 핸들러 자체 권한을 가진 메시지를 넣으십시오.

CTX

메시지 컨텍스트에서 사용자 ID의 권한을 가진 메시지를 넣으십시오. PUTAUT(CTX)를 지정하면, 기타 사용자의 ID를 추측할 수 있는 권한이 지정되어야 합니다.

RETRY (RetryCount|1 (기본값))

제어 데이터의 RETRYINT 키워드에 지정된 간격에서 조치를 시도하기 위한 횟수(1 - 999,999,999 범위)입니다. DLQ 핸들러가 특정 규칙을 구현하기 위해 시도한 횟수는 DLQ 핸들러의 현재 인스턴스에 따라 다릅니다. 재시작하는 동안에는 계수가 지속되지 않습니다. DLQ 핸들러가 재시작되면, 규칙에 적용되는 시도 횟수는 0으로 재설정됩니다.

예 규칙

다음은 DLQ 핸들러 규칙 테이블의 규칙 예입니다.

PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +
ACTION (RETRY) RETRY (3)

이 규칙에 따라 DLQ 핸들러는 MQPUT 및 MQPUT1이 금지되어 DLQ에 있는 지속적인 메시지를 목적지 큐에 이동하도록 세 번 시도합니다.

규칙에 사용할 수 있는 모든 키워드에 대해서는 이 절 나머지 부분에서 설명됩니다. 다음에 유의하십시오.

- 키워드가 있는 경우 기본값은 밑줄 그어져 있습니다. 대부분의 키워드의 경우 기본값이 *(별표)이며 어느 값과도 일치합니다.
- 수직 행()은 대안을 분리하며 이들 중 하나만 지정할 수 있습니다.
- 모든 키워드 예외 ACTION은 선택사항입니다.

DLQ 규칙 테이블 규약

데드-레터 큐 핸들러 규칙 테이블의 구문, 구조 및 콘텐츠는 이러한 규약을 지켜야 합니다.

규칙 테이블은 다음 규약을 고수해야 합니다.

- 규칙 테이블에는 최소 하나의 규칙이 있어야 합니다.
- 키워드는 임의의 순서로 발생할 수 있습니다.
- 키워드는 규칙에 한 번만 포함될 수 있습니다.
- 키워드는 대소문자를 구분하지 않습니다.
- 키워드 및 매개변수 값은 최소 하나의 공백이나 쉼표로 다른 키워드와 구분해야 합니다.
- 공백은 규칙 처음이나 끝에 넣을 수 있으며, 키워드, 구두점, 값 사이에 공백을 넣을 수 있습니다.
- 각 규칙은 새 행에서 시작해야 합니다.
- Windows 시스템의 표에서 마지막 규칙은 캐리지 리턴/라인 공급 문자로 끝나야만 합니다. 테이블의 마지막 행이 빈 줄이도록 규칙의 끝에서 Enter 키를 입력하는지 확인하여 이를 얻을 수 있습니다.
- 이식성의 이유로 행의 중요한 길이는 72자를 넘어서는 안됩니다.
- 행에서 마지막 공백이 아닌 문자로서 더하기 부호(+)를 사용하여 다음 행에서 첫 번째 공백이 아닌 문자에서 규칙이 계속되는지 표시하십시오. 행에서 마지막 공백이 아닌 문자로서 빼기 부호(-)를 사용하여 다음 행의 시작에서 규칙이 계속되는지 표시하십시오. 키워드 및 매개변수 내에서 연속 문자가 발생할 수 있습니다.

예를 들면, 다음과 같습니다.

```
APPLNAME('ABC+  
D')
```

결과적으로 'ABCD'가 되고

```
APPLNAME('ABC-  
D')
```

결과: 'ABC D'.

- 별표(*)로 시작되는 주석 행은 규칙 테이블에서 어디에나 발생할 수 있습니다.
- 빈 줄은 무시됩니다.
- DLQ 핸들러 규칙 테이블의 각 입력 항목은 하나 이상의 키워드 및 연관된 매개변수로 구성됩니다. 매개변수는 다음 구문 규칙을 따라야 합니다.
 - 각 매개변수 값은 최소 하나의 유효 문자를 포함해야 합니다. 따옴표로 묶이는 값을 분리하는 작은따옴표는 중요한 것으로 여겨지지 않습니다. 예를 들어, 올바른 매개변수는 다음과 같습니다.

FORMAT('ABC')	3개의 유효 문자
FORMAT(ABC)	3개의 유효 문자
FORMAT('A')	1개의 유효 문자

FORMAT (A)	1개의 유효 문자
FORMAT (' ')	1개의 유효 문자

이러한 매개변수는 중요한 문자를 포함하지 않으므로 올바르지 않습니다.

```
FORMAT (' ')
FORMAT ( )
FORMAT ()
FORMAT
```

- 와일드카드 문자가 지원됩니다. 후미 공백을 제외하고는 단일 문자 대신 물음표(?)를 사용할 수 있습니다. 0 개 이상의 인접 문자 대신 별표(*)를 사용할 수 있습니다. 별표(*) 및 물음표(?)는 매개변수 값에서 항상 와일드카드 문자로 해석됩니다.
- 와일드카드 문자는 키워드(ACTION, HEADER, RETRY, FWDQ, FWDQM 및 PUTAUT)의 매개변수에 포함될 수 없습니다.
- 매개변수 값과 DLQ 메시지의 해당 필드에 있는 후미 공백은 와일드카드 일치를 수행할 때 유의미하지 않습니다. 그러나, 작은따옴표로 묶인 문자열 내의 선두 문자 및 임베드된 공백은 와일드카드 일치에 중요합니다.
- 숫자 매개변수는 물음표(?) 와일드카드 문자를 포함할 수 없습니다. 전체 숫자 매개변수 대신 별표(*)를 사용할 수 있지만, 숫자 매개변수의 일부로서 사용할 수 없습니다. 예를 들어, 다음은 올바른 숫자 매개변수입니다.

MSGTYPE (2)	응답 메시지만 적합합니다.
MSGTYPE (*)	모든 메시지 유형이 적합합니다.
MSGTYPE ('*')	모든 메시지 유형이 적합합니다.

그러나 MSGTYPE ('2*')은(는) 숫자 매개변수의 일부로 별표(*)를 포함하므로 유효하지 않습니다.

- 숫자 매개변수는 범위 0-999 999 999에 있어야 합니다. 매개변수 값이 이 범위에 있으면 키워드와 관련된 필드에서 현재 그 값이 올바르지 않더라도 허용됩니다. 숫자 매개변수에 기호 이름을 사용할 수 있습니다.
- 문자열 값이 키워드와 관련된 MQDLH 또는 MQMD에 있는 필드보다 짧으면, 필드 길이에 맞게 값이 공백으로 채워집니다. 별표(*)를 제외하고 값이 필드보다 길면 오류가 진단됩니다. 예를 들어, 다음은 모두 8개 문자 필드에서 올바른 문자열 값입니다.

'ABCDEFGH'	8자
'A*C*E*G*I'	별표를 제외한 5문자
'*A*C*E*G*I*K*M*O*'	별표를 제외한 8문자

- 공백, 소문자 또는 마침표(.), 정방향 슬래시(/), 밑줄(_) 및 퍼센트 부호(%) 이외의 특수 문자가 포함된 문자열을 작은따옴표로 묶으십시오. 작은따옴표로 묶이지 않은 소문자는 대문자로 접힙니다. 문자열에 따옴표가 포함되면, 두 개의 작은따옴표를 사용하여 따옴표 시작과 끝을 모두 나타내야 합니다. 문자열의 길이를 계산할 때, 큰따옴표는 각각 하나의 문자로 계수됩니다.

DLQ 규칙 테이블이 처리되는 방법

데드-레터 큐 핸들러는 패턴이 DLQ의 메시지와 일치하는 규칙에 대해 규칙 테이블을 검색합니다.

검색은 테이블의 첫 번째 규칙부터 시작되며 테이블 전체에서 순차적으로 계속됩니다. DLQ 핸들러가 일치하는 패턴을 가진 규칙을 찾을 때, 해당 규칙으로부터 조치를 취합니다. DLQ 핸들러는 해당 규칙에 적용될 때마다 규칙에 대한 재시도 수가 1씩 증가합니다. 첫 번째 시도에 실패한 경우, DLQ 핸들러는 RETRY 키워드에 지정되는 수와 시도 수가 일치할 때까지 다시 시도합니다. 모든 시도가 실패하면, DLQ 핸들러는 테이블에서 그 다음으로 일치하는 규칙을 검색합니다.

이 프로세스는 조치가 성공할 때까지 일치하는 후속 규칙에 대해 반복됩니다. 일치하는 각 규칙이 RETRY 키워드에 지정된 횟수만큼 시도되고 모든 시도가 실패하면, ACTION(IGNORE)이 가정됩니다. 일치하는 규칙을 찾지 못한 경우에도 ACTION(IGNORE)이 가정됩니다.

참고:

1. DLQ에서 MQDLH로 시작하는 메시지에 대해서만 일치하는 규칙 패턴을 찾습니다. MQDLH로 시작하지 않는 메시지는 정기적으로 오류가 있는 것으로 보고되며, 무기한 DLQ에 남게 됩니다.
2. 규칙이 조치로만 구성될 수 있도록, 모든 패턴 키워드는 기본값에 허용될 수 있습니다. 그러나 조치만으로 구성된 규칙은 MQDLH가 있고 테이블의 다른 규칙에 따라 아직 처리되지 않은 큐의 모든 메시지에 적용됩니다.
3. DLQ 핸들러가 시작될 때 규칙 테이블의 유효성이 검증되고 그 때 오류가 플래그됩니다. 언제든지 규칙 테이블로 변경할 수 있지만, DLQ 핸들러가 다시 시작할 때까지 해당 변경사항이 발효되지 않습니다.
4. DLQ 핸들러는 메시지, MQDLH 또는 메시지 디스크립터의 콘텐츠를 대체하지 않습니다. DLQ 핸들러는 메시지 옵션 MQPMO_PASS_ALL_CONTEXT를 사용하여 항상 다른 큐에 메시지를 넣습니다.
5. 규칙 테이블의 연속 구문 오류는 규칙 테이블이 유효성 검증 동안 반복적인 오류의 생성을 제거하도록 설계되기 때문에 인식되지 않을 수도 있습니다.
6. DLQ 핸들러는 MQOO_INPUT_AS_Q_DEF 옵션을 사용하여 DLQ를 엽니다.
7. DLQ 핸들러의 다중 인스턴스는 동일한 규칙 테이블을 사용하여 동일한 큐에 대해 동시에 실행될 수 있습니다. 그러나, 일반적으로 DLQ와 DLQ 핸들러 간에는 1:1 관계가 성립됩니다.

관련 개념

데드-레터 큐

관련 태스크

[미배달 메시지 문제점 해결](#)

 DLQ 핸들러 규칙 테이블 예

단일 제어 데이터 입력 항목 및 여러 규칙을 포함하는 **runmqdlq** 명령에 대한 데드-레터 큐 규칙 테이블의 예입니다.

```
*****
*   An example rules table for the runmqdlq command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)
```

```
* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2
```

```
DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

```
* The CCCC corporation considers itself very security
* conscios, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.
```

```
REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

관련 개념

[데드-레터 큐](#)

관련 태스크

[미배달 메시지 문제점 해결](#)

관련 참조

[runmqdlq\(데드-레터 큐 핸들러 실행\)](#)

IBM i에서 데드-레터 큐 핸들러 호출

IBM MQ for IBM i에서는 **STRMQMDLQ** 명령을 설정하여 DLQ 핸들러를 호출합니다.

시작하기 전에

DLQ 핸들러를 실행하기 위해서는 DLQ 자체 및 DLQ에 있는 메시지가 전달되는 모든 메시지 큐에 대해 권한이 있어야 합니다. 또한 메시지 컨텍스트에서 사용자 ID의 권한을 사용하여 큐에 메시지를 넣기 위해 DLQ에 대한 기타 사용자의 ID를 가정하기 위해서도 권한이 있어야 합니다.

참고: 대부분 메시지를 DLQ에 넣는 것을 피하는 것을 선호합니다. DLQ의 사용과 회피에 대한 정보는 [145 페이지](#)의 『데드-레터 큐에 대한 작업』의 내용을 참조하십시오.

이 태스크 정보

데드-레터 큐(DLQ)(때때로 미전달 메시지 큐로 불림)는 목적지 큐로 전달되지 못한 메시지에 대한 큐를 보유합니다. 네트워크 내의 모든 큐 관리자는 연관된 DLQ가 있어야 합니다.

큐 관리자, 메시지 채널 에이전트 및 애플리케이션은 DLQ에 메시지를 넣을 수 있습니다. DLQ에 대한 모든 메시지는 데드 레터 헤더 구조, MQDLH를 접두부로 사용합니다. 메시지는 항상 MQDLH를 가지는 메시지 채널 채널 에이전트 또는 큐 관리자에 의해 DLQ에 넣어집니다. DLQ에 메시지를 넣는 애플리케이션의 경우 MQDLH를 제공해야 합니다.

MQDLH 구조의 *Reason* 필드에는 메시지가 DLQ에 있는 이유를 식별하는 이유 코드가 포함됩니다.

모든 IBM MQ 환경에서, DLQ에 대한 메시지를 처리하기 위해 정기적으로 실행하는 루틴이 있어야 합니다. IBM MQ는 **STRMQMDLQ** 명령을 사용하여 호출하는 데드-레터 큐 핸들러 (DLQ 핸들러) 라는 기본 루틴을 제공합니다. 사용자 작성 규칙 테이블은 DLQ의 메시지를 처리하도록 DLQ 핸들러에 지시사항을 제공합니다. 즉, DLQ 핸들러는 DLQ의 메시지가 규칙 테이블의 입력 항목과 일치하는지 확인합니다. DLQ 메시지가 규칙 테이블의 입력 항목과 일치하면 DLQ 핸들러가 해당 입력 항목과 연관된 조치를 수행합니다.

프로시저

- DLQ 핸들러 호출

DLQ 핸들러를 호출하려면 **STRMQMDLQ** 명령을 사용하십시오. 처리하려는 DLQ 및 사용하려는 큐 관리자의 이름을 두 가지 방식으로 지정할 수 있습니다.

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

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QTXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- 규칙 테이블에서. 예를 들면, 다음과 같습니다.

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

참고: 규칙 테이블은 임의의 이름을 사용할 수 있는 소스 실제 파일 내의 멤버입니다.

이 예는 기본 큐 관리자가 소유하는 ABC1.DEAD.LETTER.QUEUE(이)라는 DLQ에 적용됩니다.

표시된 대로 DLQ 또는 큐 관리자를 지정하지 않으면, 설치를 위한 기본 큐 관리자가 해당 큐 관리자에 속하는 DLQ와 함께 사용됩니다.

관련 개념

데드-레터 큐

관련 태스크

[미배달 메시지 문제점 해결](#)

IBM i IBM i의 DLQ 핸들러 규칙 테이블

데드-레터 큐 핸들러 규칙 테이블은 DLQ 핸들러가 IBM i DLQ에 도착한 메시지를 처리하는 방법을 정의합니다.

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

- 표의 첫 번째 항목(선택적)에는 제어 데이터가 포함됩니다.
- 테이블의 기타 모든 입력 항목은 DLQ 핸들러가 따라야 할 규칙입니다. 각 규칙은 메시지가 일치되는 패턴(메시지 특성 세트)과, DLQ의 메시지가 지정된 패턴과 일치할 때 수행되는 조치로 구성됩니다. 규칙 테이블에는 최소 하나의 규칙이 있어야 합니다.

규칙 테이블의 각 입력 항목은 하나 이상의 키워드로 구성됩니다.

제어 데이터

이 절에서는 DLQ 핸들러 규칙 테이블의 제어 데이터 입력 항목에 포함시킬 수 있는 키워드에 대해 설명합니다. 다음에 유의하십시오.

- 키워드가 있는 경우 기본값은 밑줄 그어져 있습니다.
- 세로 선(|)은 대안을 분리합니다. 다음 중 하나만 지정할 수 있습니다.

- 모든 키워드는 선택사항입니다.

INPUTQ (QueueName| ' ' (기본값)

처리하려는 DLQ의 이름입니다.

1. 매개변수로 **STRMQMDLQ** 명령에 지정하는 UDLMMSGQ 값(또는 *DFT)은 규칙 테이블에서 INPUTQ 값을 대체합니다.
2. 매개변수로 **STRMQMDLQ** 명령에 공백 UDLMMSGQ 값을 지정하는 경우, 규칙 테이블의 INPUTQ 값이 사용됩니다.
3. 매개변수로 **STRMQMDLQ** 명령에 공백 UDLMMSGQ 값을 지정하고 규칙 테이블에 공백 INPUTQ 값을 지정하는 경우, 시스템 기본 데드-레터 큐가 사용됩니다.

INPUTQM (QueueManager이름| ' ' (기본값)

INPUTQ 키워드에 이름 지정된 DLQ를 소유하는 큐 관리자의 이름입니다.

큐 관리자를 지정하지 않거나 INPUTQM(' ')을 규칙 테이블에 지정하는 경우, 시스템이 설치를 위한 기본 큐 관리자를 사용합니다.

RETRYINT (간격| 60 (기본값)

DLQ 핸들러가 첫 번째 시도에서 처리될 수 없는 메시지를 DLQ에서 처리하려고 시도하고 반복되는 시도가 요청되는 간격(초)입니다. 기본적으로, 재시도 간격은 60초입니다.

WAIT (YES (기본값) |NO|nnn)

DLQ 핸들러가 처리할 수 있는 추가 메시지가 없다는 것을 감지할 때 DLQ에 도착할 추가 메시지를 기다려야 하는지 여부입니다.

YES

DLQ 핸들러가 무기한으로 대기하도록 합니다.

NO

DLQ가 비어 있거나 처리할 수 있는 메시지를 포함하지 않는다는 것을 감지할 때 DLQ 핸들러가 종료하도록 합니다.

nnn

DLQ 핸들러가 큐가 비어 있거나 처리할 수 있는 메시지가 없음을 감지한 후 새 작업이 종료하기 전에 도착하도록 nnn초 동안 대기하도록 합니다.

사용이 많은 DLQ의 경우 WAIT(YES)를 지정하고, 활동 레벨이 낮은 DLQ의 경우 WAIT(NO) 또는 WAIT(nnn)를 지정하십시오. DLQ 핸들러를 종료할 수 있는 경우, 트리거를 사용하여 이를 다시 호출하십시오.

규칙에서 제어 데이터를 포함하여 목록화하기 위한 대안으로서, **STRMQMDLQ** 명령에 입력 매개변수로서 DLQ의 이름을 제공할 수 있습니다. 임의의 값이 규칙 테이블 및 **STRMQMDLQ**에 대한 입력 모두에 지정되는 경우, **STRMQMDLQ** 명령에 지정된 값이 우선입니다.

참고: 제어 데이터 입력 항목이 규칙 테이블에 포함되는 경우 테이블의 첫 번째 입력 항목이어야 합니다.

IBM i IBM i에서 DLQ 규칙(패턴 및 조치)

IBM i 데드-레터 큐 규칙 각각에 대한 패턴 및 조치의 설명입니다.

다음은 DLQ 핸들러 규칙 테이블의 규칙 예입니다.

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +
ACTION (RETRY) RETRY (3)
```

이 규칙에 따라 DLQ 핸들러는 MQPUT 및 MQPUT1이 금지되기 때문에 DLQ에 있는 지속적인 메시지를 목적지 큐에 이동하도록 세 번 시도합니다.

이 섹션은 규칙에 포함할 수 있는 키워드를 설명합니다. 다음에 유의하십시오.

- 키워드가 있는 경우 기본값은 밑줄 그어져 있습니다. 대부분의 키워드의 경우 기본값이 *(별표)이며 어느 값과도 일치합니다.
- 세로 선(|)은 대안을 분리합니다. 다음 중 하나만 지정할 수 있습니다.
- 모든 키워드 예외 ACTION은 선택사항입니다.

이 섹션은 DLQ의 메시지가 일치되는 패턴 일치 키워드의 설명으로 시작됩니다. 그런 다음, 조치 키워드에 대해 설명합니다(DLQ 핸들러가 일치 메시지를 처리하는 방법을 판별하는 것).

IBM i IBM i에서 DLQ 패턴 일치 키워드

패턴 일치 키워드는 다음 예제에서 설명됩니다. 이 키워드를 사용하여 IBM i 데드-레터 큐의 메시지가 일치하는지 확인할 값을 지정할 수 있습니다. 모든 패턴 일치 키워드는 선택사항입니다.

APPLIDAT (*ApplIdentity*데이터|* (기본값))

메시지 디스크립터 MQMD에 지정된 DLQ에 있는 메시지의 *ApplIdentityData* 값입니다.

APPLNAME (*PutAppl*이름|* (기본값))

DLQ에 메시지의 메시지 디스크립터 MQMD의 *PutApplName* 필드에 지정된 것처럼, MQPUT 또는 MQPUT1 호출을 실행한 애플리케이션의 이름입니다.

APPLTYPE (*PutAppl*유형|* (기본값))

DLQ에 있는 메시지의 메시지 디스크립터 MQMD에 지정된 *PutApplType* 값입니다.

DESTQ (*QueueName*|* (기본값))

메시지의 목적지로 지정된 메시지 큐의 이름입니다.

DESTQM (*QueueManager*이름|* (기본값))

메시지의 목적지로 지정된 메시지 큐의 큐 관리자 이름입니다.

FEEDBACK (*Feedback*|* (기본값))

MsgType 값이 MQMT_REPORT일 때, *Feedback*에 보고서의 성질에 대해 설명합니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQFB_COA를 사용하여 목적지 큐에서 도착을 확인해야 하는 DLQ의 메시지를 식별할 수 있습니다.

FORMAT (형식|* (기본값))

메시지 송신자가 메시지 데이터 형식을 설명하는 데 사용하는 이름입니다.

MSGTYPE (*MsgType*|* (기본값))

DLQ에 있는 메시지의 메시지 유형입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQMT_REQUEST를 사용하여 응답이 필요한 DLQ의 해당 메시지를 식별할 수 있습니다.

PERSIST (*Persistence*|* (기본값))

메시지의 지속성 값입니다. (메시지의 지속성은 큐 관리자 재시작 후에도 지속되는지 여부를 판별합니다.)

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQPER_PERSISTENT를 사용하여 DLQ에 있는 지속적인 메시지를 식별할 수 있습니다.

REASON (*ReasonCode*|* (기본값))

메시지가 DLQ에 놓여진 이유에 대해 설명하는 이유 코드입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQRC_Q_FULL을 사용하여 목적지 큐가 가득 찼기 때문에 DLQ에 배치된 메시지를 식별할 수 있습니다.

REPLYQ (*QueueName*|* (기본값))

DLQ에 있는 메시지의 메시지 디스크립터 MQMD에 지정된 응답 대상 큐 이름입니다.

REPLYQM (*QueueManager*이름|* (기본값))

REPLYQ 키워드에서 지정된 응답 대상 큐의 큐 관리자 이름입니다.

USERID (*UserIdentifier*|* (기본값))

메시지 디스크립터 MQMD에 지정된 대로 DLQ에서 메시지를 생성한 사용자의 사용자 ID입니다.

IBM i IBM i에서 DLQ 조치 키워드

이러한 데드-레터 큐 조치 키워드를 사용하여 IBM i 데드-레터 큐의 일치 메시지가 처리되는 방법을 판별하십시오.

ACTION (DISCARD|IGNORE|RETRY|FWD)

이 규칙에서 정의된 패턴과 일치하는 DLQ의 메시지에 대해 수행된 조치입니다.

DISCARD

DLQ에서 메시지를 삭제합니다.

IGNORE

메시지가 DLQ에 보관되도록 합니다.

RETRY

DLQ 핸들러가 목적지 큐에 메시지를 다시 넣으려고 합니다.

FWD

DLQ 핸들러가 FWDQ 키워드에서 이름이 지정된 큐로 메시지를 전달합니다.

ACTION 키워드를 지정해야 합니다. 조치를 구현하기 위해 시도한 횟수 RETRY 키워드에 의해 제어됩니다. 제어 데이터의 RETRYINT 키워드는 시도 사이의 간격을 제어합니다.

FWDQ (QueueName|&DESTQ|&REPLYQ)

ACTION 키워드를 선택할 때 메시지가 전달되는 메시지 큐의 이름입니다.

QueueName

메시지 큐의 이름입니다. FWDQ(' ')가 올바르지 않습니다.

&DESTQ

MQDLH 구조의 *DestQName* 필드에서 큐 이름을 가져옵니다.

&REPLYQ

메시지 디스크립터(MQMD)의 *ReplyToQ* 필드에서 큐 이름을 가져옵니다.

FWDQ (&REPLYQ)를 지정하는 규칙이 메시지를 비어 있는 *ReplyToQ* 필드에 일치시킬 때 오류 메시지를 방지하도록 메시지 패턴에 REPLYQ(?)를 지정할 수 있습니다.

FWDQM (QueueManager이름| & DESTQM | & REPLYQM | '' (기본값))

메시지가 전달되는 큐의 큐 관리자입니다.

QueueManagerName

ACTION(FWD) 키워드를 선택할 때 메시지가 전달되는 큐에 대한 큐 관리자 이름입니다.

&DESTQM

MQDLH 구조의 *DestQMGrName* 필드에서 큐 관리자 이름을 가져옵니다.

&REPLYQM

메시지 디스크립터(MQMD)의 *ReplyToQMGr* 필드에서 큐 관리자 이름을 가져옵니다.

''

기본값인 FWDQM(' ')은 로컬 큐 관리자를 식별합니다.

HEADER (YES (기본값) |NO)

MQDLH가 ACTION(FWD)이 요청된 메시지에 남아 있어야 하는지 여부를 지정합니다. 기본적으로 MQDLH는 메시지에 남아 있습니다. HEADER 키워드는 FWD 이외의 다른 조치에는 유효하지 않습니다.

PUTAUT (DEF (기본값) | CTX)

DLQ 핸들러가 메시지를 넣을 때 필요한 권한입니다.

DEF

DLQ 핸들러 자체의 권한으로 메시지를 넣습니다.

CTX

메시지 컨텍스트에서 사용자 ID의 권한을 사용하여 메시지를 넣습니다. PUTAUT(CTX)를 지정한 경우 다른 사용자의 ID를 가정할 수 있는 권한이 있어야 합니다.

RETRY (RetryCount|1 (기본값))

조치를 시도하기 위한 횟수(1 - 999,999,999 범위)(제어 데이터의 RETRYINT 키워드에 지정되는 간격)

참고: DLQ 핸들러가 특정 규칙을 구현하기 위해 시도한 횟수는 DLQ 핸들러의 현재 인스턴스에 따라 다릅니다. 재시작하는 동안에는 계수가 지속되지 않습니다. DLQ 핸들러를 재시작하면 규칙을 적용하기 위해 시도한 횟수가 0으로 재설정됩니다.

IBM i 데드-레터 큐 규칙 테이블은 해당 구분, 구조 및 콘텐츠에 관하여 다음 규약을 지켜야 합니다.

- 규칙 테이블에는 최소 하나의 규칙이 있어야 합니다.
- 키워드는 임의의 순서로 발생할 수 있습니다.
- 키워드는 모든 규칙에 한 번만 포함될 수 있습니다.
- 키워드는 대소문자를 구분하지 않습니다.
- 키워드 및 매개변수 값은 최소 하나의 공백이나 쉼표로 다른 키워드와 구분해야 합니다.
- 규칙의 시작 또는 끝 부분과 키워드, 구두점 및 값 사이에 여러 개의 공백이 있을 수 있습니다.
- 각 규칙은 새 행에서 시작해야 합니다.
- 이식성의 경우, 행의 중요한 길이는 72자를 넘어서는 안 됩니다.
- 행에서 마지막 공백이 아닌 문자로서 더하기 부호(+)를 사용하여 다음 행에서 첫 번째 공백이 아닌 문자에서 규칙이 계속되는지 표시하십시오. 행에서 마지막 공백이 아닌 문자로서 빼기 부호(-)를 사용하여 다음 행의 시작에서 규칙이 계속되는지 표시하십시오. 키워드 및 매개변수 내에서 연속 문자가 발생할 수 있습니다.

예를 들면, 다음과 같습니다.

```
APPLNAME(' ABC+
D')
```

결과: 'ABCD'.

```
APPLNAME(' ABC-
D')
```

결과: 'ABC D'.

- 별표(*)로 시작되는 주석 행은 규칙 테이블에서 어디에나 발생할 수 있습니다.
- 빈 줄은 무시됩니다.
- DLQ 핸들러 규칙 테이블의 각 입력 항목은 하나 이상의 키워드 및 연관된 매개변수로 구성됩니다. 매개변수는 다음 구분 규칙을 따라야 합니다.
 - 각 매개변수 값은 최소 하나의 유효 문자를 포함해야 합니다. 작은따옴표로 묶인 값을 분리하는 따옴표는 중요한 것으로 여겨지지 않습니다. 예를 들어, 올바른 매개변수는 다음과 같습니다.

FORMAT(' ABC')	3개의 유효 문자
FORMAT(ABC)	3개의 유효 문자
FORMAT(' A')	1개의 유효 문자
FORMAT(A)	1개의 유효 문자
FORMAT(' ')	1개의 유효 문자

이러한 매개변수는 중요한 문자를 포함하지 않으므로 올바르지 않습니다.

```
FORMAT(' ')
FORMAT( )
FORMAT()
FORMAT
```

- 와일드카드 문자가 지원됩니다. 후미 문자 공백을 제외하고 단일 문자 대신 물음표(?)를 사용할 수 있습니다. 0개 이상의 인접 문자 대신 별표(*)를 사용할 수 있습니다. 별표(*) 및 물음표(?)는 매개변수 값에서 항상 와일드카드 문자로 해석됩니다.

- ACTION, HEADER, RETRY, FWDQ, FWDQM 및 PUTAUT 키워드의 매개변수에는 와일드카드 문자를 포함할 수 없습니다.
- 매개변수 값과 DLQ 메시지의 해당 필드에 있는 후미 공백은 와일드카드 일치룰 수행할 때 유의미하지 않습니다. 그러나 따옴표로 묶인 문자열 내에 있는 선두 공백과 임베드된 공백은 와일드카드 일치에서 유의미합니다.
- 숫자 매개변수는 물음표(?) 와일드카드 문자를 포함할 수 없습니다. 전체 숫자 매개변수 대신 별표(*)를 사용할 수 있지만, 별표를 숫자 매개변수의 일부로서 사용할 수 없습니다. 예를 들어, 다음은 올바른 숫자 매개변수입니다.

MSGTYPE(2) 응답 메시지만 적합합니다.
 MSGTYPE(*) 모든 메시지 유형이 적합합니다.
 MSGTYPE('*') 모든 메시지 유형이 적합합니다.

그러나 MSGTYPE('2*')은(는) 숫자 매개변수의 일부로 별표(*)를 포함하므로 유효하지 않습니다.

- 숫자 매개변수는 범위 0-999 999 999에 있어야 합니다. 매개변수 값이 이 범위에 있으면 키워드와 관련된 필드에서 현재 그 값이 올바르지 않더라도 허용됩니다. 숫자 매개변수에 기호 이름을 사용할 수 있습니다.
- 문자열 값이 키워드와 관련된 MQDLH 또는 MQMD에 있는 필드보다 짧으면, 필드 길이에 맞게 값이 공백으로 채워집니다. 별표(*)를 제외하고 값이 필드보다 길면 오류가 진단됩니다. 예를 들어, 다음은 모두 8개 문자 필드에서 올바른 문자열 값입니다.

'ABCDEFGH' 8자
 'A*C*E*G*I' 별표를 제외한 5문자
 '*A*C*E*G*I*K*M*O*' 별표를 제외한 8문자

- 공백, 소문자 또는 마침표(.), 슬래시(/), 밑줄(_) 및 퍼센트 부호(%) 이외의 특수 문자를 포함하는 문자열은 작은따옴표로 묶여야 합니다. 따옴표로 묶지 않은 소문자는 대문자로 변경됩니다. 문자열에 따옴표가 포함되는 경우, 두 개의 작은따옴표는 따옴표의 시작과 끝을 표시하는 데 사용되어야 합니다. 문자열의 길이를 계산할 때, 큰따옴표는 각각 하나의 문자로 계수됩니다.

IBM i IBM i에서 DLQ 규칙 테이블이 처리되는 방법

데드-레터 큐 핸들러는 IBM i 데드-레터 큐의 메시지와 일치하는 패턴을 가진 규칙에 대해 규칙 테이블을 검색합니다.

검색은 테이블의 첫 번째 규칙부터 시작되며 테이블 전체에서 순차적으로 계속됩니다. 일치하는 패턴이 있는 규칙이 발견되면, 규칙 테이블은 해당 규칙에서 조치를 시도합니다. DLQ 핸들러가 해당 규칙을 적용하려고 시도할 때마다 규칙에 대한 재시도 수가 1씩 증가합니다. 첫 번째 시도가 실패하면 시도 횟수가 RETRY 키워드에서 지정된 수와 일치할 때까지 시도가 반복됩니다. 모든 시도가 실패하면, DLQ 핸들러는 테이블에서 그 다음으로 일치하는 규칙을 검색합니다.

이 프로세스는 조치가 성공할 때까지 일치하는 후속 규칙에 대해 반복됩니다. 일치하는 각 규칙이 RETRY 키워드에 지정된 횟수만큼 시도되고 모든 시도가 실패하면, ACTION(IGNORE)이 가정됩니다. 일치하는 규칙을 찾지 못한 경우에도 ACTION(IGNORE)이 가정됩니다.

참고:

1. DLQ에서 MQDLH로 시작하는 메시지에 대해서만 일치하는 규칙 패턴을 찾습니다. MQDLH로 시작하지 않는 메시지는 정기적으로 오류가 있는 것으로 보고되며, 무기한 DLQ에 남게 됩니다.
2. 모든 패턴 키워드는 기본값일 수 있으므로 규칙은 조치로만 구성될 수 있습니다. 그러나 조치만으로 구성된 규칙은 MQDLH가 있고 테이블의 다른 규칙에 따라 아직 처리되지 않은 큐의 모든 메시지에 적용됩니다.
3. 규칙 테이블은 DLQ 핸들러가 시작될 때 유효성 검증되며, 그 시간에 오류 플래그가 지정됩니다. (DLQ 핸들러가 발행한 오류 메시지가 메시지 및 이유 코드에서 설명됩니다.) 언제든지 규칙 테이블로 변경할 수 있지만, DLQ 핸들러가 다시 시작될 때까지 해당 변경사항이 적용되지 않습니다.
4. DLQ 핸들러는 MQDLH 또는 메시지 디스크립터의 메시지 콘텐츠를 대체하지 않습니다. DLQ 핸들러는 항상 메시지 옵션 MQPMO_PASS_ALL_CONTEXT를 사용하여 다른 큐에 메시지를 넣습니다.

5. 규칙 테이블의 유효성 검사가 반복 오류의 생성을 제거하기 때문에, 규칙 테이블의 연속적 구문 오류는 인지되지 않을 수도 있습니다.
6. DLQ 핸들러는 **MQOO_INPUT_AS_Q_DEF** 옵션을 사용하여 DLQ를 엽니다.
7. DLQ 핸들러의 다중 인스턴스는 동일한 규칙 테이블을 사용하여 동일한 큐에 대해 동시에 실행될 수 있습니다. 그러나, 일반적으로 DLQ와 DLQ 핸들러 간에는 1:1 관계가 성립됩니다.

IBM i IBM i의 예제 DLQ 핸들러 규칙 테이블

IBM i에서 데드-레터 큐 핸들러 규칙 테이블의 예제 코드입니다. 이 예제 규칙 테이블에는 단일 제어 데이터 항목 및 여러 규칙이 있습니다.

```

*****
*   An example rules table for the STRMQMDLQ command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* STRMQMDLQ, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to STRMQMDLQ,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation is always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never does things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it must be able
* to cope with the message being lost, so we can afford to
* discard the message.

```

```
PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)

* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We do not have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.

REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

모든 DLQ 메시지가 처리되었는지 확인

데드-레터 큐 핸들러는 보이지만 제거되지 않은 DLQ의 모든 메시지 레코드를 유지합니다. DLQ에 가능한 적은 수의 메시지가 포함되어 있는지 확인하십시오.

이 태스크 정보

DLQ 핸들러를 필터로 사용하여 DLQ에서 메시지의 작은 서브세트를 추출하는 경우, DLQ 핸들러는 처리하지 않은 해당 메시지의 레코드를 DLQ에 계속 보관합니다. 또한 DLQ 핸들러는 DLQ가 FIFO(First-In-First-Out)로 정의된 경우에도 DLQ에 도착하는 새 메시지를 볼 수 있다고 보장할 수 없습니다. 큐가 비어있지 않으면, DLQ는 모든 메시지를 검사하기 위해 주기적으로 재검색됩니다.

이러한 이유로, DLQ가 가능한 한 적은 메시지를 포함하는지 확인해야 합니다. 어떤 이유로든 제거하거나 다른 큐로 전달할 수 없는 메시지가 큐에 누적되도록 허용하면 DLQ 핸들러의 워크로드가 증가하고 DLQ 자체가 가득 찰 위험이 있습니다.

DLQ 핸들러가 DLQ를 비울 수 있도록 하려면 다음 조치를 수행하십시오.

프로시저

- 그렇지 않으면 무시할 메시지의 경우 메시지를 다른 큐로 이동하는 조치를 사용하십시오.

DLQ에 메시지를 남기는 **ACTION (IGNORE)** 명령을 사용하지 마십시오. 테이블의 다른 규칙에 의해 명시적으로 주소 지정되지 않은 메시지에 대해서는 **ACTION (IGNORE)** 가 가정됩니다. 대신 메시지를 다른 큐로 이동하는 조치를 사용하십시오. 예를 들면, 다음과 같습니다.

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

- 테이블의 마지막 규칙을 "catch-all" 로 설정하여 테이블의 이전 규칙으로 처리되지 않은 메시지를 처리하십시오.

예를 들어, 테이블의 최종 규칙은 다음과 같을 수 있습니다.

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

테이블의 최종 규칙에 해당하는 메시지를 수동으로 처리할 수 있는 큐 REALLY.DEAD.QUEUE에 전달합니다. 이러한 규칙이 없으면 메시지가 DLQ에 무기한 남아 있을 가능성이 높습니다.

관리 토픽에 대한 작업

MQSC 명령을 사용하여 관리 토픽을 관리하십시오.

이러한 명령에 대한 자세한 정보는 [MQSC 명령](#)을 참조하십시오.

관련 개념

[관리 토픽 오브젝트](#)

관련 태스크

[163 페이지의 『관리 토픽 정의』](#)

MQSC 명령 **DEFINE TOPIC**을 사용하여 관리 토픽을 작성하십시오. 관리 토픽을 정의할 때, 각 토픽 속성을 선택적으로 설정할 수 있습니다.

[164 페이지의 『관리 토픽 오브젝트 속성 표시』](#)

MQSC 명령 **DISPLAY TOPIC**을 사용하여 관리 토픽 오브젝트를 표시하십시오.

[164 페이지의 『관리 토픽 속성 변경』](#)

ALTER TOPIC 명령 또는 **DEFINE TOPIC** 명령을 **REPLACE** 속성과 함께 사용하여 두 가지 방법으로 토픽 속성을 변경할 수 있습니다.

[165 페이지의 『관리 토픽 정의 복사』](#)

DEFINE 명령에서 **LIKE** 속성을 사용하여 토픽 정의를 복사할 수 있습니다.

[166 페이지의 『관리 토픽 정의 삭제』](#)

MQSC 명령 **DELETE TOPIC**을 사용하여 관리 토픽을 삭제할 수 있습니다.

관리 토픽 정의

MQSC 명령 **DEFINE TOPIC**을 사용하여 관리 토픽을 작성하십시오. 관리 토픽을 정의할 때, 각 토픽 속성을 선택적으로 설정할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

명백하게 설정되지 않는 토픽의 속성도 시스템 설치가 설치되었을 때 작성된 기본 관리 토픽, SYSTEM.DEFAULT.TOPIC으로부터 상속됩니다.

예

예를 들어, 뒤따르는 **DEFINE TOPIC** 명령은 ORANGE.TOPIC이라는 토픽을 이러한 특성으로 정의합니다.

- 토픽 문자열 ORANGE로 해결됩니다. 토픽 문자열을 사용하는 방법에 대한 정보는 [토픽 문자열 결합](#)을 참조하십시오.
- ASPARENT로 설정되는 속성은 이 토픽의 상위 토픽에 의해 정의된 대로 속성을 사용합니다. 이 조치는 루트 토픽(SYSTEM.BASE.TOPIC)이 발견되는 한 토픽 트리가 반복됩니다. 자세한 정보는 [토픽 트리](#)를 참조하십시오.

```
DEFINE TOPIC (ORANGE.TOPIC) +  
TOPICSTR (ORANGE) +  
DEFPRTY (ASPARENT) +  
NPMGDLV (ASPARENT)
```

참고:

- 토픽 문자열의 값을 제외하고, 표시되는 모든 속성값은 기본값입니다. 이는 설명으로만 여기에 표시됩니다. 기본값이 원하는 것이거나 변경되지 않는 것이라고 확인하는 경우 생략할 수 있습니다. [164 페이지의 『관리 토픽 오브젝트 속성 표시』](#)도 참조하십시오.
- 이름 ORANGE.TOPIC으로 동일한 큐 관리자에 이미 관리 토픽이 있는 경우, 이 명령은 실패합니다. 기존 토픽 정의를 겹쳐쓰려면 REPLACE 속성을 사용하십시오. [164 페이지의 『관리 토픽 속성 변경』](#)의 내용을 참조하십시오.

관련 참조

[DEFINE TOPIC](#)

관리 토픽 오브젝트 속성 표시

MQSC 명령 **DISPLAY TOPIC**을 사용하여 관리 토픽 오브젝트를 표시하십시오.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

예

이 명령은 모든 주제를 표시합니다.

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

DISPLAY TOPIC 명령으로 속성을 개별적으로 지정하여 선택적으로 표시할 수 있습니다. 예를 들면, 다음과 같습니다.

```
DISPLAY TOPIC(ORANGE.TOPIC) +  
TOPICSTR +  
DEFPRTY +  
NPMSGDLV
```

이 명령은 세 개의 지정된 속성을 표시합니다.

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC)                                TYPE(LOCAL)  
TOPICSTR(ORANGE)                                   DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

런타임 시 사용되는 토픽 ASPARENT 값을 표시하려면 **DISPLAY TPSTATUS** 명령을 사용하십시오. 예를 들어, 다음을 사용하십시오.

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

명령은 다음 세부사항을 표시합니다.

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE)                                DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

관리 토픽을 정의할 때, 기본 관리 토픽에서 명시적으로 지정하지 않는 속성을 사용합니다. 이는 SYSTEM.DEFAULT.TOPIC이라고 합니다. 이러한 기본 속성이 무엇인지 보려면 다음 명령을 사용하십시오.

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

관련 참조

[DISPLAY TOPIC](#)

[DISPLAY TPSTATUS](#)

관리 토픽 속성 변경

ALTER TOPIC 명령 또는 **DEFINE TOPIC** 명령을 **REPLACE** 속성과 함께 사용하여 두 가지 방법으로 토픽 속성을 변경할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보 [다 관리IBM MQ MQSC 명령 사용](#).

예

예를 들어, ORANGE.TOPIC가 5가 되려면 다음 명령 중 하나를 사용하십시오.

- **ALTER** 명령 사용:

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

이 명령은 하나의 속성, 즉 이 토픽에 전달된 메시지의 우선순위를 5로 변경합니다. 다른 모든 속성들은 그대로 남습니다.

- **DEFINE** 명령 사용:

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

이 명령은 이 토픽으로 전달된 메시지의 기본 우선순위를 변경합니다. 기타 모든 속성에는 기본값이 제공됩니다.

이 토픽에 보내진 메시지의 우선순위를 변경하면, 기존 메시지는 영향을 받지 않습니다. 그러나 새 메시지는 발행 애플리케이션에서 제공하지 않는 경우 지정된 우선순위를 사용합니다.

관련 참조

[ALTER TOPIC](#)

[DISPLAY TOPIC](#)

관리 토픽 정의 복사

DEFINE 명령에서 LIKE 속성을 사용하여 토픽 정의를 복사할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보 [다 관리IBM MQ MQSC 명령 사용](#).

예

다음 명령은 시스템 기본 관리 토픽의 속성이 아닌 원래 토픽 ORANGE.TOPIC와 동일한 속성을 사용하여 MAGENTA.TOPIC토픽을 작성합니다. 토픽을 작성했을 때 입력된 것처럼 정확하게 복사될 토픽의 이름을 입력하십시오. 이름이 소문자를 포함하면, 이름을 작은따옴표로 묶으십시오.

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

DEFINE 명령의 이 양식을 사용하여 토픽 정의를 복제할 수도 있지만, 원래 속성으로 변경하십시오. 예를 들면, 다음과 같습니다.

```
DEFINE TOPIC(BLUE.TOPIC) +  
TOPICSTR(BLUE) +  
LIKE(ORANGE.TOPIC)
```

또한 토픽 BLUE.TOPIC에서 토픽 GREEN.TOPIC의 속성을 복사하고 발행물이 해당 정확한 구독자 큐에 전달될 수 없을 때 데드-레터 큐 위에 배치되지 않는 것을 지정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
DEFINE TOPIC(GREEN.TOPIC) +  
TOPICSTR(GREEN) +
```

LIKE (BLUE.TOPIC) +
USEDLQ (NO)

관련 참조

[DEFINE TOPIC](#)

관리 토픽 정의 삭제

MQSC 명령 **DELETE TOPIC**을 사용하여 관리 토픽을 삭제할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

예

```
DELETE TOPIC (ORANGE.TOPIC)
```

애플리케이션이 발행을 위한 토픽을 더 이상 열 수 없거나 오브젝트 이름(ORANGE.TOPIC)을 사용하여 새 구독을 더 이상 작성하지 않습니다. 토픽을 열어 둔 발행 애플리케이션은 해석된 토픽 문자열을 계속 발행할 수 있습니다. 이미 이 토픽에 작성된 구독은 토픽이 삭제된 이후에도 계속 발행물을 수신합니다.

이 토픽 오브젝트를 참조하지 않지만 이 토픽 오브젝트가, 이 예에서 'ORANGE'로, 표시하는 해석된 토픽 문자열을 사용 중인 애플리케이션은 계속 작동합니다. 이 경우, 토픽 트리에 있는 더 높은 상위 토픽 오브젝트로부터 특성을 상속합니다. 자세한 정보는 [토픽 트리](#)를 참조하십시오.

관련 참조

[DELETE TOPIC](#)

구독에 대한 작업

MQSC 명령을 사용하여 구독을 관리하십시오.

이 태스크 정보

구독은 **SUBTYPE** 속성에서 정의되는 세 가지 유형 중 하나가 될 수 있습니다.

ADMIN

사용자에 의해 관리적으로 정의됩니다.

PROXY

큐 관리자 사이에 발행물을 라우팅하기 위해 내부적으로 작성된 구독.

API

예를 들어, MQI MQSUB 호출을 사용하여 프로그래밍 방식으로 작성됩니다.

이러한 명령에 대한 자세한 정보는 [MQSC 명령](#)을 참조하십시오.

관리 구독 정의

MQSC 명령 **DEFINE SUB**를 사용하여 관리 구독을 작성하십시오. 기본 로컬 구독 정의에 정의되는 기본값을 사용할 수도 있습니다. 또는, 시스템이 설치될 때 작성되는 기본 로컬 구독(SYSTEM.DEFAULT.SUB)의 특성에서 구독 특성을 수정할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

SUBID는 구독이 작성될 때 큐 관리자에 의해 지정되는 고유한 ID입니다. 이는 일부 구독 이름이 길거나 비실용적일 수 있는 다른 문자 세트에 있으므로 표시하기에 유용한 속성입니다.

구독을 표시하기 위한 대체 방법은 SUBID를 사용하는 것입니다.

```
DISPLAY SUB +
SUBID(414D51204141412020202020202020EE921E4E20002A03) +
TOPICSTR +
DURABLE
```

이 명령은 이전 것과 동일한 다음과 같은 출력을 제공합니다.

```
AMQ8096: IBM MQ subscription inquired.
SUBID(414D512041414120202020202020EE921E4E20002A03)
SUB(ORANGE)
DURABLE(YES)
TOPICSTR(ORANGE)
```

큐 관리자의 프록시 구독은 기본적으로 표시되지 않습니다. 이를 표시하려면 PROXY 또는 ALL의 **SUBTYPE**을 지정하십시오.

`DISPLAY SBSTATUS` 명령을 사용하여 런타임 속성을 표시할 수 있습니다. 예를 들어, 다음 명령을 사용하십시오.

```
DISPLAY SBSTATUS(ORANGE) NUMMSGS
```

다음 출력이 표시됩니다.

```
AMQ8099: IBM MQ subscription status inquired.
SUB(ORANGE)
SUBID(414D512041414120202020202020EE921E4E20002A03)
NUMMSGS(0)
```

관리 구독을 정의할 때, 기본 구독에서 명시적으로 지정하지 않는 속성을 사용하고 이는 `SYSTEM.DEFAULT.SUB`이라고 합니다. 이러한 기본 속성이 무엇인지 보려면 다음 명령을 사용하십시오.

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

관련 참조

[DISPLAY SUB](#)

로컬 구독 속성 변경

`ALTER SUB` 명령 또는 `DEFINE SUB` 명령을 `REPLACE` 속성과 함께 사용하여 구독 속성을 두 가지 방법으로 변경할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

예

ORANGE라는 구독에 전달되는 메시지의 우선순위를 5로 변경하려면 다음 명령 중 하나를 사용하십시오.

- **ALTER** 명령 사용:

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

이 명령은 하나의 속성, 즉 이 구독에 전달된 메시지의 우선순위를 5로 변경합니다. 다른 모든 속성들은 그대로 남습니다.

• DEFINE 명령 사용:

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

이 명령은 이 구독으로 전달된 메시지의 우선순위가 아니라, 기본값이 제공되는 기타 모든 속성을 변경합니다. 이 구독에 보내진 메시지의 우선순위를 변경하면, 기존 메시지는 영향을 받지 않습니다. 그러나, 새 메시지도 지정된 우선권을 가집니다.

관련 참조

[ALTER SUB](#)

[DEFINE SUB](#)

로컬 구독 정의 복사

DEFINE 명령에서 **LIKE** 속성을 사용하여 구독 정의를 복사할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

예

```
DEFINE SUB(BLUE) +  
LIKE(ORANGE)
```

또한 하위 REAL의 속성을 하위 THIRD.SUB에 복사하고 전달된 발행물의 **correlID** 이 발행자가 아닌 THIRD임을 지정할 수 있습니다 **correlID**. 예를 들면, 다음과 같습니다.

```
DEFINE SUB(THIRD.SUB) +  
LIKE(BLUE) +  
DESTCORL(ORANGE)
```

관련 참조

[DEFINE SUB](#)

로컬 구독 삭제

MQSC 명령 **DELETE SUB**을 사용하여 로컬 구독을 삭제할 수 있습니다.

시작하기 전에

참고: 이 태스크의 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

예

```
DELETE SUB(ORANGE)
```

SUBID를 사용하여 구독을 삭제할 수도 있습니다.

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

관련 참조

[DELETE SUB](#)

구독에서 메시지 검사

구독이 정의될 때 큐와 연관됩니다. 이 구독과 일치하는 발행된 메시지가 이 큐에 놓입니다. MQSC 명령을 사용하여 구독을 위해 현재 큐에 대기된 메시지를 확인할 수 있습니다.

시작하기 전에

참고: 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리 IBM MQ MQSC 명령 사용](#).

이 태스크 정보

다음 MQSC 명령은 메시지를 수신한 subscription만 표시합니다.

구독을 위해 최근에 큐되는 메시지를 검사하려면 다음 단계를 수행하십시오.

프로시저

1. 구독 유형 `DISPLAY SBSTATUS(sub_name) NUMMSGS`에 대해 큐에 대기된 메시지를 확인하려면 [167 페이지의 『구독의 속성 표시』](#)의 내용을 참조하십시오.
2. `NUMMSGS` 값이 0보다 더 크면 `DISPLAY SUB(sub_name)DEST`를 입력하여 구독과 연관된 큐를 식별하십시오.
3. 리턴된 큐의 이름을 사용하면서 [135 페이지의 『샘플 프로그램을 사용하여 큐 찾아보기』](#)에서 설명된 기술을 수행하여 메시지를 볼 수 있습니다.

관련 참조

[DISPLAY SBSTATUS](#)

서비스에 대한 작업

서비스 오브젝트는 추가 프로세스가 큐 관리자의 일부로서 관리될 수 있는 수단입니다. 서비스를 사용하면 큐 관리자가 시작되고 종료될 때 시작되고 중지되는 프로그램을 정의할 수 있습니다. IBM MQ 서비스는 큐 관리자를 시작한 사용자의 사용자 ID에서 항상 시작됩니다.

이 태스크 정보

서비스 오브젝트는 다음 유형 중 하나가 될 수 있습니다.

서버

서버는 `SERVER`로 지정된 매개변수 `SERVTYPE`을 가지는 서비스 오브젝트입니다. 서버 서비스 오브젝트는 지정된 큐 관리자가 시작될 때 실행되는 프로그램의 정의입니다. 서버 서비스 오브젝트는 일반적으로 오랜 기간 동안 실행되는 프로그램을 정의합니다. 예를 들어, 서버 서비스 오브젝트는 트리거 모니터 프로세스를 실행하기 위해 사용될 수 있습니다(예: `runmqtrm`).

서버 서비스 오브젝트의 하나의 인스턴스만 동시에 실행될 수 있습니다. 실행 중인 서버 서비스 오브젝트의 상태는 MQSC 명령(`DISPLAY SVSTATUS`)을 사용하여 모니터링될 수 있습니다.

명령

명령은 `COMMAND`로 지정된 매개변수 `SERVTYPE`을 가지는 서비스 오브젝트입니다. 명령 서비스 오브젝트는 서버 서비스 오브젝트와 유사하지만, 명령 서비스 오브젝트의 다중 인스턴스는 동시에 실행될 수 있고, 해당 상태가 MQSC 명령 `DISPLAY SVSTATUS`를 사용하여 모니터링될 수 없습니다.

MQSC 명령 `STOP SERVICE`를 실행하면 MQSC 명령 `START SERVICE`로 시작된 프로그램이 프로그램을 중지하기 전에 여전히 활성 상태인지 확인할 수 없습니다.

관련 참조

[DEFINE SERVICE](#)

[DISPLAY SVSTATUS](#)

[START SERVICE](#)

[STOP SERVICE](#)

서비스 오브젝트 정의

MQSC 명령 **DEFINE SERVICE**로 서비스 오브젝트를 정의합니다.

시작하기 전에

참고: 이 태스크를 수행하려면 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

프로시저

- MQSC 명령 **DEFINE SERVICE**를 사용하여 서비스 오브젝트를 정의하십시오.
정의해야 하는 속성은 다음과 같습니다.

SERVTYPE

서비스 오브젝트의 유형을 정의합니다. 가능한 값은 다음과 같습니다.

SERVER

서버 서비스 오브젝트.

한 번에 하나의 서버 서비스 오브젝트 인스턴스를 실행할 수 있습니다. 서버 서비스 오브젝트의 상태는 MQSC 명령 **DISPLAY SVSTATUS**를 사용하여 모니터링할 수 있습니다.

명령

명령 서비스 오브젝트.

명령 서비스 오브젝트의 다중 인스턴스가 동시에 실행될 수 있습니다. 명령 서비스 오브젝트의 상태는 모니터링될 수 없습니다.

STARTCMD

서비스를 시작하기 위해 실행되는 프로그램. 프로그램으로에 대한 완전한 경로가 지정되어야 합니다.

STARTARG

시작 프로그램으로 전달되는 인수.

STDERR

서비스 프로그램의 표준 오류(stderr)의 경로가 재지정되어야 하는 파일에 경로를 지정합니다.

STDOUT

서비스 프로그램의 표준 출력(stdout)의 경로가 재지정되어야 하는 파일에 경로를 지정합니다.

STOPCMD

서비스를 중지하기 위해 실행되는 프로그램. 프로그램으로에 대한 완전한 경로가 지정되어야 합니다.

STOPARG

중지 프로그램으로 전달되는 인수.

CONTROL

서비스가 시작되고 정지되는 방식을 지정합니다.

MANUAL

서비스가 자동으로 시작되거나 자동으로 중지되지 않습니다. **START SERVICE** 및 **STOP SERVICE** 명령을 사용하여 제어됩니다. 이는 기본값입니다.

QMGR

큐 관리자가 시작되고 중지되는 것과 동시에 정의한 서비스가 시작되고 중지됩니다.

STARTONLY

큐 관리자가 시작되는 것과 동시에 서비스가 시작되지만 큐 관리자가 중지될 때 중지가 요청되지 않습니다.

관련 태스크

172 페이지의 『서비스 관리』

서비스 오브젝트의 인스턴스는 큐 관리자에 의해 자동으로 시작 및 중지되거나 MQSC 명령 **START SERVICE** 및 **STOP SERVICE**를 사용하여 시작 및 중지될 수 있습니다.

관련 참조

[DEFINE SERVICE](#)

[DISPLAY SVSTATUS](#)

[START SERVICE](#)

[STOP SERVICE](#)

서비스 관리

서비스 오브젝트의 인스턴스는 큐 관리자에 의해 자동으로 시작 및 중지되거나 MQSC 명령 **START SERVICE** 및 **STOP SERVICE**를 사용하여 시작 및 중지될 수 있습니다.

시작하기 전에

참고: 이 태스크를 수행하려면 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 [관리IBM MQ MQSC 명령 사용](#).

프로시저

- 큐 관리자에서 **CONTROL** 매개변수를 설정하여 서비스 오브젝트의 인스턴스를 자동으로 시작 또는 중지하거나 MQSC 명령 **START SERVICE** 및 **STOP SERVICE** 를 사용하여 수동으로 이를 수행하십시오.

서비스 오브젝트의 인스턴스가 시작될 때, 서비스 오브젝트의 이름 및 시작된 프로세스의 프로세스 ID를 포함하는 큐 관리자 오류 로그로 메시지가 작성됩니다. 다음은 시작하는 서버 서비스 오브젝트에 대한 예제 로그 항목입니다.

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).

EXPLANATION:
The Server process has started.
ACTION:
None.
```

다음은 시작하는 명령 서비스 오브젝트에 대한 예제 로그 항목입니다.

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5030: The Command 'C1' has started. ProcessId(13030).

EXPLANATION:
The Command has started.
ACTION:
None.
```

인스턴스 서버 서비스가 중지될 때, 서비스의 이름 및 종료 프로세스의 프로세스 ID를 포함하는 큐 관리자 오류 로그에 메시지가 작성됩니다. 다음은 서버 서비스 오브젝트 중지에 대한 예제 로그 항목입니다.

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5029: The Server 'S1' has ended. ProcessId(13031).

EXPLANATION:
The Server process has ended.
ACTION:
None.
```

관련 태스크

[173 페이지의 『service.env 파일에서 추가 환경 변수 정의』](#)

서비스가 시작될 때, 서비스 프로세스가 시작되는 환경은 큐 관리자의 환경으로부터 상속됩니다. 정의하려는 변수를 `service.env` 환경 대체 파일 중 하나에 추가하여 서비스 프로세스의 환경에서 설정할 추가 환경 변수를 정의할 수 있습니다.

관련 참조

- [멀티플랫폼의 STOP SERVICE\(서비스 중지\)](#)
- [멀티플랫폼의 START SERVICE\(서비스 시작\)](#)

service.env 파일에서 추가 환경 변수 정의

서비스가 시작될 때, 서비스 프로세스가 시작되는 환경은 큐 관리자의 환경으로부터 상속됩니다. 정의하려는 변수를 `service.env` 환경 대체 파일 중 하나에 추가하여 서비스 프로세스의 환경에서 설정할 추가 환경 변수를 정의할 수 있습니다.

이 태스크 정보

환경 변수를 추가할 수 있는 두 개의 가능한 파일이 있습니다.

- 시스템 범위 `service.env` 파일
- 큐 관리자 범위 `service.env` 파일

가능한 경우, 시스템 범위 파일에서 해당 정의보다 우선순위에 서는 큐 관리자 범위 파일의 정의로 양쪽 모든 파일이 처리됩니다.

`service.env` 파일에서 환경 변수를 지정할 수 있습니다. 예를 들어, IBM MQ 서비스가 여러 명령을 실행하는 경우 `service.env` 파일에서 **PATH** 사용자 변수를 설정하는 것이 유용할 수 있습니다.

참고: 변수를 설정하는 값은 환경 변수가 될 수 없습니다. 예를 들어, `CLASSPATH= %CLASSPATH%` 는 올바르지 않습니다. 마찬가지로 Linux `PATH= $PATH :/opt/mqm/bin` 에서는 예기치 않은 결과가 발생할 수 있습니다.



CLASSPATH 는 대문자여야 하며 클래스 경로 명령문은 리터럴만 포함할 수 있습니다. 일부 서비스(예: 텔레메트리)는 자체 클래스 경로를 설정합니다. `service.env` 에 정의된 **CLASSPATH** 가 여기에 추가됩니다.

`service.env` 파일에 정의된 변수의 형식은 이름 및 값 변수 쌍의 목록입니다. 각 변수는 새 행에 정의되어야 하고 공백 문자를 포함하여 명백하게 정의된 것처럼 각 변수가 사용됩니다.

프로시저



- 시스템 범위 `service.env` 파일에 환경 변수를 추가하십시오.

이 파일은 다음 위치에 있습니다.

-  AIX and Linux 시스템의 `/var/mqm`
-  Windows 시스템에서 설치 동안 선택된 데이터 디렉토리입니다.

- 큐 관리자 범위 `service.env` 파일에 환경 변수를 추가하십시오.

이 파일은 큐 관리자 데이터 디렉토리에 있습니다. 예를 들어, QMNAME(이)라는 큐 관리자에 대한 환경 대체 파일의 위치는 다음과 같습니다.

-  AIX and Linux 시스템의 경우 `/var/mqm/qmgrs/QMNAME/service.env`
-  Windows 시스템의 경우 `씨:\ProgramData \IBM \MQ\qmgrs\QMNAME\service.env`

service.env 파일의 예제

```
#####  
##  
## <N_OCO_COPYRIGHT> ##  
## Licensed Materials - Property of IBM ##  
##
```



```

## 63H9336                                     ##
## (C) Copyright IBM Corporation 2005, 2024.   ##
##                                             ##
## <NOC_COPYRIGHT>                             ##
##                                             ##
#####
#####
## Module Name: service.env                     ##
## Type       : IBM MQ service environment file ##
## Function   : Define additional environment variables to be set ##
##           : for SERVICE programs.          ##
## Usage     : <VARIABLE>=<VALUE>             ##
##           :                                 ##
#####
MYLOC=/opt/myloc/bin
MYTMP=/tmp
TRACEDIR=/tmp/trace
MYINITQ=ACCOUNTS.INITIATION.QUEUE

```

관련 태스크

174 페이지의 『서비스 정의에서 대체 가능한 삽입 사용』

서비스 오브젝트의 정의에서 토큰을 대체할 수 있습니다. 대체되는 토큰은 서비스 프로그램이 실행될 때 확장된 텍스트로 자동적으로 대체됩니다.

관련 참조

[환경 변수 설명](#)

서비스 정의에서 대체 가능한 삽입 사용

서비스 오브젝트의 정의에서 토큰을 대체할 수 있습니다. 대체되는 토큰은 서비스 프로그램이 실행될 때 확장된 텍스트로 자동적으로 대체됩니다.

이 태스크 정보

대체 토큰은 다음과 같은 공통 토큰 목록 또는 `service.env` 파일에 정의된 변수에서 가져올 수 있습니다.

프로시저

- 대체 가능한 삽입을 사용하려면 **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** 또는 **STDERR** 문자열 중 하나에 + 문자 내 토큰을 삽입하십시오.

이 예에 대해서는 175 페이지의 『서버 서비스 오브젝트 사용』 및 177 페이지의 『명령 서비스 오브젝트 사용』의 내용을 참조하십시오.



다음은 서비스 오브젝트의 정의에서 토큰을 대체하기 위해 사용할 수 있는 공통 토큰입니다.

MQ_INSTALL_PATH

IBM MQ이 설치된 위치입니다.

MQ_DATA_PATH

IBM MQ 데이터 디렉토리의 위치입니다.

-  AIX and Linux 시스템에서 IBM MQ 데이터 디렉토리 위치는 `/var/mqm/`입니다.
-  Windows 시스템에서 IBM MQ 데이터 디렉토리의 위치는 IBM MQ의 설치 중에 선택된 데이터 디렉토리입니다.

QMNAME

현재 큐 관리자 이름입니다.

MQ_SERVICE_NAME

서비스 이름입니다.

MQ_SERVER_PID

이 토큰은 **STOPARG** 및 **STOPCMD** 인수에 의해서만 사용될 수 있습니다.

서버 서비스 오브젝트의 경우, 이 토큰은 **STARTCMD** 및 **STARTARG** 인수로 시작되는 프로세스의 프로세스 ID로 대체됩니다. 그렇지 않으면, 이 토큰은 0으로 대체됩니다.

MQ_Q_MGR_DATA_PATH

큐 관리자 데이터 디렉토리의 위치입니다.

MQ_Q_MGR_DATA_NAME

큐 관리자의 변환된 이름입니다. 이름 변환에 대한 자세한 정보는 [IBM MQ 파일 이름 이해](#)를 참조하십시오.

서버 서비스 오브젝트 사용

이 예는 트리거 모니터 또는 기타 프로그램을 시작하기 위해 서버 서비스 오브젝트를 정의, 사용 및 변경하는 방법을 보여줍니다.

시작하기 전에

참고: 이러한 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. [보다 관리 IBM MQ MQSC 명령 사용](#).

이 예제는 달리 명시되지 않는 한 UNIX 스타일 경로 구분 기호 문자로 작성됩니다.

프로시저

1. **DEFINE SERVICE MQSC** 명령을 사용하여 서버 서비스 오브젝트를 정의하십시오.

```
DEFINE SERVICE(S1) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+bin/runmqtrm') +
STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +
STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +
STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

여기서,

+MQ_INSTALL_PATH+는 설치 디렉토리를 나타내는 토큰입니다.

+QMNAME+은 큐 관리자의 이름을 표시하는 토큰입니다.

ACCOUNTS.INITIATION.QUEUE는 이니시에이션 큐입니다.

amqsstop은 큐 관리자가 프로세스 ID에 대한 모든 연결을 중단하도록 요청하는 IBM MQ와 함께 제공되는 샘플 프로그램입니다. amqsstop은 PCF 명령을 생성합니다. 따라서 명령 서버가 실행되어야 합니다.

+MQ_SERVER_PID+는 중지 프로그램에 전달된 프로세스 ID를 나타내는 토큰입니다.

공통 토큰의 목록의 경우 [174 페이지의 『서비스 정의에서 대체 가능한 삽입 사용』](#)의 내용을 참조하십시오.

2. 큐 관리자가 다음에 시작될 때 서버 서비스 오브젝트의 인스턴스가 실행됩니다. 그러나 **START SERVICE MQSC** 명령을 사용하여 서버 서비스 오브젝트의 인스턴스를 즉시 시작할 수 있습니다.

```
START SERVICE(S1)
```

3. **DISPLAY SVSTATUS MQSC** 명령을 사용하여 서버 서비스 프로세스의 상태를 표시하십시오.

```
DISPLAY SVSTATUS(S1)
```

4. 서버 서비스 오브젝트를 대체하고 **ALTER SERVICE MQSC** 명령을 사용하여 서버 서비스 프로세스를 수동으로 다시 시작하여 업데이트를 선택하십시오.

이니시에이션 큐가 JUPITER.INITIATION.QUEUE로 지정되도록 서버 서비스 오브젝트가 대체됩니다.

```
ALTER SERVICE(S1) +
STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

참고: 실행 중인 서비스는 다시 시작될 때까지 해당 서비스 정의에 대한 업데이트를 선택하지 않습니다.

5. **STOP SERVICE** 및 **START SERVICE MQSC** 명령을 사용하여 대체가 선택되도록 서버 서비스 프로세스를 다시 시작하십시오.

```
STOP SERVICE(S1)
```

그 후에 다음 명령을 입력하십시오.

```
START SERVICE(S1)
```

서버 서비스 프로세스가 재시작되고 175 페이지의 『4』에서 작성된 변경을 선택합니다.

참고: MQSC 명령(**STOP SERVICE**)은 **STOPCMD** 인수가 서비스 정의에 지정되는 경우에만 사용될 수 있습니다.

인수 전달의 추가 예제

- 큐 관리자가 시작될 때 **runserv** 라는 프로그램을 시작하도록 서버 서비스 오브젝트를 정의하십시오.

DEFINE SERVICE MQSC 명령을 사용하여 이를 수행하십시오.

이 예제는 Windows 스타일 경로 구분 기호 문자로 작성됩니다.

시작 프로그램에 전달되는 인수 중 하나는 공백을 포함하는 문자열입니다. 이 인수는 단일 문자열로서 전달되어야 합니다. 이를 수행하기 위해 다음 명령에 표시된 대로 큰따옴표를 사용하여 명령 서비스 오브젝트를 정의합니다.

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

```
DEFINE SERVICE(S4) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

여기서,

+QMNAME+은 큐 관리자의 이름을 표시하는 토큰입니다.

"C:\Program Files\Tools\"는 공백을 포함하는 문자열이며, 단일 문자열로서 전달됩니다.

- 큐 관리자가 시작될 때 트리거 모니터를 자동으로 시작하는 데 사용할 수 있는 서버 서비스 오브젝트를 정의하십시오.

DEFINE SERVICE MQSC 명령을 사용하여 이를 수행하십시오.

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtrm') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

여기서,

+QMNAME+은 큐 관리자의 이름을 표시하는 토큰입니다.

+IQNAME+은 이니시에이션 큐의 이름을 표시하는 service.env 파일 중 하나에서 사용자에게 의해 정의된 환경 변수입니다.

관련 참조

[ALTER SERVICE](#)

[DEFINE SERVICE](#)

DISPLAY SVSTATUS

START SERVICE

STOP SERVICE

명령 서비스 오브젝트 사용

이 예는 큐 관리자가 시작되거나 중지될 때 운영 체제의 시스템 로그에 입력 항목을 기록하는 프로그램을 시작하기 위해 명령 서비스 오브젝트를 정의하는 방법을 보여줍니다.

시작하기 전에

참고: 이러한 예에서는 **DEFINE SERVICE MQSC** 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. [보다 관리 IBM MQ MQSC 명령 사용](#).

이 예제는 UNIX 스타일 경로 구분 기호 문자로 작성됩니다.

이 태스크 정보

다음 예제에서:

logger는 운영 체제의 시스템 로그에 입력 항목을 작성할 수 있는 IBM MQ와 함께 제공되는 샘플 프로그램입니다.

+QMNAME+은 큐 관리자의 이름을 표시하는 토큰입니다.

프로시저

- 큐 관리자가 시작되거나 중지될 때 운영 체제의 시스템 로그에 항목을 기록하는 프로그램을 시작하도록 명령 서비스 오브젝트를 정의하십시오.

```
DEFINE SERVICE(S2) +  
CONTROL(QMGR) +  
SERVTYPE(COMMAND) +  
STARTCMD('/usr/bin/logger') +  
STARTARG('Queue manager +QMNAME+ starting') +  
STOPCMD('/usr/bin/logger') +  
STOPARG('Queue manager +QMNAME+ stopping')
```

- 큐 관리자가 중지된 경우에만 운영 체제의 시스템 로그에 항목을 기록하는 프로그램을 시작하도록 명령 서비스 오브젝트를 정의하십시오.

```
DEFINE SERVICE(S3) +  
CONTROL(QMGR) +  
SERVTYPE(COMMAND) +  
STOPCMD('/usr/bin/logger') +  
STOPARG('Queue manager +QMNAME+ stopping')
```

관련 참조

DEFINE SERVICE

트리거에 대한 오브젝트 관리

이 예제에서는 큐의 특정 조건이 충족될 때 애플리케이션을 자동으로 시작하는 방법을 보여줍니다. 예를 들어, 큐의 메시지 수가 지정된 수에 도달할 때 애플리케이션을 시작하려고 할 수 있습니다. 이 기능은 트리거라고 합니다. 트리거를 지원하는 오브젝트를 정의해야 합니다.

시작하기 전에

참고: 이러한 예에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. [보다 관리 IBM MQ MQSC 명령 사용](#).

이 예제는 UNIX 스타일 경로 구분 기호 문자로 작성됩니다.

이 태스크 정보

트리거에 대한 자세한 설명은 [트리거를 사용하여 IBM MQ 애플리케이션 시작을 참조하십시오.](#)

프로시저

- 트리거할 애플리케이션 큐를 정의하십시오.

애플리케이션 큐는 MQI를 통해 메시징을 위해 애플리케이션에서 사용하는 로컬 큐입니다. 트리거는 애플리케이션 큐에 정의되는 수많은 큐 속성이 필요합니다.

트리거 자체는 **Trigger** 속성 (MQSC 명령의 TRIGGER) 에 의해 사용으로 설정됩니다. 이 예제에서, 트리거 이벤트는 다음과 같이 로컬 큐 MOTOR.INSURANCE.QUEUE에 우선순위가 5 이상인 메시지가 100개가 있을 때 생성됩니다.

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +  
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
MAXMSGL (2000) +  
DEFPSIST (YES) +  
INITQ (MOTOR.INS.INIT.QUEUE) +  
TRIGGER +  
TRIGTYPE (DEPTH) +  
TRIGDPTH (100)+  
TRIGMPRI (5)
```

설명:

QLOCAL (MOTOR.INSURANCE.QUEUE)

정의되는 애플리케이션 큐의 이름입니다.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

트리거 모니터 프로그램에서 시작하는 애플리케이션을 정의하는 프로세스 정의의 이름입니다.

MAXMSGL (2000)

큐의 최대 길이의 메시지입니다.

DEFPSIST (YES)

이 큐의 메시지가 기본적으로 지속되는지 지정합니다.

INITQ (MOTOR.INS.INIT.QUEUE)

큐 관리자가 트리거 메시지를 배치할 이니시에이션 큐의 이름입니다.

TRIGGER

트리거 속성 값입니다.

TRIGTYPE (DEPTH)

필수 우선순위(TRIGMPRI)의 메시지 수가 TRIGDPTH에 지정된 수에 도달할 때 트리거 이벤트가 생성된다는 것을 지정합니다.

TRIGDPTH (100)

트리거 이벤트를 생성하기 위해 필요한 메시지 수입니다.

TRIGMPRI (5)

트리거 이벤트를 생성할지 여부를 결정하는 데 있어서 큐 관리자가 계수해야 하는 메시지의 우선순위입니다. 우선순위 5 이상인 메시지 수만 계수됩니다.

- 이니시에이션 큐 정의

트리거 이벤트가 발생할 때, 큐 관리자는 애플리케이션 큐 정의에 지정되는 이니시에이션 큐에 트리거 메시지를 배치합니다. 이니시에이션 큐에는 특수 설정이 없지만, 자세한 내용을 위해 로컬 큐 MOTOR.INS.INIT.QUEUE의 다음 정의를 사용할 수 있습니다.

```
DEFINE QLOCAL (MOTOR.INS.INIT.QUEUE) +  
GET (ENABLED) +  
NOSHARE +  
NOTRIGGER +  
MAXMSGL (2000) +  
MAXDEPTH (1000)
```

- 프로세스 정의

DEFINE PROCESS 명령을 사용하여 프로세스 정의를 작성하십시오. 프로세스 정의는 애플리케이션 큐에서 메시지를 처리하기 위해 사용되는 애플리케이션을 정의합니다. 애플리케이션 큐 정의는 사용될 프로세스의 이름을 지정합니다. 그렇게 함으로써 메시지를 처리하기 위해 사용될 애플리케이션과 애플리케이션 큐를 연관합니다. 이는 애플리케이션 큐 MOTOR.INSURANCE.QUEUE에서 PROCESS 속성을 통해 수행됩니다. 다음 MQSC 명령은 이 예에 식별된 필수 프로세스 MOTOR.INSURANCE.QUOTE.PROCESS를 정의합니다.

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

여기서,

MOTOR.INSURANCE.QUOTE.PROCESS

프로세스 정의의 이름입니다.

DESCR ('Insurance request message processing')

이 정의가 관련되는 애플리케이션 프로그램을 설명합니다. 이 텍스트는 DISPLAY PROCESS 명령을 사용할 때 표시됩니다. 이는 프로세스가 수행하는 것을 식별하는 데 도움이 될 수 있습니다. 문자열에서 공백을 사용하는 경우, 작은따옴표로 문자열을 묶어야 합니다.

APPLTYPE (UNIX)

시작할 애플리케이션의 유형입니다.

APPLICID ('/u/admin/test/IRMP01')

완전한 파일 이름으로 지정되는 애플리케이션 실행 가능 파일의 이름입니다. Windows 시스템에서 일반 APPLICID 값은 c:\appl\test\irmp01.exe입니다.

USERDATA ('open, close, 235')

애플리케이션에서 사용할 수 있는 사용자 정의 데이터입니다.

- 프로세스 정의의 속성 표시

DISPLAY PROCESS 명령을 사용하여 사용자 정의 결과를 조사하십시오. 예를 들면, 다음과 같습니다.

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

MQSC 명령 ALTER PROCESS를 사용하여 기존 프로세스 정의를 대체할 수 있고, DELETE PROCESS 명령을 사용하여 프로세스 정의를 삭제할 수도 있습니다.

두 시스템 사이에 dmpmqmsg 유틸리티 사용

dmpmqmsg 유틸리티 (이전의 *qload*) 를 사용하면 큐의 콘텐츠 또는 해당 메시지를 파일로 복사하거나 이동할 수 있습니다.

개요

dmpmqmsg 로 작성하는 파일은 필요에 따라 저장할 수 있으며 나중에 메시지를 다시 큐에 다시 로드하는 데 사용할 수 있습니다.

중요사항:

1. 파일은 유틸리티가 이해할 수 있는 특정 형식을 갖습니다. 그러나 이 파일은 사람이 읽을 수 있으므로 다시 로드하기 전에 편집기로 업데이트할 수 있습니다. 파일을 편집하는 경우 해당 형식을 변경해서는 안 됩니다.

2. **dmpmqmsg** 유틸리티는 AIX, Linux, and Windows용 런타임 파일 세트와 함께 제공되므로 IBM MQ 서버 및 클라이언트 모두에서 사용할 수 있습니다.

가능한 용도는 다음과 같습니다.

- 큐에 있는 메시지를 파일에 저장(아카이브 용도로 또는 나중에 큐에 다시 로드하기 위해)
- 이전에 파일에 저장한 메시지로 큐 다시 로드
- 큐에서 이전 메시지 제거
- 저장된 위치에서 테스트 메시지 '재실행' - 필요한 경우 메시지 간에 올바른 시간 유지



주의: SupportPac MO03은 로컬 또는 클라이언트 바인딩을 지정하기 위해 **-1** 매개변수를 사용했습니다. **-1**는 **-c** 매개변수로 대체되었습니다.

이제 코드 페이지 정보에는 **-c** 대신 **-P**가 사용됩니다.

명령 및 사용 가능 매개변수에 대한 추가 정보는 [dmpmqmsg](#)를 참조하십시오.

Windows 시스템을 사용하여 Linux에서 dmpmqmsg 유틸리티를 사용하는 예

동일한 큐 관리자의 다른 큐 (Q2) 로 이동하려는 큐 (Q1) 에 메시지가 있는 Linux 시스템에 큐 관리자가 있습니다. Windows 시스템에서 **dmpmqmsg** 유틸리티를 시작하려고 합니다.

큐(Q1)에는 샘플 **amqsput**(로컬 큐 관리자) 또는 **amqsputc**(리모트 큐 관리자) 애플리케이션을 사용하여 추가된 네 개의 메시지가 있습니다.

Linux 시스템의 경우 다음과 같이 표시됩니다.

```
display q1(Q1) CURDEPTH
      2 : display q1(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

MQSERVER 환경 변수가 Linux의 큐 관리자를 가리키도록 설정하십시오. 예를 들면, 다음과 같습니다.

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

여기서 *veracruz*는 시스템의 이름입니다.

큐, Q1에서 읽을 **dmpmqmsg** 유틸리티를 실행하고 `c:\temp\mqqload.txt`에 출력을 저장하십시오.

MQSERVER에 의해 설정된 Linux 호스트 및 포트에서 실행 중인 큐 관리자 `QM_VER`에 리모트 클라이언트로 연결하십시오. 원격 클라이언트로서 연결하려면 `-c` 속성을 사용해야 합니다.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqload.txt -c
Read      - Files:    0  Messages:    4  Bytes:          22
Written - Files:    1  Messages:    4  Bytes:          22
```

출력 파일 `c:\temp\mqqload.txt`에는 **dmpmqmsg** 유틸리티가 이해하는 형식을 사용하는 텍스트가 포함됩니다.

Windows 시스템에서 **dmpmqmsg** 명령 (`-i` 옵션 대신 `-o` 옵션 사용) 을 실행하여 Windows 시스템의 파일에서 Linux 시스템의 큐 (Q2) 를 로드하십시오.

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqload.txt -c
Read      - Files:    1  Messages:    4  Bytes:          22
Written - Files:    0  Messages:    4  Bytes:          22
```

Linux 시스템의 경우, 이제 파일에서 복원된 네 개의 메시지가 큐에 있음에 유의하십시오.

```
display q1(Q2) CURDEPTH
```



```
6 : display ql(Q2) CURDEPTH
AMQ8409: Display Queue details.
  QUEUE(Q2)
  TYPE(QLOCAL)
  CURDEPTH(4)
```

Linux 시스템의 경우,
원래 큐에서 메시지를 삭제하십시오.

```
clear qlocal(Q1)
4 : clear qlocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

원래 큐에 더 이상 메시지가 없는지 확인하십시오.

```
display ql(Q1) CURDEPTH
5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
  QUEUE(Q1)
  TYPE(QLOCAL)
  CURDEPTH(0)
```

명령 및 해당 매개변수에 대한 설명은 [dmpmqmsg](#)를 참조하십시오.

관련 개념

181 페이지의 『[dmpmqmsg 유틸리티 사용 예](#)』

dmpmqmsg 유틸리티 (이전에는 **qload**) 를 사용할 수 있는 간단한 방법입니다.

dmpmqmsg 유틸리티 사용 예

dmpmqmsg 유틸리티 (이전에는 **qload**) 를 사용할 수 있는 간단한 방법입니다.

파일에 큐 언로드

명령행에서 다음 옵션을 사용하면 큐에 있는 메시지가 파일에 저장됩니다.

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

이 명령은 큐에서 메시지 사본을 가져와 지정된 필드에 저장합니다.

일련의 파일에 큐 언로드

파일 이름에서 **insert** 문자를 사용하여 일련의 파일에 큐를 언로드할 수 있습니다. 이 모드에서는 각 메시지가 새 파일에 기록됩니다.

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

이 명령은 큐를 **myfile1**, **myfile2**, **myfile3** 등의 파일에 언로드합니다.

파일에서 큐 로드

181 페이지의 『[파일에 큐 언로드](#)』 에 저장한 메시지로 큐를 다시 로드하려면 명령행에서 다음 옵션을 사용하십시오.

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

이 명령은 큐를 **myfile1**, **myfile2**, **myfile3** 등의 파일에 언로드합니다.

일련의 파일에서 큐 로드

파일 이름에서 `insert` 문자를 사용하여 일련의 파일에서 큐를 로드할 수 있습니다. 이 모드에서는 각 메시지가 새 파일에 기록됩니다.

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

이 명령은 큐를 `myfile1`, `myfile2`, `myfile3` 등의 파일에 로드합니다.

특정 큐의 메시지를 다른 큐에 복사

181 페이지의 『[파일에 큐 언로드](#)』의 파일 매개변수를 다른 큐 이름으로 바꾸고 다음 옵션을 사용하십시오.

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

이 명령을 사용하면 특정 큐의 메시지를 다른 큐에 복사할 수 있습니다.

특정 큐의 첫 번째 100개 메시지를 다른 큐에 복사

이전 예제의 명령을 사용하고 `-r#100` 옵션을 추가하십시오.

```
dmpmqmsg -m QM1 -i Q1 -o Q2 -r#100
```

특정 큐의 메시지를 다른 큐로 이동

181 페이지의 『[파일에서 큐 로드](#)』의 변형입니다. 큐를 찾아보기만 하는 `-i` (소문자) 사용과 큐에서 파괴적으로 가져오는 `-I` (대문자) 사용 사이의 차이에 유의하십시오.

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

특정 큐에서 1일이 지난 메시지를 다른 큐로 이동

이 예는 기간 선택 사용 방법을 보여줍니다. 기간보다 오래되거나 오래되지 않았거나 기간 범위 내에 있는 메시지를 선택할 수 있습니다.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

현재 큐에 있는 메시지의 기간 표시

명령행에서 다음 옵션을 사용하십시오.

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

메시지 파일에 대한 작업

큐에서 메시지를 언로드하면(181 페이지의 『[파일에 큐 언로드](#)』 참조) 파일을 편집하려고 할 수 있습니다.

또한 큐 언로드 시점에 지정하지 않은 표시 옵션 중 하나를 사용하도록 파일 형식을 변경할 수 있습니다.

`dmpmqmsg` 유틸리티를 사용하면 큐를 언로드한 후에도 파일을 필요한 형식으로 재처리할 수 있습니다. 명령행에서 다음 옵션을 사용하십시오.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

명령 및 해당 매개변수에 대한 설명은 `dmpmqmsg`를 참조하십시오.

리모트 IBM MQ 오브젝트에 대한 작업

MQSC 명령, PCF 명령 또는 administrative REST API를 사용하여 리모트 큐 관리자에서 IBM MQ 오브젝트를 관리할 수 있습니다. 이러한 방법 중 하나를 사용하려면, 명령을 리모트 큐 관리자로 송신하고 로컬 큐 관리자에서 응답을 수신하도록 로컬 큐 관리자와 리모트 큐 관리자 간 전송 큐와 채널을 정의해야 합니다. 또는 큐 관리자 클러스터를 구성한 다음 동일한 원격 관리 방법을 사용할 수 있습니다.

이 태스크 정보

원격 관리를 위해 큐 관리자를 준비하려면 로컬 큐 관리자에서 다음 오브젝트를 구성해야 합니다.

- 리스너.
- 리모트 큐 관리자의 이름을 갖는 전송 큐.
- 리모트 큐 관리자에 대한 연결 세부사항이 있는 송신자 채널.
- 리모트 큐 관리자의 송신자 채널과 이름이 같은 수신자 채널.

또한 리모트 큐 관리자에서 다음 오브젝트를 구성해야 합니다.

- 리스너.
- 로컬 큐 관리자의 이름을 갖는 전송 큐.
- 로컬 큐 관리자에 대한 연결 세부사항이 있는 송신자 채널.
- 로컬 큐 관리자의 송신자 채널과 이름이 같은 수신자 채널.

이러한 오브젝트 구성에 대한 자세한 정보는 [183 페이지의 『원격 관리를 위한 큐 관리자 구성』](#)의 내용을 참조하십시오.

또는 큐 관리자 클러스터를 구성할 수 있습니다. 클러스터는 큐 관리자가 복합적 송신 큐와 채널과 큐 정의 없이 단일 네트워크 위에서 서로 서로 직접적으로 통신할 수 있는 이와 같은 방식으로 설치된 한 그룹의 큐 관리자입니다. 클러스터는 쉽게 설정할 수 있으며, 일반적으로 클러스터에는 논리적으로 관련되어 데이터나 응용프로그램을 공유해야 하는 큐 관리자가 포함되어 있습니다. 심지어 가장 작은 클러스터는 시스템 관리 비용을 줄입니다.

클러스터에서 큐 관리자 네트워크를 설정하면 일반적인 분산 큐잉 환경을 설정하는 것보다 정의가 적습니다. 정의를 적게 작성하면서도 네트워크를 보다 빠르고 쉽게 설정하거나 변경할 수 있으므로, 정의를 작성할 때 오류가 발생할 위험도 줄어듭니다.

클러스터를 설정하려면, 일반적으로 각 큐 관리자마다 하나의 클러스터 송신자(CLUSSDR)와 하나의 클러스터 수신자(CLUSRCVR) 정의가 필요합니다. 트랜스미션 큐 정의나 리모트 큐 정의는 필요 없습니다. 원격 관리 기본 원칙은 클러스터 내에서 사용할 때와 동일하지만, 정의 자체가 일반적으로 상당히 간소합니다.

클러스터 구성에 대한 자세한 정보는 [큐 관리자 클러스터 구성](#)을 참조하십시오.

프로시저

- 원격 IBM MQ 오브젝트를 관리하는 방법은 다음 하위 주제를 참조하십시오.
 - [183 페이지의 『원격 관리를 위한 큐 관리자 구성』](#)
 - [187 페이지의 『원격 관리를 위한 명령 서버 관리』](#)
 - [188 페이지의 『리모트 큐 관리자에서 MQSC 명령 실행』](#)
 - [189 페이지의 『코드화 문자 세트 간 데이터 변환』](#)

원격 관리를 위한 큐 관리자 구성

administrative REST API, MQSC 또는 PCF 명령을 사용하여 로컬 큐 관리자에서 리모트 큐 관리자를 관리할 수 있습니다. 리모트 큐 관리자는 동일한 시스템에 다른 설치로 있거나 다른 IBM MQ 환경이나 동일한 환경의 다른 시스템에 있을 수 있습니다. 로컬 큐 관리자에서 큐 관리자를 원격으로 관리하려면 먼저 각 큐 관리자에서 송신자 및 수신자 채널, 리스너 및 전송 큐를 작성해야 합니다. 이러한 채널 및 큐를 사용하면 리모트 큐 관리자로 명령을 송신하고 로컬 큐 관리자에서 응답을 수신할 수 있습니다. 이러한 큐 및 채널을 작성하는 프로시저는 administrative REST API, MQSC 또는 PCF 명령 중 어느 것을 사용할지에 관계없이 동일합니다.

시작하기 전에

- 다음 프로시저에서는 큐 관리자 예 `source.queue.manager` 및 `target.queue.manager`를 사용합니다. 시스템에서 이러한 큐 관리자를 작성하고 시작하여 이러한 단계를 수행하거나 관련 단계에서 고유 큐 관리자 이름을 대체하십시오.
- 다음 프로시저에서는 TCP/IP를 전송 유형으로 사용합니다. 이 태스크를 완료하려면 두 시스템의 IP 주소를 알아야 합니다.
- 다음 프로시저에서는 로컬 시스템의 네트워크 포트 1818과 원격 시스템의 1819를 사용하는 리스너를 작성합니다. 다른 포트를 사용할 수 있지만 해당 단계에서는 포트 값을 대체해야 합니다.
- Telnet과 같은 네트워크 기능을 통해 또는 로컬로 프로시저에서 명령을 실행해야 합니다.

이 태스크 정보

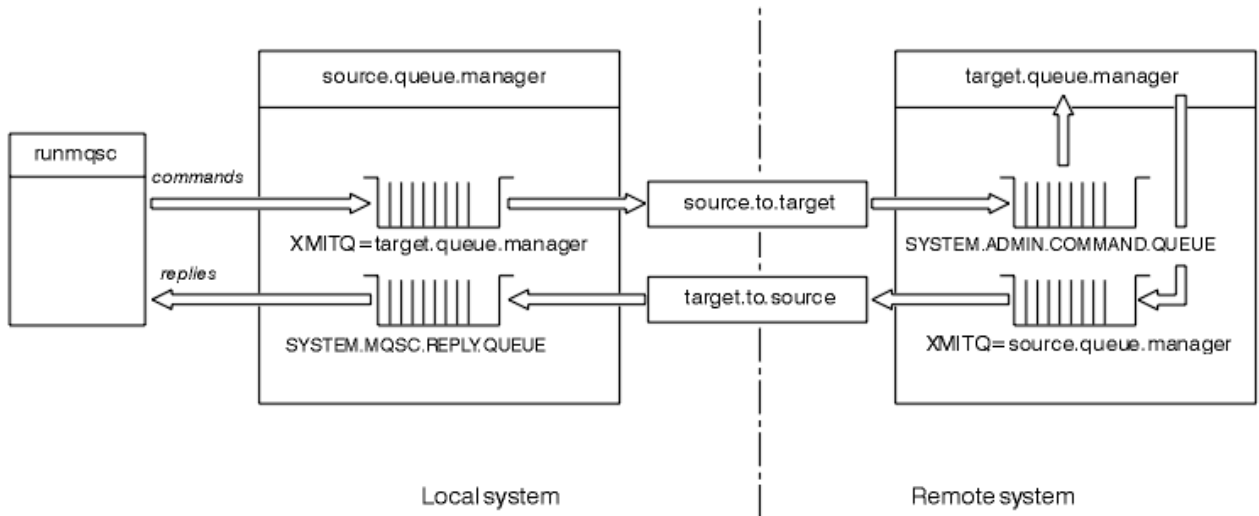


그림 15. 원격 관리를 위한 채널 및 큐 설정

184 페이지의 그림 15에서는 원격 관리에 필요한 큐 관리자, 큐 및 채널의 구성을 표시합니다.

- `source.queue.manager` 오브젝트는 administrative REST API, MQSC 또는 PCF 명령을 실행할 수 있고 이러한 명령의 결과가 리턴되는 소스 큐 관리자입니다.
- 오브젝트 `target.queue.manager`는 명령을 처리하고 운영자 메시지를 생성하는 대상 큐 관리자의 이름입니다.
- 리모트 큐 관리자와 동일한 이름의 전송 큐에 명령을 넣습니다. 이 경우 `target.queue.manager`입니다. 전송 큐는 MCA가 선택하고 리모트 큐 관리자로 이를 전송하기 전에 임시로 메시지를 보유하는 특별한 로컬 큐입니다.
- `source.to.target` 채널에서 리모트 큐 관리자의 `SYSTEM.ADMIN.COMMAND.QUEUE`로 명령을 송신합니다. 각 채널 끝에는 별도 정의가 있습니다. 하나의 끝은 송신자이고 다른 끝은 수신자입니다. 두 정의는 같은 이름을 가지고 함께 하나의 메시지 채널을 구성해야 합니다.
- 명령이 송신된 로컬 큐 관리자와 동일한 이름의 리모트 전송 큐에 명령 출력을 넣습니다. 이 경우 `source.queue.manager`입니다.
- 출력은 `target.to.source` 채널을 통해 해당 응답 큐로 송신되며, 원래 명령을 사용하여 가져오고 출력합니다.

프로시저

1. 원격 시스템 큐 관리자에서 명령 큐 `SYSTEM.ADMIN.COMMAND.QUEUE`가 있는지 확인하십시오. 이 큐는 큐 관리자가 작성될 때 기본적으로 작성됩니다.
2. 원격 시스템에서 명령 서버가 큐 관리자에서 실행 중인지 확인하십시오. 명령 서버가 실행되지 않는 경우 원격 관리가 불가능합니다.

- a) 큐 관리자에 대해 **runmqsc**를 시작하십시오. 예를 들어, 큐 관리자 `target.queue.manager`의 경우 다음 명령을 입력하십시오.

```
runmqsc target.queue.manager
```

- b) 다음 명령을 입력하여 명령 서버의 상태를 표시하십시오.

```
DISPLAY QMSTATUS CMDSERV
```

- c) 다음 명령을 입력하여 **runmqsc** 명령 프롬프트를 종료하십시오.

```
end
```

- d) 명령 서버가 시작되지 않으면 이 명령 서버를 시작하십시오. 예를 들어, 큐 관리자 `target.queue.manager`의 경우 다음 명령을 입력하십시오.

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

3. 로컬 큐 관리자에서 채널, 리스너 및 전송 큐를 정의하십시오.

- a) 큐 관리자에 대해 **runmqsc**를 시작하십시오. 예를 들어, 큐 관리자 `source.queue.manager`의 경우 다음 명령을 입력하십시오.

```
runmqsc source.queue.manager
```

- b) 송신자 채널을 정의하십시오. 이 송신자 채널의 이름은 리모트 큐 관리자의 수신자 채널 이름과 같아야 합니다. 예를 들어, 다음 MQSC 명령을 입력하십시오. **CONNAME**의 값은 리모트 큐 관리자의 IP 주소와 리스너의 포트 번호로 바꿉니다.

```
DEFINE CHANNEL ('source.to.target') +  
CHLTYPE(SDR) +  
CONNAME (localhost:1819) +  
XMITQ ('target.queue.manager') +  
TRPTYPE(TCP)
```

- c) 수신자 채널을 정의하십시오. 이 수신자 채널의 이름은 리모트 큐 관리자의 송신자 채널 이름과 같아야 합니다. 예를 들어, 다음 명령을 입력하십시오.

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

- d) 로컬 큐 관리자에서 리스너를 정의하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
DEFINE LISTENER ('source.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1818)
```

- e) 로컬 큐 관리자에서 전송 큐를 정의하십시오. 이 전송 큐의 이름은 리모트 큐 관리자의 이름과 같아야 합니다. 예를 들어, 다음 명령을 입력하십시오.

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

- f) 리스너를 시작하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
START LISTENER ('source.queue.manager')
```

- g) 다음 명령을 입력하여 **runmqsc** 명령 프롬프트를 종료하십시오.

```
end
```

4. 리모트 큐 관리자에서 채널, 리스너 및 전송 큐를 정의하십시오.

- a) 큐 관리자에 대해 **runmqsc**를 시작하십시오. 예를 들어, 큐 관리자 `target.queue.manager`의 경우 다음 명령을 입력하십시오.

```
runmqsc target.queue.manager
```

- b) 송신자 채널을 정의하십시오. 이 송신자 채널의 이름은 로컬 큐 관리자의 수신자 채널 이름과 같아야 합니다. 예를 들어, 다음 MQSC 명령을 입력하십시오. **CONNAME**의 값은 로컬 큐 관리자의 IP 주소와 리스너의 포트 번호로 바꿉니다.

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(SDR) +  
CONNAME (localhost:1818) +  
XMITQ ('source.queue.manager') +  
TRPTYPE(TCP)
```

- c) 수신자 채널을 정의하십시오. 이 수신자 채널의 이름은 로컬 큐 관리자의 송신자 채널 이름과 같아야 합니다. 예를 들어, 다음 명령을 입력하십시오.

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

- d) 리스너를 정의하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
DEFINE LISTENER ('target.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1819)
```

- e) 전송 큐를 정의하십시오. 이 전송 큐의 이름은 로컬 큐 관리자의 이름과 같아야 합니다. 예를 들어, 다음 명령을 입력하십시오.

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

- f) 리스너를 시작하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
START LISTENER ('target.queue.manager')
```

- g) 다음 명령을 입력하여 **runmqsc**를 종료하십시오.

```
end
```

5. 로컬 시스템에서 송신자 채널을 시작하십시오.

- a) 큐 관리자에 대해 **runmqsc**를 시작하십시오. 예를 들어, 큐 관리자 `source.queue.manager`의 경우 다음 명령을 입력하십시오.

```
runmqsc source.queue.manager
```

- b) 송신자 채널을 시작하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
START CHANNEL ('source.to.target')
```

- c) 다음 명령을 입력하여 **runmqsc**를 종료하십시오.

```
end
```

6. 원격 시스템에서 송신자 채널을 시작하십시오.

- a) 큐 관리자에 대해 **runmqsc**를 시작하십시오. 예를 들어, 큐 관리자 `target.queue.manager`의 경우 다음 명령을 입력하십시오.

```
runmqsc target.queue.manager
```

- b) 송신자 채널을 시작하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
START CHANNEL ('target.to.source')
```

- c) 다음 명령을 입력하여 **runmqsc**를 종료하십시오.

```
end
```

7. 로컬 시스템에서 리모트 큐 관리자로 MQSC 명령을 송신하여 구성이 완료되었는지 테스트하십시오.

- a) 로컬 시스템에서 리모트 큐 관리자에 대한 **runmqsc** 명령 프롬프트를 시작하십시오. 예를 들어, 다음 명령을 입력하십시오.

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- b) 다음 명령을 입력하여 리모트 큐 관리자에서 큐를 표시하십시오.

```
DISPLAY QUEUE (*)
```

성공 시 리모트 큐 관리자의 큐 목록이 표시됩니다.

- c) 이러한 단계가 작동하지 않으면 두 시스템의 채널이 실행 중 상태인지 확인하십시오. 채널이 실행 중이 아니고 시작되지 않으면 채널 및 전송 큐가 올바르게 구성되었고 명령 서버가 실행 중인지 확인하십시오. 예를 들어, 올바른 CONNAME이 송신자 채널에 대해 지정되었고 전송 큐에 올바른 이름이 있는지 확인하십시오. 또한 문제를 해결하는 데 도움이 될 수 있는 보안 예외는 큐 관리자 로그를 확인하십시오.

결과

큐 관리자가 로컬 시스템에서 리모트 큐 관리자를 원격으로 관리하도록 구성되었습니다.

다음에 수행할 작업

- MQSC 명령을 사용한 원격 관리에 대해 자세히 학습하십시오. [188 페이지의 『리모트 큐 관리자에서 MQSC 명령 실행』](#)
- PCF 명령을 사용한 관리 프로그램 작성에 대해 자세히 학습하십시오. [25 페이지의 『IBM MQ PCF\(Programmable Command Format\) 사용』](#).
- 원격 관리를 위한 administrative REST API 사용에 대해 자세히 학습하십시오. [75 페이지의 『REST API를 사용한 원격 관리』](#).

원격 관리를 위한 명령 서버 관리

각 큐 관리자에는 이와 연관된 명령 서버가 있습니다. 명령 서버는 리모트 큐 관리자에서 수신되는 명령 또는 애플리케이션에서 PCF 명령을 처리합니다. 또한 처리를 위해 큐 관리자에 명령을 제공하고 완료 코드 또는 운영자 메시지를 리턴합니다. 사용자는 명령 서버의 상태를 시작하고 중지하고 표시할 수 있습니다. 명령 서버는 PCF 명령, MQAI를 포함하는 모든 관리 및 원격 관리에도 필수적입니다.

시작하기 전에

큐 관리자 속성의 값(**SCMDSERV**)에 따라, 명령 서버는 큐 관리자가 시작될 때 자동으로 시작되거나 수동으로 시작되어야 합니다. 명령 서버가 자동으로 시작되면 **strmqscsv** 또는 **endmqscsv** 명령을 사용하여 명령 서버를 시작하고 중지할 수 없습니다. MQSC 명령 **ALTER QMGR**을 사용하여 **SCMDSERV** 속성 값을 변경할 수 있습니다. 기본적으로, 명령 서버가 자동으로 시작됩니다.

큐 관리자를 중지하면 이와 연관되는 명령 서버도 종료합니다.

프로시저

- 명령 서버의 상태를 표시하십시오.

- a) 다음 명령을 입력하여 적절한 큐 관리자에 대한 **runmqsc** 명령 프롬프트를 시작하십시오.

```
runmqsc target.queue.manager
```

여기서 **target.queue.manager**는 표시되는 명령 서버의 큐 관리자입니다.

- b) 다음 MQSC 명령을 입력하여 명령 서버 상태를 표시하십시오.

```
DISPLAY QMSTATUS CMDSERV
```

- c) 다음 명령을 입력하여 **runmqsc** 명령 프롬프트를 종료하십시오.

```
end
```

- 명령 서버가 자동으로 시작하도록 설정되지 않은 경우 다음 명령을 입력하여 명령 서버를 시작하십시오.

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

여기서 **target.queue.manager**는 시작되는 명령 서버의 큐 관리자입니다.

- 명령 서버가 자동으로 시작하도록 설정되지 않은 경우 다음 명령을 입력하여 명령 서버를 중지하십시오.

```
endmqcsv target.queue.manager
```

여기서 **target.queue.manager**는 중지되는 명령 서버의 큐 관리자입니다.

기본적으로 명령 서버는 제어된 방식으로 중지됩니다. 명령에 **-i** 플래그를 추가하여 명령 서버를 즉시 중지할 수 있습니다.

리모트 큐 관리자에서 MQSC 명령 실행

원격 구성을 위해 큐 관리자를 구성한 후 로컬 시스템에서 **runmqsc** 명령의 특정 양식을 사용하여 리모트 큐 관리자에서 MQSC 명령을 실행할 수 있습니다. 각 명령은 Escape PCF로서 리모트 큐 관리자의 명령 큐 SYSTEM.ADMIN.COMMAND.QUEUE에 송신됩니다. 응답은 SYSTEM.MQSC.REPLY.QUEUE 큐에 수신됩니다.

시작하기 전에

MQSC 명령을 사용하여 큐 관리자를 원격으로 관리하려면 183 페이지의 [『원격 관리를 위한 큐 관리자 구성』](#)의 단계를 완료하여 채널, 전송 큐, 리스너 및 명령 서버를 구성해야 합니다.

프로시저

- 명령 서버가 리모트 큐 관리자에서 실행 중인지 확인하십시오.
큐 관리자에서 명령 서버를 시작하는 방법에 대한 정보는 187 페이지의 [『원격 관리를 위한 명령 서버 관리』](#)의 내용을 참조하십시오.
- 소스 큐 관리자에서 다음 두 가지 방법 중 하나로 MQSC 명령을 실행할 수 있습니다.

- 다음 명령으로 **runmqsc**를 시작하여 대화식으로
 - 리모트 큐 관리자가 z/OS에 있는 경우 다음 명령을 입력하십시오.

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

- 리모트 큐 관리자가 멀티플랫폼에 있는 경우 다음 명령을 입력하십시오.

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- 명령행에서
 - 텍스트 파일에 원격 시스템에서 실행할 MQSC 명령을 배치하십시오(행마다 하나의 명령).

- b. **runmqsc** 명령에 **-v** 플래그를 사용하여 로컬 큐 관리자의 MQSC 명령을 확인하십시오. **-v** 플래그는 명령이 올바른지 확인하지만 이 명령을 실행하지는 않습니다. 일부 명령이 리모트 큐 관리자에 적용 가능하지만 로컬 큐 관리자에게는 적용할 수 없는 경우 실패할 수 있습니다.

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

myCmdFile.in에는 확인할 MQSC 명령이 포함되고 results.out 파일에는 명령에 대한 검증 결과가 포함됩니다.

- c. 다음 명령 중 하나를 입력하여 리모트 큐 관리자에서 명령 파일을 실행하십시오.
- 리모트 큐 관리자가 z/OS에 있는 경우 다음 명령을 입력하십시오.

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in > results.out
```

- 리모트 큐 관리자가 멀티플랫폼에 있는 경우 다음 명령을 입력하십시오.

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in > results.out
```

사용되는 매개변수는 다음 매개변수입니다.

-w seconds

MQSC 명령이 간접 모드에서 실행되도록 지정합니다. 여기서 명령은 명령 서버 입력 큐에 배치되며 순서대로 실행됩니다.

변수 *seconds*는 리모트 큐 관리자의 응답을 기다리는 시간 길이(초)를 지정합니다. 이 시간이 지난 후 수신된 모든 응답은 제거되지만 MQSC 명령은 여전히 리모트 큐 관리자에서 실행됩니다. 명령 제한시간이 초과되면 로컬 큐 관리자에 다음 메시지가 생성됩니다.

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

MQSC 명령을 실행하는 것을 중지할 때, 로컬 큐 관리자는 도착한 제한시간 초과 응답을 표시하고 추가 응답을 제거합니다.

-x

리모트 큐 관리자가 z/OS 큐 관리자가 되도록 지정합니다.

-m localQMGrName

리모트 큐 관리자에 명령을 제출하기 위해 사용할 로컬 큐 관리자의 이름을 지정합니다.

다음에 수행할 작업

MQSC 명령을 원격으로 실행할 때 문제가 있는 경우 다음을 수행하십시오.

- 리모트 큐 관리자가 실행되고 있는지 확인하십시오.
- 명령 서버가 원격 시스템에서 실행 중인지 확인하십시오.
- 채널 연결 끊기 간격이 만료되지 않았는지 확인하십시오. 예를 들어, 채널이 시작되었지만 잠시 후에 종료됩니다. 이는 채널을 수동으로 시작하는 경우 특히 중요합니다.
- 로컬 큐 관리자에서 송신된 요청이 대상 큐 관리자를 인식하는지 확인하십시오. 예를 들어, 리모트 큐 관리자에 지원되지 않는 매개변수가 포함된 요청이 있습니다.
- MQSC 명령으로 문제점 해결도 참조하십시오.

코드화 문자 세트 간 데이터 변환

두 문자 세트가 모두 단일 언어 또는 하나의 유사한 언어 그룹과 관련된 경우 IBM MQ 정의 형식(내장 형식이라고도 함)의 메시지 데이터는 큐 관리자에 의해 한 코드화 문자 세트에서 다른 문자 세트로 변환할 수 있습니다.

예를 들면, 코드화 문자 세트 ID(CCSID)가 850과 500인 코드화 문자 세트 간 변환이 지원되는데, 두 문자 세트 모두 서유럽 언어에 해당하기 때문입니다.

ASCII에 대한 EBCDIC 줄 바꾸기 (NL) 문자 변환의 경우, `mqs.ini` 파일의 모든 큐 관리자 스탠자 및 **AMQ_CONVEBCDICNEWLINE** 환경 변수를 참조하십시오.

지원되는 변환은 [데이터 변환 처리](#)에 정의됩니다.

CCSID 37과 500사이의 변환은 IBM MQ Appliance, Windows, Linux 및 macOS에서 지원됩니다.

큐 관리자가 내장 형식의 메시지를 변환할 수 없는 경우

CCSID가 다른 자국어(NL) 그룹을 나타내는 경우 큐 관리자는 내장 형식에서 자동으로 메시지를 변환할 수 없습니다. 예를 들어, CCSID 850과 CCSID 1025 사이의 변환(키릴 문자 스크립트를 사용하는 언어에 설정된 EBCDIC 코드화 문자임)은 지원되지 않습니다. 하나의 코드화 문자 세트의 많은 문자가 다른 코드화 문자 세트로 표시될 수 없기 때문입니다. 다른 자국어(NL)에서 작동하는 큐 관리자의 네트워크 및 일부 코드화 문자 세트 사이의 데이터 변환이 지원되지 않는 경우, 기본 변환을 사용할 수 있습니다.

`ccsid_part2.tbl`가 적용되는 플랫폼의 경우 자세한 정보는 `ccsid_part2.tbl`를 사용하는 193 페이지의 『기본 데이터 변환 지정』의 내용을 참조하십시오. `ccsid_part2.tbl` 파일이 적용되는 플랫폼 이외의 플랫폼에서의 기본 데이터 변환은 191 페이지의 『기본 데이터 변환』에 설명되어 있습니다.

개선된 유니코드 데이터 변환 지원

제품은 데이터 변환에서 유니코드 8.0 표준에 정의된 모든 유니코드 문자를 지원합니다. 여기에는 대리 쌍(U+FFFF 이상의 유니코드 코드 포인트를 나타내는 'D800' - 'DFFF' 범위의 2바이트 UTF-16 문자 쌍)을 포함하여 UTF-16에 대한 전체 지원이 포함됩니다.

문자 결합 시퀀스는 한 CCSID에서 사전 구성된 문자가 다른 CCSID에서 문자 결합 시퀀스에 맵핑되는 경우에도 지원됩니다.

유니코드와 CCSID 1388, 1390, 1399, 4933, 5488, 16884 사이의 데이터 변환은 일부 플랫폼에서 유니코드 보충 언어판의 코드 포인트에 맵핑되는 항목을 포함하여 이 CCSID에 대해 현재 정의된 모든 코드 포인트를 지원하기 위해 확장됩니다.

CCSID 1390, 1399, 16884의 경우 여기에는 JIS X 0213(JIS2004) 표준에 정의된 문자가 포함됩니다.

또한 유니코드 및 여섯 개의 새로운 CCSID(1374 - 1379) 간 변환에 대한 지원도 추가되었습니다.

ccsid_part2.tbl 파일


추가 파일 `ccsid_part2.tbl`가 제공됩니다.


`ccsid_part2.tbl` 파일은 `ccsid.tbl` 파일보다 우선합니다.

- CCSID 항목을 추가하거나 수정할 수 있도록 합니다.
- 기본 데이터 변환을 지정합니다.
- 다양한 명령 레벨에 대한 데이터를 지정합니다.

`ccsid_part2.tbl`은(는) 다음 플랫폼에만 적용 가능합니다.

-  Linux - 모든 버전
-  Windows

 IBM MQ for Windows에서 `ccsid_part2.tbl`은 기본적으로 `MQDataRoot\conv\table` 디렉토리에 있습니다. 그리고 IBM MQ for Windows에서는 지원되는 모든 코드 세트를 기록합니다.


 IBM MQ for Linux에서, `ccsid_part2.tbl`은 `MQDataRoot/conv/table` 디렉토리에 있으며, 지원되는 코드 세트는 IBM MQ에서 제공된 변환표에 유지됩니다.

`ccsid_part2.tbl` 파일이 추가 CCSID 정보를 제공하기 위해 IBM MQ의 이전 버전에서 사용된 기존 `ccsid.tbl` 파일을 대체하지만, `ccsid.tbl` 파일은 계속해서 IBM MQ에 의해 구문 분석되므로 삭제되지 않아야 합니다.

자세한 정보는 191 페이지의 『`ccsid_part2.tbl` 파일』의 내용을 참조하십시오.

ccsid.tbl 파일

ccsid_part2.tbl이 적용되는 플랫폼 외 플랫폼에서는 다음 목적으로 ccsid.tbl 파일이 사용됩니다.

-  AIX에서 지원되는 코드 세트는 내부적으로 운영 체제에 보유됩니다.
- 이는 추가 코드 세트를 지정합니다. 추가 코드 세트를 지정하려면 ccsid.tbl을 편집해야 합니다(이를 수행하는 방법에 대한 지침은 파일에 제공됨).
- 이는 기본 데이터 변환을 지정합니다.

ccsid.tbl에 기록된 정보를 업데이트할 수 있습니다. 예를 들어, 운영 체제의 향후 릴리스가 추가 코드화된 문자 세트를 지원하는 경우 이를 수행할 수도 있습니다.

기본 데이터 변환

데이터 변환이 일반적으로 지원되지 않는 두 개의 시스템 사이의 채널을 설정하는 경우, 작업할 채널에 대한 기본 데이터 변환을 사용해야 합니다.

ccsid_part2.tbl이 적용되는 플랫폼 외 플랫폼에서 기본 데이터 변환을 사용하려면 ccsid.tbl 파일을 편집하여 기본 EBCDIC CCSID 및 기본 ASCII CCSID를 지정하십시오. 수행 방법에 대한 지시사항은 파일에 포함되어 있습니다. 채널을 사용하여 연결될 모든 시스템에서 이를 수행해야 합니다. 변경사항을 적용하려면 큐 관리자를 재시작하십시오.

기본 데이터 변환 프로세스는 다음과 같습니다.

- 소스 및 대상 CCSID 사이의 변환이 지원되지 않지만, 소스 및 대상 환경의 CCSID가 양쪽 EBCDIC 또는 양쪽 ASCII인 경우, 문자 데이터는 변환되지 않고 대상 애플리케이션으로 전달됩니다.
- 하나의 CCSID가 ASCII 코드화 문자 세트를 나타내고 다른 CCSID가 EBCDIC 코드화 문자 세트를 나타내는 경우, IBM MQ는 ccsid.tbl에 정의된 기본 데이터 변환 CCSID를 사용하여 데이터를 변환합니다.

참고: 메시지에 지정된 코드화 문자 세트 및 기본 코드화 문자 세트에 동일한 코드 값이 있는 문자로 변환되는 문자를 제한하십시오. IBM MQ 오브젝트 이름에 유효한 문자 세트만 사용하는 경우(IBM MQ 오브젝트 이름 지정에 정의됨) 일반적으로 이 요구사항을 충족하게 됩니다. 일본에서 사용되는 EBCDIC CCSID 290, 930, 1279 및 5026의 경우 예외가 발생하는데, 여기서는 소문자가 다른 EBCDIC CCSID에 사용된 코드와 다른 코드를 갖기 때 문입니다.

사용자 정의 형식의 메시지 변환

큐 관리자는 하나의 코드화 문자 세트에서 다른 문자 세트로 사용자 정의 형식의 메시지를 변환할 수 없습니다. 사용자 정의 형식의 데이터를 변환해야 하는 경우, 그와 같은 각각의 형식에 데이터 변환 엑시트를 제공해야 합니다. 사용자 정의 형식의 문자 세트를 변환하기 위해 기본 CCSID를 사용하지 마십시오. 사용자 정의 형식의 데이터를 변환하고 데이터 변환 엑시트를 작성하는 것에 대한 자세한 정보는 [데이터 변환 엑시트 작성](#)을 참조하십시오.

큐 관리자 CCSID 변경

ALTER QMGR 명령의 **CCSID** 속성을 사용하여 큐 관리자의 CCSID를 변경한 경우, 명령 서버 및 채널 프로그램을 포함하여 실행 중인 모든 애플리케이션이 중지되고 재시작되도록 큐 관리자를 중지하고 재시작하십시오.

이는 큐 관리자 CCSID가 변경될 때 실행 중인 모든 애플리케이션은 기존 CCSID를 계속 사용하므로 필수적입니다.

ccsid_part2.tbl 파일

ccsid_part2.tbl 파일은 추가 CCSID 정보를 제공하는 데 사용됩니다. ccsid_part2.tbl 파일은 IBM MQ 9.0이전에 사용된 ccsid.tbl 파일을 대체합니다.

참고: IBM MQ 9.0 이전에 추가 CCSID 정보를 제공하는 데 사용된 ccsid.tbl 파일은 IBM MQ에서 계속 구문 분석되므로 삭제하지 않아야 합니다. 하지만 ccsid_part2.tbl의 입력 항목은 ccsid.tbl의 다른 입력 항목보다 우선합니다.

ccsid_part2.tbl이 다음과 같으므로 ccsid.tbl이 아닌 ccsid_part2.tbl을 사용해야 합니다.

- 유니코드 인코딩 값에 대한 지원을 포함합니다. IBM MQ 9.0부터 제품은 UTF-16의 전체 지원을 포함하여 데이터 변환에서 유니코드 8.0 표준에 정의된 모든 유니코드 문자를 지원합니다. 자세한 정보는 [189 페이지의 『코드화 문자 세트 간 데이터 변환』](#)의 내용을 참조하십시오.
- 선택된 명령 레벨에만 항목을 적용할 수 있도록 CCSID 항목의 버전을 지정할 수 있습니다.



ccsid_part2.tbl 파일을 사용하여 다음을 수행할 수 있습니다.

- CCSID 입력 항목을 추가하거나 수정합니다.
- 기본 데이터 변환을 지정합니다.
- 다양한 명령 레벨에 대한 데이터를 지정합니다.

ccsid_part2.tbl 파일은 다음 플랫폼에만 적용 가능합니다.

-  Linux - 모든 버전
-  Windows

ccsid_part2.tbl 파일의 위치는 플랫폼에 따라 다릅니다.

-  모든 버전의 Linux에서는 `MQDataRoot/conv/table` 디렉토리입니다.
-  Windows의 경우 `MQDataRoot\conv\table` 디렉토리입니다.

CCSID 항목 추가 또는 수정

ccsid_part2.tbl 파일의 항목 형식은 다음과 같습니다.

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

CCSID 1200(UTF-16)에 대한 항목의 예는 다음과 같습니다.

```
1200 1200 1200 1200 3 8 0 UTF-16
```

참고: ACRI의 값에 대한 자세한 정보는 ccsid_part2.tbl 파일의 주석을 참조하십시오.

ccsid_part2.tbl 형식에서:

유형은 다음과 같을 수 있습니다.

1=SBCS

2=DBCS

3=MBCS

인코딩은 다음과 같을 수 있습니다.

1=EBCDIC

2 = ASCII

3 = ISO

4 = UCS-2

5 = UTF-8

6 = Euc

7 = GB18030

8 = UTF-16

9 = UTF-32

파일을 편집할 때 다음을 수행할 수 있습니다.

- 행 시작 위치에 # 기호를 사용하여 주석을 지정할 수 있습니다. 이를 통해 IBM MQ에서 해당 행을 구문 분석하지 않습니다.
- 인라인 주석을 제공할 수 없습니다.
- 빈 행을 작성하지 않도록 해야 합니다.
- 파일의 끝 위치에 새 항목을 추가하지 않아야 합니다.

새 CCSID 항목은 ACRI 테이블 정보 앞에 추가되어야 합니다.

기본 데이터 변환 지정

기본 변환 CCSID를 정의할 수 있습니다. 이는 ASCII 또는 유사한 CCSID와 EBCDIC CCSID 사이의 변환이 지원되지 않는 경우 두 CCSID 사이에서 변환하는 데 사용됩니다.

이 기능을 사용하는 경우 전송 및 메시지 헤더에 기본 변환이 사용되고, 사용자 데이터 변환에도 이를 사용할 수 있습니다.

기본 변환은 다음과 유사한 두 행을 작성하여 사용으로 설정됩니다.

```
default      0      500      1      1      0
default      0      850      1      2      0
```

첫 번째 행은 EBCDIC CCSID에 대해 기본값을 500으로 설정하고 두 번째 행은 ASCII 및 유사한 CCSID에 대해 기본값을 850으로 설정합니다.

서로 다른 명령 레벨에 대한 데이터 지정

IBM MQ의 여러 명령 레벨에 대해 CCSID 항목을 지정하려면 콜론 기호와 다음 섹션을 적용 가능하게 하려는 IBM MQ의 명령 레벨을 이어서 사용합니다.

숫자는 큐 관리자 또는 클라이언트가 실행되어야 하는 최소 명령 레벨을 나타냅니다. 예를 들어, 현재 큐 관리자가 명령 레벨 900이고, 800 또는 900 명령 레벨 플래그를 발견하면 CCSID를 읽습니다.

그러나 레벨 800의 큐 관리자는 900 섹션의 CCSID를 무시합니다.

지정된 명령 레벨은 새 명령 레벨 플래그를 찾을 때까지, 명령 레벨 플래그 이후에 발견한 모든 CCSID 항목에 적용 가능합니다.

명령 레벨을 모든 명령 레벨로 설정해야 하는 경우 숫자 0을 지정하십시오.

처음 ccsid_part2.tbl을 구문 분석하는 경우 IBM MQ에서는 발견한 모든 CCSID를 IBM MQ의 모든 명령 레벨에 대해 유효한 항목으로 처리합니다.

버전화는 IBM MQ가 첫 번째 명령 레벨 플래그를 발견하는 경우에만 사용되기 시작합니다.

다음 코드 스니펫은 버전화 사용에 대한 예를 보여줍니다.

```
# Comment Block
# End of Comment Block
# Because no command level flag is specified and we're at the start of the file
# the following CCSIDs will be read on all versions
  819  819      0      819      1      3      0      IS08859-1
  923  923      0      923      1      3      0      IS08859-15
 1051 1051      0      1051     1      3      0      IBM-1051
# The colon :900 below shows that the CCSIDs after will only be for MQ cmd level 900 and above
:900
  8629 437      0      437      1      2      0      IBM-437
 12725 437      0      437      1      2      0      IBM-437
 16821 437      0      437      1      2      0      IBM-437
 20917 437      0      437      1      2      0      IBM-437
# The colon :0 below shows that the CCSIDs after will be for all version of MQ
:0
  4946 850      0      850      1      2      0      IBM-850
 33618 850      0      850      1      2      0      IBM-850
 61697 850      0      850      1      2      0      IBM-850
 61698 850      0      850      1      2      0      IBM-850
```


Managed File Transfer 관리

Managed File Transfer 명령을 사용하여 Managed File Transfer를 관리할 수 있습니다. 또한 일부 관리 태스크에는 IBM MQ Explorer를 사용할 수도 있습니다.

에이전트 명령 큐에 메시지를 넣어 전송 시작

소스 에이전트의 명령 큐에 파일 전송 메시지를 넣어 파일 전송을 시작할 수도 있습니다. 예제 명령 큐 이름은 SYSTEM.FTE.COMMAND.AGENT01입니다. 메시지가 올바른 소스 에이전트의 명령 큐에 도달하는지 확인해야 합니다. XML의 소스 정보와 일치하지 않는 에이전트에서 메시지를 수신하면 메시지가 거부됩니다.

전송 요청 XML은 FileTransfer.xsd 스키마를 따라야 하며 <request> 요소를 루트 요소로 사용해야 합니다. 전송 요청 메시지의 구조 및 콘텐츠에 대한 정보는 [파일 전송 요청 메시지 형식](#)을 참조하십시오. 에이전트 명령 큐에 전송 요청 메시지를 추가하는 방법은 태스크에 따라 다릅니다. 예를 들어, IBM MQ Java API를 사용하여 프로그래밍 방식으로 큐에 메시지를 넣을 수 있습니다.

관련 개념

[242 페이지의 『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

[256 페이지의 『메시지에서 파일로 데이터 전송』](#)

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

[266 페이지의 『프로토콜 브릿지』](#)

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

[302 페이지의 『IBM Integration Bus 에서 MFT 에 대한 작업』](#)

FTEOutput 및 FTEInput 노드를 사용하여 IBM Integration Bus 에서 Managed File Transfer 에 대해 작업할 수 있습니다.

[302 페이지의 『MFT 복구 및 재시작』](#)

에이전트 또는 큐 관리자가 사용 불가능한 경우(예: 전원 문제 또는 네트워크 장애), Managed File Transfer는 다음과 같이 복구합니다.

관련 태스크

[195 페이지의 『MFT 에이전트 시작』](#)

파일 전송을 위해 Managed File Transfer 에이전트를 사용하려면 먼저 에이전트를 시작해야 합니다.

[202 페이지의 『새 파일 전송 시작』](#)

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

[208 페이지의 『진행 중인 파일 전송 모니터링』](#)

IBM MQ Explorer의 [관리 파일 전송-현재 전송 진행](#) 탭을 사용하여 진행 중인 파일 전송을 모니터링할 수 있습니다. 이 파일 전송은 IBM MQ Explorer 또는 명령행에서 시작된 것입니다. 또한 탭에는 스케줄된 전송이 시작된 때의 스케줄된 전송의 진행 상태가 표시됩니다.

[210 페이지의 『전송 로그에서 파일 전송 상태 보기』](#)

IBM MQ Explorer의 [전송 로그](#) 를 사용하여 파일 전송의 세부사항을 볼 수 있습니다. 이는 명령행 또는 IBM MQ Explorer에서 시작된 전송일 수 있습니다. 또한 [전송 로그](#)에 표시되는 내용을 사용자 정의할 수 있습니다.

[211 페이지의 『MFT 자원 모니터링』](#)

Managed File Transfer 자원(예: 큐 또는 디렉토리)을 모니터링할 수 있습니다. 이 자원의 조건이 충족되면 자원 모니터가 파일 전송과 같은 태스크를 시작합니다. IBM MQ Explorer용 Managed File Transfer 플러그인에서 **fteCreateMonitor** 명령 또는 [모니터 보기](#)를 사용하여 자원 모니터를 작성할 수 있습니다.

240 페이지의 『파일 전송 템플릿에 대한 작업』

파일 전송 템플릿을 사용하여 반복된 전송 또는 복합 전송에 적합한 공용 파일 전송 설정을 저장할 수 있습니다. **fteCreateTemplate** 명령을 사용하여 명령행에서 전송 템플릿을 작성하거나 IBM MQ Explorer에서 **관리 파일 전송의 새 템플릿 작성** 마법사를 사용하여 전송 템플릿을 작성하거나 **템플릿으로 전송 설정 저장** 선택란을 선택하여 파일 전송 작성 중에 템플릿을 저장합니다. **전송 템플릿** 창에 Managed File Transfer 네트워크에서 사용자가 작성한 전송 템플릿이 모두 표시됩니다.

200 페이지의 『MFT 에이전트 나열』

명령행 또는 IBM MQ Explorer를 사용하여 특정 큐 관리자에 등록된 Managed File Transfer 에이전트를 나열할 수 있습니다.

200 페이지의 『MFT 에이전트 중지』

명령행에서 Managed File Transfer 에이전트를 중지할 수 있습니다. 에이전트를 중지하는 경우, 에이전트가 중지되기 전에 에이전트를 일시중지하여 현재 파일 전송을 완료할 수 있도록 합니다. 명령행에서 **-i** 매개변수를 지정하여 에이전트를 즉시 중지시킬 수도 있습니다. 에이전트가 중지되면 에이전트를 재시작할 때까지는 해당 에이전트를 사용하여 파일을 전송할 수 없습니다.

MFT 로거 구성

관련 참조

파일 전송을 위한 지침

MFT 에이전트 시작

파일 전송을 위해 Managed File Transfer 에이전트를 사용하려면 먼저 에이전트를 시작해야 합니다.

이 태스크 정보

명령행에서 Managed File Transfer Agent를 시작할 수 있습니다. 이 경우 시스템을 로그오프하면 에이전트 프로세스가 중지됩니다.

ALW AIX, Linux, and Windows에서는 시스템을 로그오프할 때 에이전트를 계속 실행하고 파일 전송을 계속 수신할 수 있도록 에이전트를 구성할 수 있습니다.

z/OS z/OS에서 대화식 세션 없이도 에이전트를 JCL에서 시작된 태스크로 시작하도록 구성할 수 있습니다.

에이전트 실행 중에 복구할 수 없는 오류가 발생하는 경우 FFDC(First Failure Data Capture)가 생성되고 에이전트가 중지됩니다.

프로시저

- 명령행에서 에이전트를 시작하려면 **fteStartAgent** 명령을 사용하십시오.

자세한 정보는 **fteStartAgent**의 내용을 참조하십시오.

ALW

시스템에서 로그오프할 때 에이전트를 계속 실행하도록 구성하려면 다음을 수행하십시오.

- Windows** Windows에서 에이전트를 Windows 서비스로 실행되도록 구성하십시오. 자세한 정보는 [196 페이지의 『MFT 에이전트를 Windows 서비스로 시작』](#)의 내용을 참조하십시오.

- Linux** **AIX** AIX and Linux에서 스크립트 파일을 사용하여 재부팅하는 동안 에이전트가 자동으로 시작되도록 구성하십시오. 자세한 정보는 [198 페이지의 『AIX and Linux 시스템 시작 시 MFT 에이전트 시작』](#)의 내용을 참조하십시오.

z/OS

z/OS에서 대화식 세션 없이도 에이전트를 JCL에서 시작된 태스크로 시작하도록 구성하십시오.

자세한 정보는 [199 페이지의 『Starting an MFT agent on z/OS』](#)의 내용을 참조하십시오.

관련 태스크

200 페이지의 『MFT 에이전트 나열』

명령행 또는 IBM MQ Explorer를 사용하여 특정 큐 관리자에 등록된 Managed File Transfer 에이전트를 나열할 수 있습니다.

200 페이지의 『MFT 에이전트 중지』

명령행에서 Managed File Transfer 에이전트를 중지할 수 있습니다. 에이전트를 중지하는 경우, 에이전트가 중지되기 전에 에이전트를 일시중지하여 현재 파일 전송을 완료할 수 있도록 합니다. 명령행에서 **-i** 매개변수를 지정하여 에이전트를 즉시 중지시킬 수도 있습니다. 에이전트가 중지되면 에이전트를 재시작할 때까지는 해당 에이전트를 사용하여 파일을 전송할 수 없습니다.

관련 참조

MFT 에이전트 상태 값

fteStartAgent

Windows MFT 에이전트를 Windows 서비스로 시작

사용자가 Windows에서 로그오프해도 에이전트는 계속 실행되고 파일 전송을 수신할 수 있도록 에이전트를 Windows 서비스로 시작할 수 있습니다.

이 태스크 정보

Windows의 경우 명령행에서 에이전트를 시작하면 사용자가 Windows에 로그인하는 데 사용한 사용자 이름을 사용하여 에이전트 프로세스가 실행됩니다. 시스템에서 로그오프하면 에이전트 프로세스가 중지됩니다. 에이전트가 중지되지 않도록 하기 위해 에이전트를 Windows 서비스로 실행되도록 구성할 수 있습니다. Windows 서비스로 실행하면 Windows 환경이 시작되거나 재시작될 때 에이전트가 자동으로 시작되도록 구성할 수도 있습니다.

Windows 서비스로 실행되는 에이전트를 시작하려면 다음 단계를 완료하십시오. 에이전트를 Windows 서비스로 실행하려면 지원되는 Windows 버전 중 하나에서 Managed File Transfer 를 실행 중이어야 합니다. 지원되는 환경 목록은 [IBM MQ의 시스템 요구사항](#)을 참조하십시오.

정확한 단계는 에이전트를 이미 작성했는지 또는 작성 중인지에 따라 달라집니다. 두 옵션 모두 다음 단계에 설명되어 있습니다.

프로시저

1. Managed File Transfer 에이전트를 작성하는 경우 **fteCreateAgent**, **fteCreateCDAgent** 또는 **fteCreateBridgeAgent** 명령을 사용하십시오. 에이전트를 Windows 서비스로 실행하려면 **-s** 매개변수를 지정하십시오. 다음 예에서는 에이전트 큐 관리자 QMGR1이 있는 AGENT1 에이전트가 작성됩니다. Windows 서비스는 비밀번호 **ftpassword**가 연관된 **fteuser**라는 사용자 이름을 사용하여 실행됩니다.

```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftpassword
```

선택사항으로 **-s** 매개변수 뒤에 서비스의 이름을 지정할 수 있습니다. 이름을 지정하지 않으면 서비스 이름이 **mqmftAgentAGENTQMGR**로 지정됩니다. 여기서 **AGENT**는 지정한 에이전트 이름이며 **QMGR**은 에이전트 큐 관리자 이름입니다. 이 예제에서 서비스의 기본 이름은 **mqmftAgentAGENT1QMGR1**입니다.

참고: **-su** 매개변수를 사용하여 지정하는 Windows 사용자 계정에는 **Log on as a service** 권한이 있어야 합니다. 이를 구성하는 방법에 대한 정보는 [Windows 서비스로 실행 중인 MFT 에이전트 또는 로거 문제점 해결](#)을 참조하십시오.

자세한 정보는 [fteCreateAgent](#), [fteCreateCDAgent: Connect:Direct® 브릿지 에이전트 작성](#) 또는 [fteCreateBridgeAgent \(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)를 참조하십시오.

2. 이전 단계에 따라 에이전트를 작성한 경우 **fteCreateAgent**, **fteCreateCDAgent** 또는 **fteCreateBridgeAgent** 명령으로 생성된 MQSC 명령을 실행하십시오. 이러한 명령을 실행하면 에이전트에 필요한 IBM MQ 큐가 작성됩니다.

예를 들어, **AGENT1**이라는 에이전트, **QMGR1**이라는 에이전트 큐 관리자 및 **COORDQMGR1**이라는 조정 큐 관리자의 경우 다음 명령을 실행하십시오.

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

- 이전 단계에 따라 에이전트를 작성하지 않았으며 대신 기존 에이전트를 Windows 서비스로 실행되도록 구성하려면 먼저 실행 중인 에이전트를 중지한 후 에이전트의 구성을 수정하십시오.

a) 다음 예에서는 **AGENT1**이라는 에이전트를 사용합니다. 다음 명령을 실행하십시오.

```
fteStopAgent AGENT1
```

- fteModifyAgent** 명령을 사용하여 Windows 서비스로 실행되도록 에이전트를 구성하십시오.

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

자세한 정보는 **fteModifyAgent: MFT 에이전트를 Windows 서비스로서 실행**을 참조하십시오.

- fteStartAgent** 명령을 사용하여 에이전트를 시작하십시오. 또는 제어판의 관리 도구에서 사용할 수 있으며 Windows 데스크탑 시작 메뉴에서 선택하는 Windows 서비스 도구를 사용하여 서비스를 시작할 수 있습니다.

```
fteStartAgent AGENT1
```

사용자가 Windows에서 로그오프해도 서비스는 계속 실행됩니다. 시스템 종료 후 Windows 가 다시 시작될 때도 서비스가 다시 시작되도록 하기 위해 Windows 서비스 도구의 **시작 유형 필드**는 기본적으로 **자동**으로 설정됩니다. Windows 가 다시 시작될 때 서비스가 다시 시작되지 않도록 하려면 이를 **수동**으로 변경하십시오.

- 옵션: 에이전트를 중지하려면 **fteStopAgent** 명령을 사용하거나 Windows 서비스 도구를 사용하십시오. 예를 들면, 명령행에서 다음 명령을 실행하십시오.

```
fteStopAgent AGENT1
```

- fteStopAgent** 명령을 서비스로 실행하는 경우 이 매개변수를 지정했는지 여부에 관계없이 이 명령은 항상 **-i** 매개변수를 사용하여 실행됩니다. **-i** 매개변수는 진행 중인 전송을 완료하지 않고 즉시 에이전트를 중지합니다. Windows 서비스의 제한사항으로 인해 이러한 상태가 발생합니다.

다음에 수행할 작업

Windows 서비스를 시작하는 데 문제가 있는 경우 [Windows 서비스로 실행 중인 MFT 에이전트 또는 로거 문제점 해결](#)을 참조하십시오. 이 주제에서는 Windows 서비스 로그 파일의 위치에 대해서도 설명합니다.

관련 태스크

200 페이지의 [『MFT 에이전트 나열』](#)

명령행 또는 IBM MQ Explorer를 사용하여 특정 큐 관리자에 등록된 Managed File Transfer 에이전트를 나열할 수 있습니다.

200 페이지의 [『MFT 에이전트 중지』](#)

명령행에서 Managed File Transfer 에이전트를 중지할 수 있습니다. 에이전트를 중지하는 경우, 에이전트가 중지되기 전에 에이전트를 일시중지하여 현재 파일 전송을 완료할 수 있도록 합니다. 명령행에서 **-i** 매개변수를 지정하여 에이전트를 즉시 중지시킬 수도 있습니다. 에이전트가 중지되면 에이전트를 재시작할 때까지는 해당 에이전트를 사용하여 파일을 전송할 수 없습니다.

관련 참조

[fteCreateAgent\(MFT 에이전트 작성\)](#)

[fteCreateCDAgent\(Connect:Direct 브릿지 에이전트 작성\)](#)

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

[fteModify 에이전트\(실행MFT 대리인으로서의Windows 서비스\)](#)

관련 정보

MFTagent.properties 파일

Linux

AIX

AIX and Linux 시스템 시작 시 MFT 에이전트 시작

Managed File Transfer Agent는 AIX and Linux에서 시스템 시동 시 시작하도록 구성될 수 있습니다. 로그오프 하면 에이전트가 계속 실행되고 파일 전송을 받을 수 있습니다.

이러한 Managed File Transfer 명령 (**fteCreateAgent**, **fteCreateCDAgent** 또는 **fteCreateBridgeAgent**) 중 하나를 사용하여 에이전트를 작성하고 구성한 경우, 다음 명령을 단순히 실행하는 스크립트 파일을 사용하여 AIX and Linux 시스템에서 재부팅하는 동안 자동으로 시작되도록 에이전트를 구성할 수 있습니다.

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

여기서 `mq_install_root`는 필수 Managed File Transfer 설치의 루트 디렉토리입니다. 기본값은 `/opt/mqm`입니다. `agent_name`은 시작할 Managed File Transfer Agent의 이름입니다. 이 스크립트 파일의 사용은 특정 운영 체제에 따라 다릅니다. 예를 들어 Linux에서 사용할 수 있는 추가 옵션이 있습니다.

Linux

Linux

Linux 시스템의 경우 시스템 시작 프로세스 중에 애플리케이션을 시작할 수 있는 여러 방법이 있습니다. 일반적으로 다음 단계를 수행하는 것을 고려하십시오.

1. 다음 콘텐츠가 포함된 `/etc/rc.mqmft` 파일을 작성하십시오.

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

여기서 `mqmft_user`는 에이전트 프로세스를 실행할 사용자 ID입니다. 이 사용자 ID는 `mqm` 그룹의 멤버여야 합니다.

2. 파일 실행 파일을 작성하십시오. 예를 들어 다음과 같습니다.

```
chmod 755 /etc/rc.mqmft
```

3. 그런 다음, 다음 행을 `/etc/inittab`에 추가하십시오.

```
mqmft:5:boot:/etc/rc.mqmft
```

Linux에서 부팅 중에 에이전트를 시작하는 다른 방법으로는 `/etc/rc.d/rc.local` 파일에 스크립트 행을 추가하거나 Linux SuSe에서 `/etc/init.d/boot.local` 파일에 스크립트 행을 추가하는 방법이 있습니다. 환경에 가장 적합한 방법을 선택해야 합니다. 다음은 지원되는 특정 Linux 배포판에서 시동 중에 에이전트를 시작하는 다른 방법에 대한 자세한 정보입니다.

SLES 10 및 11

SLES(SUSE Linux Enterprise Server) 10 및 11 시스템의 경우 다음 단계를 수행하십시오.

1. 시스템 루트 사용자 ID로 고유한 `/etc/init.d/rc.rclocal` 파일을 작성하십시오.
2. 다음 행을 `rc.rclocal` 파일에 추가하십시오.

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
```

```
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

3. 다음 명령을 실행하십시오.

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

systemd 를 사용하여 Linux 에서 Managed File Transfer 에이전트 시작

Linux

다음 프로시저를 수행하십시오.

1. /etc/systemd/ 시스템 폴더에 파일을 작성하고 이름을 지정하십시오 (예: <agentname>.service). 다음 콘텐츠를 추가하십시오. 여기서 <agentname>은 MFT_AGT_LNX_0입니다.

```
# vi /etc/systemd/system/MFT_AGT_LNX_0.service
[Unit]
Description=IBM MQ MFT MFT_AGT_LNX_0
[Service]
ExecStart=/opt/mqm/bin/fteStartAgent MFT_AGT_LNX_0
ExecStop=/opt/mqm/bin/fteStopAgent MFT_AGT_LNX_0
Type=forking
User=mqm
Group=mqm
KillMode=none
```

2. 서비스를 활성화하려면 다음 명령을 실행하십시오.

```
# systemctl enable MFT_AGT_LNX_0
# systemctl daemon-reload
```

3. 에이전트를 시작하고 상태를 확인하려면 다음 명령을 실행하십시오.

```
# systemctl start MFT_AGT_LNX_0
# systemctl status MFT_AGT_LNX_0
```

관련 태스크

200 페이지의 『MFT 에이전트 중지』

명령행에서 Managed File Transfer 에이전트를 중지할 수 있습니다. 에이전트를 중지하는 경우, 에이전트가 중지되기 전에 에이전트를 일시중지하여 현재 파일 전송을 완료할 수 있도록 합니다. 명령행에서 **-i** 매개변수를 지정하여 에이전트를 즉시 중지시킬 수도 있습니다. 에이전트가 중지되면 에이전트를 재시작할 때까지는 해당 에이전트를 사용하여 파일을 전송할 수 없습니다.

관련 참조

[fteCreateAgent](#)

[fteCreateCDAgent: Connect:Direct 브릿지 에이전트 작성](#)

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

z/OS

Starting an MFT agent on z/OS

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

Note: Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The

fteStartAgent, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

As a Java application, an agent is a z/OS UNIX System Services application that you can run from JCL by using the BFGAGSTP member, from a generated Managed File Transfer command PDSE library data set for an agent. For more information about how to create an MFT command PDSE library data set, and customize it for the required agent, see [Creating an MFT Agent or Logger command data set](#).

Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

Related reference

[“Stopping an MFT agent on z/OS” on page 201](#)

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

[The MFT agent.properties file](#)

MFT 에이전트 나열

명령행 또는 IBM MQ Explorer를 사용하여 특정 큐 관리자에 등록된 Managed File Transfer 에이전트를 나열할 수 있습니다.

이 태스크 정보

명령행을 사용하여 에이전트를 나열하려면 **fteListAgents** 명령을 참조하십시오.

IBM MQ Explorer를 사용하여 에이전트를 나열하려면 네비게이터 보기에서 조정 큐 관리자 이름 아래에 있는 **에이전트**를 클릭하십시오.

에이전트가 **fteListAgents** 명령에 의해 나열되지 않거나 IBM MQ Explorer에 표시되지 않는 경우, 다음 주제의 진단 플로우차트를 사용하여 문제점을 찾아 수정하십시오. [MFT 에이전트가 fteListAgents 명령에 의해 나열되지 않는 경우 수행할 작업](#).

관련 참조

[fteListAgents: 조정 큐 관리자에 대한 MFT 에이전트 나열](#)

[MFT 에이전트 상태 값](#)

[fteShowAgentDetails](#)

MFT 에이전트 중지

명령행에서 Managed File Transfer 에이전트를 중지할 수 있습니다. 에이전트를 중지하는 경우, 에이전트가 중지되기 전에 에이전트를 일시중지하여 현재 파일 전송을 완료할 수 있도록 합니다. 명령행에서 **-i** 매개변수를 지정하여 에이전트를 즉시 중지시킬 수도 있습니다. 에이전트가 중지되면 에이전트를 재시작할 때까지는 해당 에이전트를 사용하여 파일을 전송할 수 없습니다.

시작하기 전에

큐 관리자와 연관된 에이전트의 이름을 확인하려는 경우 IBM MQ Explorer 또는 명령행을 사용하여 에이전트를 나열할 수 있습니다([fteListAgents](#) 명령 참조).

이 태스크 정보

명령행에서 에이전트를 중지하려면 [fteStopAgent](#)를 참조하십시오.

에이전트가 **fteStopAgent**를 사용하여 제어된 방식으로 중지되면 에이전트는 새 관리 전송 요청을 승인하지 않고 진행 중인 전송이 완료될 때까지 대기한 후 실제로 종료됩니다. IBM MQ 9.3.0부터 에이전트가 여전히 임시 상태이므로 아직 종료되지 않았고 아직 재시작할 수 없음을 표시하기 위해 에이전트는 진행 중인 전송이 완료될 때까지 STOPPING 상태가 됩니다. 이 상태는 [fteListAgents](#) 및 [fteShowAgentDetails](#) 명령의 출력과 MFT REST API 조회 및 IBM MQ Explorer의 MFT 플러그인의 에이전트 보기에 표시됩니다.

Windows 에이전트를 Windows 서비스로 실행되도록 구성한 경우 **fteStopAgent** 명령을 실행하면 Windows 서비스도 중지됩니다. 또는 Windows 서비스 도구를 사용하여 서비스를 중지하여 에이전트를 중지할 수 있습니다. 자세한 정보는 196 페이지의 『MFT 에이전트를 Windows 서비스로 시작』의 내용을 참조하십시오.

관련 태스크

195 페이지의 『MFT 에이전트 시작』

파일 전송을 위해 Managed File Transfer 에이전트를 사용하려면 먼저 에이전트를 시작해야 합니다.

관련 참조

[MFT 에이전트 상태 값](#)

[fteStopAgent](#)

201 페이지의 『Stopping an MFT agent on z/OS』

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

z/OS Stopping an MFT agent on z/OS

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

Note: Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

Controlled agent shutdown by using the z/OS MODIFY command (F)

The **MODIFY** command allows you to stop an agent in a controlled way as an alternative to the **fteStopAgent** command. The agent completes any transfers currently in progress but the agent does not start any new transfers.

For example:

```
F job_name,APPL=STOP
```

where *job_name* is the job that the agent process is running under.

Immediate agent shutdown by using the z/OS STOP command (P)

The **STOP** command is equivalent to an immediate stop by using the **fteStopAgent** command with the **-i** parameter. The agent is stopped immediately even if the agent is currently transferring a file.

For example:

```
P job_name
```

where *job_name* is the job that the agent process is running under.

Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

Related reference

[“Starting an MFT agent on z/OS” on page 199](#)

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

The [MFT agent.properties](#) file

새 파일 전송 시작

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

이 태스크 정보

명령행에서 새 파일 전송을 시작하려면 [fteCreateTransfer](#) 명령을 참조하십시오.

IBM MQ Explorer에서 새 관리 파일 전송 작성 마법사를 사용하여 새 파일 전송을 시작하려면 다음 단계를 사용하십시오.

프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 클릭하십시오. 콘텐츠 보기에 **관리 파일 전송 센터**가 표시됩니다.
2. 모든 조정 큐 관리자가 네비게이터 보기에 표시됩니다. 전송에 사용하려는 에이전트가 등록되는 조정 큐 관리자의 이름을 펼치십시오. 현재 전송에 사용하려는 조정 큐 관리자가 아닌 조정 큐 관리자에 연결되어 있는 경우, 네비게이터 보기에서 해당 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭하고 **연결 끊기**를 클릭하십시오. 그런 다음, 사용하려는 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭하고 **연결**을 클릭하십시오.
3. 다음 메소드 중 하나를 사용하여 새 관리 파일 전송 작성 마법사를 시작하십시오.
 - a) 탐색 보기에서 관련 조정 큐 관리자, **전송 템플릿**, **전송 로그** 또는 **보류 중인 전송** 노드 중 하나의 이름을 마우스 오른쪽 단추로 클릭하십시오. 그런 다음 **새 전송**을 클릭하여 마법사를 시작하십시오.
 - b) **파일 > 새로 작성 > 기타 > 관리 파일 전송 마법사 > 새 전송 마법사**를 클릭하십시오.
4. 마법사 패널의 지시사항을 따르십시오. 각 패널에 대한 컨텍스트 도움말도 제공됩니다. Windows에서 컨텍스트 도움말에 액세스하려면 F1을 누르십시오. Linux에서는 Ctrl+F1 또는 Shift+F1을 누르십시오.

관련 개념

203 페이지의 『전송 정의 파일 사용』

파일 전송 작성 시 사용할 수 있는 전송 정의 파일을 지정할 수 있습니다. 전송 정의 파일은 전송 작성에 필요한 정보의 전부 또는 일부를 정의하는 XML 파일입니다.

관련 태스크

205 페이지의 『스케줄된 파일 전송 작성』

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 스케줄할 수 있습니다. 스케줄된 전송에 단일 파일 또는 다중 파일이 그룹으로 포함될 수 있습니다. 스케줄된 파일 전송을 한 번 수행하거나 전송을 여러 번 반복할 수 있습니다.

207 페이지의 『파일 전송 트리거』

해당 전송이 진행되기 전에 참여해야 하는 파일 전송에 대해 특정 트리거 조건을 설정할 수 있습니다. 트리거 조건이 참이 아닌 경우, 파일 전송이 진행되지 않으며 전송이 발생하지 않았음을 기록하는 로그 메시지가 선택적으로 제출됩니다. 그런 다음, 파일 전송 요청이 제거됩니다. 예를 들어, 소스 에이전트가 위치한 시스템에 있는 이름 지정된 파일이 지정된 크기를 초과하는 경우 또는 특정 이름 지정된 파일이 소스 에이전트가 위치한 시스템에 존재하는 경우에만 수행되는 파일 전송을 설정할 수 있습니다. IBM MQ Explorer 또는 명령행에서 트리거된 파일 전송을 설정할 수 있습니다.

303 페이지의 『중단된 전송의 복구에 대한 제한시간 설정』

한 소스 에이전트의 모든 전송에 적용되는 중단된 파일 전송의 전송 복구 제한시간을 설정할 수 있습니다. 개별 전송에 대한 전송 복구 제한시간을 설정할 수도 있습니다. 소스 에이전트가 중단된 파일 전송을 복구하려고 계속 시도하는 특정 시간을 초 단위로 설정하고 에이전트가 이 제한시간에 도달할 때 전송에 성공하지 못하면 전송은 실패합니다.

관련 참조

fteCreateTransfer: 새 파일 전송 시작

[파일 전송 요청 메시지 형식](#)

[파일 전송을 위한 지침](#)

전송 정의 파일 사용

파일 전송 작성 시 사용할 수 있는 전송 정의 파일을 지정할 수 있습니다. 전송 정의 파일은 전송 작성에 필요한 정보의 전부 또는 일부를 정의하는 XML 파일입니다.

전송 정의 파일은 단일 전송 조작으로 여러 소스 파일 및 여러 목적지 파일을 지정할 때 유용합니다. 전송 정의 파일을 사용하여 복합 파일 전송을 제출할 수 있습니다. 전송 정의 파일을 재사용 및 공유할 수 있습니다.

두 가지 형식의 전송 정의 파일을 사용할 수 있으며, 이러한 형식은 약간 다르지만 둘 다 `FileTransfer.xsd` 스키마를 따릅니다. 이 스키마는 Managed File Transfer 설치의 `samples\schema` 디렉토리에서 찾을 수 있습니다.

다음과 같은 두 가지 형식의 전송 정의 파일이 지원됩니다.

- 전송을 위한 소스 및 목적지 파일 정의. 이 정의는 **transferSpecifications** 요소를 루트로 사용합니다.
- 소스와 목적지 파일 및 소스와 목적지 에이전트 등 전체 전송 정의. 이 정의는 **request** 요소를 루트로 사용합니다.
 - 이러한 형식의 파일은 **fteCreateTransfer** 명령에서 **-gt** 매개변수를 사용하여 생성할 수 있습니다.

다음 예에서는 전송의 소스 및 목적지 파일만 지정하는 전송 정의 파일 형식을 보여 줍니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <item checksumMethod="MD5" mode="text">
    <source recursive="false" disposition="leave">
      <file>textTransferTest.txt</file>
    </source>
    <destination type="directory" exist="overwrite">
      <file>c:\targetfiles</file>
    </destination>
  </item>
</transferSpecifications>
```

이러한 형식의 전송 정의 파일을 제출하려면 명령행에서 소스 및 목적지 에이전트를 지정해야 합니다.

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

다음 예는 전송에 필요한 모든 정보를 지정하는 전송 정의 파일 형식입니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>fteuser</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
    <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:\sourcefiles\*.jpg</file>
        </source>
        <destination type="directory" exist="error">
          <file>/targetfiles/images</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

fteCreateTransfer 명령에서 **-gt** 매개변수를 사용하여 이 형식의 파일을 생성할 수 있습니다. 이 형식으로 전송 정의 파일을 제출하는 경우 명령행에 다른 것을 지정할 필요가 없습니다.

```
fteCreateTransfer -td c:\definitions\example2.xml
```

전송 정의 파일과 함께 일반 매개변수를 전달하여 명령행에 대한 소스 및 목적지 에이전트 정보를 대체할 수 있습니다. 예를 들면, 다음과 같습니다.

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

이 예에서는 명령행 옵션을 사용하여 전송 정의 파일 내에 정의된 목적지 에이전트를 **AGENT9**으로 대체하고 전송 정의 파일에 정의된 목적지 큐 관리자를 **agent9qm**으로 대체합니다.

설명된 두 형식 모두에 하나 이상의 `<item>` 요소가 포함될 수 있습니다. `<item>` 요소에 대한 추가 정보는 [파일 전송 요청 메시지 형식](#)을 참조하십시오. 이러한 각 전송 항목은 전송의 동작을 제어하는 추가 속성을 사용하여 소스 및 목적지 파일 쌍을 정의합니다. 예를 들어, 다음 동작을 지정할 수 있습니다.

- 전송이 체크섬을 사용하는지 여부
- 전송이 텍스트 또는 2진인지 여부
- 전송이 완료된 후 소스 파일을 삭제할지 여부
- 목적지 파일이 이미 있는 경우 해당 파일을 덮어쓸지 여부

전송 정의 파일을 사용하면 명령행에서는 사용할 수 없는 추가 옵션을 지정할 수 있는 장점이 있습니다. 예를 들어, 메시지에서 파일로의 전송을 수행 중인 경우 전송 정의 파일을 사용하여 `groupId` 속성을 지정할 수 있습니다. 이 속성은 큐에서 읽는 메시지의 IBM MQ 그룹 ID를 지정합니다. 전송 정의 파일의 또 다른 장점은 각 파일 쌍에 다른 옵션을 지정할 수 있는 것입니다. 예를 들어, 파일별로 체크섬이 사용되는지 여부 또는 파일 전송 모드(텍스트 또는 2진 모드)를 지정할 수 있습니다. 명령행을 사용하는 경우에는 동일한 옵션이 전송되는 모든 파일에 적용됩니다.

예를 들면, 다음과 같습니다.

```
<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
```

```

</source>
<destination type="file" exist="error">
  <file>c:\destinationfiles\destination1.doc</file>
</destination>
</item>

<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>

```

z/OS

항목을 사용하여 분산 시스템에서 z/OS 시스템으로 파일을 전송할 수 있습니다.

z/OS

```

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>

```

z/OS

이 예에서는 소스 에이전트의 `textTransferTest.txt` 파일을 텍스트 모드로 목적지 에이전트의 데이터 세트 `//TEXT.TRANS.TEST`에 전송합니다. 이 전송에서는 소스 에이전트의 기본 인코딩(소스 인코딩 속성은 지정되지 않음)의 소스 데이터를 코드 페이지 IBM-1047로 변환합니다.

스케줄된 파일 전송 작성

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 스케줄할 수 있습니다. 스케줄된 전송에 단일 파일 또는 다중 파일이 그룹으로 포함될 수 있습니다. 스케줄된 파일 전송을 한 번 수행하거나 전송을 여러 번 반복할 수 있습니다.

이 태스크 정보

파일 전송 스케줄을 한 번 발생으로 설정하거나 다음 간격으로 발생하도록 설정할 수 있습니다.

- 매분
- 매시
- 매일
- 매주
- 매월
- 매년

그런 다음, 다음 시점에서 전송이 중지되도록 지정할 수 있습니다.

- 정의된 시간 및 날짜
- 정의된 발생 횟수 후

또는 영구적으로 전송이 이루어지도록 지정할 수 있습니다.

스케줄링된 전송이 매일 동일한 시간에 실행되면 에이전트 특성 파일의 **adjustScheduleTimeForDaylightSaving** 속성을 사용하여 시계가 변경될 때 스케줄이 발생하는 시간을 조정하십시오. 자세한 정보는 [MFT agent.properties](#) 파일을 참조하십시오.

명령행을 사용하여 스케줄링된 새 파일 전송을 작성하려면 [fteCreateTransfer](#) 명령에 대한 스케줄링 매개변수(-**tb**, -**ss**, -**oi**, -**of**, -**oc** 및 -**es**)를 사용하십시오.

IBM MQ Explorer에서 새 관리 파일 전송 작성 마법사를 사용하여 새 스케줄된 파일 전송을 작성하려면 다음 단계를 사용하십시오.

프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 클릭하십시오. 콘텐츠 보기에 **관리 파일 전송 센터**가 표시됩니다.
2. 모든 조정 큐 관리자가 네비게이터 보기에 표시됩니다. 전송에 사용하려는 에이전트가 등록되는 조정 큐 관리자의 이름을 펼치십시오. 현재 전송에 사용하려는 조정 큐 관리자가 아닌 조정 큐 관리자에 연결되어 있는 경우, 네비게이터 보기에서 해당 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭하고 **연결 끊기**를 클릭하십시오. 그런 다음, 사용하려는 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭하고 **연결**을 클릭하십시오.
3. 다음 메소드를 사용하여 새 관리 파일 전송 작성 마법사를 시작하십시오.
 - a) 탐색 보기에서 관련 조정 큐 관리자, **전송 템플릿**, **전송 로그** 또는 **보류 중인 전송** 노드 중 하나의 이름을 마우스 오른쪽 단추로 클릭하십시오. 그런 다음 **새 전송**을 클릭하여 마법사를 시작하십시오.
 - b) **파일 > 새로 작성 > 기타 > 관리 파일 전송 마법사 > 새 전송 마법사**를 클릭하십시오.
4. 마법사 패널의 지시사항을 따르십시오. **스케줄 전송 사용** 선택란을 선택하고 **스케줄** 탭에 스케줄 세부사항을 입력하십시오. 스케줄된 파일 전송은 전송에 영향을 줄 수 있는 문제점이 없는 경우 스케줄 시작 후 일 분 내에 시작됩니다. 예를 들어, 스케줄된 전송의 시작을 막는 네트워크 또는 에이전트의 문제점이 있을 수 있습니다. 각 패널에 대한 컨텍스트 도움말이 제공됩니다. Windows에서 컨텍스트 도움말에 액세스하려면 F1을 누르십시오. Linux에서는 Ctrl+F1 또는 Shift+F1을 누르십시오.

결과

스케줄된 파일 전송에 관련된 메시지에 대한 정보는 [스케줄된 파일 전송 로그 메시지 형식](#)을 참조하십시오.

보류 중인 파일 전송에 대한 작업

IBM MQ Explorer에서 보류 중인 스케줄된 파일 전송을 볼 수 있습니다. **보류 중인 전송** 창에 현재 연결되어 있는 조정 큐 관리자에 등록된 보류 중인 전송이 모두 표시됩니다.

이 태스크 정보


아직 시작되지 않은 스케줄된 파일 전송의 상태를 보려면 다음 단계를 수행하십시오.

프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 펼치십시오. 콘텐츠 보기에 **관리 파일 전송 센터**가 표시됩니다.
2. 모든 조정 큐 관리자가 네비게이터 보기에 표시됩니다. 스케줄된 전송에 사용된 조정 큐 관리자의 이름을 펼치십시오. 연결된 조정 큐 관리자를 변경하려면, 네비게이터 보기에서 사용할 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭한 후 **연결**을 클릭하십시오.
3. **보류 중인 전송**을 클릭하십시오. **보류 중인 전송** 창이 콘텐츠 보기에 표시됩니다.
4. **보류 중인 전송** 창에 스케줄된 파일 전송에 대한 다음과 같은 자세한 내용이 표시됩니다.
 - a) **이름**: 스케줄된 파일 전송의 번호입니다. 이 번호는 자동으로 지정됩니다.
 - b) **소스**: 소스 에이전트의 이름입니다.
 - c) **소스 파일**: 호스트 시스템에서 전송할 파일의 이름입니다.
 - d) **목적지**: 목적지 에이전트의 이름입니다.
 - e) **목적지 파일**: 목적지 시스템에 전송된 후의 파일 이름입니다.

- f) **스케줄된 시작(선택한 시간대):** 관리자가 선택한 시간대에서 시작되도록 스케줄된 파일 전송 시간 및 날짜입니다. 표시되는 시간대를 변경하려면 **창 > 환경 설정 > IBM MQ Explorer > Managed File Transfer**를 클릭하고 **시간대:** 목록에서 대체 시간대를 선택하십시오. **확인**을 클릭하십시오.
- g) **반복 간격:** 스케줄된 전송을 반복하도록 선택한 경우, 전송을 반복할 지정된 간격으로 숫자로 표시됩니다.
- h) **반복 유형:** 스케줄된 전송을 반복하도록 선택한 경우, 파일 전송을 위해 지정한 반복 간격의 유형입니다. 유형 값은 분, 시간, 일, 주, 개월 또는 년 중 하나일 수 있습니다.
- i) **특정 시점까지 반복:** 스케줄된 전송을 반복하도록 선택한 경우, 파일 전송 반복을 중지하려 할 때의 세부사항입니다. 예를 들어, 지정된 날짜 및 시간 또는 지정된 발생 횟수입니다.

결과

보류 중인 전송 창에 표시된 내용을 새로 고치려면 콘텐츠 보기 도구 모음에서 새로 고치기 단추 를 클릭하십시오.

보류 중인 파일 전송을 취소하려면 특정 전송을 마우스 오른쪽 단추로 클릭한 다음, **취소**를 클릭하십시오. 전송을 완전히 취소하면 파일 전송 요청이 제거됩니다.

파일 전송 트리거

해당 전송이 진행되기 전에 참여해야 하는 파일 전송에 대해 특정 트리거 조건을 설정할 수 있습니다. 트리거 조건이 참이 아닌 경우, 파일 전송이 진행되지 않으며 전송이 발생하지 않았음을 기록하는 로그 메시지가 선택적으로 제출됩니다. 그런 다음, 파일 전송 요청이 제거됩니다. 예를 들어, 소스 에이전트가 위치한 시스템에 있는 이름 지정된 파일이 지정된 크기를 초과하는 경우 또는 특정 이름 지정된 파일이 소스 에이전트가 위치한 시스템에 존재하는 경우에만 수행되는 파일 전송을 설정할 수 있습니다. IBM MQ Explorer 또는 명령행에서 트리거된 파일 전송을 설정할 수 있습니다.

이 태스크 정보

트리거 조건을 충족시키기 위해 지속적으로 자원을 모니터링할 수도 있습니다. 자원 모니터링에 대한 추가 정보는 [211 페이지의 『MFT 자원 모니터링』](#)의 내용을 참조하십시오.

설정할 수 있는 세 가지 트리거 조건이 있습니다. 조건은 다음과 같습니다.

- 특정 파일이 소스 에이전트와 동일한 시스템에 있는 경우
- 특정 파일이 소스 에이전트와 동일한 시스템에 없는 경우
- 특정 파일이 소스 에이전트가 위치한 시스템에서 특정 크기를 초과하는 경우(크기는 바이트, KB, MB 또는 GB로 표현될 수 있음). 이러한 측정 단위는 2¹⁰ 규칙을 사용합니다(예: 1KB = 1024바이트, 1MB = 1024KB).

선행 목록에 있는 트리거 유형은 두 가지 방식으로 결합될 수 있습니다.

- 단일 조건의 경우 소스 에이전트가 위치한 시스템에 둘 이상의 파일을 지정할 수 있습니다. 이렇게 되면 지정된 파일 중 하나가 조건(Boolean 연산자 OR)을 충족하는 경우 전송이 트리거됩니다.
- 다중 조건을 지정할 수 있습니다. 이렇게 되면 모든 조건(Boolean 연산자 AND)이 충족되는 경우에만 전송이 트리거됩니다.

또한 트리거된 전송을 스케줄된 전송과 결합할 수 있습니다. 자세한 정보는 [스케줄된 파일 전송 작성](#)을 참조하십시오. 이 경우 트리거 조건이 스케줄이 시작될 시간에 또는 스케줄이 시작될 때마다(반복 스케줄의 경우) 평가됩니다.

프로토콜 브릿지 에이전트에서는 트리거된 전송이 지원되지 않습니다.

명령행을 사용하여 트리거된 파일 전송을 작성하려면 `fteCreateTransfer` 명령에서 **-tr** 매개변수를 사용하십시오.

IBM MQ Explorer에서 새 관리 파일 전송 작성 마법사를 사용하여 스케줄된 파일 전송을 작성하려면 다음 단계를 사용하십시오.

프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 클릭하십시오. 콘텐츠 보기에 **관리 파일 전송 센터**가 표시됩니다.


2. 모든 조정 큐 관리자가 네비게이터 보기에 표시됩니다. 스케줄된 전송에 사용된 조정 큐 관리자의 이름을 펼치십시오. 연결된 조정 큐 관리자를 변경하려면, 네비게이터 보기에서 사용할 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭한 후 **연결**을 클릭하십시오.
3. 다음 메소드 중 하나를 사용하여 **새 관리 파일 전송 작성** 마법사를 시작하십시오.
 - a) 탐색 보기에서 관련 조정 큐 관리자, **전송 템플릿**, **전송 로그** 또는 **보류 중인 전송** 노드 중 하나의 이름을 마우스 오른쪽 단추로 클릭하십시오. 그런 다음 **새 전송**을 클릭하여 마법사를 여십시오.
 - b) **파일 > 새로 작성 > 기타 > 관리 파일 전송 마법사 > 새 전송 마법사**를 클릭하십시오.
4. 마법사 패널의 지시사항을 따르십시오. **트리거** 탭에서 **트리거된 전송 사용** 선택란을 선택하고 해당 탭의 필드를 기입하여 트리거를 설정하십시오. 각 패널에 대한 컨텍스트 도움말이 제공됩니다. Windows에서 컨텍스트 도움말에 액세스하려면 F1을 누르십시오. Linux에서는 **Ctrl+F1** 또는 **Shift+F1**을 누르십시오.

진행 중인 파일 전송 모니터링

IBM MQ Explorer의 **관리 파일 전송-현재 전송 진행** 탭을 사용하여 진행 중인 파일 전송을 모니터링할 수 있습니다. 이 파일 전송은 IBM MQ Explorer 또는 명령행에서 시작된 것입니다. 또한 탭에는 스케줄된 전송이 시작된 때의 스케줄된 전송의 진행 상태가 표시됩니다.

이 태스크 정보

IBM MQ Explorer를 사용하여 원격 시스템의 조정 큐 관리자와 연관된 전송을 모니터링하려면 [209 페이지의 『원격 조정 큐 관리자를 모니터링하도록 IBM MQ Explorer 구성』](#) 주제의 지시사항을 따르십시오.


IBM MQ Explorer를 중지한 후 재시작한 후에는 이전 파일 전송 정보가 보존되지 않습니다. 재시작될 때 **현재 전송 진행 상황** 탭에서 과거 전송에 대한 정보가 지워집니다. IBM MQ Explorer가 열려 있을 때 언제든지 **완료된 전송 제거** 를 사용하여 완료된 전송을 지울 수 있습니다.


프로시저

IBM MQ Explorer 또는 명령행을 사용하여 새 파일 전송을 시작한 후에는 **현재 전송 진행 상태** 탭에서 전송의 진행 상태를 모니터링할 수 있습니다. 진행 중인 각 전송에 대해 다음 정보가 표시됩니다.

- a) **소스.** 소스 시스템에서 파일을 전송하는 데 사용되는 에이전트의 이름입니다.
- b) **대상.** 목적지 시스템에서 파일을 수신하는 데 사용되는 에이전트의 이름입니다.
- c) **현재 파일:** 현재 전송 중인 파일의 이름입니다. 이미 전송된 개별 파일의 일부가 괄호 안에 있는 총 파일 크기와 함께 B, KiB, MiB, GiB 또는 TiB 단위로 표시됩니다. 표시된 측정 단위는 파일의 크기에 따라 다릅니다.
B는 초당 바이트 수입입니다. KB/s는 초당 킬로바이트이고 여기서 1킬로바이트는 1024바이트와 같습니다. MB/s는 초당 메가바이트이고 여기서 1메가바이트는 1 048 576바이트와 같습니다. GB/s는 초당 기가바이트이고 여기서 1기가바이트는 1 073 741 824바이트입니다. TB/s는 초당 테라바이트이고 여기서 1테라바이트는 1 099 511 627 776바이트입니다.
- d) **파일 번호:** 둘 이상의 파일을 전송하는 경우, 이 번호는 전송이 총 파일 그룹에서 어디까지 이루어졌는지의 순서를 나타냅니다.
- e) **진행:** 진행 표시줄은 현재 파일 전송이 완료된 정도를 백분율로 표시합니다.
- f) **비율:** 파일이 전송되는 비율이며, KB/s(초당 킬로바이트, 여기서 1킬로바이트는 1024바이트와 같음)로 표시됩니다.
- g) **시작됨(선택한 시간대):** 파일 전송이 시작된 시간으로서 관리자가 선택한 시간대로 표시됩니다. 표시되는 시간대를 변경하려면 **창 > 환경 설정 > IBM MQ Explorer > Managed File Transfer**를 클릭하고 **시간대:** 목록에서 대체 시간대를 선택하십시오. **확인**을 클릭하십시오.
파일을 전송하는 동안 전송이 복구 상태로 되면, 시작된 시간은 파일 전송이 재개된 시간을 반영하도록 업데이트됩니다.

결과

이 탭은 정기적으로 해당 정보를 자동으로 새로 고치지만, **현재 전송 진행 상태** 탭에 표시되는 내용을 강제로 새로 고치려면 콘텐츠 도구 모음의 **새로 고치기** 를 클릭하십시오.

현재 전송 진행 상황 탭에서 파일 전송을 삭제하려면, 콘텐츠 보기 도구 모음에서 **완료된 전송 제거** 를 클릭하십시오. 이 단추를 클릭하면 탭에서 파일 전송 세부사항만 제거하며 현재 또는 스케줄된 전송을 중지하거나 취소하지 않습니다.

이를 닫고 **현재 전송 진행 상황** 탭으로 되돌아가려면 **창 > 보기 표시 > 기타 > 기타 > 관리 파일 전송 - 현재 전송 진행 상황**을 클릭하십시오. **확인**을 클릭하십시오.

다음에 수행할 작업

또한 사용자 정의 파일 전송 모니터링을 위한 애플리케이션을 개발할 수도 있습니다. 이는 해당 Managed File Transfer 관리 토픽에 대한 구독을 (프로그래밍 방식으로 또는 관리식으로) 작성하여 수행할 수 있고 모니터 애플리케이션은 토픽에 대한 Managed File Transfer 파일 전송 활동 발행물을 수신할 수 있습니다. 구독 토픽 및 발행 메시지 형식에 대한 자세한 정보는 [파일 전송 진행 메시지 예](#)를 참조하십시오.

관련 태스크

209 페이지의 [『원격 조정 큐 관리자를 모니터하도록 IBM MQ Explorer 구성』](#)

IBM MQ Explorer를 사용하여 원격 시스템에서 실행 중인 조정 큐 관리자와 연관된 파일 전송을 모니터링할 수 있습니다. IBM MQ Explorer를 실행할 수 있는 시스템이 필요합니다. 원격 조정 큐 관리자에 연결할 수 있으려면 IBM MQ Explorer 컴포넌트가 설치되어 있어야 합니다.

210 페이지의 [『전송 로그에서 파일 전송 상태 보기』](#)

IBM MQ Explorer의 **전송 로그**를 사용하여 파일 전송의 세부사항을 볼 수 있습니다. 이는 명령행 또는 IBM MQ Explorer에서 시작된 전송일 수 있습니다. 또한 **전송 로그**에 표시되는 내용을 사용자 정의할 수 있습니다.

원격 조정 큐 관리자를 모니터하도록 IBM MQ Explorer 구성

IBM MQ Explorer를 사용하여 원격 시스템에서 실행 중인 조정 큐 관리자와 연관된 파일 전송을 모니터링할 수 있습니다. IBM MQ Explorer를 실행할 수 있는 시스템이 필요합니다. 원격 조정 큐 관리자에 연결할 수 있으려면 IBM MQ Explorer 컴포넌트가 설치되어 있어야 합니다.

이 태스크 정보

가정: 원격 연결을 허용하도록 큐 관리자를 구성하여 원격 조정 큐 관리자에 연결할 권한을 획득했습니다.

구성 방법에 대한 자세한 정보는 [채널 인증으로 클라이언트 모드의 큐 관리자에 연결 및 MFT 특정 자원에 대한 권한 관리](#)를 참조하십시오.

Windows 또는 Linux를 실행하지 않는 시스템에서 에이전트 사이의 파일 전송과 큐 관리자를 모니터링하려면 다음 단계를 사용하여 원격 시스템에 연결하도록 IBM MQ Explorer를 구성하십시오.

프로시저

1. 로컬 IBM MQ Explorer를 시작하십시오.
2. IBM MQ Explorer가 로드되면 **Managed File Transfer** 폴더를 마우스의 오른쪽 단추로 클릭한 후 **새 구성**을 선택하십시오.
3. 조정 및 명령 큐 관리자를 선택하여 마법사를 진행한 후 구성의 이름을 정의하십시오.
4. 정의를 완료하려면 **완료**를 클릭하십시오.
5. 정의가 완료되면 정의를 마우스 오른쪽 단추로 클릭한 후 **연결**을 선택하십시오.

결과

이제 IBM MQ Explorer를 시작하고 사용하여 조정 큐 관리자와 연관된 Managed File Transfer 네트워크의 전송 활동을 모니터링할 수 있습니다.

관련 태스크

208 페이지의 [『진행 중인 파일 전송 모니터링』](#)

IBM MQ Explorer의 **관리 파일 전송-현재 전송 진행** 탭을 사용하여 진행 중인 파일 전송을 모니터링할 수 있습니다. 이 파일 전송은 IBM MQ Explorer 또는 명령행에서 시작된 것입니다. 또한 탭에는 스케줄된 전송이 시작된 때의 스케줄된 전송의 진행 상태가 표시됩니다.


210 페이지의 『전송 로그에서 파일 전송 상태 보기』

IBM MQ Explorer의 전송 로그를 사용하여 파일 전송의 세부사항을 볼 수 있습니다. 이는 명령행 또는 IBM MQ Explorer에서 시작된 전송일 수 있습니다. 또한 전송 로그에 표시되는 내용을 사용자 정의할 수 있습니다.

전송 로그에서 파일 전송 상태 보기

IBM MQ Explorer의 전송 로그를 사용하여 파일 전송의 세부사항을 볼 수 있습니다. 이는 명령행 또는 IBM MQ Explorer에서 시작된 전송일 수 있습니다. 또한 전송 로그에 표시되는 내용을 사용자 정의할 수 있습니다.



프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 펼친 다음, 전송 로그를 보려는 조정 큐 관리자의 이름을 펼치십시오.
2. 네비게이터 보기에서 **전송 로그**를 클릭하십시오. 콘텐츠 보기에 **전송 로그**가 표시됩니다.
3. **전송 로그** 창에 파일 전송에 대해 다음과 같은 자세한 내용이 표시됩니다.
 - a) **소스**: 소스 파일이 위치한 시스템의 에이전트의 이름입니다.
 - b) **목적지**: 파일을 전송할 시스템에 있는 에이전트의 이름입니다.
 - c) **완료 상태** 파일 전송의 상태입니다. 상태 값은 "시작됨", "진행 중", "성공", "부분적 성공", "취소됨" 또는 "실패" 중 하나일 수 있습니다.
 - d) **소유자**: 전송 요청을 제출한 호스트의 사용자 ID입니다.
 - e) **시작됨(선택한 시간대)**: Managed File Transfer 에이전트가 파일 전송 요청을 허용한 시간과 날짜로서 관리자가 선택한 시간대로 표시됩니다. 표시되는 시간대를 변경하려면 **창 > 환경 설정 > IBM MQ Explorer > Managed File Transfer**를 클릭하고 **시간대**: 목록에서 대체 시간대를 선택하십시오. **확인**을 클릭하십시오.
 - f) **상태 기록(시간대)**: (이 열은 기본적으로 표시되지 않습니다. **전송 로그 열 구성**  창을 사용하여 열을 표시하도록 선택할 수 있습니다. 관리자가 선택한 시간대에서 완료 상태가 기록된 시간 및 날짜입니다.
 - g) **작업 이름** `fteCreateTransfer`의 `-jn` 매개변수를 사용하여 사용자가 지정하거나 Ant 스크립트에서 지정하는 ID입니다.
 - h) **전송 ID**: 파일 전송의 고유 ID입니다.
 - i) **연결: 직접**: 프로세스 번호, 프로세스 이름, 기본 노드, 보조 노드, 소스 유형 및 목적지 유형에 대한 상세 정보입니다.

결과

참고: 전송 로그의 내부 형식은 APAR IC99545의 IBM MQ 8.0.0 Fix Pack 1에서 변경되었습니다. 결과적으로 IBM MQ Explorer가 V8.0.01 이상으로 업그레이드된 후 V8.0.0.0으로 복원되면, IBM MQ Explorer가 V8.0.0.1 인 동안 발생한 전송에 대해 감사 XML이 표시되지 않습니다. 이러한 전송을 위한 **특성** 창의 XML 패널에는 빈 텍스트 상자가 들어갑니다.

완료된 전송에 대한 추가 세부사항을 보려면 더하기 부호(+)를 클릭하여 원하는 전송을 펼치십시오. 그러면 해당 전송에 포함된 모든 소스 및 목적지 파일 이름을 볼 수 있습니다. 하지만 전송이 현재 진행 중이고 많은 파일로 구성되어 있는 경우에는 지금까지 이미 전송된 파일만 볼 수 있습니다.

전송 로그에 표시된 내용을 새로 고치려면 콘텐츠 보기 도구 모음에서 **새로 고치기** 단추 를 클릭하십시오. IBM MQ Explorer를 중지하고 재시작한 후에도 전송 로그의 파일 전송 정보가 로그에 남아 있습니다. 로그에서 완료된 파일 전송을 모두 삭제하려면 콘텐츠 보기 도구 모음에서 **완료된 전송 제거** 를 클릭하십시오.

로그에서 완료된 개별 파일 전송을 삭제하려면 전송을 마우스 오른쪽 단추로 클릭한 다음, **삭제**를 클릭하십시오. 전송을 삭제하더라도 진행 중이거나 스케줄된 전송은 중지되거나 취소되지 않으며 저장된 실행 기록 데이터만 삭제됩니다.

전송의 고유 ID를 클립보드에 복사하려면 해당 전송을 마우스의 오른쪽 단추로 클릭한 후 **ID 복사**를 클릭하십시오.

전송의 메타데이터 및 전체 감사 XML은 **특성** 조치 아래의 팝업 메뉴에서 사용할 수 있습니다.

관련 태스크

208 페이지의 『진행 중인 파일 전송 모니터링』

IBM MQ Explorer의 **관리 파일 전송-현재 전송 진행** 탭을 사용하여 진행 중인 파일 전송을 모니터링할 수 있습니다. 이 파일 전송은 IBM MQ Explorer 또는 명령행에서 시작된 것입니다. 또한 탭에는 스케줄된 전송이 시작된 때의 스케줄된 전송의 진행 상태가 표시됩니다.

211 페이지의 『전송 로그 구성』

표시되는 정보 및 정보가 IBM MQ Explorer의 **전송 로그**에 표시되는 방법을 구성할 수 있습니다.

303 페이지의 『중단된 전송의 복구에 대한 제한시간 설정』

한 소스 에이전트의 모든 전송에 적용되는 중단된 파일 전송의 전송 복구 제한시간을 설정할 수 있습니다. 개별 전송에 대한 전송 복구 제한시간을 설정할 수도 있습니다. 소스 에이전트가 중단된 파일 전송을 복구하려고 계속 시도하는 특정 시간을 초 단위로 설정하고 에이전트가 이 제한시간에 도달할 때 전송에 성공하지 못하면 전송은 실패합니다.


전송 로그 구성

표시되는 정보 및 정보가 IBM MQ Explorer의 **전송 로그**에 표시되는 방법을 구성할 수 있습니다.


이 태스크 정보

전송 로그에서 열의 순서를 재정렬하려면 이동할 열의 제목을 클릭한 후 새 위치로 끌어 놓으십시오. 새 열의 순서는 다음에 IBM MQ Explorer를 중지한 후 재시작할 때까지만 유지됩니다.

전송 로그에서 입력 항목을 필터하려면 **표시된 로그 입력 항목 필터** 필드에 문자열을 입력하십시오. 로그의 입력 항목을 모두 복원하려면 필드에서 입력한 문자열을 삭제하십시오. 이 필드에서 올바른 Java 정규식을 사용할 수 있습니다. 자세한 정보는 [MFT에서 사용하는 정규식을 참조하십시오](#).

전송 로그에 표시되는 열을 사용자 정의하려면 **전송 로그 열 구성** 을 사용하십시오. 다음 단계를 사용하여 **전송 로그 열 구성** 창을 시작 및 사용하십시오.

프로시저

1. 콘텐츠 보기에서 **전송 로그**가 열려 있는지 확인하십시오. 콘텐츠 보기 도구 모음에서 **전송 로그 열 구성** 을 클릭하십시오. **전송 로그 열 구성** 창이 열립니다.
2. **전송 로그**의 보기를 사용자 정의하려면 표시하거나 숨길 열에 해당하는 개별 선택란을 선택하거나 선택 취소하십시오. **모두 선택**을 클릭한 후 **확인**을 클릭하여 모든 선택란을 선택하거나, **모두 선택 취소**를 클릭한 후 **확인**을 클릭하여 모든 선택란을 선택 취소할 수 있습니다.

관련 태스크

208 페이지의 『진행 중인 파일 전송 모니터링』

IBM MQ Explorer의 **관리 파일 전송-현재 전송 진행** 탭을 사용하여 진행 중인 파일 전송을 모니터링할 수 있습니다. 이 파일 전송은 IBM MQ Explorer 또는 명령행에서 시작된 것입니다. 또한 탭에는 스케줄된 전송이 시작된 때의 스케줄된 전송의 진행 상태가 표시됩니다.

210 페이지의 『전송 로그에서 파일 전송 상태 보기』

IBM MQ Explorer의 **전송 로그**를 사용하여 파일 전송의 세부사항을 볼 수 있습니다. 이는 명령행 또는 IBM MQ Explorer에서 시작된 전송일 수 있습니다. 또한 **전송 로그**에 표시되는 내용을 사용자 정의할 수 있습니다.

MFT 자원 모니터링

Managed File Transfer 자원(예: 큐 또는 디렉토리)을 모니터링할 수 있습니다. 이 자원의 조건이 충족되면 자원 모니터가 파일 전송과 같은 태스크를 시작합니다. IBM MQ Explorer용 Managed File Transfer 플러그인에서 **fteCreateMonitor** 명령 또는 **모니터** 보기를 사용하여 자원 모니터를 작성할 수 있습니다.

이 태스크 정보

Managed File Transfer 자원 모니터링에는 다음 용어가 사용됩니다.

자원 모니터

자원 모니터는 사전정의된 정기적 간격으로 자원(예: 디렉토리 또는 큐)을 폴링하여 자원 콘텐츠가 변경되었는지 확인하는 프로세스입니다. 콘텐츠가 변경된 경우, 콘텐츠는 이 모니터에 대한 조건 세트와 비교됩니다. 일치があれば 이 모니터의 태스크가 시작됩니다.

자원

자원 모니터가 트리거 조건과 비교하기 위해 폴 간격마다 조사하는 시스템 자원입니다. 큐, 디렉토리 또는 중첩 디렉토리 구조가 모니터할 자원이 될 수 있습니다.

조건 및 트리거 조건

조건은 평가되는(일반적으로 모니터되는 자원의 콘텐츠에 대해) 표현식입니다. 표현식이 true로 평가되는 경우, 조건은 전체 트리거 조건에 기여합니다.

트리거 조건은 모든 조건이 충족되면 충족되는 전체 조건입니다. 트리거 조건이 충족되면 태스크를 진행할 수 있습니다.

태스크

태스크는 트리거 조건 또는 조건 세트가 충족되면 시작되는 조작입니다. 지원되는 태스크는 파일 전송 및 명령 호출입니다.

트리거 파일

트리거 파일은 모니터되는 디렉토리에 배치되어 태스크(일반적으로 전송)를 시작할 수 있는지 표시하는 파일입니다. 예를 들어, 처리할 모든 파일이 알려진 위치에 도달했으므로 전송할 수 있거나 달리 작동될 수 있음을 표시할 수 있습니다. 트리거 파일의 이름은 변수 대체를 사용하여 전송할 파일을 지정하는 데 사용할 수 있습니다. 자세한 정보는 222 페이지의 『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』의 내용을 참조하십시오.

트리거 파일은 준비 파일 또는 이동 파일이라고도 합니다. 그러나 이 문서에서는 일반적으로 트리거 파일이라고 합니다.

자원 모니터링은 프로토콜 브릿지 에이전트 또는 Connect:Direct 브릿지 에이전트에서 지원되지 않습니다.

관련 개념

[에이전트 오버로딩을 방지하도록 MFT 자원 모니터 구성에 대한 자세한 내용](#)

관련 참조

[fteCreateMonitor: MFT 자원 모니터 작성](#)

[fteListMonitors: MFT 자원 모니터 나열](#)

[fteDeleteMonitor: MFT 자원 모니터 삭제](#)

[MFT 모니터 요청 메시지 형식](#)

MFT 자원 모니터링 개념

Managed File Transfer 자원 모니터링 기능의 주요 개념에 대한 개요입니다.

자원 모니터

명령행에서 새 자원 모니터를 작성하고 시작하는 **fteCreateMonitor** 명령을 사용하여 자원 모니터를 작성합니다. 자원 모니터는 Managed File Transfer 에이전트와 연결되며 해당 에이전트가 실행 중일 때만 활성화됩니다. 모니터링 에이전트가 중지되면 자원 모니터도 중지됩니다. 자원 모니터가 작성될 때 에이전트가 이미 실행 중이면 자원 모니터가 즉시 시작됩니다. 모니터링 에이전트는 자원 모니터가 시작하는 태스크의 소스 에이전트이기도 해야 합니다.

자원 모니터 이름은 해당 에이전트 내에서 고유해야 합니다. 자원 모니터 이름은 최소 1자 길이어야 하고 별표(*), 퍼센트(%) 또는 물음표(?) 문자를 포함할 수 없습니다. 자원 모니터 이름의 제공에 사용되는 대소문자 구분은 무시되고 자원 이름은 대문자로 변환됩니다. 이미 있는 이름을 사용하여 자원 모니터를 작성하려고 시도하면 요청이 무시되고 그 시도는 자원 모니터 로그 토크에 로깅됩니다.

참고: 스케줄된 전송을 포함하는 태스크 정의를 사용하여 자원 모니터를 작성할 수 없습니다.

IBM MQ 9.3.0 이전에 자원 모니터를 중지하는 유일한 방법은 모니터 조작을 실행 중인 에이전트를 중지하는 것입니다. 자원 모니터를 다시 시작하려면 에이전트를 함께 다시 시작해야 합니다. IBM MQ 9.3.0에서 에이전트를 중지하거나 다시 시작할 필요 없이 자원 모니터를 시작 및 중지할 수 있습니다. 자세한 정보는 215 페이지의 『자원 모니터 시작 및 중지』의 내용을 참조하십시오.

에이전트에서 작성할 수 있고 모두 동일한 우선순위로 실행되는 자원 모니터의 수에 대한 제한은 없습니다. 겹치는 모니터링된 자원, 충돌하는 트리거 조건 및 자원 폴링 빈도의 의미를 고려하십시오.

자원 모니터가 겹치면 다음이 발생할 수 있습니다.

- 소스 위치/항목에 대한 경합 가능성이 있습니다.
- 동일한 소스 항목에 대해 중복 전송 요청이 있을 수 있습니다.
- 소스 항목의 충돌로 인한 예기치 않은 오류 또는 전송 실패.

여러 모니터가 동일한 위치를 스캔하고 동일한 항목에 대해 트리거할 수 있는 경우 동일한 항목에 대한 관리 전송 요청을 제출하는 두 개의 다른 모니터 문제가 발생할 수 있습니다.

자원 모니터는 모든 폴 간격 기간 후에 자원의 콘텐츠를 살펴봅니다. 자원의 콘텐츠는 트리거 조건과 비교되고 해당 조건이 충족되는 경우 자원 모니터와 연관된 태스크가 호출됩니다.

태스크는 비동기식으로 시작됩니다. 조건 일치와 있고 태스크가 시작되는 경우 자원 모니터는 자원 콘텐츠에 대한 추가적인 변경을 위해 계속해서 폴링합니다. 따라서 예를 들어 `reports.go` 파일이 모니터된 디렉토리에 도착하여 일치가 발생한 경우 태스크는 한번 시작됩니다. 다음 폴 간격에서는 파일이 계속 존재하는 경우에도 태스크가 다시 시작되지 않습니다. 하지만 파일이 삭제된 후 다시 디렉토리에 추가되거나, 마지막 수정 날짜 속성이 변경 등으로 인해 파일이 업데이트된 경우 다음 트리거 조건 검사를 수행하면 태스크가 다시 호출됩니다.

IBM MQ 9.1.5 이전에는 자원 모니터가 폴링 간격보다 긴 시간이 소요되는 폴을 수행하는 경우 이는 현재 폴이 완료되는 즉시 다음 폴이 시작됨을 의미하여 이는 자원 모니터가 에이전트에 대한 작업을 제출하는 속도에 영향을 미칠 수 있습니다. 이로 인해, 두 번째 폴이 발생할 때 첫 번째 폴에서 찾은 항목이 계속해서 그 위치에 있는 경우 성능 문제가 발생할 수 있습니다.

자원 모니터는 `ScheduledExecutor` 서비스를 사용하고 이전 폴링 완료 후에만 다음 폴링을 시작하고 구성된 폴링 간격 시간을 더합니다. 이는 폴 시간이 폴 간격보다 긴 경우 이전 폴 후에 즉시 다른 폴이 시작되지 않고 폴 간격 사이에는 항상 갭이 있음을 의미합니다.

파일 전송에 실패한 경우 자원 모니터 히스토리를 지울 수 있습니다. 이를 통해 파일을 삭제하고 디렉토리에 다시 배치하거나 파일을 업데이트하여 마지막으로 수정된 날짜 속성을 변경할 필요 없이 다른 전송 요청을 제출할 수 있습니다. 예를 들어 실행 기록 지우기는 파일을 전송해야 하지만 파일 수정이 불가능한 경우에 유용합니다. 자세한 정보는 238 페이지의 [『자원 모니터 실행 기록 지우기』](#)의 내용을 참조하십시오.

자원

Managed File Transfer에서의 자원 모니터는 다음 두 가지 유형의 자원 콘텐츠를 폴링할 수 있습니다.

디렉토리 또는 중첩된 디렉토리 구조

공용 시나리오는 트리거 파일의 존재 여부에 대해 디렉토리를 모니터하는 것입니다. 외부 애플리케이션은 여러 개의 파일을 처리하고 이 파일을 알려진 소스 디렉토리에 놓을 수 있습니다. 애플리케이션이 처리를 완료한 경우, 애플리케이션은 트리거 파일을 모니터할 위치에 놓아서 파일이 전송되거나 달리 작동할 준비가 되었음을 표시합니다. 트리거 파일은 Managed File Transfer 자원 모니터에 의해 감지될 수 있고 소스 디렉토리에서 다른 Managed File Transfer Agent로의 해당 파일 전송이 시작됩니다.

기본적으로, 지정된 디렉토리가 모니터됩니다. 서브디렉토리도 조사하려면 `fteCreateTransfer` 명령에서 자원 레벨을 설정하십시오.

디렉토리 모니터링의 두 예는 다음과 같습니다.

- 트리거 파일(예: `trigger.file`)을 모니터한 다음 와일드카드(예: `*.zip`)를 전송합니다.
- `*.zip`을 모니터한 다음 ``${FilePath}`(예: 전송을 트리거한 파일)를 전송합니다. 변수 대체에 대한 자세한 정보는 222 페이지의 [『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』](#)의 내용을 참조하십시오.

참고: `*.zip`을 모니터한 다음 `*.zip`을 전송하는 모니터를 작성하지 마십시오. 모니터에서 시스템의 모든 `.zip` 파일에 대한 `*.zip`의 전송을 시작하려고 합니다. 즉, 모니터가 `*.zip`에 대해 *개의 전송을 생성합니다.

디렉토리를 모니터하기 위한 자원 모니터 작성의 예는 220 페이지의 [『디렉토리 모니터링 및 변수 대체 사용』](#)의 내용을 참조하십시오.

IBM MQ 큐

모니터링 에이전트의 예에서는 외부 애플리케이션이 메시지를 생성 중이고 이러한 메시지를 동일한 그룹 ID를 가진 알려진 큐에 배치합니다. 애플리케이션이 큐에 메시지 넣기를 완료한 경우 이는 그룹이 완료된 것을 나타내는 것입니다. 전체 메시지 그룹은 Managed File Transfer 자원 모니터에 의해 감지될 수 있고 소스 큐에서 파일로의 메시지 그룹 전송이 시작됩니다. 큐를 모니터링할 자원 모니터 작성의 예는 [221 페이지의 『예: MFT 자원 구성』](#)의 내용을 참조하십시오.

참고: 큐당 하나의 모니터만 지정할 수 있습니다. IBM MQ 큐를 폴링하기 위해 하나 이상의 모니터링을 지정하는 경우 예측할 수 없는 작동이 발생합니다.

데이터 세트 모니터링은 지원되지 않습니다.

조건 및 트리거 조건

조건은 자원에 다른 문자열 또는 패턴과 일치하는 값이 있을 때 충족됩니다. 조건은 다음 중 하나가 될 수 있습니다.

- 파일 이름(패턴)에 대해 일치
- 파일 이름(패턴)에 대해 일치 안함
- 파일 크기
- 여러 폴에서 파일 크기가 동일하게 유지되는 경우 일치합니다.

파일 이름 일치하는 다음과 같이 표현할 수 있습니다.

- 정확한 문자열 일치
- **MFT와 함께 와일드카드 문자 사용에 설명된 단순 와일드카드** 일치
- 정규식 일치

절대 일치하지 않는 파일 이름을 식별하는 와일드카드 또는 Java 정규식을 사용하여 파일 이름 일치에서 파일 이름을 제외할 수도 있습니다.

일치하는 파일이 감지되면, 마지막 수정된 시간소인이 보유됩니다. 후속 폴에서 파일이 변경되었음을 감지하면 트리거 조건이 다시 충족되고 태스크가 시작됩니다. 조건이 파일이 존재하지 않는 경우를 감지하는 것이면, 언급된 디렉토리의 어떤 파일도 파일 이름 패턴과 일치하지 않는 경우 태스크가 시작됩니다. 파일이 파일 이름 패턴과 일치하는 디렉토리에 추가되면 태스크는 파일이 삭제되는 경우에만 시작됩니다.

태스크

Managed File Transfer는 자원 모니터에 의해 시작되도록 구성할 수 있는 다음 두 가지 유형의 태스크를 지원합니다.

파일 전송 태스크

파일 전송은 다른 파일 전송과 같은 방식으로 정의됩니다. 모니터에 필요한 태스크 XML을 생성하려면 **-gt** 매개변수와 함께 **fteCreateTransfer** 명령을 실행하는 방법이 유용합니다. 이 명령은 태스크 정의를 XML 문서로 생성합니다(전송 스펙 포함). 그런 다음 태스크 XML 문서의 이름을 **fteCreateMonitor** 명령에서 **-mt** 매개변수의 값으로 전달합니다. **fteCreateMonitor**가 실행되는 경우 태스크 XML 문서를 읽습니다.

fteCreateMonitor가 실행된 후 태스크 XML 파일에 대해 변경된 사항은 모니터에서 사용하지 않습니다.

파일 전송 태스크를 사용하는 경우 태스크에 배치되는 트리거 조건 수를 선택할 수 있습니다. 기본 설정은 하나의 트리거 조건마다 하나의 태스크를 시작하는 것입니다. **-bs** 옵션과 함께 **fteCreateMonitor** 명령을 실행하여 하나의 태스크로 일괄처리되는 트리거 조건 수를 선택할 수 있습니다.

명령 태스크

명령 태스크는 Ant 스크립트를 실행하거나, 실행 프로그램을 호출하거나 JCL작업을 실행할 수 있습니다. 자세한 정보는 [216 페이지의 『명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성』](#)의 내용을 참조하십시오.

트리거 파일

자원 모니터의 트리거 파일 콘텐츠를 사용하여 단일 전송 요청으로 전송할 파일 세트를 정의할 수 있습니다. 일치하는 트리거 파일이 감지될 때마다 해당 콘텐츠가 소스 파일 경로 및 선택적으로 목적지 파일 경로로 구문 분석됨

니다. 그런 다음 이러한 파일 경로는 사용자가 지정하는 태스크 전송 XML 파일에 파일 항목을 정의하는 데 사용 됩니다. 이 XML 파일은 단일 전송 요청으로 에이전트에 제출됩니다. 자원 모니터의 정의는 트리거 콘텐츠가 사용 가능한지 여부를 판별합니다.

각 트리거 파일의 양식은 각 텍스트 행에 전송할 단일 파일 경로가 있는 것입니다. 행의 기본 양식은 단일 소스 파일 경로이거나 쉘표로 구분된 소스 및 목적지 파일 경로입니다.

자세한 정보 및 예는 [230 페이지의 『트리거 파일 사용』](#)의 내용을 참조하십시오.

자원 모니터 시작 및 중지

IBM MQ 9.3.0이전에 자원 모니터를 중지하는 유일한 방법은 모니터 조작을 실행 중인 에이전트를 중지하는 것입니다. 자원 모니터를 다시 시작하려면 에이전트를 함께 다시 시작해야 합니다. 자세한 정보는 [195 페이지의 『MFT 에이전트 시작』](#) 및 [200 페이지의 『MFT 에이전트 중지』](#)의 내용을 참조하십시오.

IBM MQ 9.3.0부터는 **fteStartMonitor** 및 **fteStopMonitor** 명령을 사용하여 에이전트를 중지하거나 다시 시작할 필요 없이 자원 모니터를 시작하고 중지할 수 있습니다. 예를 들어 다음과 같은 경우 유용합니다.

- 에이전트에 여러 자원 모니터가 있으며 일부에서만 오류가 발생하고 나머지 자원 모니터는 여전히 제대로 작동하므로 실패한 자원 모니터만 다시 시작하면 되는 경우.
- 유지보수 작업을 수행하기 위해 자원 모니터를 중지하는 경우 또는 특정 시간 동안 자원 모니터가 필요하지 않으므로 불필요하게 실행하여 소중한 시스템 자원을 소비하지 않으려는 경우.

자세한 정보는 [235 페이지의 『MFT 자원 모니터 시작』](#) 및 [236 페이지의 『MFT 자원 모니터 중지』](#)의 내용을 참조하십시오.

표 9. 실행되는 명령에 따른 자원 모니터의 동작	
명령	자원 모니터의 동작
fteStartMonitor	에이전트가 실행 중인 경우 자원 모니터가 현재 중지되어 있으면 시작됩니다.
fteStopMonitor	에이전트가 실행 중인 경우 자원 모니터가 현재 시작되어 있으면 중지됩니다.
fteStartAgent	자원 모니터는 에이전트 시작(fteStopMonitor 에 대한 이전 호출과 무관)의 일부로 시작됩니다.
fteStopAgent	실행 중인 모든 자원 모니터가 중지됩니다.

자원 모니터 백업 및 복원

이후 다시 사용할 수 있도록 이미 정의된 자원 모니터를 백업할 수 있습니다. 다음과 같이 사용할 수 있는 여러 옵션이 있습니다.

- **fteCreateMonitor** 명령을 **-ox** 매개변수와 함께 사용하여 XML 파일에 자원 모니터 구성을 내보내고, **-ix** 매개변수와 함께 사용하는 경우 XML 파일에서 자원 모니터 구성을 가져와 자원 모니터를 복원합니다.
- **fteListMonitors** 명령을 **-ox**와 함께 사용하여 단일 자원 모니터의 정의를 XML 파일로 내보냅니다.
- **fteListMonitors** 명령을 **-od**와 함께 사용하여 다중 자원 모니터 정의를 지정된 디렉토리로 내보냅니다. 각 자원 모니터 정의는 별도의 XML 파일에 저장됩니다. 또한 **-od** 옵션을 사용하여 단일 자원 모니터 정의를 지정된 디렉토리로 내보낼 수 있습니다.

자세한 정보는 [237 페이지의 『MFT 자원 모니터의 백업 및 복원』](#)의 내용을 참조하십시오.

자원 모니터 로깅

Managed File Transfer에는 자원 모니터 로깅이 포함되어 있습니다. 자세한 정보는 [233 페이지의 『MFT 자원 모니터 로깅』](#)의 내용을 참조하십시오.

관련 개념

[222 페이지의 『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』](#)

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

관련 태스크

216 페이지의 『명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성』

자원 모니터는 연관된 태스크로 파일 전송을 수행하도록 제한되지 않습니다. 실행 가능 프로그램, Ant 스크립트 또는 JCL 작업을 포함하여 모니터링 에이전트에서 다른 명령을 호출하도록 모니터를 구성할 수도 있습니다. 명령을 호출하려면 인수 및 특성과 같은 해당되는 명령 호출 매개변수와 함께 하나 이상의 명령 요소를 포함하도록 모니터 태스크 정의 XML을 편집하십시오.

221 페이지의 『예: MFT 자원 구성』

fteCreateMonitor 명령과 함께 **-mq** 매개변수를 사용하여 자원 모니터가 모니터링할 자원으로 IBM MQ 큐를 지정할 수 있습니다.

227 페이지의 『큐 모니터링 및 변수 대체 사용』

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

관련 참조

fteCreateMonitor: MFT 자원 모니터 작성

fteListMonitors: MFT 자원 모니터 나열

fteDeleteMonitor: MFT 자원 모니터 삭제

명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성

자원 모니터는 연관된 태스크로 파일 전송을 수행하도록 제한되지 않습니다. 실행 가능 프로그램, Ant 스크립트 또는 JCL 작업을 포함하여 모니터링 에이전트에서 다른 명령을 호출하도록 모니터를 구성할 수도 있습니다. 명령을 호출하려면 인수 및 특성과 같은 해당되는 명령 호출 매개변수와 함께 하나 이상의 명령 요소를 포함하도록 모니터 태스크 정의 XML을 편집하십시오.

이 태스크 정보

모니터링 에이전트가 호출하게 하려는 실행 프로그램, Ant 스크립트 또는 JCL 작업에 대한 파일경로는 모니터링 에이전트의 `commandPath`에 포함되어야 합니다. 명령 경로 특성에 대한 정보는 [commandPath MFT 특성을 참조하십시오](#).

다음 방법 중 한 가지 방법으로 태스크 정의 XML 문서를 작성할 수 있습니다.

- `FileTransfer.xsd` 스키마에 따라 수동으로 태스크 정의 XML 문서를 작성하십시오.
- 생성된 XML 문서를 태스크 정의의 기초로 사용하십시오.

전송 태스크 또는 명령 태스크를 원하는지에 관계없이 태스크 정의는 `<request>` 루트 요소로 시작해야 합니다. `<request>`의 하위 요소는 `<managedTransfer>` 또는 `<managedCall>`이어야 합니다. 일반적으로 실행할 단일 명령 또는 스크립트가 있는 경우 `<managedCall>`을 선택하고, 태스크가 하나의 파일 전송과 선택적으로 네 개까지의 명령 호출을 포함하도록 하려면 `<managedTransfer>`를 선택합니다.

프로시저

- `FileTransfer.xsd` 스키마에 따라 수동으로 태스크 정의 XML 문서를 작성하려면 [216 페이지의 『스키마에 따라 수동으로 태스크 정의 XML 작성』](#)의 내용을 참조하십시오.
- 생성된 문서를 수정하여 태스크 정의를 작성하려면 **fteCreateTransfer -gt** 매개변수에 의해 생성된 XML 문서를 편집하십시오. 자세한 정보는 [218 페이지의 『생성된 문서를 수정하여 태스크 정의 문서 작성』](#)의 내용을 참조하십시오.

스키마에 따라 수동으로 태스크 정의 XML 작성

`FileTransfer.xsd` 스키마에 따라 태스크 정의 XML 파일을 수동으로 작성할 수 있습니다.

이 태스크 정보

FileTransfer.xsd 스키마는 `MQ_INSTALLATION_PATH/mqft/samples/schema`에서 찾을 수 있습니다. 이 스키마에 대한 자세한 정보는 [파일 전송 요청 메시지 형식](#)을 참조하십시오.

예

다음 예에서는 `<managedCall>` 요소를 사용하여 `RunCleanup.xml`(이)라는 Ant 스크립트를 호출하는 `cleanuptask.xml`, (으)로 저장된 예제 태스크 정의 XML 문서를 보여줍니다. `RunCleanup.xml` Ant 스크립트는 모니터링 에이전트의 `commandPath`에 있어야 합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <target>copy_to_archive</target>
          <target>rename_temps</target>
          <target>delete_files</target>
          <property name="trigger.filename" value="{FileName}"/>
          <property name="trigger.path" value="{FilePath}"/>
        </command>
      </call>
    </transferSet>
  </job>
  <name>JOBCLEAN1</name>
</managedCall>
</request>
```

`<agent>` 요소는 `commandPath`에서 Ant(이)라는 스크립트로 구성된 Managed File Transfer Agent(를) 지정합니다.

`<call><command>`... 구조는 실행할 실행 파일 또는 스크립트를 정의합니다. 명령은 값이 다음 중 하나일 수 있는 선택적 `type` 속성을 사용합니다.

antscript

별도의 JVM에서 Ant 스크립트를 실행하십시오.

executable

실행 가능 프로그램을 호출합니다.

jcl

JCL 작업을 호출합니다.

`type` 속성을 생략하면 기본값 실행 파일이 사용됩니다.

`name` 속성은 경로 정보 없이 실행하려는 Ant 스크립트, 실행 파일 또는 JCL 작업의 이름을 지정합니다. 에이전트는 에이전트의 `agent.properties` 파일에 있는 `commandPath` 특성에 지정된 위치에서 스크립트 또는 프로그램을 검색합니다.

`retrycount` 속성은 프로그램이 성공 리턴 코드를 리턴하지 않는 경우 프로그램을 다시 호출하려고 시도하는 횟수를 지정합니다. 이 속성에 지정된 값은 음수가 아니어야 합니다. `retrycount` 속성을 지정하지 않으면 기본값 0이 사용됩니다.

`retrywait` 속성은 프로그램 호출을 다시 시도하기 전에 대기하는 시간(초)을 지정합니다. 이 속성에 지정된 값은 음수가 아니어야 합니다. `retrywait` 속성을 지정하지 않으면 기본값 0이 사용됩니다.

successrc 속성은 프로그램 호출이 성공적으로 실행되는 시기를 판별하는 데 사용되는 표현식입니다. 이 표현식을 사용하여 명령의 프로세스 리턴 코드를 평가합니다. 값은 Boolean OR을 나타내는 막대 기호(!) 또는 Boolean AND를 나타내는 앰퍼샌드(&) 문자가 결합된 하나 이상의 표현식으로 구성될 수 있습니다. 각 표현식은 다음 유형의 표현식 중 하나입니다.

- 프로세스 리턴 코드와 숫자 사이에 동등 테스트를 표시하는 숫자
- 숫자와 프로세스 리턴 코드 사이에 보다 큼(greater-than) 테스트를 표시하기 위해 보다 큼 문자(>)가 접두부에 사용된 숫자
- 숫자와 프로세스 리턴 코드 사이에 미만 테스트를 표시하기 위해 보다 작음 문자(<)가 접두부에 사용된 숫자
- 숫자와 프로세스 리턴 코드 사이에 같지 않음(not-equal-to) 테스트를 표시하기 위해 느낌표(!)가 접두부에 사용된 숫자. 예를 들어, >2&<7&!5|0|14는 0, 3, 4, 6, 14와 같은 성공 리턴 코드로 해석됩니다. 다른 모든 리턴 코드는 실패로 해석됩니다.

successrc 속성을 지정하지 않으면 기본값 0이 사용됩니다. 이것은 코드 0을 리턴하는 경우에만 명령이 성공적으로 실행된 것으로 평가됨을 의미합니다.

Ant 스크립트의 경우 일반적으로 <target> 및 <property> 요소를 지정합니다. <target> 요소 값은 Ant 스크립트의 대상 이름과 일치해야 합니다.

실행 가능 프로그램의 경우 <argument> 요소를 지정할 수 있습니다. 중첩 argument 요소는 프로그램 호출의 일부로 호출되는 프로그램에 전달할 인수를 지정합니다. 프로그램 인수는 argument 요소가 발견된 순서대로 argument 요소에 지정된 값을 통해 빌드됩니다. 0개 이상의 argument 요소를 프로그램 호출의 중첩 요소로 지정할 수 있습니다.

관리자는 <managedCall> 요소를 포함하는 태스크 정의 XML 문서를 사용하여 모니터를 정상적으로 정의하고 시작합니다. 예를 들면, 다음과 같습니다.

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt /tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*.go
```

전송 정의 XML 문서의 경로는 **fteCreateMonitor** 명령을 실행하는 로컬 파일 시스템에 있어야 합니다(이 예에서는 /tasks/cleanuptask.xml). cleanuptask.xml 문서는 자원 모니터를 작성하는 데만 사용됩니다. cleanuptask.xml 문서가 참조하는 모든 태스크(Ant 스크립트 또는 JCL 작업)는 모니터링 에이전트의 명령 경로에 있어야 합니다. 모니터 트리거 조건이 충족되면, 태스크 정의 XML의 변수가 모니터의 실제 값으로 대체됩니다. 따라서 예를 들어 \${FilePath}는 에이전트로 전송되는 요청 메시지에서 /monitored/cleanup.go로 대체됩니다. 요청 메시지는 에이전트 명령 큐에 추가됩니다. 명령 프로세서는 요청이 프로그램 호출에 대한 요청임을 감지하고 지정된 프로그램을 시작합니다. antscript 유형의 명령이 호출되면 새 JVM이 시작되고 Ant 태스크는 새 JVM에서 실행됩니다. 변수 대체 사용에 대한 자세한 정보는 [변수 대체로 태스크 사용자 정의를 참조하십시오](#).

관련 개념

222 페이지의 『[변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의](#)』

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

관련 참조

[파일 전송 요청 메시지 형식](#)

[commandPath MFT 특성](#)

생성된 문서를 수정하여 태스크 정의 문서 작성

fteCreateTransfer의 **-gt** 옵션에 의해 생성된 XML 문서를 수정하여 모니터 태스크 정의 문서를 작성할 수 있습니다.

이 태스크 정보

생성된 문서에는 <request> 다음에 <managedTransfer> 요소가 있습니다. 이 태스크 정의를 유효한 <managedCall> 구조로 변환하려면 다음 단계를 수행하십시오.

프로시저

1. <managedTransfer> 시작 및 종료 태그를 <managedCall> 태그로 바꾸십시오.
2. <schedule> 요소 및 하위 노드를 제거하십시오.
3. 모니터링 에이전트 구성 세부사항과 일치하도록 <sourceAgent> 시작 및 종료 태그를 <agent>(으)로 바꾸십시오.
4. <destinationAgent> 및 <trigger> 요소를 제거하십시오.
5. <item> 요소를 제거하십시오.
6. preSourceCall, postSourceCall, preDestinationCall 또는 postDestinationCall 요소를 제거하십시오.
7. <transferSet> 요소에 새 <call>...</call> 구조를 삽입하십시오. 이 구조에는 다음 예에 표시된 대로 명령 정의가 포함되어 있습니다.

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="{FileName}"/>
    <property name="trigger.path" value="{FilePath}"/>
  </command>
</call>
```

예

또한 모든 파일 전송 세부사항을 포함하여 <managedTransfer> 요소를 보유할 수 있으며 최대 네 개의 명령 호출을 삽입할 수 있습니다. 이 경우 <metadataSet> 및 <item> 요소 사이에 다음 호출 요소의 선택사항을 삽입합니다.

preSourceCall

전송을 시작하기 전에 소스 에이전트에 대해 프로그램을 호출합니다.

postSourceCall

전송을 완료한 후 소스 에이전트에 대해 프로그램을 호출합니다.

preDestinationCall

전송을 시작하기 전에 목적지 에이전트에 대해 프로그램을 호출합니다.

postDestinationCall

전송을 완료한 후 목적지 에이전트에 대해 프로그램을 호출합니다.

해당 요소는 각각 이전 예에 설명된 대로 <command> 요소 구조를 사용합니다. FileTransfer.xsd 스키마는 다양한 호출 요소에 사용되는 유형을 정의합니다.

다음 예제는 태스크 정의 문서의 preSourceCall, postSourceCall, preDestinationCall 및 postDestinationCall을 보여줍니다.

```
⋮
<transferSet priority="1">
  <metadataSet>
    <metaData key="key1">value1</metaData>
  </metadataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
    type="executable">
      <argument>report1.pdf</argument>
      <argument>>true</argument>
    </command>
  </preSourceCall>
  <postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
    type="jcl">
      <argument>argument</argument>
    </command>
  </postSourceCall>
```

```

<preDestinationCall>
  <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
    type="antscript">
    <target>step1</target>
    <property name="name" value="value"/>
  </command>
</preDestinationCall>
<postDestinationCall>
  <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
</postDestinationCall>
<item checksumMethod="none" mode="binary">

```

다른 유형의 명령을 전송에 혼합할 수 있습니다. argument, target 및 property 요소는 선택사항입니다.

디렉토리 모니터링 및 변수 대체 사용

fteCreateMonitor 명령을 사용하여 디렉토리를 모니터링할 수 있습니다. 대체 변수의 값은 태스크 XML 정의에서 대체하고 전송 동작을 정의하는 데 사용할 수 있습니다.

이 태스크 정보

이 예에서 소스 에이전트는 AGENT_HOP입니다. AGENT_HOP 모니터가 모니터링하는 디렉토리는 /test/monitored입니다. 에이전트는 5분마다 디렉토리를 폴링합니다.

.zip 파일이 디렉토리에 기록된 후 파일을 디렉토리에 기록하는 애플리케이션이 동일한 디렉토리에 트리거 파일을 기록합니다. 트리거 파일의 이름은 .zip 파일의 이름과 동일하지만 파일 확장자가 다릅니다. 예를 들어, file1.zip 파일이 디렉토리에 기록된 후 file1.go 파일이 디렉토리에 기록됩니다. 자원 모니터는 디렉토리에 *.go 패턴과 일치하는 파일이 있는지 모니터링한 후 변수 대체를 사용하여 연관된 .zip 파일의 전송을 요청합니다.

프로시저

1. 모니터가 트리거되는 경우 수행하는 태스크를 정의하는 태스크 XML을 작성하십시오.

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/${fileName}{token=1}{separator=.}.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/${fileName}{token=1}{separator=.}.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

트리거 파일과 연관된 값으로 바뀌는 변수는 굵게 강조표시되어 있습니다. 이 태스크 XML은 /home/USER1/task.xml 파일에 저장됩니다.

2. /test/monitored 디렉토리를 모니터링하도록 자원 모니터를 작성하십시오. 다음 명령을 제출하십시오.

```

fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*.go -pi 5 -pu minutes

```

3. 사용자 또는 프로그램이 `jump.zip` 파일을 `/test/monitored` 디렉토리에 기록한 후 `jump.go` 파일을 디렉토리에 기록합니다.
4. `jump.go` 파일이 존재하면 모니터가 트리거됩니다. 에이전트가 트리거 파일에 대한 정보를 태스크 XML로 대체합니다.
이 결과로 태스크 XML이 다음과 같이 변환됩니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/jump.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/jump.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

결과

태스크 XML로 정의한 전송이 수행됩니다. `jump.zip` 파일은 `AGENT_HOP`가 `/test/monitored` 디렉토리에 서 읽고 `AGENT_SKIP`이 실행 중인 시스템에 있는 `/out/jump.zip` 파일로 전송됩니다.

관련 개념

222 페이지의 『[변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의](#)』

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

관련 태스크

216 페이지의 『[명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성](#)』

자원 모니터는 연관된 태스크로 파일 전송을 수행하도록 제한되지 않습니다. 실행 가능 프로그램, Ant 스크립트 또는 JCL 작업을 포함하여 모니터링 에이전트에서 다른 명령을 호출하도록 모니터를 구성할 수도 있습니다. 명령을 호출하려면 인수 및 특성과 같은 해당되는 명령 호출 매개변수와 함께 하나 이상의 명령 요소를 포함하도록 모니터 태스크 정의 XML을 편집하십시오.

관련 참조

[fteCreateMonitor](#): MFT 자원 모니터 작성

예: MFT 자원 구성

`fteCreateMonitor` 명령과 함께 `-mq` 매개변수를 사용하여 자원 모니터가 모니터링 자원으로 IBM MQ 큐를 지정할 수 있습니다.

이 태스크 정보

이 예에서 모니터링 자원은 `MONITORED_QUEUE` 큐입니다. 이 큐는 모니터링 에이전트의 큐 관리자 `QM_NEPTUNE`에 있어야 합니다. 큐를 모니터링하는 조건은 메시지의 완료 그룹이 존재하는지 여부입니다. 조건이 충족되는 경우 수행될 태스크는 `task.xml` 파일에 정의되어 있습니다.

참고: 둘 이상의 자원 모니터를 작성하여 개별 큐를 모니터링하지 마십시오. 그렇지 않으면 예상할 수 없는 작동이 발생합니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

`completeGroups` 조건이 `true`인지 확인하기 위해 모니터가 5분마다 큐를 검사합니다. 큐에 하나 이상의 완전한 그룹이 있는 경우 모니터가 각각의 완전한 그룹에 대해 한 번씩 `task.xml` 파일에 정의된 태스크를 실행합니다.

관련 개념

222 페이지의 『[변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의](#)』

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

관련 태스크

216 페이지의 『[명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성](#)』

자원 모니터는 연관된 태스크로 파일 전송을 수행하도록 제한되지 않습니다. 실행 가능 프로그램, Ant 스크립트 또는 JCL 작업을 포함하여 모니터링 에이전트에서 다른 명령을 호출하도록 모니터를 구성할 수도 있습니다. 명령을 호출하려면 인수 및 특성과 같은 해당되는 명령 호출 매개변수와 함께 하나 이상의 명령 요소를 포함하도록 모니터 태스크 정의 XML을 편집하십시오.

227 페이지의 『[큐 모니터링 및 변수 대체 사용](#)』

fteCreateMonitor 명령을 사용하여 큐를 모니터하고 메시지를 모니터되는 큐에서 파일로 전송할 수 있습니다. 모니터되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

관련 참조

fteCreateMonitor: MFT 자원 모니터 작성

변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.



주의: 변수 이름은 대소문자를 구분하지 않습니다.

대체에 사용되는 변수는 긍정적인 트리거 조건에만 사용할 수 있습니다. `match` 및 `fileSize` 트리거 조건만 변수가 대체되도록 합니다. `noMatch` 조건이 사용되고 태스크 정의에 대체 변수 이름이 있는 경우, 태스크가 호출되지 않고 모니터가 리턴 코드 110 및 오류 메시지 BFGDM0060E를 발생시킵니다.

모니터되는 자원이 큐인 경우

모니터되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있습니다. 사용자 정의 메시지 특성의 접두부는 `usr.`이지만 변수 이름에 이 접두부는 포함되지 않습니다. 변수 이름 앞에는 달러 기호(\$)가 오고 중괄호{}로 묶어야 합니다.

예를 들면, `${destFileName}`은 소스 큐에서 읽을 첫 번째 메시지의 `usr.destFileName` 메시지 특성 값으로 바뀝니다. 자세한 정보는 소스 큐의 메시지에서 MFT가 읽은 MQ 메시지 특성 및 227 페이지의 『[큐 모니터링 및 변수 대체 사용](#)』의 내용을 참조하십시오.

변수가 메시지 특성으로 정의되지 않은 경우 모니터에서 BFGDM0060E 오류를 보고하고 리턴 코드 110(모니터 태스크 변수 대체에 실패함)을 리턴합니다. 또한 에이전트가 다음 오류 메시지를 이벤트 로그(outputN.log)에 기록합니다.

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

모니터에 대한 중간 또는 상세 자원 모니터 로깅이 사용으로 설정된 경우 모니터가 다음 메시지를 에이전트의 자원 모니터 이벤트 로그(resmoneventN.log)에 기록합니다.

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

자원 모니터 로깅에 대한 자세한 정보는 233 페이지의 『MFT 자원 모니터 로깅』의 내용을 참조하십시오.

다음 테이블에서는 기본적으로 제공되는 대체 변수를 표시합니다. 예를 들어 \${AGENTNAME}은 자원 모니터 에이전트의 이름으로 대체됩니다.

표 10. 기본값으로 제공되는 대체 변수	
변수	설명
AGENTNAME	자원 모니터 에이전트의 이름입니다.
QUEUENAME	모니터 중인 큐의 이름입니다.
ENCODING	큐에 있는 첫 번째 메시지 또는 그룹에 있는 첫 번째 메시지의 문자 인코딩입니다.
MESSAGEID	큐에 있는 첫 번째 메시지 또는 그룹에 있는 첫 번째 메시지의 IBM MQ 메시지 ID입니다.
GROUPID	단일 메시지만 찾은 경우 그룹의 IBM MQ 그룹 ID 또는 메시지 ID입니다. 이 변수는 완료 그룹을 모니터 중인 경우에만 설정됩니다.
CurrentTimeStamp	모니터가 트리거되는 로컬 시간을 기반으로 하는 시간소인입니다. 시간소인 값은 에이전트에 대해 고유합니다.
CurrentTimeStamp UTC	모니터가 트리거되는 UTC 시간대의 시간을 기반으로 하는 시간소인입니다. 시간소인 값은 에이전트에 대해 고유합니다.

모니터되는 자원이 디렉토리인 경우

다음 테이블에서는 태스크 XML 정의에서 대체될 수 있는 변수 이름 세트를 표시합니다.

표 11. 대체될 수 있는 변수	
변수	설명
FilePath	트리거 파일의 전체 경로 이름입니다.
FileName	트리거의 파일 이름 부분입니다.
LastModifiedTime	트리거 파일을 마지막으로 수정한 시간입니다. 이 시간은 에이전트가 실행되는 시간대의 로컬 시간으로 표시되고 ISO 8601 시간으로 형식화됩니다.
LastModifiedDate	트리거 파일을 마지막으로 수정한 날짜입니다. 이 날짜는 에이전트가 실행 중인 시간대의 로컬 날짜로 표시되고 ISO 8601 날짜로 형식화됩니다.
LastModifiedTimeUTC	트리거 파일을 마지막으로 수정한 시간입니다. 이 시간은 UTC 시간대로 변환된 로컬 시간으로 표시되고 ISO 8601 시간으로 형식화됩니다.
LastModifiedDateUTC	트리거 파일을 마지막으로 수정한 날짜입니다. 이 날짜는 UTC 시간대로 변환된 로컬 날짜로 표현되고 ISO 8601 날짜로 양식화됩니다.
AgentName	자원 모니터 에이전트의 이름입니다.

표 11. 대체될 수 있는 변수 (계속)	
변수	설명
CurrentTimeSt amp	모니터가 트리거된 로컬 시간을 기반으로 하는 시간소인입니다. 시간소인 값은 에이전트에 대해 고유합니다.
CurrentTimeSt ampUTC	모니터가 트리거된 UTC 시간대의 시간을 기반으로 하는 시간소인입니다. 시간소인 값은 에이전트에 대해 고유합니다.

모니터되는 자원이 트리거 파일인 경우

다음 테이블에서는 자원 모니터가 트리거 파일의 콘텐츠를 사용하여 전송해야 하는 파일을 판별할 때 대체될 수 있는 변수 이름 세트를 표시합니다.

표 12. 트리거 파일을 사용할 때 대체될 수 있는 변수	
변수	설명
contentSource	소스 파일의 전체 경로 이름입니다.
contentDestina tion	목적지 파일의 전체 경로 이름입니다.

변수 이름 앞에는 달러 기호(\$) 문자가 있어야 하고 변수 이름은 중괄호({})로 묶어야 합니다. 예를 들어, \${FilePath}는 일치하는 트리거 파일의 완전한 파일 경로로 대체됩니다.

추가적인 세밀함을 제공하기 위해 변수 이름에 적용할 수 있는 두 가지의 특수 키워드가 있습니다. 이러한 항목은 다음과 같습니다.

토큰

대체할 토큰 색인(왼쪽으로부터는 1에서 시작하고 오른쪽으로부터는 -1에서 시작됨)

구분 기호

변수 값을 토큰화하기 위한 단일 문자. 기본값은 AIX and Linux 플랫폼의 경우 슬래시 문자 (/) 또는 Windows 플랫폼의 경우 백슬래시 문자 (\) 이지만, 구분 기호는 변수 값에 표시될 수 있는 유효한 문자일 수 있습니다.

separator 키워드를 변수 이름에 지정한 경우 변수 값은 separator 문자에 해당되는 토큰으로 분리됩니다.

token 키워드에 지정되는 값은 변수 이름을 바꾸기 위해 사용할 토큰을 선택하는 데 필요한 색인으로 사용됩니다. 토큰 색인은 변수의 첫 번째 문자에 상대적이며 1부터 시작합니다. token 키워드를 지정하지 않으면 전체 변수가 삽입됩니다.

메시지 XML에서 에이전트 이름으로 대체되는 값은 대소문자를 구분하지 않는 방식으로 처리됩니다. 모든 Managed File Transfer Agent 이름은 대문자입니다. Paris 값이 메시지 XML에서 에이전트 속성으로 대체되는 경우, 이 값은 에이전트 PARIS에 대한 참조로 해석됩니다.

관련 개념

224 페이지의 『예제: 자원 모니터 정의에 대한 변수 대체』

XML 및 IBM MQ Explorer을(를) 사용한 자원 모니터 정의의 변수 대체 예제입니다.

관련 태스크

변수 대체로 인해 여러 파일이 단일 파일 이름이 되는 경우 수행할 작업

예제: 자원 모니터 정의에 대한 변수 대체

XML 및 IBM MQ Explorer을(를) 사용한 자원 모니터 정의의 변수 대체 예제입니다.

변수 대체 작동 방법을 표시하는 예

일치하는 트리거 파일에 대한 파일 경로가 c:\MONITOR\REPORTS\Paris\Report2009.doc (Windows의 경우) 및 /MONITOR/REPORTS/Paris/Report2009.doc (AIX and Linux 플랫폼의 경우) 라고 가정하면 변수가 다음 표에 표시된 대로 대체됩니다.

표 13. 변수 대체 방법	
변수 스펙	변수 대체 이후
<code>\${FilePath}</code>	Windows : c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=1}{separator=.}}</code>	Windows : c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=2}{separator=.}}</code>	Windows : 문서 AIX and Linux : 문서
<code>\${FilePath{token=3}}</code>	Windows : 보고서 AIX and Linux : 파리

다음 테이블에 표시된 대로 음수 토큰 색인을 지정하여 변수의 마지막 문자에 관련된 토큰을 선택할 수도 있습니다. 표의 예에서는 동일한 변수 값 (Windows의 경우 c:\MONITOR\REPORTS\Paris\Report2009.doc, AIX and Linux의 경우 /MONITOR/REPORTS/Paris/Report2009.doc) 을 사용합니다.

표 14. 음수 토큰 색인 사용 예제	
변수 스펙	변수 대체 이후
<code>\${FilePath}</code>	Windows : c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows : c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\}}</code>	Windows : 파리 AIX and Linux : 파리
<code>\${FilePath{token=-4}}</code>	Windows : 모니터 AIX and Linux : 모니터

대체에 사용되는 변수는 다음과 같은 긍정적인 트리거 조건과 긍정적인 트리거 조건 규칙에 대한 예외인 noSizeChange 옵션에만 사용할 수 있습니다.

- match
- fileSize

- noSizeChange

noMatch 조건이 사용되고 태스크 정의에 대체 변수 이름이 있는 경우, 태스크가 호출되지 않고 모니터가 리턴 코드 110 및 오류 메시지 BFGDM0060E를 발생시킵니다.

XML 사용 예제

다음 예제 태스크 정의 XML에서는 전송을 위한 소스 에이전트로서 모니터 에이전트 이름(Paris)을 사용하고, 전송을 위한 목적지 에이전트 이름으로서 파일 경로에 있는 끝에서 두 번째에 있는 디렉토리 이름 (Report2009)을 사용하며, 확장자가 .rpt인 트리거 파일 이름의 루트가 되도록 전송된 파일의 이름을 바꿉니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="{AgentName}" QMgr="QM1" />
    <destinationAgent agent="{FilePath}{token=-2}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{FileName}{token=1}{separator=}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

이 결과로 태스크 XML이 다음과 같이 변환됩니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="QM1" />
    <destinationAgent agent="Paris" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/Report2009.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

<destinationAgent> 요소의 agent 속성에 있는 \${FilePath}{token=-2} 변수가 Paris 값으로 바뀝니다. 이 값은 대소문자를 구분하지 않는 방식으로 처리되고 에이전트 PARIS에 대한 참조로서 해석됩니다.

IBM MQ Explorer 사용 예제

IBM MQ Explorer를 통해 자원 모니터를 작성하고 일단 모니터 특성과 트리거 조건이 지정되면, 모니터에 전송 항목을 추가하기 위한 옵션이 제공됩니다. 다음 예제에서는 \${FilePath} 및 \${FileName} 변수가 "전송 항목 추가 패널"에서 사용되어 자원 모니터 일치에서 발생하는 전송을 사용자 정의하는 방법에 대해 설명합니다.

예제 1

트리거 조건을 충족할 때 소스 파일을 다른 위치로 간단히 전송하기 위해서는 `${FilePath}` 변수가 사용될 수 있습니다.

- 소스 파일 이름이 `${FilePath}`가 되도록 설정합니다.
- 목적지의 유형 드롭 다운 메뉴에서 **디렉토리**를 선택하십시오.
- 목적지 **파일 이름**을 소스 파일이 전송될 위치로 설정하십시오. 예를 들어, `C:\MFT\out\`이 될 수 있습니다.

예 2

소스 파일을 다른 위치로 전송하고 파일의 확장자를 변경하기 위해 `${FileName}` 변수를 `${FilePath}` 변수와 함께 사용할 수 있습니다.

다음 예제에서는 소스 파일의 파일 경로가 `C:\MONITOR\REPORTS\Paris\Report2009.doc`와 동일하다고 가정합니다.

- 소스 파일 이름이 `${FilePath}`가 되도록 설정합니다.
- 소스 파일을 전송할 위치 다음에 `${FileName}{token=1}{separator=.}`가 오고 그 다음에 파일의 새 확장자가 오도록 목적지 **파일 이름**을 설정하십시오. 예를 들어, `C:\MFT\out\${FileName}{token=1}{separator=.}.rpt`이고, 이는 소스 파일 이름이 포함된 `C:\MFT\out\Report2009.rpt`와 동일합니다.

예제 3

소스 파일의 파일 경로 일부를 사용하여 전송 목적지를 판별하기 위해, `${FilePath}` 변수를 토큰 및 구분 기호 스펙과 함께 사용할 수 있습니다.

다음 예제에서는 소스 파일의 파일 경로가 `C:\MONITOR\REPORTS\Paris\Report2009.doc`와 동일하다고 가정합니다.

소스 파일 경로의 일부를 사용하여 파일의 목적지를 판별할 수 있습니다. 소스 파일의 위치에 따라 파일이 폴더 (이 예제에서는 `Paris`)에 전송된 경우 파일 경로 예제 `C:\MONITOR\REPORTS\Paris\Report2009.doc`를 사용하면 다음이 수행될 수 있습니다.

- 소스 파일 이름이 `${FilePath}`가 되도록 설정합니다.
- 각 위치에 대한 폴더가 있는 목적지가 되도록 목적지 **파일 이름**을 설정하고, 파일 경로의 목적지 파트와 파일 이름을 추가하십시오. 예를 들어, `C:\MFT\out\${FilePath}{token=-2}{separator=\}\${FileName}`이고, 이는 소스 파일 이름이 포함된 `C:\MFT\out\Paris\Report2009.doc`와 동일합니다.

관련 개념

222 페이지의 『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

관련 태스크

[변수 대체로 인해 여러 파일이 단일 파일 이름이 되는 경우 수행할 작업](#)

큐 모니터링 및 변수 대체 사용

`fteCreateMonitor` 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

이 태스크 정보

이 예제에서는 소스 에이전트를 `AGENT_VENUS`라고 하며 이 에이전트가 `QM_VENUS`에 연결됩니다. `AGENT_VENUS`가 모니터링하는 큐는 `START_QUEUE`이며 `QM_VENUS`에 있습니다. 에이전트는 30분마다 큐를 폴링합니다.

메시지의 완료 그룹이 큐에 기록되면 모니터 태스크가 여러 목적지 에이전트 중 하나에 있는 파일에 메시지 그룹을 보내며 모든 목적지 에이전트는 큐 관리자 QM_MARS에 연결됩니다. 메시지 그룹이 전송되는 파일의 이름은 그룹에 있는 첫 번째 메시지의 IBM MQ 메시지 특성 `usr.fileName`을 통해 정의됩니다. 메시지 그룹이 송신되는 에이전트의 이름은 그룹에 있는 첫 번째 메시지의 IBM MQ 메시지 특성 `usr.toAgent`를 통해 정의됩니다. `usr.toAgent` 헤더가 설정되지 않은 경우 목적지 에이전트에서 사용될 기본값은 AGENT_MAGENTA입니다.

`useGroups="true"`를 지정할 경우 `groupId="${GROUPID}"`를 지정하지 않으면, 전송에서는 큐의 첫 번째 메시지만 사용합니다. 따라서 예를 들어, 변수 대체를 사용하여 `fileName`을 생성하는 경우 `a.txt`의 콘텐츠가 올바르게 바뀔 수 있습니다. 모니터에서 `fileName`을 생성하지만 전송 시 `fileName`이라는 파일을 생성해야 하는 메시지가 아닌 다른 메시지를 실제로 가져오기 때문입니다.

프로시저

1. 모니터가 트리거되는 경우 수행하는 태스크를 정의하는 태스크 XML을 작성하십시오.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="${toAgent}" QMgr="QM_MARS"/>
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

IBM MQ 메시지 헤더의 값으로 바뀌는 변수는 굵은체로 강조표시되어 있습니다. 이 태스크 XML은 `/home/USER1/task.xml` 파일에 저장됩니다.

2. 자원 모니터를 작성하여 START_QUEUE 큐를 모니터하십시오.

다음 명령을 제출하십시오.

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. 사용자 또는 프로그램이 메시지 그룹을 START_QUEUE 큐에 기록합니다.

이 그룹의 첫 번째 메시지에 다음 IBM MQ 메시지 특성이 설정되어 있습니다.

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. 완료 그룹이 기록되면 모니터가 트리거됩니다. 에이전트가 IBM MQ 메시지 특성을 태스크 XML로 대체합니다.

이 결과로 태스크 XML이 다음과 같이 변환됩니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
```

```

    <userID>USER1</userID>
  </originator>
  <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
  <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
  <transferSet>
    <item mode="binary" checksumMethod="none">
      <source>
        <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
      </source>
      <destination type="file" exist="overwrite">
        <file>/reports/larmer.rpt</file>
      </destination>
    </item>
  </transferSet>
</managedTransfer>
</request>

```

결과

태스크 XML에 의해 정의되는 전송이 수행됩니다. AGENT_VENUS가 START_QUEUE에서 읽은 전체 메시지 그룹은 AGENT_VIOLET이 실행 중인 시스템의 /reports/larmer.rpt 파일에 기록됩니다.

다음에 수행할 작업

개별 파일에 각 메시지 전송

큐를 모니터링하려는 경우 모든 메시지를 개별 파일로 전송했으면 이 주제에서 이전에 설명한 것과 비슷한 기법을 사용할 수 있습니다.

1. **ftCreateMonitor** 명령에 **-tr completeGroups** 매개변수를 지정하여 이전에 설명한 대로 모니터를 작성하십시오.
2. 태스크 XML에서 다음을 지정하십시오.

```
<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
```

그러나 소스 큐에 메시지를 넣는 경우 IBM MQ 그룹에 넣지 마십시오. IBM MQ 메시지 특성을 각 메시지에 추가하십시오. 예를 들어, 각 메시지마다 고유 파일 이름 값을 사용하여 **usr.filename** 특성을 지정하십시오. 이 경우 Managed File Transfer Agent가 효과적으로 소스 큐의 각 메시지를 개별 그룹으로 처리합니다.

관련 개념

[256 페이지의 『메시지에서 파일로 데이터 전송』](#)

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

[222 페이지의 『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』](#)

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

큐 자원 모니터에 의해 시작된 전송에 의해 작성된 목적지 파일에 잘못된 데이터가 포함된 경우 수행할 작업

관련 태스크

[216 페이지의 『명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성』](#)

자원 모니터는 연관된 태스크로 파일 전송을 수행하도록 제한되지 않습니다. 실행 가능 프로그램, Ant 스크립트 또는 JCL 작업을 포함하여 모니터링 에이전트에서 다른 명령을 호출하도록 모니터를 구성할 수도 있습니다. 명령을 호출하려면 인수 및 특성과 같은 해당되는 명령 호출 매개변수와 함께 하나 이상의 명령 요소를 포함하도록 모니터 태스크 정의 XML을 편집하십시오.

[221 페이지의 『예: MFT 자원 구성』](#)

ftCreateMonitor 명령과 함께 **-mq** 매개변수를 사용하여 자원 모니터가 모니터링할 자원으로 IBM MQ 큐를 지정할 수 있습니다.

관련 참조

fteCreateMonitor: MFT 자원 모니터 작성

[소스 큐의 메시지에서 MFT가 읽은 MQ 메시지 특성](#)

메시지에서 파일로의 전송에 대한 재시도 작동 모니터링 구성

자원 모니터에 의해 트리거되는 메시지에서 파일로의 전송이 실패하고 큐에서 모니터를 트리거한 메시지 그룹을 남겨두는 경우 해당 전송은 후속 폴링 간격에서 다시 제출됩니다. 전송이 다시 제출되는 횟수는 모니터링 에이전트의 **monitorGroupRetryLimit** 특성에 의해 제한됩니다.

이 태스크 정보

새로운 메시지에서 파일로의 전송이 트리거될 때마다 전송 태스크를 위한 새 전송 ID가 생성됩니다.

에이전트가 다시 시작되면 전송이 트리거된 횟수가 **agent.properties** 파일의 **monitorGroupRetryLimit** 값을 초과한 경우에도 모니터가 전송을 다시 트리거합니다.

monitorGroupRetryLimit 특성의 값은 메시지 그룹이 여전히 큐에 존재하는 경우 모니터가 메시지에서 파일로의 전송을 다시 트리거하는 최대 횟수입니다. 이 특성의 기본값은 10입니다. 이 특성의 값은 양의 정수 값 또는 -1로 설정할 수 있습니다. 이 특성의 값으로 -1이 지정되면 모니터는 트리거 조건이 충족되지 않을 때까지 횟수 제한 없이 전송을 다시 트리거합니다.

전송 시도로 인해 전송이 트리거된 횟수가 **monitorGroupRetryLimit**의 값을 초과하면 에이전트는 이벤트 로그에 오류를 기록합니다.

단일 메시지는 단일 그룹인 것처럼 처리되고 메시지가 큐에 남아 있고 전송이 트리거된 횟수가 **monitorGroupRetryLimit**의 값 미만인 경우 각각의 폴링 간격에서 전송이 다시 트리거됩니다.

모니터링 에이전트에서 **monitorGroupRetryLimit** 특성을 설정하려면 다음 단계를 수행하십시오.

프로시저

1. **fteStopAgent** 명령을 사용하여 모니터링 에이전트를 중지하십시오.
2. 다음 행을 포함하도록 모니터링 에이전트에 대한 **agent.properties** 파일을 편집하십시오.

```
monitorGroupRetryLimit=number_of_retries
```

agent.properties 파일은 **MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name** 디렉토리에 있습니다.

3. **fteStartAgent** 명령을 사용하여 모니터링 에이전트를 시작하십시오.

관련 태스크

221 페이지의 『예: MFT 자원 구성』

fteCreateMonitor 명령과 함께 **-mq** 매개변수를 사용하여 자원 모니터가 모니터할 자원으로 IBM MQ 큐를 지정할 수 있습니다.

트리거 파일 사용

자원 모니터의 트리거 파일 콘텐츠를 사용하여 단일 전송 요청으로 전송할 파일 세트를 정의할 수 있습니다. 일치하는 트리거 파일이 감지될 때마다 해당 콘텐츠가 소스 파일 경로 및 선택적으로 목적지 파일 경로로 구문 분석됩니다. 그런 다음 이러한 파일 경로는 사용자가 지정하는 태스크 전송 XML 파일에 파일 항목을 정의하는 데 사용됩니다. 이 XML 파일은 단일 전송 요청으로 에이전트에 제출됩니다. 자원 모니터의 정의는 트리거 콘텐츠가 사용 가능한지 여부를 판별합니다.

-tc(트리거 콘텐츠) 매개변수를 지정하여 모니터 작성 시 파일 콘텐츠 트리거를 사용 가능하게 할 수 있습니다. 이 **-tc** 매개변수는 파일 트리거 옵션 **match** 및 **noSizeChange**에만 적용됩니다. 모니터 작성에 대한 자세한 정보는 **fteCreateMonitor**: MFT 자원 모니터 작성을 참조하십시오.

트리거 콘텐츠 파일을 사용할 때 각 행의 다음 중 하나입니다.

- 단일 소스 파일 경로 또는

- 쉘표로 구분된 소스 파일 경로 및 목적지 파일 경로

여기서 공백 문자는 파일 경로의 일부로 처리됩니다. **fteCreateMonitor** 명령에 **-tcr** 및 **-tcc** 매개변수를 지정하여 기본 행 형식을 변경할 수 있습니다. 자세한 정보는 232 페이지의 『고급 옵션』의 내용을 참조하십시오.

트리거 파일이 구문 분석된 후 파일 경로 목록이 생성되고 사용자가 지정한 전송 태스크 XML에 적용됩니다. 모든 모니터와 마찬가지로 전송 태스크 XML의 양식은 단일 항목 또는 파일이 정의된 **fteCreateTransfer** 명령을 통해 생성된 전체 전송 태스크 XML입니다. 단일 항목은 대체 변수 `${contentSource}`와 선택적으로 `{contentDestination}`을 소스 및 목적지 파일 경로에 대한 대체로 사용해야 합니다. 모니터가 트리거 파일에 각 행(파일 경로)의 파일 항목을 포함하도록 전송 태스크 XML을 확장합니다.

-tc 매개변수가 각 트리거 파일당 하나의 전송 요청을 나타내므로 파일 콘텐츠 트리거를 **-bs** 매개변수와 함께 사용할 수 없습니다.

예

다음 예에서는 **trig**로 끝나는 파일에 대해 트리거할 모니터를 정의하고 해당 파일의 파일 경로를 읽습니다.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

fteCreateTransfer 명령은 소스 파일 경로가 `${contentSource}`인 단일 파일에 대해 `task.xml` 라는 파일을 작성합니다. 예를 들면, 다음과 같습니다.

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

fteCreateMonitor 명령은 `/home/trigdir` 디렉토리에서 **trig**로 끝나는 파일을 스캔하고 콘텐츠를 사용하여 해당 트리거 파일의 모든 경로에 대해 `task.xml`을 기반으로 하는 단일 전송 요청을 작성합니다. 트리거 파일의 양식은 쉘표 구분 기호 없이 각 행마다 하나의 파일 경로(소스만)이어야 합니다. 예를 들면, 다음과 같습니다.

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

모든 파일은 해당 파일 경로가 아니라 해당 파일 이름으로 `/file/destdir` 디렉토리에 전달됩니다. 즉, `/home/file/first.txt`는 `/file/destdir/first.txt`로 전달됩니다.

또는 **fteCreateTransfer** 명령의 **-dd /file/destdir** 매개변수를 **-df \${contentDestination}**으로 변경하고 트리거 파일의 콘텐츠 양식을 소스 파일 경로, 목적지 파일 경로로 변경하면 동일한 목적지 에이전트에 대해 여러 목적지 경로를 정의할 수 있습니다. 예를 들면, 다음과 같습니다.

```
/home/file/first.txt,/home/other/sixth.txt
```

그러면 목적지 위치가 `/home/other/sixth.txt`가 됩니다.

대체 변수를 토큰화할 수 있습니다. 예를 들어, `${contentDestination{token=-1}}`을 사용하여 제공된 경로에서 파일 이름 부분을 분리할 수 있습니다. 따라서 **fteCreateTransfer** 대상이 `-df /file/destdir/${contentDestination{token=-1}}`로 정의된 경우 `/home/file/first.txt`의 새 대상은 `/file/destdir/sixth.txt`입니다.

고급 옵션

-**tc**r *regex* 매개변수를 사용하여 트리거 파일의 콘텐츠에 대한 기본 행 양식을 변경할 수 있습니다. 필수 행 양식과 일치하고 하나 또는 두 개의 캡처 그룹을 제공하는 정규식을 제공하십시오. 첫 번째 캡처 그룹은 소스이고 두 번째 선택적 캡처 그룹은 목적지입니다. 예를 들면, 다음과 같습니다.

- 소스 및 목적지 경로는 하이픈으로 구분됩니다.

```
((?:[^-]+)-)((?:[^-]+)
```

이 예에서는 구분 기호가 세 개의 위치에 정의되어 있고 세 개의 하이픈(-) 인스턴스 모두 임의의 문자로 변경할 수 있습니다. 특수 문자를 이스케이프하는지 확인하십시오.

- 소스 및 목적지 경로는 쉼표로 구분되며 후미 공백이 있습니다. 번호 부호(#)로 표시되는 주석은 무시됩니다.

```
((?:[^\,]+),)((?:[^\,]+) *(?:#\.[*])+
```

파일 경로에는 숫자 부호(#)가 포함될 수 없습니다. 일반적으로 항목은 `/home/source/from.txt,/home/destination/to.txt # some comment`입니다.

-**tc**r 매개변수를 사용하는 경우 표현식이 오류를 감지하고 트리거 파일을 올바르게 구문 분석할 수 있도록 정규식이 잘 디자인되고 테스트되었는지 확인하십시오.

-**tcc destSrc** 매개변수를 사용하여 캡처의 순서를 뒤바꿀 수 있습니다. 이 매개변수를 지정하는 경우 첫 번째 캡처 그룹이 목적지 파일 경로이고 두 번째 그룹이 소스 파일 경로입니다.

오류가 핸들링되는 방법

비어 있는 트리거 파일

트리거 파일이 비어 있는 경우 결과가 파일 전송이 아닙니다. 즉, 모니터가 전송 요청을 작성하지만 파일 항목이 지정되지 않습니다.

오류가 있는 트리거 파일

트리거 파일의 입력 항목이 예상된 양식에 대해 구문 분석하는 데 실패한 경우 전송 요청이 생성되지 않습니다. 모니터 오류 로그가 발행되고 오류가 이벤트 로그에도 로깅됩니다. 트리거 파일이 처리됨으로 표시되고 파일이 업데이트될 때까지 모니터가 파일을 다시 처리하려고 시도하지 않습니다.

전송 태스크 XML 불일치

전송 태스크 XML은 트리거 파일과 일치해야 합니다. 전송 태스크 XML에 `${contentSource}` 및 `${contentDestination}`이 둘 다 있는 경우 해당 모니터에 대한 모든 트리거 파일에 소스 및 목적지 파일 경로가 있어야 하고 반대의 경우도 마찬가지입니다. 첫 번째 경우에 트리거 파일이 소스 파일 경로만 제공하면 모니터가 `${contentDestination}`의 대체 실패를 보고합니다.

예

다음 예제는 트리거 파일의 콘텐츠에 소스 파일 경로만 포함된 기본 콘텐츠 트리거입니다.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

-**tc**r 매개변수는 공백 문자로 구분된 문자 순서의 두 캡처 그룹을 정의합니다. -**tcc destSrc** 매개변수 및 옵션은 캡처 그룹이 목적지로 처리된 후 소스로 처리됨을 나타냅니다.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df ${contentDestination} ${
contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
-tcr "((?:[^\ ]+)((?:[^\ ]+))" -tcc destSrc
```

관련 개념

[222 페이지의 『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』](#)

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

관련 태스크

227 페이지의 『큐 모니터링 및 변수 대체 사용』

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터되는 큐에서 파일로 전송할 수 있습니다. 모니터되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

관련 참조

fteCreateMonitor: MFT 자원 모니터 작성

fteCreateTransfer: 새 파일 전송 시작

MFT 자원 모니터 로깅

로깅을 사용하여 자원 모니터에 대한 진단 정보를 확보할 수 있습니다.

이 태스크 정보

자원 모니터에 대한 로깅을 사용하기 위해 **fteSetAgentLogLevel** 명령 또는 `agent.properties` 파일을 사용하여 자원 모니터 로깅을 제어할 수 있습니다.

기존 추적 지점도 정보 캡처에 계속 사용됩니다.

자원 모니터 로그는 `resmoneventN.log`(이)라는 파일에 기록됩니다. 여기서 *N*은 숫자를 나타냅니다(예: `resmonevent0.log`). 이벤트 로그 파일은 모니터가 자원(예: 디렉토리 또는 큐)을 폴링할 때 발생하는 여러 조치를 기록합니다.



주의: 에이전트의 모든 자원 모니터는 동일한 로그 파일에 기록합니다.

`resmoneventN.log` 파일의 일부 예제 출력은 [MFT 디렉토리 자원 모니터가 파일을 트리거하지 않는 경우 수행할 작업을 참조하십시오.](#)

다음 표에는 자원 모니터가 로그 파일에 기록하는 이벤트의 유형이 나열되어 있습니다. 세 번째 열에서는 각 이벤트를 캡처하는 데 필요한 로그 레벨에 대해 설명하며 여기서 최하위 레벨은 INFO이고 최상위 레벨은 VERBOSE입니다.

로그 레벨을 더 높게 설정하면 더 낮은 레벨의 이벤트가 기록됩니다. 예를 들어, 로그 레벨을 MODERATE로 설정하면 INFO 레벨 이벤트를 기록하고 VERBOSE 레벨 이벤트는 기록되지 않습니다.

번호	이벤트	로그 레벨	설명
1	모니터 작성됨	INFO	자원 모니터가 작성되었습니다.
2	모니터 삭제됨	INFO	자원 모니터가 삭제되었습니다.
3	모니터 중지됨	INFO	자원 모니터가 중지되었습니다.
4	모니터 시작됨	INFO	자원 모니터가 시작되었습니다.
5	폴 시작	INFO	자원 모니터가 새 폴 주기를 시작했습니다.
6	폴 종료	INFO	자원 모니터 폴 주기가 종료되었습니다.
7	패턴 일치	VERBOSE	트리거 모니터 디렉토리의 파일 또는 지정된 패턴과 일치하는 큐의 메시지를 찾았습니다.
8	패턴 불일치	VERBOSE	트리거 모니터 디렉토리의 일치하지 않는 파일 또는 지정된 패턴과 일치하지 않는 큐의 메시지를 찾았습니다.
9	전송 요청	INFO	자원 모니터가 전송을 시작했습니다.

번호	이벤트	로그 레벨	설명
10	디렉토리가 너무 깊음	VERBOSE	자원 모니터가 모니터링하는 디렉토리에 폴링할 서브디렉토리가 자원 모니터 구성에 지정된 숫자보다 너무 많이 포함되어 있습니다.
11	파일이 잠김	중간	자원 모니터가 모니터링하는 트리거 파일이 다른 프로세스에 의해 잠겼습니다.
12	파일 크기 작음	중간	트리거 파일이 자원 모니터 구성에 지정된 크기보다 작습니다.
13	파일 크기 불안정	중간	트리거 파일이 자원 모니터 구성에 지정된 것보다 너무 빈번하게 변경되고 있습니다.
14	폴이 너무 많음	중간	자원 모니터가 안정적인지 않은 트리거 파일을 너무 여러 번 폴링했습니다.
15	일치하는 항목 수	INFO	자원 모니터가 폴링한 디렉토리에서 찾은 트리거 파일의 총 수입니다.
16	전송 항목	INFO	전송 요청의 총 항목 수입니다.
17	FDC 생성됨	중간	자원 모니터가 예외를 생성했습니다.
18	전송 요청	INFO	자원 모니터가 전송 요청을 제출했습니다.
19	모니터 시작 실패	중간	자원 모니터를 시작하는 데 실패했습니다.
20	기록을 지움	INFO	모니터 히스토리 정보를 지웠습니다.
21	모니터 기록 지우기 실패	INFO	모니터 기록 정보를 지우는 데 실패했습니다.
22	전송 ID	INFO	모니터에서 전송 요청의 ID를 제출했습니다.
23	일괄처리	INFO	일치된 항목의 총 전송 요청 수는 N 입니다. 여기서 N 은 숫자입니다.
> V 9.4.0 24	연결됨	VERBOSE	자원 모니터가 에이전트 큐 관리자에 연결되었습니다.
> V 9.4.0 25GB	연결 끊어짐	VERBOSE	자원 모니터가 에이전트 큐 관리자에서 연결이 끊어졌습니다.
> V 9.4.0 26	연결 끊기 중에 오류 발생	VERBOSE	에이전트 큐 관리자에서 연결을 끊을 때 자원 모니터에 문제가 발생했습니다.

프로시저

- **fteSetAgentLogLevel** 명령을 사용하여 자원 모니터 로깅을 켜고 끄려면 [fteSetAgentLogLevel](#)에서 **logMonitor** 매개변수에 대한 설명 및 다른 옵션을 사용하는 방법에 대한 예제를 참조하십시오.
- `agent.properties` 파일을 사용하여 자원 모니터 로깅을 제어하려면 `MFT agent.properties` 파일에서 다음 로깅 활동을 수행할 수 있도록 하는 추가 특성에 대한 설명을 참조하십시오.
 - 로깅 켜기 또는 끄기
 - 각 로그 파일의 크기 제한
 - 자원 모니터가 생성할 수 있는 로그 수 제한

예

다음 샘플 메시지는 MFTDEMO 큐 관리자의 HA2 에이전트에 대한 verbose 레벨 로깅을 설정합니다.

```
<?xml version="1.0"?>
```

```
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
  <log:originator>
    <log:request>
      <log:hostName>192.168.7.1</log:hostName>
      <log:userID>johndoe</log:userID>
    </log:request>
  </log:originator>
  <log:endpoint agent="HA2" QMgr="MFTDEMO"/>
  <log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

관련 참조

[fteSetAgentLogLevel 명령](#)

[MFTagent.properties 파일](#)

MFT 자원 모니터 시작

IBM MQ 9.3.0부터는 **fteStartMonitor** 명령을 사용하여 에이전트를 중지하거나 다시 시작할 필요 없이 자원 모니터를 시작할 수 있습니다.

시작하기 전에

`agent.properties` 파일에서 **authorityChecking** 속성을 `true` 로 설정하여 사용자 권한 관리가 사용 가능한 경우, 자원 모니터를 시작하려면 모니터 또는 모니터 조작 권한이 있어야 합니다. 사용자 권한 관리에 관한 자세한 정보는 [MFT 에이전트 조치에서 사용자 권한 제한을 참조하십시오](#).

이 태스크 정보

Managed File Transfer 명령 컴포넌트가 설치된 시스템에서 **fteStartMonitor** 명령을 실행할 수 있습니다. 이는 어디서든 자원 모니터를 시작할 수 있으며 자원 모니터를 소유하는 에이전트가 실행 중인 시스템으로 제한되지 않음을 의미합니다. 이 명령의 필수 매개변수와 선택적 매개변수에 관한 정보는 [fteStartMonitor\(MFT 자원 모니터 시작\)](#)를 참조하십시오.

프로시저

- **fteStartMonitor** 명령을 실행하기 전이나 후에 에이전트의 상태를 확인하려면 다음 예제에 표시된 대로 **-v** 매개변수와 함께 **fteListMonitors** 명령을 사용하십시오.

```
fteListMonitors -ma monitoring_agent_name -v
```

- 같은 시스템에서 실행 중인 에이전트에서 자원 모니터를 시작하려면 다음과 같이 **fteStartMonitor** 명령을 입력하십시오.

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- 다른 시스템에서 실행 중인 에이전트에서 자원 모니터를 시작하려면 다음과 같이 **fteStartMonitor** 명령을 입력하십시오.

```
fteStartMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

명령 큐 관리자도 모니터링 에이전트의 에이전트 큐 관리자인 경우 **-mm** 매개변수는 선택사항이지만, 그렇지 않으면 **-mm** 매개변수와 함께 에이전트 큐 관리자를 지정해야 합니다.

결과

에이전트가 실행 중인 경우 자원 모니터가 현재 중지되어 있으면 시작됩니다. 명령은 다음 메시지를 출력하고 에이전트의 `output0.log`에 이벤트를 로그합니다.

```
BFGCL0816I: 'agent_name' 에이전트의 자원 모니터 'monitor_name' 시작 요청이 발행되었습니다.
BFGCL0251I: 요청이 완료되었습니다.
```


자원 모니터를 시작할 수 없는 경우 명령이 출력하는 메시지에 관한 정보는 [fteStartMonitor\(MFT 자원 모니터 시작\)](#)를 참조하십시오.

관련 개념

212 페이지의 『MFT 자원 모니터링 개념』

Managed File Transfer 자원 모니터링 기능의 주요 개념에 대한 개요입니다.

관련 태스크

236 페이지의 『MFT 자원 모니터 중지』

IBM MQ 9.3.0부터는 **fteStopMonitor** 명령을 사용하여 에이전트를 중지하거나 다시 시작할 필요 없이 자원 모니터를 중지할 수 있습니다.

관련 참조

[fteStartMonitor \(MFT 자원 모니터 시작\)](#)

MFT 자원 모니터 중지

IBM MQ 9.3.0부터는 **fteStopMonitor** 명령을 사용하여 에이전트를 중지하거나 다시 시작할 필요 없이 자원 모니터를 중지할 수 있습니다.

시작하기 전에

`agent.properties` 파일에서 **authorityChecking** 속성을 `true` 로 설정하여 사용자 권한 관리가 가능한 경우, 자원 모니터를 중지하려면 모니터 또는 모니터 조작 권한이 있어야 합니다. 사용자 권한 관리에 관한 자세한 정보는 [MFT 에이전트 조치에서 사용자 권한 제한을 참조하십시오](#).

이 태스크 정보

Managed File Transfer 명령 구성요소가 설치된 시스템에서 **fteStopMonitor** 명령을 실행할 수 있습니다. 이는 어디서든 자원 모니터를 중지할 수 있으며 자원 모니터를 소유하는 에이전트가 실행 중인 시스템으로 제한되지 않음을 의미합니다. 이 명령의 필수 매개변수와 선택적 매개변수에 관한 정보는 [fteStopMonitor\(MFT 자원 모니터 중지\)](#)를 참조하십시오.

자원 모니터가 중지되면 에이전트의 자원 모니터 이벤트 로그 `resmoneventnumber.log`에 메시지를 기록합니다. **fteStopMonitor** 명령으로 자원 모니터를 중지하면 메시지에 중지 요청을 실행한 사용자의 이름이 포함됩니다.

'`<mquser_id>`' 사용자가 자원 모니터를 중지함

fteStopMonitor 명령을 사용하여 이전에 자원 모니터가 중지된 경우에도 에이전트가 다시 시작하면 자원 모니터가 자동으로 시작됩니다.

에이전트에서는 모니터 중지 요청을 병렬이 아니라 직렬로 처리하므로, 에이전트가 M1 모니터 중지 요청을 받은 직후에 M2 모니터 중지 요청을 받으면 M2를 중지하기 전에 먼저 M1을 중지합니다.

프로시저

- **fteStopMonitor** 명령을 실행하기 전이나 후에 에이전트의 상태를 확인하려면 다음 예제에 표시된 대로 **-v** 매개변수와 함께 **fteListMonitors** 명령을 사용하십시오.

```
fteListMonitors -ma monitoring_agent_name -v
```

- 동일한 시스템에서 실행 중인 에이전트에서 자원 모니터를 중지하려면 다음과 같이 **fteStopMonitor** 명령을 입력하십시오.

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- 다른 시스템에서 실행 중인 에이전트에서 자원 모니터를 중지하려면 다음과 같이 **fteStopMonitor** 명령을 입력하십시오.

```
fteStopMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```


명령 큐 관리자도 모니터링 에이전트의 에이전트 큐 관리자인 경우 **-mm** 매개변수는 선택사항이지만, 그렇지 않으면 **-mm** 매개변수와 함께 에이전트 큐 관리자를 지정해야 합니다.

결과

에이전트가 실행 중인 경우 자원 모니터가 현재 시작되어 있으면 중지됩니다. 명령은 다음 메시지를 출력하고 에이전트의 `output0.log`에 이벤트를 로그합니다.

```
BFGCL0813I: 'SOURCE' 에이전트의 자원 모니터 'MNTR' 중지 요청이 발행되었습니다.  
BFGCL0251I: 요청이 완료되었습니다.
```

자원 모니터를 중지할 수 없는 경우 명령이 출력하는 메시지에 관한 정보는 [fteStopMonitor\(MFT 자원 모니터 중지\)](#)를 참조하십시오.

관련 개념

212 페이지의 『MFT 자원 모니터링 개념』

Managed File Transfer 자원 모니터링 기능의 주요 개념에 대한 개요입니다.

관련 태스크

235 페이지의 『MFT 자원 모니터 시작』

IBM MQ 9.3.0부터는 **fteStartMonitor** 명령을 사용하여 에이전트를 중지하거나 다시 시작할 필요 없이 자원 모니터를 시작할 수 있습니다.

관련 참조

[fteStopMonitor\(MFT 자원 모니터 중지\)](#)

MFT 자원 모니터의 백업 및 복원

백업에서 새 자원 모니터를 작성할 때 가져올 수 있는 XML 파일로 해당 정의를 내보냄으로써 이후 사용할 수 있도록 자원 모니터를 백업할 수 있습니다.

이 태스크 정보

예를 들어 다른 인프라에서 자원 모니터를 다시 작성하거나 큐 관리자 문제로 인해 자원 모니터를 다시 작성해야 하는 경우 이후 정의를 다시 사용할 수 있도록 이전에 정의했던 자원 모니터를 백업해야 하는 경우가 있습니다.

fteCreateMonitor 명령 또는 **fteListMonitors** 명령을 **-ox** 매개변수와 함께 사용하여 단일 자원 관리자 정의를 백업할 수 있습니다. 두 경우 모두 자원 관리자 정의는 XML 파일로 내보내어 백업합니다. 그런 다음 XML 파일에서 정의를 가져오면 **fteCreateMonitor** 명령의 **-ix** 매개변수를 사용하여 새 자원 관리자를 작성할 수 있습니다.

-ox 매개변수를 사용하면 한 번에 하나의 자원 모니터 정의만 백업할 수 있습니다.

-od 매개변수가 **fteListMonitors** 명령에 추가됩니다. 이 매개변수를 지정하면 지정된 디렉토리로 해당 정의를 벌크로 내보냄으로써 한 번에 둘 이상의 자원 모니터를 백업할 수 있습니다. 각 자원 모니터 정의는 *agent name.monitor name.xml* 형식의 이름으로 별도의 XML 파일에 저장됩니다.

각 자원 정의에 대해 **fteListMonitors -ox** 명령을 개별적으로 실행하거나 각 자원 모니터에 대해 **fteListMonitors -ox** 명령을 실행하는 별도의 스크립트를 사용하는 대신 **fteListMonitors -od** 명령을 한 번만 실행해야 하므로 백업하려는 자원 모니터 수가 많은 경우 **-od** 매개변수는 특히 유용합니다.

프로시저

- 한 자원 모니터의 정의를 XML 파일로 내보내어 백업하려면 다음 명령 중에서 사용하십시오.
 - **-ox** 매개변수가 있는 **fteCreateMonitor** 명령.
 - **-ox** 매개변수가 있는 **fteListMonitors** 명령.
- **-ox** 매개변수를 사용하는 경우 다음 예에 표시된 대로 **-ma** 및 **-mn** 매개변수도 지정해야 합니다.

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

- 여러 자원 모니터 정의를 지정된 디렉토리의 XML 파일로 반출하여 백업하려면 다음 예제에 표시된 대로 **-od** 매개변수와 함께 **fteListMonitors** 명령을 사용하십시오.

```
fteListMonitors -od /usr/mft/resmonbackup
```

자원 모니터를 벌크로 백업하는 경우 올바른 대상 디렉토리를 지정해야 합니다. 대상 경로를 지정하지 않으면 다음 예에 표시된 대로 오류 메시지가 발생합니다.

BFGCL0762E: 출력 디렉토리가 지정되지 않았습니다. 올바른 경로를 지정하는 명령을 다시 실행하십시오.

-od 매개변수는 **-ox** 매개변수와 결합되지 않아야 합니다. 그렇지 않으면 다음 오류 메시지가 표시됩니다.

BFGCL0761E: '-od'와 '-ox' 매개변수를 동시에 지정하는 것은 올바르지 않습니다.

백업에 포함시킬 특정 자원 모니터 세트를 정의할 수 있습니다. 예를 들어, **-ma** 매개변수를 사용하여 에이전트의 이름을 지정하면 다음 예에 표시된 대로 해당 에이전트의 모든 자원 모니터를 백업할 수 있습니다.

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

또한 일치하는 에이전트 이름 또는 모니터 이름 또는 이들 모두에 대해 사용할 패턴을 정의할 때 별표(*) 문자를 포함시켜 와일드카드 일치를 사용할 수도 있습니다. 다음 예제는 이름이 지정된 패턴과 일치하고 지정된 패턴에 맞는 이름을 가진 에이전트에 있는 모든 자원 모니터를 백업합니다.

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

명령을 실행하는 동안 다음 진행 보고 메시지가 표시됩니다.

총 *number*개의 일치하는 자원 모니터 정의를 발견했습니다.
*index/number*개의 자원 모니터 정의를 파일 시스템으로 저장했습니다.

상세 옵션을 사용하는 경우 현재 합계가 계속 표시되지만 이 명령이 다음을 표시하는 대신

*index/number*개의 자원 모니터 정의가 파일 시스템에 저장되었습니다.

저장 중인 모니터 정의의 이름을 표시합니다. 예를 들면, 다음과 같습니다.

BFGCL0762I: 에이전트 'XFERAGENT' 의 모니터 'FILEMON' 정의가 FILEMON.XFERAGENT.XML 을 파일 시스템으로 복사 하십시오.

- 특정 에이전트에 대한 하나의 자원 모니터를 지정된 디렉토리의 XML 파일로 반출하여 백업하려면 **-od** 매개변수와 함께 **fteListMonitors** 명령을 사용하십시오.

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

-od 매개변수를 사용하여 단일 자원 모니터를 백업하는 것은 출력 파일 이름이 *agent name.monitor name.xml* 형식이라는 점을 제외하고 **-ox** 매개변수를 사용하는 것과 유사합니다.

- 백업에서 자원 모니터 정의를 복원하려면 다음 예제에 표시된 대로 **fteCreateMonitor** 명령을 **-ix** 매개변수와 함께 사용하십시오.

```
fteCreateMonitor -ix file name
```

-od 매개변수 사용 방법에 대한 추가 예제는 [fteListMonitors: 목록 MFT 자원 모니터](#)를 참조하십시오.

관련 참조

fteCreateMonitor: MFT 자원 모니터 작성

fteListMonitors: MFT 자원 모니터 나열

자원 모니터 실행 기록 지우기

오류 때문에 이전에 전송되지 않은 파일에 대해 전송 요청을 제출할 수 있도록 자원 모니터 실행 기록을 지울 수 있습니다. 자원 모니터 실행 기록을 지우는 데 **fteClearMonitorHistory** 명령 또는 IBM MQ Explorer를 사용할 수 있습니다.

시작하기 전에

`agent.properties` 파일에서 **authorityChecking** 속성을 `true` 로 설정하여 사용자 권한 관리가 사용으로 설정된 경우, 모니터 히스토리를 지우는 사용자는 다음 표에 표시된 대로 적절한 권한을 가지고 있어야 합니다.

표 15. fteClearMonitorHistory 명령을 실행하는 데 필요한 사용자 권한		
모니터 실행 기록을 지우는 사용자	MFT 액세스 권한	필수 권한
자원 모니터를 작성한 사용자와 동일한 사용자	모니터	SYSTEM.FTE.AUTHMON1.<monitor_agent_name>에 대한 BROWSE 이 권한은 자원 모니터를 작성 또는 삭제하는 데 필요한 권한과 같은 권한입니다.
자원 모니터를 작성한 사용자를 제외한 사용자	모니터 조작	SYSTEM.FTE.AUTHPS1.<agent_name>에 대한 SET 이 권한은 자원 모니터를 삭제하는 데 필요한 권한과 같은 권한입니다.

사용자 권한 관리에 관한 자세한 정보는 MFT 에이전트 조치에서 사용자 권한 제한을 참조하십시오.


필수 권한이 없는 사용자가 자원 모니터 실행 기록을 지우려고 시도하면 **fteClearMonitorHistory** 명령이 오류 메시지를 출력하고 에이전트의 `output0.log` 파일에 실패를 로그합니다. 자세한 정보는 [fteClearMonitorHistory: 자원 모니터 실행 기록 지우기를 참조하십시오.](#)

이 태스크 정보

파일 전송이 시작되었는데 어떤 이유로 파일을 전송할 수 없는 경우, 자원 모니터는 다음 폴링에서 이 파일을 전송 대상으로 다시 선택하지 않습니다. 그 이유는 해당 파일이 이전 폴링에 표시되어 있었고 그 이후로 수정된 적이 없다고 모니터 실행 기록에 표시되어 있기 때문입니다(212 페이지의 『MFT 자원 모니터링 개념』 참조).

IBM MQ 9.1.3 이전에는 파일 전송에 실패할 경우, 파일을 삭제한 후 디렉토리에 다시 저장하거나 마지막 수정 날짜 속성이 변경되도록 파일을 업데이트하거나 자원 모니터 자체를 다시 작성한 경우에만 파일 전송을 다시 시작할 있습니다.

그러나 **fteClearMonitorHistory** 명령을 사용하거나 IBM MQ Explorer를 사용하여 자원 모니터 실행 기록을 지울 수 있습니다. 실행 기록을 지우면 파일을 삭제한 후 디렉토리에 다시 저장하거나, 마지막 수정 날짜 속성을 변경하기 위해 파일을 업데이트할 필요 없이 전송에 실패한 파일에 대해 다른 전송 요청을 제출할 수 있습니다. 예를 들어 이 기능은 파일을 전송해야 하지만 파일 수정이 불가능한 경우에 유용합니다. 또한 자원 모니터 실행 기록을 지울 수 있으므로 전송에 실패한 파일에 대한 다른 전송 요청을 제출하기 위해 자원 모니터를 다시 작성할 필요도 없습니다.

 z/OS 에서 Managed File Transfer 와 함께 제공되는 샘플 SCSQFCMD 멤버에는 모니터의 히스토리를 지우기 위한 JCL 스크립트가 포함되어 있습니다.

프로시저

- **fteClearMonitorHistory** 명령을 사용하여 자원 모니터 실행 기록을 지우려면 다음 형식으로 명령을 입력하십시오.

```
fteClearMonitorHistory -p <configuration> -ma <agent name> -mn <monitor name> -w 1000
```

-ma 및 **-mn** 매개변수만 필수입니다. 그 외의 모든 매개변수는 선택사항입니다. 예제를 포함하여 **fteClearMonitorHistory** 명령을 사용하는 방법에 대한 자세한 정보는 [fteClearMonitorHistory: 자원 모니터 실행 기록 지우기를 참조하십시오.](#)

실행 기록이 성공적으로 지워지면 명령은 다음 메시지를 출력합니다.

BFGCL0780I: 'agent name' 에이전트의 자원 모니터
'monitor name'에 대한 실행 기록 지우기 요청이 발행되었습니다.
BFGCL0251I: 요청이 완료되었습니다.

에이전트의 output0.log 파일에 성공을 로그합니다.

자원 모니터 실행 기록을 지우려는 시도가 실패하면 **fteClearMonitorHistory** 는 오류 메시지를 출력하고 에이전트의 output0.log 파일에 실패를 로그합니다.

- IBM MQ Explorer MFT 플러그인의 자원 모니터 보기를 사용하여 자원 모니터 실행 기록을 지우려면 자원 모니터를 마우스 오른쪽 단추로 클릭하고 드롭 다운 메뉴에서 **실행 기록 지우기**를 선택하십시오.

실행 기록을 지우면 다음 메시지가 표시됩니다.

BFGUI00171: 자원 모니터 실행 기록이 지워졌습니다.

실행 기록을 지우는 데 실패할 경우 오류 메시지가 표시됩니다. 예를 들면, 다음과 같습니다.

BFGUI0016E 지정된 자원 모니터의 실행 기록을 지우는 데 실패함 - 이유 2059

파일 전송 템플릿에 대한 작업

파일 전송 템플릿을 사용하여 반복된 전송 또는 복합 전송에 적합한 공용 파일 전송 설정을 저장할 수 있습니다. **fteCreateTemplate** 명령을 사용하여 명령행에서 전송 템플릿을 작성하거나 IBM MQ Explorer에서 **관리 파일 전송의 새 템플릿 작성** 마법사를 사용하여 전송 템플릿을 작성하거나 **템플릿으로 전송 설정 저장** 선택란을 선택하여 파일 전송 작성 중에 템플릿을 저장합니다. **전송 템플릿** 창에 Managed File Transfer 네트워크에서 사용자가 작성한 전송 템플릿이 모두 표시됩니다.

이 태스크 정보


명령행에서 전송 템플릿을 작성하려면 **fteCreateTemplate** 명령을 사용하십시오. 그런 다음 명령행에서 작성한 전송 템플릿을 제출하려면 IBM MQ Explorer에서 **제출** 을 클릭하십시오.

IBM MQ Explorer에서 전송 템플릿을 보려면 다음 단계를 수행하십시오.

프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 펼치십시오. 콘텐츠 보기에 **관리 파일 전송 센터**가 표시됩니다.
2. 모든 조정 큐 관리자가 네비게이터 보기에 나열됩니다. 스케줄된 전송에 사용된 조정 큐 관리자의 이름을 펼치십시오. 연결된 조정 큐 관리자를 변경하려면, 네비게이터 보기에서 사용할 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭한 후 **연결**을 클릭하십시오.
3. **전송 템플릿**를 클릭하십시오. **전송 템플릿** 창이 콘텐츠 보기에 표시됩니다.
4. **전송 템플릿** 창에 파일 전송에 대한 다음과 같은 자세한 내용이 나열됩니다.
 - a) **이름**: 파일 전송 템플릿의 이름입니다.
 - b) **소스**: 소스 시스템에서 파일을 전송하는 데 사용되는 에이전트의 이름입니다.
 - c) **소스 파일**: 호스트 시스템에서 전송할 파일의 이름입니다.
전송 템플릿 정보를 펼치면 이 필드를 볼 수 있습니다.
 - d) **목적지**: 목적지 시스템에서 파일을 수신하는 데 사용되는 에이전트의 이름입니다.
 - e) **목적지 파일**: 목적지 시스템에 전송된 후의 파일 이름입니다.
전송 템플릿 정보를 펼치면 이 필드를 볼 수 있습니다.
 - f) **스케줄된 시작(선택한 시간대)**: 관리자가 사용하는 시간대에 시작되도록 스케줄된 파일 전송 시간 및 날짜입니다. 표시되는 시간대를 변경하려면 **창 > 환경 설정 > IBM MQ Explorer > Managed File Transfer**를 클릭하고 **시간대**: 목록에서 대체 시간대를 선택하십시오. **확인**을 클릭하십시오.
 - g) **이벤트 트리거**: 시작할 파일 전송을 트리거하는 이벤트의 유형입니다. 유형 값은 존재함, 존재하지 않음 또는 초과 중 하나일 수 있습니다.

결과

전송 템플리트 창에 표시된 내용을 새로 고치려면 콘텐츠 보기 도구 모음에서 새로 고치기 단추 를 클릭하십시오.

전송 템플리트를 제출하고 템플리트에 정의된 전송을 시작하려면 템플리트 이름을 마우스 오른쪽 단추로 클릭한 다음, **제출**을 클릭하십시오.

전송 템플리트를 변경하려면 템플리트 이름을 마우스의 오른쪽 단추로 클릭하고 **편집**을 클릭하십시오. 원래 템플리트에 포함된 모든 파일은 원래 템플리트에 있는 그룹의 부분으로 포함되지 않은 경우에도 전송 그룹의 부분으로 나열됩니다. 템플리트에서 파일을 제거하려는 경우 그룹에서 파일 스펙을 선택하고 **선택 제거**를 클릭해야 합니다. 새 파일 스펙을 템플리트에 추가하려는 경우 템플리트 패널에 있는 필드를 사용하고 **그룹에 추가** 단추를 클릭하십시오. 편집 시 편집한 템플리트의 이름을 새로 작성하도록 프롬프트됩니다.

전송 템플리트에서 파일 전송을 작성하려면 템플리트 이름을 마우스의 오른쪽 단추로 클릭하고 **새 전송으로 편집**을 클릭하십시오.

전송 템플리트의 복제본을 작성하려면 템플리트 이름을 마우스 오른쪽 단추로 클릭한 다음, **복제**를 클릭하십시오. 복제 전송 템플리트는 원래 템플리트와 동일한 이름에 "(copy)"가 추가되어 자동 저장됩니다.

전송 템플리트를 삭제하려면 템플리트 이름을 마우스 오른쪽 단추로 클릭한 다음, **삭제**를 클릭하십시오.

관련 태스크

241 페이지의 『[IBM MQ Explorer를 사용하여 파일 전송 템플리트 작성](#)』

IBM MQ Explorer 또는 명령행에서 파일 전송 템플리트를 작성할 수 있습니다. 그런 다음 해당 템플리트를 사용하여 템플리트 세부사항으로 새 파일 전송을 작성하거나, 템플리트를 제출하여 파일 전송을 시작할 수 있습니다.

관련 참조

[fteCreateTemplate](#): 새 파일 전송 템플리트 작성

[fteListTemplates](#)

[fteDeleteTemplates](#)

IBM MQ Explorer를 사용하여 파일 전송 템플리트 작성

IBM MQ Explorer 또는 명령행에서 파일 전송 템플리트를 작성할 수 있습니다. 그런 다음 해당 템플리트를 사용하여 템플리트 세부사항으로 새 파일 전송을 작성하거나, 템플리트를 제출하여 파일 전송을 시작할 수 있습니다.

이 태스크 정보

명령행에서 파일 전송 템플리트를 작성하려면 [fteCreateTemplate](#) 명령을 사용하십시오.

IBM MQ Explorer에서 **관리 파일 전송을 위한 새 템플리트 작성** 마법사를 사용하여 파일 전송 템플리트를 작성하려면 다음 단계를 사용하십시오.

프로시저

1. 네비게이터 보기에서 **관리 파일 전송**을 클릭하십시오. 콘텐츠 보기에 **관리 파일 전송 센터**가 표시됩니다.
2. 모든 조정 큐 관리자가 네비게이터 보기에 표시됩니다. 스케줄된 전송에 사용된 조정 큐 관리자의 이름을 펼치십시오. 연결된 조정 큐 관리자를 변경하려면, 네비게이터 보기에서 사용할 조정 큐 관리자의 이름을 마우스 오른쪽 단추로 클릭한 후 **연결**을 클릭하십시오.
3. **전송 템플리트**를 마우스 오른쪽 단추로 클릭한 다음 **새 템플리트**를 클릭하여 **관리 파일 전송의 새 템플리트 작성** 마법사를 시작하십시오.
4. 마법사 패널의 지시사항을 따르십시오. 각 패널에 대한 컨텍스트 도움말이 제공됩니다. Windows에서 컨텍스트 도움말에 액세스하려면 F1을 누르십시오. Linux에서는 Ctrl+F1 또는 Shift+F1을 누르십시오.

모든 필수 전송 세부사항이 포함된 템플리트를 작성한 경우 **전송 요약** 페이지의 **템플리트로 전송 설정 저장** 선택란이 아직 선택되어 있지 않으면 해당 선택란을 선택하십시오. 이름 필드에 템플리트의 이름도 입력하십시오. 모든 필수 전송 세부사항이 포함되지 않은 템플리트를 작성한 경우에는 **템플리트로 전송 설정 저장** 선택란이 자동으로 선택됩니다.

관련 태스크

240 페이지의 『파일 전송 템플릿에 대한 작업』

파일 전송 템플릿을 사용하여 반복된 전송 또는 복합 전송에 적합한 공용 파일 전송 설정을 저장할 수 있습니다. **fteCreateTemplate** 명령을 사용하여 명령행에서 전송 템플릿을 작성하거나 IBM MQ Explorer에서 **관리 파일 전송의 새 템플릿 작성** 마법사를 사용하여 전송 템플릿을 작성하거나 **템플릿으로 전송 설정 저장** 선택란을 선택하여 파일 전송 작성 중에 템플릿을 저장합니다. **전송 템플릿** 창에 Managed File Transfer 네트워크에서 사용자가 작성한 전송 템플릿이 모두 표시됩니다.

관련 참조

fteCreateTemplate: 새 파일 전송 템플릿 작성

[fteListTemplates](#)

[fteDeleteTemplates](#)

파일 전송 템플릿 정의 백업하기

파일 전송 템플릿에는 전송을 위한 소스 및 대상 파일 스펙을 정의하는 XML 문서가 들어 있습니다. 이 XML 파일을 **fteCreateTemplate** 명령에 대한 입력으로 사용하여 파일 전송 템플릿을 다시 작성할 수 있습니다.

이 태스크 정보

전송 템플릿의 소스 및 대상 파일을 포함하는 XML 문서를 백업하려면 [fteCreateTransfer command](#) 명령 또는 IBM MQ Explorer를 사용하십시오. 전송 템플릿 XML 형식의 백업 파일을 작성하려면 다음 단계를 사용하십시오.

프로시저

- 방법 1: [fteCreateTransfer](#) 명령에서 **-gt** 매개변수를 사용하여 새 파일에 전송 템플릿 XML 메시지를 생성하십시오.
- 메소드 2: IBM MQ Explorer를 사용하여 템플릿을 작성하십시오.
전송 템플릿 요약 페이지로 이동하면 다음을 수행하십시오.
 - a) 요청 메시지 XML 미리 보기를 복사하십시오.
 - b) 이 전송 템플릿 XML 메시지를 새 파일에 저장하십시오.
- 메소드 3: IBM MQ Explorer를 사용하여 기존 템플릿을 백업하십시오.
 - a) **관리 파일 전송 > 큐 관리자 이름 > 전송 템플릿**으로 이동하십시오.
 - b) 전송 패널에서 백업해야 하는 템플릿을 강조표시하고 마우스 오른쪽 단추를 클릭한 다음 팝업 메뉴에서 **편집**을 선택하십시오.
 - c) 전송 템플릿 요약 페이지로 이동할 때까지 **다음**을 클릭하십시오.
 - d) 요청 메시지 XML 미리 보기를 복사하십시오.
 - e) 이 전송 템플릿 XML 메시지를 새 파일에 저장하십시오.

결과

위의 메소드 중 하나를 사용하여 작성된 전송 템플릿 XML 메시지 파일을 [fteCreateTemplate](#) 명령에 대한 입력으로 사용할 수 있습니다. 이 명령을 사용하는 방법에 대한 자세한 내용은 **fteCreateTemplate** 명령을 참조하십시오.

관련 참조

[fteCreateTemplate](#) 명령

[fteListTemplates](#) 명령

파일에서 메시지로 데이터 전송

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

메시지에서 파일로 전송에 대한 정보는 [256 페이지의 『메시지에서 파일로 데이터 전송』](#)의 내용을 참조하십시오.

파일에서 메시지로 전송의 목적지 에이전트는 프로토콜 브릿지 에이전트 또는 Connect:Direct 브릿지 에이전트가 될 수 없습니다.

파일 데이터를 IBM MQ 메시지 데이터로 전송할 수 있습니다. 애플리케이션이 IBM MQ 메시지를 읽고 사용할 수 있습니다. 다음과 같은 유형의 파일에서 메시지로 전송이 지원됩니다.

- 단일 파일에서 단일 메시지로. 메시지는 IBM MQ 그룹 ID가 설정되지 않습니다.
- 파일을 주어진 길이의 메시지로 분할하여 단일 파일을 다중 메시지로. 모든 메시지에 동일한 IBM MQ 그룹 ID가 포함됩니다.
- Java 정규식 구분 기호에서 텍스트 파일을 분할하여 단일 파일에서 다중 메시지로. 모든 메시지에 동일한 IBM MQ 그룹 ID가 포함됩니다.
- 16진 구분 기호에서 2진 파일을 분할하여 단일 파일에서 다중 메시지로. 모든 메시지에 동일한 IBM MQ 그룹 ID가 포함됩니다.

일련의 바이트를 구분 기호로 사용하여 2진 파일을 분할하려는 경우 **fteCreateTransfer** 명령의 **-sqdb** 매개변수를 사용하십시오. 자세한 정보는 [-sqdb 매개변수를 참조하십시오](#).

기본적으로 파일에서 메시지로 전송에 의해 작성되는 메시지는 지속 메시지입니다. 메시지를 비지속 메시지로 설정하거나 목적지 큐에 의해 정의된 지속 값을 갖도록 설정할 수 있습니다.

파일이 다중 메시지로 분할되도록 지정하는 경우 해당 파일에서 작성되는 모든 메시지의 IBM MQ 그룹 ID는 동일합니다. 파일이 다중 메시지로 분할되도록 지정하지 않으면 해당 파일에서 하나의 메시지만 작성되고 이 메시지는 IBM MQ 그룹 ID가 설정되지 않습니다.

파일을 대형 메시지 또는 다수의 작은 메시지에 전송하는 경우, 일부 IBM MQ 또는 Managed File Transfer 특성을 변경해야 합니다. 정보는 [메시지 크기와 연관된 MQ 속성 및 MFT 특성 설정에 대한 자세한 내용을 참조하십시오](#).

참고: 목적지 큐가 클러스터된 큐 또는 클러스터된 큐의 알리어스인 경우, `enableClusterQueueInputOutput` 에이전트 특성이 true로 설정되면 파일을 큐로 전송할 때 오류 메시지가 표시됩니다. 자세한 정보는 [목적지 큐가 클러스터된 큐이거나 클러스터된 큐에 대한 알리어스인 경우 수행할 작업을 참조하십시오](#).

관련 태스크

[244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 `enableQueueInputOutput`을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 `enableClusterQueueInputOutput`도 True로 설정해야 합니다.

[245 페이지의 『예: 단일 파일에서 단일 메시지로 전송』](#)

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

[246 페이지의 『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』](#)

fteCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

[249 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

[248 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 매개변수를 사용하십시오.

[251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』](#)

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은

처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

202 페이지의 『새 파일 전송 시작』

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

255 페이지의 『파일에서 메시지로 전송 실패』

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

목적지 큐에 기록된 메시지에서 MFT가 설정한 MQ 메시지 특성

[메시지 크기와 연관된 MQ 속성 및 MFT 특성 설정에 대한 자세한 내용](#)

파일에서 메시지로 전송을 수행하도록 에이전트 구성

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 `enableQueueInputOutput`을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 `enableClusterQueueInputOutput`도 True로 설정해야 합니다.

이 태스크 정보

`enableQueueInputOutput` 특성이 True로 설정되지 않은 목적지 에이전트에 대해 파일에서 메시지로 전송을 수행하는 경우 전송에 실패합니다. 조정 큐 관리자에 발행되는 전송 로그 메시지에 다음 메시지가 포함됩니다.

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

에이전트가 큐에 쓰고 읽을 수 있도록 하려면 다음 단계를 수행하십시오.

프로시저

1. **fteStopAgent** 명령을 사용하여 목적지 에이전트를 중지하십시오.
2. `agent.properties` 파일을 편집하여 `enableQueueInputOutput=true` 행을 포함시키십시오.
`agent.properties` 파일은 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` 디렉토리에 있습니다.
3. 옵션: `agent.properties` 파일을 편집하여 `enableClusterQueueInputOutput=true` 행을 포함시키십시오. `agent.properties` 파일은 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` 디렉토리에 있습니다.
4. **fteStartAgent** 명령을 사용하여 목적지 에이전트를 시작하십시오.

관련 개념

242 페이지의 『파일에서 메시지로 데이터 전송』

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

245 페이지의 『예: 단일 파일에서 단일 메시지로 전송』

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

246 페이지의 『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』

fteCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

249 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치를 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

248 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 매개변수를 사용하십시오.

251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

관련 참조

fteStopAgent

fteStartAgent

MFTagent.properties 파일

255 페이지의 『파일에서 메시지로 전송 실패』

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

예: 단일 파일에서 단일 메시지로 전송

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

이 태스크 정보

소스 파일은 /tmp/single_record.txt이며 소스 에이전트 AGENT_NEPTUNE과 동일한 시스템에 있습니다. 소스 에이전트 AGENT_NEPTUNE은 큐 관리자 QM_NEPTUNE을 사용합니다. 목적지 에이전트는 AGENT_VENUS이며 이 에이전트는 큐 관리자 QM_VENUS에 연결됩니다. 목적지 큐(RECEIVING_QUEUE)는 큐 관리자 QM_MERCURY에 위치합니다. QM_MERCURY는 큐 관리자 QM_VENUS와 동일한 IBM MQ 네트워크에 있으며 큐 관리자 QM_VENUS에 의해 액세스될 수 있습니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

목적지 큐가 목적지 에이전트에서 사용하는 큐 관리자와 다른 큐 관리자에 있는 경우에는 **-dq** 매개변수의 값을 *queue_name@queue_manager_name* 형식으로 지정해야 합니다. 값에 *@queue_manager_name*을 지정하지 않으면 목적지 에이전트는 목적지 큐가 목적지 에이전트 큐 관리자에 있다고 가정합니다.

enableClusterQueueInputOutput 에이전트 특성이 True로 설정된 경우는 예외입니다. 이 경우 목적지 에이전트는 표준 IBM MQ 해결 프로시저를 사용하여 큐의 위치를 판별합니다.

소스 에이전트 AGENT_NEPTUNE은 /tmp/single_record.txt 파일에서 데이터를 읽고 이 데이터를 목적지 에이전트 AGENT_VENUS에 전송합니다. 목적지 에이전트 AGENT_VENUS는 RECEIVING_QUEUE@QM_MERCURY 큐의 지속 메시지에 데이터를 송신합니다. 메시지에는 IBM MQ 그룹 ID가 설정되지 않습니다.

관련 개념

242 페이지의 『파일에서 메시지로 데이터 전송』

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 enableQueueInputOutput을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 enableClusterQueueInputOutput도 True로 설정해야 합니다.

246 페이지의 『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』

ftcCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

249 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **ftcCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

248 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **ftcCreateTransfer** 명령의 **-dqdt** 매개변수를 사용하십시오.

251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』

ftcCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

202 페이지의 『새 파일 전송 시작』

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

255 페이지의 『파일에서 메시지로 전송 실패』

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

예: 길이를 단위로 단일 파일을 여러 메시지로 분할

ftcCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

이 태스크 정보

소스 파일의 이름은 /tmp/source.file이며 크기는 36KB입니다. 소스 파일은 소스 에이전트 (AGENT_NEPTUNE)와 동일한 시스템에 위치합니다. 소스 에이전트 AGENT_NEPTUNE은 큐 관리자 QM_NEPTUNE에 연결됩니다. 목적지 에이전트는 큐 관리자 QM_MERCURY에 연결되는 AGENT_MERCURY입니다. 목적지 큐 RECEIVING_QUEUE도 큐 관리자 QM_MERCURY에 위치합니다. 전송 시 소스 파일이 1KB 크기의 섹션으로 분할되고 이들 각 섹션은 RECEIVING_QUEUE에 메시지로 기록됩니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

소스 에이전트 AGENT_NEPTUNE은 /tmp/source.file 파일에서 데이터를 읽고 이 데이터를 목적지 에이전트 AGENT_MERCURY에 전송합니다. 목적지 에이전트 AGENT_MERCURY는 큐 RECEIVING_QUEUE@QM_MERCURY에 있는 36개의 1KB 지속 메시지에 데이터를 기록합니다. 이러한 메시지는 모두 동일한 IBM MQ 그룹 ID를 가지며 그룹의 마지막 메시지에는 IBM MQ LAST_MSG_IN_GROUP 플래그 세트가 있습니다.

관련 개념

[242 페이지의 『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

[244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 enableQueueInputOutput을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 enableClusterQueueInputOutput도 True로 설정해야 합니다.

[245 페이지의 『예: 단일 파일에서 단일 메시지로 전송』](#)

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

[249 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치를 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

[248 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 매개변수를 사용하십시오.

[251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』](#)

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

[252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』](#)

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

[202 페이지의 『새 파일 전송 시작』](#)

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

[255 페이지의 『파일에서 메시지로 전송 실패』](#)

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 `fteCreateTransfer` 명령의 `-dqdt` 매개변수를 사용하십시오.

이 태스크 정보

파일은 가변 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다. 파일의 텍스트가 주어진 정규식과 일치하는 각 지점에서 텍스트 파일이 분할됩니다. 소스 파일은 `/tmp/names.text`이며 다음 콘텐츠가 있습니다.

```
Jenny Jones,John Smith,Jane Brown
```

파일 분할 위치를 지정하는 정규식은 쉼표 문자(,)입니다.

소스 파일은 큐 관리자 `QM_NEPTUNE`에 연결되는 소스 에이전트 `AGENT_NEPTUNE`과 동일한 시스템에 있습니다. 목적지 큐(`RECEIVING_QUEUE`)는 큐 관리자 `QM_MERCURY`에 위치합니다. `QM_MERCURY`는 목적지 에이전트(`AGENT_MERCURY`)에서 사용하는 큐 관리자이기도 합니다. 전송 시 소스 파일이 섹션으로 분할되고 이들 각 섹션은 `RECEIVING_QUEUE`에 메시지로 기록됩니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

소스 에이전트 `AGENT_NEPTUNE`은 `/tmp/names.text` 파일에서 데이터를 읽고 이 데이터를 목적지 에이전트 `AGENT_MERCURY`에 전송합니다. 목적지 에이전트 `AGENT_MERCURY`는 큐 `RECEIVING_QUEUE`에 있는 세 개의 지속 메시지에 데이터를 기록합니다. 이러한 메시지는 모두 동일한 IBM MQ 그룹 ID를 가지며 그룹의 마지막 메시지에 IBM MQ `LAST_MSG_IN_GROUP` 플래그 세트가 있습니다.

메시지의 데이터는 다음과 같습니다.

- 첫 번째 메시지:

```
Jenny Jones
```

- 두 번째 메시지:

```
John Smith
```

- 세 번째 메시지:

```
Jane Brown
```

관련 개념

[242 페이지의 『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

[244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 `enableQueueInputOutput`을 `True`로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 `enableClusterQueueInputOutput`도 `True`로 설정해야 합니다.

[245 페이지의 『예: 단일 파일에서 단일 메시지로 전송』](#)

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

246 페이지의 『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』

fteCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

249 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치를 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

202 페이지의 『새 파일 전송 시작』

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

255 페이지의 『파일에서 메시지로 전송 실패』

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

[MFT에서 사용하는 정규식](#)

예: 정규식 구분 기호를 사용하여 텍스트 파일 분할

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치를 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

이 태스크 정보

단일 텍스트 파일을 큐의 다중 메시지로 전송합니다. 파일은 가변 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다. 파일의 텍스트가 주어진 정규식과 일치하는 각 지점에서 텍스트 파일이 분할됩니다. 소스 파일은 /tmp/customers.text이며 다음 콘텐츠가 있습니다.

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314

Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42

Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

파일 분할 위치를 지정하는 정규식은 Customer\snumber:\s\d+이고 이는 임의의 숫자가 뒤에 오는 텍스트 "Customer number:"와 일치합니다. 명령행에 지정하는 정규식은 큰따옴표로 묶어서 명령 셸이 정규식을 평가하지 못하도록 해야 합니다. 정규식은 Java 정규식으로 평가됩니다. 자세한 정보는 [MFT에서 사용하는 정규식을 참조하십시오](#).

기본적으로 정규식이 일치시킬 수 있는 문자 수는 5로 설정되어 있습니다. 이 예에 사용된 정규식은 다섯자를 초과하는 문자열을 일치시킵니다. 5자보다 긴 일치를 사용하려면 에이전트 특성 파일을 편집하여 **maxDelimiterMatchLength** 특성을 포함시키십시오.

기본적으로 정규식과 일치하는 텍스트는 메시지에 포함되지 않습니다. 정규식과 일치하는 텍스트를 메시지에 포함시키려면 아래 예와 같이 **-qi** 매개변수를 사용하십시오. 소스 파일은 큐 관리자 QM_NEPTUNE에 연결되는 소스 에이전트 AGENT_NEPTUNE과 동일한 시스템에 있습니다. 목적지 큐(RECEIVING_QUEUE)는 큐 관리자 QM_MERCURY에 위치합니다. QM_MERCURY는 목적지 에이전트(AGENT_MERCURY)에서 사용하는 큐 관리자이기도 합니다. 전송 시 소스 파일이 섹션으로 분할되고 이들 각 섹션은 RECEIVING_QUEUE에 메시지로 기록됩니다.

프로시저

1. 다음 명령을 사용하여 목적지 에이전트를 중지하십시오.

```
fteStopAgent AGENT_MERCURY
```

2. AGENT_MERCURY의 에이전트 특성 파일에 다음 행을 추가하십시오.

```
maxDelimiterMatchLength=25
```

참고: **maxDelimiterMatchLength**의 값을 증가시키면 성능이 저하될 수 있습니다.

3. 다음 명령을 사용하여 목적지 에이전트를 시작하십시오.

```
fteStartAgent AGENT_MERCURY
```

4. 다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE  
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

소스 에이전트 AGENT_NEPTUNE은 /tmp/customers.text 파일에서 데이터를 읽고 이 데이터를 목적지 에이전트 AGENT_MERCURY에 전송합니다. 목적지 에이전트 AGENT_MERCURY는 데이터를 큐 RECEIVING_QUEUE에 있는 세 개의 지속 메시지에 기록합니다. 이러한 메시지는 모두 동일한 IBM MQ 그룹 ID를 가지며 그룹의 마지막 메시지에는 IBM MQ LAST_MSG_IN_GROUP 플래그 세트가 있습니다.

메시지의 데이터는 다음과 같습니다.

- 첫 번째 메시지:

```
Customer name: John Smith  
Customer contact details: john@example.net  
Customer number: 314
```

- 두 번째 메시지:

```
Customer name: Jane Brown  
Customer contact details: jane@example.com  
Customer number: 42
```

- 세 번째 메시지:

```
Customer name: James Jones  
Customer contact details: jjones@example.net  
Customer number: 26
```

관련 개념

[242 페이지의 『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 `enableQueueInputOutput`을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 `enableClusterQueueInputOutput`도 True로 설정해야 합니다.

245 페이지의 『예: 단일 파일에서 단일 메시지로 전송』

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

246 페이지의 『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』

fteCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

248 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 매개변수를 사용하십시오.

251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

202 페이지의 『새 파일 전송 시작』

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

MFTagent.properties 파일

MFT에서 사용하는 정규식

예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

이 태스크 정보

fteCreateTransfer 명령에 **-qmp true** 매개변수를 포함시키십시오. 이 예에서, 명령을 제출하는 사용자의 MQMD 사용자 ID는 `larmer`입니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true -t text /tmp/source_file.txt
```

목적지 에이전트 AGENT_SATURN에 의해 큐 관리자 MyQM의 큐 MY_QUEUE에 기록되는 첫 번째 메시지의 IBM MQ 메시지 특성은 다음 값으로 설정됩니다.

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=text
usr.WMQFTESourceAgent=AGENT_JUPITER
usr.WMQFTEDestinationAgent=AGENT_SATURN
usr.WMQFTEFileName=source_file.txt
usr.WMQFTEFileSize=1024
usr.WMQFTEFileLastModified=1273740879040
usr.WMQFTEFileIndex=0
usr.WMQFTEQMqmdUser=larmer
```

관련 개념

[242 페이지의 『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

[244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 enableQueueInputOutput을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 enableClusterQueueInputOutput도 True로 설정해야 합니다.

[245 페이지의 『예: 단일 파일에서 단일 메시지로 전송』](#)

fteCreateTransfer 명령에서 **-dq** 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

[246 페이지의 『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』](#)

fteCreateTransfer 명령의 **-qs** 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

[249 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 및 **-qi** 매개변수를 사용하십시오.

[248 페이지의 『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 **-dqdt** 매개변수를 사용하십시오.

[252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』](#)

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

[202 페이지의 『새 파일 전송 시작』](#)

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

[255 페이지의 『파일에서 메시지로 전송 실패』](#)

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

[목적지 큐에 기록된 메시지에서 MFT가 설정한 MQ 메시지 특성](#)

예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정

사용자 정의 메타데이터는 전송에 의해 목적지 큐에 기록된 첫 번째 메시지에 IBM MQ 메시지 특성으로 설정됩니다. IBM MQ 메시지 특성을 사용하면 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

이 태스크 정보

fteCreateTransfer 명령에 `-qmp true` 및 `-md account=123456` 매개변수를 포함하여 RFH2 헤더에서 `usr.account` 특성을 123456 로 설정하십시오.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM  
-qmp true -md account=123456 /tmp/source_file.txt
```

IBM MQ 메시지 특성의 표준 세트 외에도, 목적지 에이전트 `AGENT_SATURN`이 기록한 첫 번째 메시지의 메시지 헤더에서 큐 관리자 `MyQM`의 `MY_QUEUE` 큐에 사용자 정의 특성이 설정됩니다. 헤더에 다음 값이 설정됩니다.

```
usr.account=123456
```

접두부 `usr`이 사용자 정의 메타데이터 이름의 시작 부분에 추가됩니다.

관련 개념

242 페이지의 [『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

244 페이지의 [『파일에서 메시지로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 `enableQueueInputOutput`을 `True`로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 `enableClusterQueueInputOutput`도 `True`로 설정해야 합니다.

245 페이지의 [『예: 단일 파일에서 단일 메시지로 전송』](#)

fteCreateTransfer 명령에서 `-dq` 매개변수를 사용하여 큐를 파일 전송의 목적지로 지정할 수 있습니다. 소스 파일은 목적지 큐의 최대 메시지 길이 세트보다 작아야 합니다. 목적지 큐는 목적지 에이전트가 연결되는 큐 관리자와 동일한 큐 관리자에 있지 않아도 되지만 이러한 두 큐 관리자는 통신할 수 있어야 합니다.

246 페이지의 [『예: 길이를 단위로 단일 파일을 여러 메시지로 분할』](#)

fteCreateTransfer 명령의 `-qs` 매개변수를 사용하여 파일을 여러 IBM MQ 메시지로 분할할 수 있습니다. 파일이 고정 길이 섹션으로 분할되고 각 섹션은 개별 메시지로 기록됩니다.

249 페이지의 [『예: 정규식 구분 기호를 사용하여 텍스트 파일 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 다중 메시지로 전송하고 정규식 일치를 결과 메시지에 포함시킵니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 `-dqdt` 및 `-qi` 매개변수를 사용하십시오.

248 페이지의 [『예: 정규식 구분 기호를 사용하여 텍스트 파일을 여러 메시지로 분할』](#)

제공된 Java 정규식의 각 일치에서 파일을 분할하여 단일 텍스트 파일을 복수의 메시지에 전송합니다. 이를 수행하기 위해서는 **fteCreateTransfer** 명령의 `-dqdt` 매개변수를 사용하십시오.

251 페이지의 [『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』](#)

fteCreateTransfer 명령에서 `-qmp` 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지 특성을 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

202 페이지의 [『새 파일 전송 시작』](#)

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

목적지 큐에 기록된 메시지에서 MFT가 설정한 MQ 메시지 특성

예: 파일에서 메시지로 전송에 대한 사용자 정의 메시지 특성 추가

Managed File Transfer를 메시지에서 파일로 관리 전송에 사용 중인 경우 결과 메시지에 대한 사용자 정의 메시지 특성을 포함할 수 있습니다.

이 태스크 정보

다음 메소드 중 하나를 사용하여 사용자 정의 메시지 특성을 정의할 수 있습니다.

- 전송 요청에 대한 **-md** 매개변수를 지정하십시오. 자세한 정보는 252 페이지의 『예: 파일에서 메시지로 전송 시 사용자 정의 특성 설정』의 내용을 참조하십시오.
- Ant 태스크를 사용하십시오. `fte:filecopy` 또는 `fte:filemove`를 사용할 수 있습니다. 다음은 `fte:filecopy` 태스크 예제입니다.

```
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
  <property name="src.file" value="/home/user/file1.bin"/>
  <property name="dst.queue" value="TEST.QUEUE@qm2"/>
  <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="${FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="${src.file}" dstqueue="${dst.queue}"
dstmsgprops="true"/>

</fte:filecopy>

</target>
</project>
```

- 자원 모니터 및 변수 대체를 사용합니다. 다음 예는 일부 전송 태스크 XML을 나타냅니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
<name>METADATA</name>
<pollInterval units="minutes">5</pollInterval>
<batch maxSize="5"/>
<agent>AGENT1</agent>
<resources>
  <directory recursionLevel="0">e:\temp</directory>
</resources>
<triggerMatch>
  <conditions>
    <allOf>
      <condition>
        <fileMatch>
          <pattern>*.txt</pattern>
        </fileMatch>
      </condition>
    </allOf>
  </conditions>
</triggerMatch>
<tasks>
  <task>
    <name/>
    <transfer>
      <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
```

```

xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>mqjason.raleigh.ibm.com.</hostName>
      <userID>administrator</userID>
    </originator>
    <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
    <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
    <transferSet priority="0">
      <metaDataSet>
        <metaData key="FileName">${FileName}</metaData>
      </metaDataSet>
      <item checksumMethod="MD5" mode="text">
        <source disposition="delete" recursive="false">
          <file>${FilePath}</file>
        </source>
        <destination type="queue">
          <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
        </destination>
      </item>
    </transferSet>
    <job>
      <name>Metadata_example</name>
    </job>
  </managedTransfer>
</request>
</transfer>
</task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>

```

관련 태스크

251 페이지의 『예: 파일에서 메시지로 전송 시 IBM MQ 메시지 특성 설정』

fteCreateTransfer 명령에서 **-qmp** 매개변수를 사용하여 전송에 의해 목적지 큐에 기록되는 첫 번째 메시지에 IBM MQ 메시지 특성이 설정되는지 여부를 지정할 수 있습니다. IBM MQ 메시지를 통해 애플리케이션은 처리할 메시지를 선택하거나, IBM MQ 메시지 디스크립터(MQMD) 또는 MQRFH2 헤더에 액세스하지 않고도 메시지에 대한 정보를 검색할 수 있습니다.

관련 참조

[fte:filecopy Ant 태스크](#)

[fte:filemove Ant 태스크](#)

파일에서 메시지로 전송 실패

에이전트가 파일 데이터를 목적지 큐에 쓰기 시작한 이후 파일에서 메시지로 전송이 실패하는 경우 실패가 발생한 메시지를 이용하는 애플리케이션에 표시하기 위해 에이전트는 큐에 메시지를 씁니다.

실패가 발생한 경우 목적지 큐에 기록되는 메시지는 다음과 같습니다.

- 공백입니다.
- 에이전트가 목적지 큐에 기록한 이전 메시지와 동일한 IBM MQ 그룹 ID를 갖습니다.
- IBM MQ LAST_MSG_IN_GROUP 플래그 세트가 있습니다.
- 메시지 특성이 사용 가능한 경우 추가의 IBM MQ 메시지 특성을 포함합니다. 자세한 정보는 [목적지 큐에 기록된 메시지](#)에서 MFT가 설정한 MQ 메시지 특성 주제를 참조하십시오.

예

다음 명령을 사용하여 전송이 요청되었습니다.

```

fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE
-qmp true -qs 1K /tmp/source1.txt

```

source1.txt 파일은 48KB입니다. 전송 시 이 파일은 1KB 메시지로 분할되어 목적지 큐 (RECEIVING_QUEUE)에 기록됩니다.

전송이 진행 중인 동안 에이전트가 16개의 메시지를 RECEIVING_QUEUE에 모두 기록한 후 소스 에이전트에서 실패가 발생합니다.

에이전트가 공백 메시지를 RECEIVING_QUEUE에 기록합니다. 표준 메시지 특성 세트 외에 공백 메시지에는 다음과 같은 메시지 특성 세트가 있습니다.

```
usr.WMQFTEResultCode = 40
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

IBM MQ 9.3.0부터 파일로부터의 전송이 실패하면 구분 기호 크기 검사 오류로 인해 하나의 비어 있는 메시지만 전송됩니다. 또한 구분 기호가 대상 에이전트의 크기를 초과했기 때문에 전송에 실패한 경우 이 메시지에 사용자 특성을 추가합니다.

관련 개념

[242 페이지의 『파일에서 메시지로 데이터 전송』](#)

Managed File Transfer의 파일에서 메시지로 기능을 사용하여 파일의 데이터를 IBM MQ 큐의 단일 메시지 또는 다중 메시지에 전송할 수 있습니다.

관련 태스크

[244 페이지의 『파일에서 메시지로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 파일에서 메시지로 또는 메시지에서 파일로의 전송을 수행할 수 없습니다. 이 기능을 사용으로 설정하려면 에이전트 특성 enableQueueInputOutput을 True로 설정해야 합니다. IBM MQ 클러스터된 큐에 대한 쓰기를 사용으로 설정하려면 에이전트 특성 enableClusterQueueInputOutput도 True로 설정해야 합니다.

[202 페이지의 『새 파일 전송 시작』](#)

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

관련 참조

[MFTagent.properties 파일](#)

[목적지 큐에 기록된 메시지에서의 MFT가 설정한 MQ 메시지 특성](#)

메시지에서 파일로 데이터 전송

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

파일에서 메시지로 전송에 대한 정보는 [242 페이지의 『파일에서 메시지로 데이터 전송』](#)의 내용을 참조하십시오.



주의: 메시지에서 파일로 전송의 소스 에이전트는 프로토콜 브릿지 에이전트 또는 Connect:Direct 브릿지 에이전트가 될 수 없습니다.

IBM MQ 메시지 데이터를 파일로 전송할 수 있습니다. 다음과 같은 유형의 메시지에서 파일로 전송이 지원됩니다.

- 단일 메시지에서 단일 파일로
- 다중 메시지에서 단일 파일로
- 동일한 IBM MQ 그룹 ID를 갖는 다중 메시지에서 단일 파일로
- 다중 메시지에서 단일 파일로(각 메시지의 데이터 사이에 있는 텍스트 또는 2진 구분 기호도 파일에 기록됨)

대형 메시지 또는 다수의 작은 메시지에서 파일을 전송하는 경우, 일부 IBM MQ 또는 Managed File Transfer 특성을 변경해야 합니다. 자세한 정보는 [메시지 크기와 연관된 MQ 속성 및 MFT 특성 설정에 대한 자세한 내용을 참조하십시오.](#)

메시지에서 파일로 전송에서 소스 에이전트는 IBM MQ의 이전 버전에서 파괴적인 GET과 달리 소스 큐에서 메시지를 찾아봅니다. 모든 메시지(메시지 그룹을 사용할 경우 그룹의 모든 메시지)를 찾아서 데이터를 대상 파일에 쓴 후에는 메시지가 소스 큐에서 제거됩니다. 이 경우 전송이 실패하거나 취소된 경우에도 메시지가 소스 큐에 남아 있을 수 있습니다. 이런 변경사항 때문에 BROWSE에 대한 권한을 GET 권한과 함께 제공해야 메시지에서 파일로 전송할 수 있습니다.

Managed File Transfer 는 전송 요청 XML 페이로드 내의 groupId 속성 값과 전송 ID를 비교합니다. 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.

관련 태스크

[221 페이지의 『예: MFT 자원 구성』](#)

fteCreateMonitor 명령과 함께 **-mq** 매개변수를 사용하여 자원 모니터가 모니터할 자원으로 IBM MQ 큐를 지정할 수 있습니다.

관련 참조

[소스 큐의 메시지에서 MFT가 읽은 MQ 메시지 특성](#)

[메시지 크기와 연관된 MQ 속성 및 MFT 특성 설정에 대한 자세한 내용](#)

메시지에서 파일로 전송을 수행하도록 에이전트 구성

기본적으로 에이전트는 메시지에서 파일로 또는 파일에서 메시지로 전송을 수행할 수 없습니다. 이 기능을 사용 가능하게 하려면 enableQueueInputOutput 에이전트 특성을 True로 설정해야 합니다.

이 태스크 정보

enableQueueInputOutput 특성이 True로 설정되지 않은 소스 에이전트에서 메시지에서 파일로 전송을 수행하는 경우 전송에 실패합니다. 조정 큐 관리자에 발행되는 전송 로그 메시지에 다음 메시지가 포함됩니다.

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.  
The agent must have enableQueueInputOutput=true set in the agent.properties file  
to support transferring from a queue.
```

에이전트가 큐에 쓰고 읽을 수 있도록 하려면 다음 단계를 수행하십시오.

프로시저

1. **fteStopAgent** 명령을 사용하여 소스 에이전트를 중지하십시오.
2. `agent.properties` 파일을 편집하여 `enableQueueInputOutput=true` 행을 포함시키십시오.
`agent.properties` 파일은 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name` 디렉토리에 있습니다.
3. **fteStartAgent** 명령을 사용하여 소스 에이전트를 시작하십시오.

관련 개념

[256 페이지의 『메시지에서 파일로 데이터 전송』](#)

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

관련 태스크

[258 페이지의 『예: 큐에서 단일 파일로 전송』](#)

fteCreateTransfer 명령과 함께 **-sq** 매개변수를 사용하여 IBM MQ 큐를 파일 전송의 소스로 지정할 수 있습니다.

[259 페이지의 『예: 메시지 그룹을 큐에서 단일 파일로 전송』](#)

fteCreateTransfer 명령과 함께 **-sq** 및 **-sqgi** 매개변수를 사용하여 IBM MQ 큐의 단일 전체 그룹을 파일 전송의 소스로 지정할 수 있습니다.

[260 페이지의 『예: 각 메시지의 데이터 앞에 텍스트 구분 기호 삽입』](#)

텍스트 모드로 소스 큐에서 파일로 전송하는 경우 **-sq**, **-sqdt** 및 **-sqdp** 매개변수를 **fteCreateTransfer** 명령과 함께 사용하여 개별 메시지의 데이터 앞에 해당 텍스트 구분 기호가 삽입되도록 지정할 수 있습니다.

261 페이지의 『예: 각 메시지의 데이터 뒤에 2진 구분 기호 삽입』

2진 모드로 소스 큐에서 파일로 전송하는 경우 **fteCreateTransfer** 명령에 **-sq**, **-sqdb** 및 **-sqdp** 매개변수를 사용하여 개별 메시지의 데이터 뒤에 2진 구분 기호가 삽입되도록 지정할 수 있습니다.

227 페이지의 『큐 모니터링 및 변수 대체 사용』

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

264 페이지의 『예: IBM MQ 메시지 특성을 사용하여 메시지에서 파일 전송 실패』

`usr.UserReturnCode` IBM MQ 메시지 특성을 0이 아닌 값으로 설정하여 메시지에서 파일로의 전송이 실패하도록 할 수 있습니다. 또한 `usr.UserSupplement` IBM MQ 메시지 특성을 설정하여 실패한 이유에 대한 추가 정보를 지정할 수 있습니다.

관련 참조

[MFTagent.properties](#) 파일

예: 큐에서 단일 파일로 전송

fteCreateTransfer 명령과 함께 **-sq** 매개변수를 사용하여 IBM MQ 큐를 파일 전송의 소스로 지정할 수 있습니다.

이 태스크 정보

소스 데이터가 `START_QUEUE` 큐의 세 메시지에 포함되어 있습니다. 이 큐는 소스 에이전트의 큐 관리자 `QM_NEPTUNE`에 있어야 합니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
                 -da AGENT_VENUS -df /out/three_to_one.txt
                 -sq START_QUEUE
```

`START_QUEUE` 큐에 있는 메시지의 데이터는 `AGENT_VENUS`가 실행 중인 시스템의 `/out/three_to_one.txt` 파일에 기록됩니다.

관련 개념

256 페이지의 『메시지에서 파일로 데이터 전송』

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

관련 태스크

257 페이지의 『메시지에서 파일로 전송을 수행하도록 에이전트 구성』

기본적으로 에이전트는 메시지에서 파일로 또는 파일에서 메시지로 전송을 수행할 수 없습니다. 이 기능을 사용 가능하게 하려면 `enableQueueInputOutput` 에이전트 특성을 `True`로 설정해야 합니다.

259 페이지의 『예: 메시지 그룹을 큐에서 단일 파일로 전송』

fteCreateTransfer 명령과 함께 **-sq** 및 **-sqgi** 매개변수를 사용하여 IBM MQ 큐의 단일 전체 그룹을 파일 전송의 소스로 지정할 수 있습니다.

260 페이지의 『예: 각 메시지의 데이터 앞에 텍스트 구분 기호 삽입』

텍스트 모드로 소스 큐에서 파일로 전송하는 경우 **-sq**, **-sqdt** 및 **-sqdp** 매개변수를 **fteCreateTransfer** 명령과 함께 사용하여 개별 메시지의 데이터 앞에 해당 텍스트 구분 기호가 삽입되도록 지정할 수 있습니다.

[261 페이지의 『예: 각 메시지의 데이터 뒤에 2진 구분 기호 삽입』](#)

2진 모드로 소스 큐에서 파일로 전송하는 경우 **fteCreateTransfer** 명령에 **-sq**, **-sqdb** 및 **-sqdp** 매개변수를 사용하여 개별 메시지의 데이터 뒤에 2진 구분 기호가 삽입되도록 지정할 수 있습니다.

[227 페이지의 『큐 모니터링 및 변수 대체 사용』](#)

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

[264 페이지의 『예: IBM MQ 메시지 특성을 사용하여 메시지에서 파일 전송 실패』](#)

`usr.UserReturnCode` IBM MQ 메시지 특성을 0이 아닌 값으로 설정하여 메시지에서 파일로의 전송이 실패하도록 할 수 있습니다. 또한 `usr.UserSupplement` IBM MQ 메시지 특성을 설정하여 실패한 이유에 대한 추가 정보를 지정할 수 있습니다.

관련 참조

소스 큐의 메시지에서 MFT가 읽은 MQ 메시지 특성

fteCreateTransfer: 새 파일 전송 시작

예: 메시지 그룹을 큐에서 단일 파일로 전송

fteCreateTransfer 명령과 함께 **-sq** 및 **-sqgi** 매개변수를 사용하여 IBM MQ 큐의 단일 전체 그룹을 파일 전송의 소스로 지정할 수 있습니다.

이 태스크 정보

이 예에서는 `START_QUEUE` 큐에 10개의 메시지가 있습니다. 이 큐는 소스 에이전트의 큐 관리자 `QM_NEPTUNE`에 있어야 합니다. 처음 세 개의 메시지는 IBM MQ 그룹 ID가

41424b3ef3a22020202020202020202020202020202020201111인 그룹에 속합니다. 이 그룹은 완전한 그룹이 아닙니다. 다음 다섯 개의 메시지는 IBM MQ 그룹 ID가

41424b3ef3a22020202020202020202020202020202020202222인 그룹에 속합니다. 이 그룹은 완전한 그룹입니다. 나머지 두 개의 메시지는 IBM MQ 그룹 ID가

41424b3ef3a22020202020202020202020202020202020203333인 그룹에 속합니다. 이 그룹은 완전한 그룹입니다.

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS
                  -df /out/group.txt -sqgi -sq START_QUEUE
```

`START_QUEUE` 큐의 첫 번째 완료 그룹이며 IBM MQ 그룹 ID가

41424b3ef3a22020202020202020202020202020202020202222인 그룹에 속하는 메시지의 데이터는 시스템의 `/out/group.txt` 파일에 기록됩니다. 여기서, `AGENT_VENUS`는 실행 중입니다.

관련 개념

[256 페이지의 『메시지에서 파일로 데이터 전송』](#)

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

관련 태스크

[257 페이지의 『메시지에서 파일로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 메시지에서 파일로 또는 파일에서 메시지로 전송을 수행할 수 없습니다. 이 기능을 사용 가능하게 하려면 `enableQueueInputOutput` 에이전트 특성을 `True`로 설정해야 합니다.

[258 페이지의 『예: 큐에서 단일 파일로 전송』](#)

fteCreateTransfer 명령과 함께 **-sq** 매개변수를 사용하여 IBM MQ 큐를 파일 전송의 소스로 지정할 수 있습니다.

260 페이지의 『예: 각 메시지의 데이터 앞에 텍스트 구분 기호 삽입』

텍스트 모드로 소스 큐에서 파일로 전송하는 경우 **-sq**, **-sqdt** 및 **-sqdp** 매개변수를 **fteCreateTransfer** 명령과 함께 사용하여 개별 메시지의 데이터 앞에 해당 텍스트 구분 기호가 삽입되도록 지정할 수 있습니다.

261 페이지의 『예: 각 메시지의 데이터 뒤에 2진 구분 기호 삽입』

2진 모드로 소스 큐에서 파일로 전송하는 경우 **fteCreateTransfer** 명령에 **-sq**, **-sqdb** 및 **-sqdp** 매개변수를 사용하여 개별 메시지의 데이터 뒤에 2진 구분 기호가 삽입되도록 지정할 수 있습니다.

227 페이지의 『큐 모니터링 및 변수 대체 사용』

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

264 페이지의 『예: IBM MQ 메시지 특성을 사용하여 메시지에서 파일 전송 실패』

`usr.UserReturnCode` IBM MQ 메시지 특성을 0이 아닌 값으로 설정하여 메시지에서 파일로의 전송이 실패하도록 할 수 있습니다. 또한 `usr.UserSupplement` IBM MQ 메시지 특성을 설정하여 실패한 이유에 대한 추가 정보를 지정할 수 있습니다.

관련 참조

fteCreateTransfer: 새 파일 전송 시작

예: 각 메시지의 데이터 앞에 텍스트 구분 기호 삽입

텍스트 모드로 소스 큐에서 파일로 전송하는 경우 **-sq**, **-sqdt** 및 **-sqdp** 매개변수를 **fteCreateTransfer** 명령과 함께 사용하여 개별 메시지의 데이터 앞에 해당 텍스트 구분 기호가 삽입되도록 지정할 수 있습니다.

이 태스크 정보

이 예제에서는 `START_QUEUE` 큐에 네 개의 메시지가 있습니다. 이 큐는 소스 에이전트의 큐 관리자 `QM_NEPTUNE`에 있습니다. 각 메시지의 데이터 앞에 삽입될 텍스트 구분 기호는 Java 리터럴 문자열로 표현될 수 있습니다(예: `\n\u002D\u002D\u002D\n`).

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt
-t text -sqdt "\n\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

소스 에이전트 `AGENT_NEPTUNE`에 의해 `START_QUEUE`의 네 메시지 각각에서 제공되는 데이터의 시작 부분에 텍스트 구분 기호가 추가됩니다. 이 데이터는 목적지 파일 `/out/output.txt`에 기록됩니다.

관련 개념

256 페이지의 『메시지에서 파일로 데이터 전송』

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

관련 태스크

257 페이지의 『메시지에서 파일로 전송을 수행하도록 에이전트 구성』

기본적으로 에이전트는 메시지에서 파일로 또는 파일에서 메시지로 전송을 수행할 수 없습니다. 이 기능을 사용 가능하게 하려면 `enableQueueInputOutput` 에이전트 특성을 `True`로 설정해야 합니다.

258 페이지의 『예: 큐에서 단일 파일로 전송』

fteCreateTransfer 명령과 함께 **-sq** 매개변수를 사용하여 IBM MQ 큐를 파일 전송의 소스로 지정할 수 있습니다.

259 페이지의 『예: 메시지 그룹을 큐에서 단일 파일로 전송』

fteCreateTransfer 명령과 함께 **-sq** 및 **-sqgi** 매개변수를 사용하여 IBM MQ 큐의 단일 전체 그룹을 파일 전송의 소스로 지정할 수 있습니다.

261 페이지의 『예: 각 메시지의 데이터 뒤에 2진 구분 기호 삽입』

2진 모드로 소스 큐에서 파일로 전송하는 경우 **fteCreateTransfer** 명령에 **-sq**, **-sqdb** 및 **-sqdp** 매개변수를 사용하여 개별 메시지의 데이터 뒤에 2진 구분 기호가 삽입되도록 지정할 수 있습니다.

227 페이지의 『큐 모니터링 및 변수 대체 사용』

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

264 페이지의 『예: IBM MQ 메시지 특성을 사용하여 메시지에서 파일 전송 실패』

`usr.UserReturnCode` IBM MQ 메시지 특성을 0이 아닌 값으로 설정하여 메시지에서 파일로의 전송이 실패하도록 할 수 있습니다. 또한 `usr.UserSupplement` IBM MQ 메시지 특성을 설정하여 실패한 이유에 대한 추가 정보를 지정할 수 있습니다.

관련 참조

fteCreateTransfer: 새 파일 전송 시작

예: 각 메시지의 데이터 뒤에 2진 구분 기호 삽입

2진 모드로 소스 큐에서 파일로 전송하는 경우 **fteCreateTransfer** 명령에 **-sq**, **-sqdb** 및 **-sqdp** 매개변수를 사용하여 개별 메시지의 데이터 뒤에 2진 구분 기호가 삽입되도록 지정할 수 있습니다.

이 태스크 정보

이 예제에서는 `START_QUEUE` 큐에 세 개의 메시지가 있습니다. 이 큐는 소스 에이전트의 큐 관리자 `QM_NEPTUNE`에 있습니다. 각 메시지의 데이터 뒤에 삽입될 2진 구분 기호는 십진로 구분된 16진 바이트 목록으로 표현해야 합니다(예: `x34`, `xE7`, `xAE`).

프로시저

다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

2진 구분 기호는 소스 에이전트(`AGENT_NEPTUNE`)에 의해 `START_QUEUE`의 세 메시지에서 나오는 데이터에 추가됩니다. 이 데이터는 목적지 파일 `/out/binary.file`에 기록됩니다.

관련 개념

256 페이지의 『메시지에서 파일로 데이터 전송』

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

관련 태스크

257 페이지의 『메시지에서 파일로 전송을 수행하도록 에이전트 구성』

기본적으로 에이전트는 메시지에서 파일로 또는 파일에서 메시지로 전송을 수행할 수 없습니다. 이 기능을 사용 가능하게 하려면 `enableQueueInputOutput` 에이전트 특성을 `True`로 설정해야 합니다.

258 페이지의 『예: 큐에서 단일 파일로 전송』

fteCreateTransfer 명령과 함께 **-sq** 매개변수를 사용하여 IBM MQ 큐를 파일 전송의 소스로 지정할 수 있습니다.

259 페이지의 『예: 메시지 그룹을 큐에서 단일 파일로 전송』

fteCreateTransfer 명령과 함께 **-sq** 및 **-sqgi** 매개변수를 사용하여 IBM MQ 큐의 단일 전체 그룹을 파일 전송의 소스로 지정할 수 있습니다.

260 페이지의 『예: 각 메시지의 데이터 앞에 텍스트 구분 기호 삽입』

텍스트 모드로 소스 큐에서 파일로 전송하는 경우 **-sq**, **-sqdt** 및 **-sqdp** 매개변수를 **fteCreateTransfer** 명령과 함께 사용하여 개별 메시지의 데이터 앞에 해당 텍스트 구분 기호가 삽입되도록 지정할 수 있습니다.

227 페이지의 『큐 모니터링 및 변수 대체 사용』

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

264 페이지의 『예: IBM MQ 메시지 특성을 사용하여 메시지에서 파일 전송 실패』

`usr.UserReturnCode` IBM MQ 메시지 특성을 0이 아닌 값으로 설정하여 메시지에서 파일로의 전송이 실패하도록 할 수 있습니다. 또한 `usr.UserSupplement` IBM MQ 메시지 특성을 설정하여 실패한 이유에 대한 추가 정보를 지정할 수 있습니다.

관련 참조

fteCreateTransfer: 새 파일 전송 시작

큐 모니터링 및 변수 대체 사용

fteCreateMonitor 명령을 사용하여 큐를 모니터링하고 메시지를 모니터링되는 큐에서 파일로 전송할 수 있습니다. 모니터링되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

이 태스크 정보

이 예제에서는 소스 에이전트를 `AGENT_VENUS`라고 하며 이 에이전트가 `QM_VENUS`에 연결됩니다. `AGENT_VENUS`가 모니터링하는 큐는 `START_QUEUE`이며 `QM_VENUS`에 있습니다. 에이전트는 30분마다 큐를 폴링합니다.

메시지의 완료 그룹이 큐에 기록되면 모니터 태스크가 여러 목적지 에이전트 중 하나에 있는 파일에 메시지 그룹을 보내며 모든 목적지 에이전트는 큐 관리자 `QM_MARS`에 연결됩니다. 메시지 그룹이 전송되는 파일의 이름은 그룹에 있는 첫 번째 메시지의 IBM MQ 메시지 특성 `usr.fileName`을 통해 정의됩니다. 메시지 그룹이 송신되는 에이전트의 이름은 그룹에 있는 첫 번째 메시지의 IBM MQ 메시지 특성 `usr.toAgent`를 통해 정의됩니다. `usr.toAgent` 헤더가 설정되지 않은 경우 목적지 에이전트에서 사용될 기본값은 `AGENT_MAGENTA`입니다.

`useGroups="true"`를 지정할 경우 `groupId="${GROUPID}"`를 지정하지 않으면, 전송에서는 큐의 첫 번째 메시지만 사용합니다. 따라서 예를 들어, 변수 대체를 사용하여 `fileName`을 생성하는 경우 `a.txt`의 콘텐츠가 올 바르지 않을 수 있습니다. 모니터에서 `fileName`을 생성하지만 전송 시 `fileName`이라는 파일을 생성해야 하는 메시지가 아닌 다른 메시지를 실제로 가져오기 때문입니다.

프로시저

1. 모니터가 트리거되는 경우 수행하는 태스크를 정의하는 태스크 XML을 작성하십시오.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="${toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

```

    </item>
  </transferSet>
</managedTransfer>
</request>

```

IBM MQ 메시지 헤더의 값으로 바뀌는 변수는 **굵은체**로 강조표시되어 있습니다. 이 태스크 XML은 /home/USER1/task.xml 파일에 저장됩니다.

2. 자원 모니터를 작성하여 START_QUEUE 큐를 모니터하십시오.

다음 명령을 제출하십시오.

```

fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA

```

3. 사용자 또는 프로그램이 메시지 그룹을 START_QUEUE 큐에 기록합니다.

이 그룹의 첫 번째 메시지에 다음 IBM MQ 메시지 특성이 설정되어 있습니다.

```

usr.fileName=larmer
usr.toAgent=AGENT_VIOLET

```

4. 완료 그룹이 기록되면 모니터가 트리거됩니다. 에이전트가 IBM MQ 메시지 특성을 태스크 XML로 대체합니다.

이 결과로 태스크 XML이 다음과 같이 변환됩니다.

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

결과

태스크 XML에 의해 정의되는 전송이 수행됩니다. AGENT_VENUS가 START_QUEUE에서 읽은 전체 메시지 그룹은 AGENT_VIOLET이 실행 중인 시스템의 /reports/larmer.rpt 파일에 기록됩니다.

다음에 수행할 작업

개별 파일에 각 메시지 전송

큐를 모니터하려는 경우 모든 메시지를 개별 파일로 전송했으면 이 주제에서 이전에 설명한 것과 비슷한 기법을 사용할 수 있습니다.

1. **fteCreateMonitor** 명령에 **-tr completeGroups** 매개변수를 지정하여 이전에 설명한 대로 모니터를 작성하십시오.
2. 태스크 XML에서 다음을 지정하십시오.

```

<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>

```


그러나 소스 큐에 메시지를 넣는 경우 IBM MQ 그룹에 넣지 마십시오. IBM MQ 메시지 특성을 각 메시지에 추가하십시오. 예를 들어, 각 메시지마다 고유 파일 이름 값을 사용하여 `usr.filename` 특성을 지정하십시오. 이 경우 Managed File Transfer Agent가 효과적으로 소스 큐의 각 메시지를 개별 그룹으로 처리합니다.

관련 개념

256 페이지의 『메시지에서 파일로 데이터 전송』

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

222 페이지의 『변수 대체를 사용하여 MFT 자원 모니터 태스크 사용자 정의』

활성 자원 모니터의 트리거 조건이 충족되면, 정의된 태스크가 호출됩니다. 매번 목적지 에이전트 또는 목적지 파일 이름이 동일한 전송 또는 명령 태스크를 호출하는 것 외에도 런타임 시 태스크 정의를 수정할 수도 있습니다. 태스크 정의의 XML에 변수 이름을 삽입하여 수정할 수 있습니다. 모니터가 트리거 조건이 충족되고 태스크 정의에 변수 이름이 포함되어 있다고 판별하면 변수 이름을 변수 값으로 대체한 후 태스크를 호출합니다.

큐 자원 모니터에 의해 시작된 전송에 의해 작성된 목적지 파일에 잘못된 데이터가 포함된 경우 수행할 작업

관련 태스크

216 페이지의 『명령 및 스크립트를 시작하도록 MFT 모니터 태스크 구성』

자원 모니터는 연관된 태스크로 파일 전송을 수행하도록 제한되지 않습니다. 실행 가능 프로그램, Ant 스크립트 또는 JCL 작업을 포함하여 모니터링 에이전트에서 다른 명령을 호출하도록 모니터를 구성할 수도 있습니다. 명령을 호출하려면 인수 및 특성과 같은 해당되는 명령 호출 매개변수와 함께 하나 이상의 명령 요소를 포함하도록 모니터 태스크 정의의 XML을 편집하십시오.

221 페이지의 『예: MFT 자원 구성』

fteCreateMonitor 명령과 함께 **-mq** 매개변수를 사용하여 자원 모니터가 모니터링할 자원으로 IBM MQ 큐를 지정할 수 있습니다.

관련 참조

fteCreateMonitor: MFT 자원 모니터 작성

소스 큐의 메시지에서 MFT가 읽은 MQ 메시지 특성

예: IBM MQ 메시지 특성을 사용하여 메시지에서 파일 전송 실패

`usr.UserReturnCode` IBM MQ 메시지 특성을 0이 아닌 값으로 설정하여 메시지에서 파일로의 전송이 실패하도록 할 수 있습니다. 또한 `usr.UserSupplement` IBM MQ 메시지 특성을 설정하여 실패한 이유에 대한 추가 정보를 지정할 수 있습니다.

이 태스크 정보

이 예제에서는 INPUT_QUEUE 큐와 /home/user/output.file 파일 간에 전송이 진행 중입니다.

사용자는 메시지를 작성하여 INPUT_QUEUE 큐에 배치하는 중입니다. 소스 에이전트가 INPUT_QUEUE 큐의 메시지를 이용하며 목적지 에이전트에 전송 데이터를 보내는 중입니다. 목적지 에이전트는 이 데이터를 /home/user/output.file 파일에 기록하는 중입니다.

메시지를 INPUT_QUEUE 큐에 기록하는 사용자가 진행 중인 전송을 중지하고 목적지 파일에 이미 기록된 모든 데이터를 삭제하려 합니다.

프로시저

1. 사용자가 다음 IBM MQ 메시지 특성이 설정된 INPUT_QUEUE 큐에 메시지를 기록합니다.

```
usr.UserReturnCode=1
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. 소스 에이전트가 IBM MQ 메시지 특성을 읽고 큐에서 메시지 처리를 중지합니다. 목적지 에이전트는 목적지 디렉토리에 기록된 모든 파일 데이터를 삭제합니다.
3. 소스 에이전트가 전송 로그 메시지를 조정 큐 관리자에게 보내 전송 실패를 보고합니다.

메시지에는 다음 정보가 포함됩니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020207e970d4920008702" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T21:28:09.593Z">progress</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </destinationAgent>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <transferSet index="0" size="1"
    startTime="2008-11-02T21:28:09.281Z"
    total="1">
    <item mode="binary">
      <source>
        <queue>INPUT_QUEUE@QM1</queue>
      </source>
      <destination exist="error">
        <file>/home/user/output.file</file>
      </destination>
      <status resultCode="1">
        <supplement>Cancelling transfer - sent wrong data.</supplement>
      </status>
    </item>
  </transferSet>
</transaction>
```

관련 개념

[256 페이지의 『메시지에서 파일로 데이터 전송』](#)

Managed File Transfer의 메시지에서 파일로 기능을 이용하면 IBM MQ 큐에 있는 하나 이상의 메시지의 데이터를 파일, 데이터 세트(z/OS), 또는 사용자 파일 공간으로 전송할 수 있습니다. IBM MQ 메시지를 작성 또는 처리하는 애플리케이션이 있는 경우, Managed File Transfer의 메시지에서 파일로 기능을 사용하여 이러한 메시지를 Managed File Transfer 네트워크의 시스템에 있는 파일에 전송할 수 있습니다.

관련 태스크

[257 페이지의 『메시지에서 파일로 전송을 수행하도록 에이전트 구성』](#)

기본적으로 에이전트는 메시지에서 파일로 또는 파일에서 메시지로 전송을 수행할 수 없습니다. 이 기능을 사용 가능하게 하려면 enableQueueInputOutput 에이전트 특성을 True로 설정해야 합니다.

[258 페이지의 『예: 큐에서 단일 파일로 전송』](#)

fteCreateTransfer 명령과 함께 **-sq** 매개변수를 사용하여 IBM MQ 큐를 파일 전송의 소스로 지정할 수 있습니다.

[259 페이지의 『예: 메시지 그룹을 큐에서 단일 파일로 전송』](#)

fteCreateTransfer 명령과 함께 **-sq** 및 **-sqgi** 매개변수를 사용하여 IBM MQ 큐의 단일 전체 그룹을 파일 전송의 소스로 지정할 수 있습니다.

[260 페이지의 『예: 각 메시지의 데이터 앞에 텍스트 구분 기호 삽입』](#)

텍스트 모드로 소스 큐에서 파일로 전송하는 경우 **-sq**, **-sqdt** 및 **-sqdp** 매개변수를 **fteCreateTransfer** 명령과 함께 사용하여 개별 메시지의 데이터 앞에 해당 텍스트 구분 기호가 삽입되도록 지정할 수 있습니다.

[261 페이지의 『예: 각 메시지의 데이터 뒤에 2진 구분 기호 삽입』](#)

2진 모드로 소스 큐에서 파일로 전송하는 경우 **fteCreateTransfer** 명령에 **-sq**, **-sqdb** 및 **-sqdp** 매개변수를 사용하여 개별 메시지의 데이터 뒤에 2진 구분 기호가 삽입되도록 지정할 수 있습니다.

[227 페이지의 『큐 모니터링 및 변수 대체 사용』](#)

fteCreateMonitor 명령을 사용하여 큐를 모니터하고 메시지를 모니터되는 큐에서 파일로 전송할 수 있습니다. 모니터되는 큐에서 읽을 첫 번째 메시지의 IBM MQ 메시지 특성 값을 태스크 XML 정의에서 대체할 수 있으며 이 값을 사용하여 전송 동작을 정의할 수 있습니다.

관련 참조

소스 큐의 메시지에서 MFT가 읽은 MQ 메시지 특성

프로토콜 브릿지

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

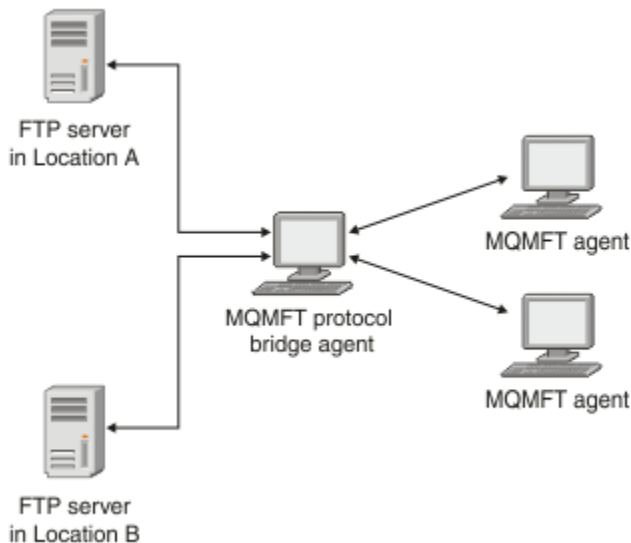
프로토콜 브릿지는 Managed File Transfer의 서비스 컴포넌트의 일부로 사용 가능합니다. MFT를 실행 중인 단일 시스템에 서로 다른 파일 서버에 연결하는 복수의 전용 에이전트가 있을 수 있습니다.

프로토콜 브릿지 에이전트를 사용하여 동시에 여러 엔드포인트로 파일을 전송할 수 있습니다. MFT는 파일을 전송할 다른 프로토콜 파일 서버를 정의하기 위해 편집할 수 있는 **ProtocolBridgeProperties.xml**이라는 파일을 제공합니다. **fteCreateBridgeAgent** 명령은 기본 프로토콜 파일 서버에 대한 자세한 내용을 **ProtocolBridgeProperties.xml**에 추가합니다. 이 파일은 [프로토콜 브릿지 특성 파일 형식](#)에 설명되어 있습니다.

프로토콜 브릿지 에이전트를 사용하여 다음과 같은 조치를 수행할 수 있습니다.

- FTP, FTPS 또는 SFTP를 사용하여 MFT 네트워크에서 원격 서버로 파일 업로드
- FTP, FTPS 또는 SFTP를 사용하여 원격 서버에서 MFT 네트워크로 파일 다운로드

참고: 프로토콜 브릿지 에이전트는 해당 절대 파일 경로를 통해 파일에 액세스할 수 있도록 허용하는 FTP, FTPS 또는 SFTP 서버만 지원할 수 있습니다. 전송 요청에 상대 파일 경로가 지정되어 있으면, 프로토콜 브릿지 에이전트는 프로토콜 서버에 로그인하는 데 사용되는 홈 디렉토리를 기반으로 하여 상대 경로를 절대 파일 경로로 변환하려고 시도합니다. 현재 디렉토리 기반의 파일에만 액세스할 수 있도록 하는 이러한 프로토콜 서버는 프로토콜 브릿지 에이전트에서 지원되지 않습니다.



이 다이어그램은 서로 다른 위치에 있는 두 개의 FTP 서버를 보여줍니다. FTP 서버는 Managed File Transfer 에이전트와 파일을 교환하는 데 사용됩니다. 프로토콜 브릿지 에이전트는 FTP 서버와 나머지 MFT 네트워크 사이에 위치하며 두 FTP 서버 모두와 통신하도록 구성됩니다.

MFT 네트워크에 프로토콜 브릿지 에이전트 외에 다른 에이전트가 있는지 확인하십시오. 프로토콜 브릿지 에이전트는 FTP, FTPS 또는 SFTP 서버에 대한 전용 브릿지이며 전송된 파일을 로컬 디스크에 기록하지 않습니다. FTP, FTPS 또는 SFTP 서버 간에 파일을 전송하려는 경우, 프로토콜 브릿지 에이전트를 FTP, FTPS 또는 SFTP 서

버를 나타내는 파일 전송에 대한 목적지 또는 소스로 사용해야 하고 다른 표준 에이전트를 해당 소스 또는 목적지로 사용해야 합니다.

프로토콜 브릿지를 사용하여 파일을 전송하는 경우 브릿지는 전송할 파일이 있는 소스 또는 목적지 디렉토리를 읽을 수 있는 권한이 있어야 합니다. 예를 들어, 실행 권한(d-x-x-x)만 있는 /home/fte/bridge 디렉토리에서 파일을 전송하려는 경우 이 디렉토리에서 시도하는 전송이 실패하며 다음과 같은 오류 메시지가 표시됩니다.

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

프로토콜 브릿지 에이전트 구성

프로토콜 브릿지 에이전트는 표준 MFT 에이전트와 비슷합니다. **fteCreateBridgeAgent** 명령을 사용하여 프로토콜 브릿지 에이전트를 작성하십시오. [프로토콜 브릿지 특성 파일 형식에 설명된 ProtocolBridgeProperties.xml](#) 파일을 사용하여 프로토콜 브릿지 에이전트를 구성할 수 있습니다. 이전 버전을 사용 중인 경우 [고급 에이전트 특성: 프로토콜 브릿지 및 고급 에이전트 특성: 프로토콜 브릿지 에이전트](#) 로깅에 설명된 특정 프로토콜 브릿지 특성을 사용하여 에이전트를 구성하십시오. [274 페이지의 『파일 서버의 신임 정보 매핑』](#)에서 설명한 바와 같이 모든 버전에 대해 신임 정보 매핑을 구성할 수도 있습니다. 특정 프로토콜 파일 서버의 프로토콜 브릿지 에이전트를 구성한 후에는 해당 용도로만 해당 에이전트를 사용할 수 있습니다.

프로토콜 브릿지 복구

파일 서버가 사용 불가능하여 프로토콜 브릿지 에이전트가 파일 서버에 연결할 수 없는 경우 파일 서버가 사용 가능해질 때까지 모든 파일 전송 요청이 큐에서 대기합니다. 프로토콜 브릿지 에이전트가 올바르게 않은 신임 정보를 사용하여 파일 서버에 연결할 수 없는 경우 전송이 실패하고 전송 로그 메시지에 이 오류가 반영됩니다. 어떤 이유로 프로토콜 브릿지 에이전트가 종료된 경우라도, 요청된 모든 파일 전송이 보유하고 프로토콜 브릿지를 재 시작하면 계속 수행합니다.

파일 전송 시 일반적으로 파일은 목적지에 임시 파일로 기록된 후 전송이 완료되면 이름이 바뀝니다. 그러나 전송 목적지가 쓰기 전용으로 구성된 프로토콜 파일 서버인 경우(사용자가 프로토콜 파일 서버에 파일을 업로드할 수 있지만 업로드한 파일을 어떤 식으로든 변경할 수 없는 경우) 전송된 파일이 목적지에 직접 기록됩니다. 따라서 전송 중 문제가 발생한 경우, 부분적으로 기록된 파일이 목적지 프로토콜 파일 서버에 남아 있으며 Managed File Transfer가 이러한 파일을 삭제하거나 편집할 수 없음을 의미합니다. 이 상황에서는 전송이 실패합니다.

관련 태스크

[278 페이지의 『예: UNIX 서버에서 개인 키 신임을 사용하도록 브릿지 에이전트를 구성하는 방법』](#) 이 예는 ProtocolBridgeCredentials.xml 파일을 생성하고 구성하는 방법을 보여줍니다. 이 예는 일반적인 예이며 세부사항은 플랫폼에 따라 다를 수 있지만 원칙은 동일합니다.

[267 페이지의 『ProtocolBridgeProperties.xml 파일을 사용하여 프로토콜 파일 서버의 특성 정의』](#) 에이전트 구성 디렉토리에서 Managed File Transfer가 제공하는 ProtocolBridgeProperties.xml 파일을 사용하여 파일을 주고 받을 하나 이상의 프로토콜 파일 서버에 대한 특성을 정의하십시오.

관련 참조

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

[274 페이지의 『파일 서버의 신임 정보 매핑』](#)

프로토콜 브릿지 에이전트의 기본 신임 정보 매핑 기능을 사용하거나 고유의 사용자 엑시트를 작성하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 매핑합니다. Managed File Transfer는 사용자 신임 매핑을 수행하는 샘플 사용자 엑시트를 제공합니다.

[ProtocolBridgeCredentialExit.java 인터페이스](#)

[샘플 프로토콜 브릿지 신임 사용자 엑시트](#)

[프로토콜 브릿지의 FTPS 서버 지원](#)

ProtocolBridgeProperties.xml 파일을 사용하여 프로토콜 파일 서버의 특성 정의

에이전트 구성 디렉토리에서 Managed File Transfer가 제공하는 ProtocolBridgeProperties.xml 파일을 사용하여 파일을 주고 받을 하나 이상의 프로토콜 파일 서버에 대한 특성을 정의하십시오.

이 태스크 정보

ftcCreateBridgeAgent 명령은 에이전트 구성 디렉토리 *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name*에서 *ProtocolBridgeProperties.xml* 파일을 작성합니다. 명령이 실행될 때 기본값이 지정된 경우 이 명령은 기본 프로토콜 파일 서버의 파일에도 입력 항목을 작성합니다.

메시지 BFGCL0392I에 *ProtocolBridgeProperties.xml* 파일의 위치가 제공됩니다.

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.

Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or
<tns:sftpServer>
element - depending on the protocol used to communicate with the server. When the protocol
bridge agent participates in a managed file transfer it will determine which server to used
based on
the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/
file.txt' would
be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name
is compared
to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element
in this
XML document and the first match is used to determine which protocol server the protocol bridge
agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this
XML
document can specify a default server to use for the managed file transfer. To specify a
default server use the <tns:defaultServer> element as the first element inside the
<tns:serverProperties>
element. The default server will be used whenever the protocol bridge agent participates in
a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element
contains
attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each serverProperties
definition. This
element contains a path to a file containing credentials to be used when connecting to defined
servers.

An example ProtocolBridgeProperties.xml file is as follows:

<?xml version="1.0" encoding="UTF-8"?>
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">

  <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />

  <tns:defaultServer name="myFTPserver" />

  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234"
platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  listFormat="unix" limitedWrite="false">

    <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"
      maxReconnectRetry="2" reconnectWaitPeriod="10"
      maxSessions="60" socketTimeout="30" />

  </tns:ftpServer>

  <tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false" ftpsType="explicit"
  trustStore="C:\FTE\keystores\myFTPSserver\FTPSKeyStore.jks"
  trustStorePassword="password">

    <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>

  </tns:ftpsServer>
```

```

<tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  limitedWrite="false">

  <tns:limits connectionTimeout="60"/>

</tns:sftpServer>
</tns:serverProperties>

```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to the command line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```

-->
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">

  <!-- By default the location of the credentials file is in the home directory of the user
  that started the -->
  <!-- protocol bridge agent. If you wish to specify a different location use the
  credentialsFile element to -->
  <!-- do this. For
  example:
  <!-- <tns:credentialsFile path="/test/
  ProtocolBridgeCredentials.xml"/>
  -->

  <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM" />
  <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
  platform="unix"
    timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
    listFormat="unix" limitedWrite="false" />

  <!-- Define servers here -->
</tns:serverProperties>

```

명령은 BFGCL0532I 메시지를 생성할 수 있습니다.

이 에이전트가 작동하려면 수동으로 추가 신임 정보 파일을 작성해야 합니다. 기본적으로 이 파일은 ProtocolBridgeCredentials.xml 이며 홈에 있습니다. 에이전트를 시작하는 사용자의 디렉토리입니다. 예를 들어, 이 사용자가 에이전트를 시작한 경우 위치는 \$HOME/ProtocolBridgeCredentials.xml 입니다.

신임 정보 파일을 사용하는 경우

1. 작성 방법에 대한 추가 정보는 다음 텍스트를 참조하십시오.
2. 신임 정보 파일은 제한된 권한이 있는 디렉토리에 있어야 합니다. 예를 들어, 기타 사용자의 경우 읽기 액세스 권한이 없어야 합니다.

3. 시작된 에이전트의 사용자 ID에 대해 \$HOME 환경 변수에 신임 정보 파일의 디렉토리 위치를 지정하거나 ProtocolBridgeProperties.xml 파일을 편집하고 다음의 위치를 지정하십시오.

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

기본값이 아닌 프로토콜 서버를 추가할 경우 이 파일을 편집하여 특성을 정의하십시오. 이 예는 추가 FTP 서버를 추가합니다.

참고: 프로토콜 브릿지 에이전트는 파일 잠금을 지원하지 않습니다. 이는 Managed File Transfer가 파일 서버에서 파일 잠금 메커니즘을 지원하지 않기 때문입니다.

프로시저

1. 다음 행을 <tns:serverProperties>의 하위 요소로 파일에 삽입하여 프로토콜 파일 서버를 정의하십시오.

```
<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
platform="windows"
timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>
```

2. 그런 다음, 속성의 값을 변경하십시오.

- name은 프로토콜 파일 서버의 이름입니다.
- host는 프로토콜 파일 서버의 호스트 이름 또는 IP 주소입니다.
- port는 프로토콜 파일 서버의 포트 번호입니다.
- platform은 프로토콜 파일 서버가 실행되는 플랫폼입니다.
- timeZone은 프로토콜 파일 서버를 실행하는 시간대입니다.
- locale은 프로토콜 파일 서버에서 사용되는 언어입니다.
- fileEncoding은 프로토콜 파일 서버의 문자 인코딩입니다.
- listFormat은 프로토콜 파일 서버에서 리턴된 파일 목록 양식입니다.
- limitedWrite는 파일 서버에 기록할 때 임시 파일을 작성한 후 전송이 완료되면 해당 파일의 이름을 바꾸는 기본 모드를 따르는지 여부를 결정합니다. 쓰기 전용으로 구성된 파일 서버의 경우 파일의 최종 이름을 사용하여 파일이 바로 작성됩니다. 이 특성의 값은 true 또는 false입니다. 프로토콜 브릿지 에이전트의 경우 limitedWrite 속성 및 doNotUseTempOutputFile 에이전트 특성이 함께 사용됩니다. 임시 파일을 사용하려면 doNotUseTempOutputFile의 값을 설정하지 않아야 하며 limitedWrite의 값을 False로 설정해야 합니다. 또 다른 설정 결함은 임시 파일이 사용되지 않음을 의미합니다.
- maxListFileNames는 파일 이름의 프로토콜 파일 서버에서 디렉토리 스캔 시 수집되는 최대 이름 수입니다.
- maxListDirectoryLevels는 프로토콜 파일 서버의 디렉토리에서 파일 이름을 스캔할 때 반복할 최대 디렉토리 레벨 수입니다.

필수사항인지 아니면 선택사항인지와 해당 기본값을 포함하여 이러한 속성에 대한 자세한 정보는 [프로토콜 브릿지 특성 파일 형식](#)을 참조하십시오.

관련 참조

[프로토콜 브릿지 특성 파일 형식](#)

[MFT에서 사용하는 정규식](#)

프로토콜 파일 서버 특성 검색: ProtocolBridgePropertiesExit2

프로토콜 파일 서버가 많이 있는 경우

com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2 인터페이스를 구현하여 전송에서 참조되는 프로토콜 파일 서버 특성을 검색할 수 있습니다. ProtocolBridgeProperties.xml 파일 유지보수보다 우선적으로 이 인터페이스를 구현할 수 있습니다.

이 태스크 정보

Managed File Transfer는 프로토콜 파일 서버 특성을 검색하는 샘플 사용자 엑시트를 제공합니다. 자세한 정보는 271 페이지의 『[샘플 사용자 엑시트를 사용하여 프로토콜 파일 서버 특성 검색](#)』의 내용을 참조하십시오.

프로토콜 브릿지 특성을 검색하는 사용자 엑시트는

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` 인터페이스를 구현해야 합니다. 자세한 정보는 `ProtocolBridgePropertiesExit.java` 인터페이스를 참조하십시오.

다른 사용자 엑시트와 비슷한 방식으로 여러 개의 프로토콜 서버 특성 엑시트들을 함께 연결할 수 있습니다. 엑시트는 에이전트 특성 파일에서 `protocolBridgePropertiesExitClasses` 특성을 사용하여 지정된 순서대로 호출됩니다. `initialize` 메소드는 모두 개별적으로 리턴되고 `false` 값을 리턴하는 메소드가 하나 이상일 경우에 이진트가 시작되지 않습니다. 에이전트 이벤트 로그에 오류가 보고됩니다.

모든 엑시트의 `getProtocolServerProperties` 메소드에 대해서는 전체 결과가 하나만 리턴됩니다. 메소드가 특성 오브젝트를 결과 코드로 리턴하는 경우, 이 값이 리턴된 결과이므로 후속 엑시트의 `getProtocolServerProperties` 메소드는 호출되지 않습니다. 메소드가 결과 코드로 널값을 리턴하는 경우, 다음 엑시트의 `getProtocolServerProperties` 메소드가 호출됩니다. 후속 엑시트가 없는 경우, 널 결과가 리턴됩니다. 프로토콜 브릿지 에이전트는 널의 전체 결과 코드를 검색 실패로 간주합니다.

`ProtocolBridgePropertiesExit2.java` 인터페이스를 사용하는 것이 좋습니다. 하지만 `ProtocolBridgePropertiesExit.java` 인터페이스에 대한 정보는 272 페이지의 『[프로토콜 파일 서버 특성 검색: ProtocolBridgePropertiesExit](#)』의 내용을 참조하십시오.

엑시트를 실행하려면 다음 단계를 완료하십시오.

프로시저

1. 프로토콜 서버 특성 사용자 엑시트를 컴파일하십시오.
2. 컴파일된 엑시트 및 패키지 구조가 포함된 Java 아카이브(JAR) 파일을 작성하십시오.
3. 엑시트 클래스가 포함된 JAR 파일을 프로토콜 브릿지 에이전트의 `exits` 디렉토리에 배치하십시오. 이 디렉토리는 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 디렉토리에 있습니다.
4. 프로토콜 브릿지 에이전트의 특성 파일을 편집하여 `protocolBridgePropertiesExitClasses` 특성을 포함하십시오. 프로토콜 브릿지 서버 특성 사용자 엑시트를 구현하는 클래스의 심볼로 구분된 목록을 이 특성의 값으로 지정하십시오. 엑시트 클래스는 이 목록에 지정된 순서대로 호출됩니다. 자세한 정보는 `MFT agent.properties` 파일을 참조하십시오.
5. 선택적으로 `protocolBridgePropertiesConfiguration` 특성을 지정할 수 있습니다. 이 특성에 지정하는 값은 `protocolBridgePropertiesExitClasses`에 지정된 엑시트 클래스의 `initialize()` 메소드에 문자열로 전달됩니다. 자세한 정보는 `MFT agent.properties` 파일을 참조하십시오.

샘플 사용자 엑시트를 사용하여 프로토콜 파일 서버 특성 검색

Managed File Transfer는 프로토콜 파일 서버 특성을 검색하는 샘플 사용자 엑시트를 제공합니다.

이 태스크 정보

프로토콜 브릿지 특성을 검색하는 샘플 사용자 엑시트는 `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 디렉토리 및 [샘플 프로토콜 브릿지 특성 사용자 엑시트](#) 주제에서 제공됩니다.

`SamplePropertiesExit2.java` 엑시트는 프로토콜 서버의 특성을 포함하는 특성 파일을 읽습니다. 특성 파일에서 각 입력 항목의 형식은 다음과 같습니다.

```
serverName=type://host:port
```

특성 파일의 위치는 프로토콜 브릿지 에이전트 특성 `protocolBridgePropertiesConfiguration`에서 알 수 있습니다.

샘플 사용자 엑시트를 실행하려면 다음 단계를 완료하십시오.

프로시저

1. SamplePropertiesExit2.java 파일을 컴파일하십시오.
2. 컴파일된 엑시트 및 패키지 구조가 있는 jar 파일을 작성하십시오.
3. JAR 파일을 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits` 디렉토리에 배치하십시오.
4. 다음 행을 포함하도록 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 파일을 편집하십시오.

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 디렉토리에 프로토콜 브릿지 특성 파일(예: `protocol_bridge_properties.properties`)을 작성하십시오. 다음 형식의 입력 항목을 포함하도록 이 파일을 편집하십시오.

```
serverName=type://host:port
```

6. 다음 행을 포함하도록 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` 파일을 편집하십시오.

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

`protocol_bridge_properties.properties` 파일에 대한 절대 경로를 사용해야 합니다.

7. **fteStartAgent** 명령을 사용하여 프로토콜 브릿지 에이전트를 시작하십시오.

관련 개념

266 페이지의 『프로토콜 브릿지』

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

관련 참조

[ProtocolBridgePropertiesExit.java 인터페이스](#)

[샘플 프로토콜 브릿지 특성 사용자 엑시트](#)

[MFTagent.properties 파일](#)

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

프로토콜 파일 서버 특성 검색: ProtocolBridgePropertiesExit

프로토콜 파일 서버가 많이 있는 경우

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 인터페이스를 구현하여 전송에서 참조되는 프로토콜 파일 서버 특성을 검색할 수 있습니다.

이 태스크 정보

`ProtocolBridgeProperties.xml` 파일 유지보수보다 우선적으로

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 인터페이스를 구현할 수 있습니다. `ProtocolBridgePropertiesExit2.java` 인터페이스를 사용하십시오. `ProtocolBridgePropertiesExit2.java`의 **getCredentialLocation** 메소드는 홈 디렉토리인 `ProtocolBridgeCredentials.xml` 파일의 기본 위치를 사용합니다.

프로토콜 브릿지 특성을 검색하는 사용자 엑시트는 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 인터페이스를 구현해야 합니다.

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *        The values of properties defined for the protocol bridge.
     *        These values can only be read, they cannot be updated by the
     *        implementation.
     * @return {@code true} if the initialization is successful and {@code
     *         false} if unsuccessful. If {@code false} is returned from an exit
     *         the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Obtains a set of properties for the specified protocol server name.
     * <p>
     * The returned {@link Properties} must contain entries with key names
     * corresponding to the constants defined in
     * {@link ProtocolServerPropertyConstants} and in particular must include an
     * entry for all appropriate constants described as required.
     *
     * @param protocolServerName
     *        The name of the protocol server whose properties are to be
     *        returned. If a null or a blank value is specified, properties
     *        for the default protocol server are to be returned.
     * @return The {@link Properties} for the specified protocol server, or null
     *         if the server cannot be found.
     */
    public Properties getProtocolServerProperties(
        final String protocolServerName);

    /**
     * Invoked once when a protocol bridge agent is shut down. It is intended to
     * release any resources that were allocated by the exit.
     *
     * @param bridgeProperties
     *        The values of properties defined for the protocol bridge.
     *        These values can only be read, they cannot be updated by the
     *        implementation.
     */
    public void shutdown(final Map<String, String> bridgeProperties);
}
```

다른 사용자 엑시트와 비슷한 방식으로 여러 개의 프로토콜 서버 특성 엑시트들을 함께 연결할 수 있습니다. 엑시트는 에이전트 특성 파일에서 `protocolBridgePropertiesExitClasses` 특성을 사용하여 지정된 순서대로 호출됩니다. `initialize` 메소드는 모두 개별적으로 리턴되고 `false` 값을 리턴하는 메소드가 하나 이상일 경우에 이진트가 시작되지 않습니다. 에이전트 이벤트 로그에 오류가 보고됩니다.

모든 엑시트의 `getProtocolServerProperties` 메소드에 대해서는 전체 결과가 하나만 리턴됩니다. 메소드가 특성 오브젝트를 결과 코드로 리턴하는 경우, 이 값이 리턴된 결과이므로 후속 엑시트의 `getProtocolServerProperties` 메소드는 호출되지 않습니다. 메소드가 결과 코드로 널값을 리턴하는 경우, 다음

엑시트의 `getProtocolServerProperties` 메소드가 호출됩니다. 후속 엑시트가 없는 경우, 널 결과가 리턴됩니다. 프로토콜 브릿지 에이전트는 널의 전체 결과 코드를 검색 실패로 간주합니다.

프로시저

엑시트를 실행하려면 다음 단계를 완료하십시오.

1. 프로토콜 서버 특성 사용자 엑시트를 컴파일하십시오.
2. 컴파일된 엑시트 및 패키지 구조가 포함된 Java 아카이브(JAR) 파일을 작성하십시오.
3. 엑시트 클래스가 포함된 JAR 파일을 프로토콜 브릿지 에이전트의 `exits` 디렉토리에 배치하십시오.
이 디렉토리는 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 디렉토리에 있습니다.
4. 프로토콜 브릿지 에이전트의 특성 파일을 편집하여 `protocolBridgePropertiesExitClasses` 특성을 포함하십시오.
프로토콜 브릿지 서버 특성 사용자 엑시트를 구현하는 클래스의 심볼로 구분된 목록을 이 특성의 값으로 지정하십시오. 엑시트 클래스는 이 목록에 지정된 순서대로 호출됩니다. 자세한 정보는 [MFT agent.properties](#) 파일을 참조하십시오.
5. 선택적으로 `protocolBridgePropertiesConfiguration` 특성을 지정할 수 있습니다.
이 특성에 지정하는 값은 `protocolBridgePropertiesExitClasses`에 지정된 엑시트 클래스의 `initialize()` 메소드에 문자열로 전달됩니다. 자세한 정보는 [MFT agent.properties](#) 파일을 참조하십시오.

파일 서버의 신임 정보 맵핑

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하거나 고유의 사용자 엑시트를 작성하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑합니다. Managed File Transfer는 사용자 신임 맵핑을 수행하는 샘플 사용자 엑시트를 제공합니다.

관련 개념

[266 페이지의 『프로토콜 브릿지』](#)

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

관련 태스크

[275 페이지의 『ProtocolBridgeCredentials.xml 파일을 사용하여 파일 서버의 신임 정보 맵핑』](#)

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑합니다. Managed File Transfer는 신임 정보를 포함하기 위해 편집할 수 있는 XML 파일을 제공합니다.

[276 페이지의 『엑시트 클래스를 사용하여 파일 서버에 대한 신임 정보 맵핑』](#)

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하지 않으려는 경우에는 고유의 사용자 엑시트를 작성하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑할 수 있습니다. 신임 정보 맵핑 사용자 엑시트를 구성하는 경우 기본 신임 정보 맵핑 기능을 대신합니다.

[278 페이지의 『예: UNIX 서버에서 개인 키 신임을 사용하도록 브릿지 에이전트를 구성하는 방법』](#)

이 예는 `ProtocolBridgeCredentials.xml` 파일을 생성하고 구성하는 방법을 보여줍니다. 이 예는 일반적인 예이며 세부사항은 플랫폼에 따라 다를 수 있지만 원칙은 동일합니다.

관련 참조

[ProtocolBridgeCredentialExit.java 인터페이스](#)

[샘플 프로토콜 브릿지 신임 사용자 엑시트](#)

[MFTagent.properties 파일](#)

ProtocolBridgeCredentials.xml 파일을 사용하여 파일 서버의 신임 정보 맵핑

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑합니다. Managed File Transfer는 신임 정보를 포함하기 위해 편집할 수 있는 XML 파일을 제공합니다.

이 태스크 정보

ProtocolBridgeCredentials.xml 파일은 사용자가 수동으로 작성해야 합니다. 기본적으로 이 파일의 위치는 프로토콜 브릿지 에이전트를 시작한 사용자의 홈 디렉토리입니다. 하지만 에이전트가 액세스할 수 있는 파일 시스템의 임의의 위치에 이를 저장할 수 있습니다. 다른 위치를 지정하려면 ProtocolBridgeProperties.xml 파일에 <credentialsFile> 요소를 추가하십시오. 예:

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

프로토콜 브릿지 에이전트를 사용하려면 먼저 호스트, 사용자 및 신임 정보를 포함하도록 이 파일을 편집하여 신임 정보 맵핑을 설정해야 합니다. 자세한 정보 및 샘플은 [프로토콜 브릿지 신임 정보 파일 형식을 참조하십시오](#).

프로시저

1. name 속성의 값을 ProtocolBridgeProperties.xml 파일에 있는 서버 이름으로 변경하려면 <tns:server name="server name"> 행을 편집하십시오.

패턴 속성을 사용하여 와일드카드 또는 정규식이 포함된 서버 이름을 사용했음을 지정할 수 있습니다. 예:

```
<tns:server name="serverA*" pattern="wildcard">
```

2. 사용자 ID 및 신임 정보를 <tns:server>의 하위 요소로 파일에 삽입하십시오.

다음 요소 중 하나 이상을 파일에 삽입할 수 있습니다.

- 프로토콜 파일 서버가 FTP, FTPS 또는 SFTP 서버인 경우에는 비밀번호를 사용하여 전송을 요청하는 사용자를 인증할 수 있습니다. 파일에 다음 행을 삽입하십시오.

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

그런 다음 속성의 값을 변경하십시오.

- name 는 MFT 전송 요청과 연관된 MQMD 사용자 ID와 일치하는 Java 정규식입니다.
- serverUserId는 로그인 사용자 ID로 프로토콜 파일 서버에 전달되는 값입니다. serverUserId 속성을 지정하지 않을 경우 MFT 전송 요청과 연관된 MQMD 사용자 ID가 대신 사용됩니다.
- serverPassword는 serverUserId와 연관된 비밀번호입니다.

name 속성에는 Java 정규식이 포함될 수 있습니다. 신임 정보 맵핑 프로그램은 이 정규식과 일치하는 MFT 전송 요청의 MQMD 사용자 ID를 찾으려고 시도합니다. 프로토콜 브릿지 에이전트는 파일에 <tns:user> 요소가 존재하는 순서대로 해당 요소의 name 속성에 있는 정규식에 MQMD 사용자 ID를 일치시킵니다. 일치 항목이 발견되면 프로토콜 브릿지 에이전트는 추가로 일치 항목을 검색하지 않습니다. 일치가 발견되면 해당 serverUserId 및 serverPassword 값이 로그인 사용자 ID 및 비밀번호로 프로토콜 파일 서버에 전달됩니다. MQMD 사용자 ID 일치는 대소문자를 구분합니다.

- 프로토콜 파일 서버가 SFTP 서버인 경우에는 공개 키 및 개인 키를 사용하여 전송을 요청하는 사용자를 인증할 수 있습니다. 파일에 다음 행을 삽입하고 속성의 값을 변경하십시오. <tns:user> 요소는 하나 이상의 <tns:privateKey> 요소를 포함할 수 있습니다.

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  hostKey="Host Key">
  <tns:privateKey associationName="association"
    keyPassword="Private key password">
    Private key file text
```

```
</tns:privateKey>
</tns:user>
```

- name 는 MFT 전송 요청과 연관된 MQMD 사용자 ID와 일치하는 Java 정규식입니다.
- serverUserId는 로그인 사용자 ID로 프로토콜 파일 서버에 전달되는 값입니다. serverUserId 속성을 지정하지 않을 경우 MFT 전송 요청과 연관된 MQMD 사용자 ID가 대신 사용됩니다.
- hostKey는 로그인 시 서버에서 리턴되는 예상 키입니다.
- key는 serverUserId의 개인 키입니다.
- keyPassword는 공개 키를 생성하는 데 필요한 키의 비밀번호입니다.
- associationName은 추적 및 로깅 용도로 식별하는 데 사용되는 값입니다.

name 속성에는 Java 정규식이 포함될 수 있습니다. 신임 정보 맵핑 프로그램은 이 정규식과 일치하는 MFT 전송 요청의 MQMD 사용자 ID를 찾으려고 시도합니다. 프로토콜 브릿지 에이전트는 파일에 <tns:user> 요소가 존재하는 순서대로 해당 요소의 name 속성에 있는 정규식에 MQMD 사용자 ID를 일치시킵니다. 일치 항목이 발견되면 프로토콜 브릿지 에이전트는 추가로 일치 항목을 검색하지 않습니다. 일치가 발견되면 해당 serverUserId 및 key 값을 사용하여 프로토콜 파일 서버에서 MFT 사용자를 인증합니다. MQMD 사용자 ID 일치는 대소문자를 구분합니다.

프로토콜 브릿지 에이전트에 개인 키 사용에 대한 자세한 정보는 278 페이지의 『예: UNIX 서버에서 개인 키 신임을 사용하도록 브릿지 에이전트를 구성하는 방법』의 내용을 참조하십시오.

참고:

전송 요청이 명령 큐에 기록될 때, 소스 에이전트 명령 큐가 z/OS 또는 IBM i 시스템에 있으면 MQMD 사용자 ID가 대문자로 변환될 수 있습니다. 따라서 동일한 원래 사용자에 대한 MQMD 사용자 ID는 전송 요청에서 지정된 소스 에이전트에 따라 신임 정보 엑시트에 원래 대소문자로 도달하거나 대문자로 변환되어 도달할 수 있습니다. 기본 신임 정보 맵핑 엑시트는 제공되는 MQMD 사용자 ID에 대해 대소문자 구분 일치를 수행하며, 사용자는 맵핑 파일에서 이를 허용해야 합니다.

관련 참조

- [프로토콜 브릿지 신임 정보 파일 형식](#)
- [프로토콜 브릿지 특성 파일 형식](#)
- [MFT에서 사용하는 정규식](#)

엑시트 클래스를 사용하여 파일 서버에 대한 신임 정보 맵핑

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하지 않으려는 경우에는 고유의 사용자 엑시트를 작성하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑할 수 있습니다. 신임 정보 맵핑 사용자 엑시트를 구성하는 경우 기본 신임 정보 맵핑 기능을 대신합니다.

이 태스크 정보

Managed File Transfer는 사용자 신임 맵핑을 수행하는 샘플 사용자 엑시트를 제공합니다. 자세한 정보는 277 페이지의 『샘플 프로토콜 브릿지 신임 사용자 엑시트 사용』의 내용을 참조하십시오.

프로토콜 브릿지 신임 정보 맵핑을 위한 사용자 엑시트는 다음 인터페이스 중 하나를 구현해야 합니다.

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`를 사용하면 프로토콜 브릿지 에이전트는 하나의 기본 프로토콜 파일 서버와의 사이에 파일을 전송할 수 있습니다.
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`를 사용하면 여러 엔드 포인트와의 사이에서 파일을 전송할 수 있습니다.

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` 인터페이스에는 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`와 동일한 기능이 있으며 확장 기능도 있습니다. 자세한 정보는 `ProtocolBridgeCredentialExit.java` 인터페이스 및 `ProtocolBridgeCredentialExit2.java` 인터페이스를 참조하십시오.

신임 엑시트들을 다른 사용자 엑시트와 비슷한 방식으로 연결할 수 있습니다. 엑시트는 에이전트 특성 파일에서 `protocolBridgeCredentialConfiguration` 특성을 사용하여 지정된 순서대로 호출됩니다. `initialize` 메

소드는 모두 개별적으로 리턴되고 false 값을 리턴하는 메소드가 하나 이상일 경우 에이전트가 시작되지 않습니다. 에이전트 이벤트 로그에 오류가 보고됩니다.

모든 엑시트의 mapMQUserId 메소드에 대해서는 다음과 같이 전체 결과가 하나만 리턴됩니다.

- 메소드가 USER_SUCCESSFULLY_MAPPED 또는 USER_DENIED_ACCESS 값을 결과 코드로 리턴하는 경우, 이 값이 리턴된 결과이므로 후속 엑시트의 mapMQUserId 메소드는 호출되지 않습니다.
- 메소드가 결과 코드로 NO_MAPPING_FOUND의 값을 리턴하는 경우 다음 엑시트의 mqMQUserId 메소드가 호출됩니다.
- 후속 엑시트가 없는 경우 NO_MAPPING_FOUND 결과가 리턴됩니다.
- 브릿지 에이전트는 USER_DENIED_ACCESS 또는 NO_MAPPING_FOUND의 전체 결과 코드를 전송 실패로 간주합니다.

엑시트를 실행하려면 다음 단계를 완료하십시오.

프로시저

1. 프로토콜 브릿지 신임 사용자 엑시트를 컴파일하십시오.
2. 컴파일된 엑시트 및 패키지 구조가 포함된 Java 아카이브(JAR) 파일을 작성하십시오.
3. 엑시트 클래스가 포함된 JAR 파일을 브릿지 에이전트의 exits 디렉토리에 배치하십시오. 이 디렉토리는 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 디렉토리에 있습니다.
4. 프로토콜 브릿지 에이전트의 특성 파일을 편집하여 protocolBridgeCredentialExitClasses 특성을 포함시키십시오. 프로토콜 브릿지 신임 정보 엑시트 루틴을 구현하는 클래스의 심볼로 구분된 목록을 이 특성의 값으로 지정하십시오. 엑시트 클래스는 이 목록에 지정된 순서대로 호출됩니다. 자세한 정보는 [MFT agent.properties](#) 파일을 참조하십시오.
5. 다음을 포함하도록 프로토콜 브릿지 에이전트의 특성 파일을 편집하십시오.

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

에이전트에 대한 agent.properties 파일은 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name` 디렉토리에 있습니다.

agent.properties 파일을 변경하는 경우 변경사항을 적용하려면 에이전트를 재시작해야 합니다.

6. 선택적으로 protocolBridgeCredentialConfiguration 특성을 지정할 수 있습니다. 이 특성에 지정하는 값은 protocolBridgeCredentialExitClasses에 지정된 엑시트 클래스의 initialize() 메소드에 문자열 오브젝트로 전달됩니다. 자세한 정보는 [MFT agent.properties](#) 파일을 참조하십시오.
7. **fteStartAgent** 명령을 사용하여 프로토콜 브릿지 에이전트를 시작하십시오.

샘플 프로토콜 브릿지 신임 사용자 엑시트 사용

Managed File Transfer는 사용자 신임 매핑을 수행하는 샘플 사용자 엑시트를 제공합니다.

이 태스크 정보

샘플 프로토콜 브릿지 신임 정보 엑시트는 `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 디렉토리 및 [샘플 프로토콜 브릿지 신임 정보 사용자 엑시트](#) 주제에서 제공됩니다. 이 샘플은 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` 인터페이스 기반 샘플입니다.

SampleCredentialExit.java 엑시트는 전송 요청과 연관된 MQMD 사용자 ID를 서버 사용자 ID 및 서버 비밀번호에 매핑하는 특성 파일을 읽습니다. 특성 파일의 위치는 프로토콜 브릿지 에이전트 특성 protocolBridgeCredentialConfiguration에서 알 수 있습니다.

샘플 사용자 엑시트를 실행하려면 다음 단계를 완료하십시오.

프로시저

1. SampleCredentialExit.java 파일을 컴파일하십시오.
2. 컴파일된 엑시트 및 패키지 구조가 포함된 JAR 파일을 작성하십시오.
3. JAR 파일을 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits` 디렉토리에 배치하십시오.
4. 다음 행을 포함하도록 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 파일을 편집하십시오.

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. 다음을 포함하도록 프로토콜 브릿지 에이전트의 특성 파일을 편집하십시오.

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

에이전트에 대한 `agent.properties` 파일은 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 디렉토리에 있습니다.

`agent.properties` 파일을 변경하는 경우 변경사항을 적용하려면 에이전트를 재시작해야 합니다.

6. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 디렉토리에서 신임 정보 특성 파일(`credentials.properties`)을 작성하여 다음 형식의 입력 항목을 포함하도록 편집하십시오.

```
mqUserId=serverUserId,serverPassword
```

7. 다음 행을 포함하도록 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 파일을 편집하십시오.

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

`credentials.properties` 파일에 대한 절대 경로를 사용해야 합니다.

8. **fteStartAgent** 명령을 사용하여 프로토콜 브릿지 에이전트를 시작하십시오.

관련 개념

266 페이지의 『프로토콜 브릿지』

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

관련 참조

[ProtocolBridgeCredentialExit.java 인터페이스](#)

[ProtocolBridgeCredentialExit2.java 인터페이스](#)

[샘플 프로토콜 브릿지 신임 사용자 엑시트](#)

[MFTagent.properties 파일](#)

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

예: UNIX 서버에서 개인 키 신임을 사용하도록 브릿지 에이전트를 구성하는 방법

이 예는 ProtocolBridgeCredentials.xml 파일을 생성하고 구성하는 방법을 보여줍니다. 이 예는 일반적인 예이며 세부사항은 플랫폼에 따라 다를 수 있지만 원칙은 동일합니다.

이 태스크 정보

프로시저

1. SFTP 서버를 인증하는 데 사용할 공개 및 개인 키를 생성하십시오.

예를 들어, Linux 호스트 시스템에서 'openssh' 패키지의 일부로 제공되는 **ssh-keygen** 도구를 사용하여 공개/개인 키 쌍을 작성할 수 있습니다.

기본적으로 인수가 없는 **ssh-keygen** 명령은 두 개의 키 파일에 대한 위치 및 비밀번호 문구를 프롬프트합니다. 이는 기본적으로 다음 이름으로 설정됩니다.

```
id_rsa      <-- Private key
id_rsa.pub  <-- Public key
```



주의: RHEL 8과 함께 제공하는 등 최신 버전의 OpenSSH에서 **ssh-keygen** 명령을 사용하는 경우 사용되는 키 형식이 프로토콜 브릿지 에이전트와 호환되지 않으며 SFTP 서버로의 전송 시도가 실패하며 다음 메시지가 표시됩니다.

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed
because of invalid private key.
```

이 최신 버전의 OpenSSH를 사용하여 호환 가능한 개인 키를 작성하려면 **ssh-keygen** 명령에 대한 다음 인수를 사용하여 키 형식을 지정하십시오.

```
ssh-keygen -m PEM
```

id_rsa 개인 키의 콘텐츠에는 다음과 같은 첫 번째 및 마지막 행이 있습니다.

```
-----BEGIN RSA PRIVATE KEY-----
... ..
-----END RSA PRIVATE KEY-----
```

프로토콜 브릿지 에이전트와 호환 가능합니다.

2. id_rsa.pub 파일의 전체 콘텐츠를 SFTP 서버에 있는 SFTP 사용자의 ~/.ssh/authorized_keys 파일에 복사하십시오.

키 인증을 허용하기 위해 이 파일 및 ~/.ssh 디렉토리에 대한 파일 권한이 SFTP 서버에 적합하게 설정되어 있는지 확인하십시오. 일반적으로 권한은 다음과 같습니다.

```
~/.ssh          Mode 700
~/.ssh/authorized_keys  Mode 600
```

3. Managed File Transfer에는 MD5 알고리즘을 사용하여 생성된 호스트 ssh 지문이 필요합니다. SFTP 서버의 호스트 ssh 지문을 확보하려면 다음 명령 중 하나를 실행하십시오.

- Red Hat® Enterprise Linux 버전 6.x 이하 및 Linux Ubuntu 14.04의 경우 다음 명령을 실행하십시오.

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- Red Hat Enterprise Linux 7.x, Linux Ubuntu 16.04 및 SuSE Linux 12.4부터 ssh-keygen 명령은 기본적으로 SHA256 알고리즘을 사용하여 ssh 지문을 생성합니다. MD5 알고리즘을 사용하여 ssh 지문을 생성하려면 다음 명령을 실행하십시오.

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

이 명령의 출력은 다음 예와 유사합니다.

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

ProtocolBridgeCredentials.xml 파일에서 hostKey로 사용할 출력의 16진 부분만 추출하십시오 (280 페이지의 『4』 단계 참조). 따라서 이 예에서는 64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27을(를) 추출합니다.

4. 프로토콜 브릿지 에이전트 시스템에서 ProtocolBridgeCredentials.xml 파일을 편집하십시오. 다음 예에서 이탤릭체로 표시된 값을 사용자의 값으로 바꾸십시오.

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
ProtocolBridgeCredentials.xsd ">

<tns:agent name="Agent_name">

<tns:server name="SFTP_name">

<tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"
hostKey="ssh_host_finger">
<tns:privateKey associationName="name" keyPassword="pass_phrase">
Complete contents of the id_rsa file including the entries
-----BEGIN RSA PRIVATE KEY-----

-----END RSA PRIVATE KEY-----
</tns:privateKey>
</tns:user>

</tns:server>
</tns:agent>
</tns:credentials>
```

설명:

- Agent_name은 프로토콜 브릿지 에이전트의 이름입니다.
- SFTP_host_name 은 ProtocolBridgeProperties.xml 파일에 표시된 SFTP 서버의 이름입니다.
- mq_User_ID는 전송 요청과 연관된 MQMD 사용자 ID입니다.
- SFTP_user_ID는 2단계에서 사용된 SFTP 사용자 ID이며 SFTP 서버에 로그인 사용자 ID로 전달된 값입니다.
- ssh_host_finger는 3단계에서 수집한 지문입니다.
- name은 추적 및 로깅 목적으로 사용하도록 지정할 수 있는 이름입니다.
- pass_phrase는 1단계의 ssh-keygen에 제공한 비밀번호 문구입니다.
- Complete contents of the id_rsa file은 1단계에서 생성된 id_rsa 파일의 전체 콘텐츠입니다. 연결 오류를 방지하려면 다음 두 항목을 모두 포함해야 합니다.

```
-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

<tns:privatekey> 요소를 복제하여 키를 추가할 수 있습니다.

5. 프로토콜 브릿지 에이전트가 아직 시작되지 않은 경우 해당 에이전트를 시작하십시오. 또는 프로토콜 브릿지 에이전트가 ProtocolBridgeCredentials.xml 파일을 주기적으로 폴링하여 변경사항을 채택합니다.

관련 참조

[프로토콜 브릿지 신임 정보 파일 형식](#)

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

[MFTagent.properties 파일](#)

274 페이지의 『파일 서버의 신임 정보 맵핑』

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하거나 고유의 사용자 엑시트를 작성하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑합니다. Managed File Transfer는 사용자 신임 맵핑을 수행하는 샘플 사용자 엑시트를 제공합니다.

FTPS 서버의 프로토콜 브릿지 구성

FTP 서버를 구성할 때와 비슷한 방법으로 FTPS 서버를 구성하십시오. 서버의 브릿지 에이전트를 작성하고 서버 특성을 정의하고 사용자 신임 정보를 맵핑하십시오.

이 태스크 정보

FTPS 서버를 구성하려면 다음 단계를 완료하십시오.

프로시저

1. **fteCreateBridgeAgent** 명령을 사용하여 FTPS 서버의 프로토콜 브릿지 에이전트를 작성하십시오. FTP에 적용 가능한 매개변수는 FTPS에도 적용 가능하지만 FTPS에만 해당되는 세 개의 필수 매개변수도 있습니다.
 - a) **-bt** 매개변수. FTPS를 이 매개변수의 값으로 지정하십시오.
 - b) 신뢰 저장소 파일에 대한 **-bts** 매개변수입니다. 이 명령에서는 서버 인증만 필요하며 사용자가 신뢰 저장소 파일의 위치를 지정해야 한다고 가정합니다.

FTPS 프로토콜의 명확한 양식은 기본적으로 **fteCreateBridgeAgent** 명령에서 구성하지만 프로토콜 브릿지 특성 파일을 변경하여 암시적 양식을 구성할 수 있습니다. 프로토콜 브릿지는 항상 수동 모드에서 FTPS 서버에 연결됩니다.

fteCreateBridgeAgent 명령에 대한 자세한 정보는 [fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)을 참조하십시오.

신뢰 저장소 파일을 작성하는 방법에 대한 지시사항이 필요한 경우 [Oracle keytool 문서](#)에서 keytool에 대한 정보를 참조하십시오.

2. 프로토콜 브릿지 특성 파일 ProtocolBridgeProperties.xml의 <ftpsServer> 요소 내에 FTPS 서버 특성을 정의하십시오. 자세한 정보는 267 페이지의 『ProtocolBridgeProperties.xml 파일을 사용하여 프로토콜 파일 서버의 특성 정의』의 내용을 참조하십시오. 프로토콜 브릿지 특성 파일을 편집하여 클라이언트 인증을 사용할 수도 있습니다. 모든 구성 옵션에 대한 세부사항은 [프로토콜 브릿지 특성 파일 형식](#)을 참조하십시오.
3. 프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하거나 고유의 사용자 엑시트를 작성하여 Managed File Transfer의 사용자 신임 정보를 FTPS 서버의 사용자 신임 정보에 맵핑합니다. 자세한 정보는 274 페이지의 『파일 서버의 신임 정보 맵핑』의 내용을 참조하십시오.
4. 기본적으로 신뢰 저장소 파일은 JKS 양식을 사용하도록 구성됩니다. 양식을 변경하려면 프로토콜 브릿지 특성 파일을 편집하십시오.

예

프로토콜 브릿지 특성 파일에서 FTPS 서버의 입력 항목 예는 다음과 같이 표시됩니다.

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpsserver.mycompany.com" />

  <tns:ftpsServer name="ftpsserver.mycompany.com" host="ftpsserver.mycompany.com" port="990"
  platform="windows"
  timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false"
  trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

다음에 수행할 작업

지원되는 FTPS 프로토콜 부분과 지원되지 않는 부분에 대한 정보는 [프로토콜 브릿지를 통한 FTPS 서버 지원](#)을 참조하십시오.

관련 개념

266 페이지의 『[프로토콜 브릿지](#)』

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

관련 태스크

275 페이지의 『[ProtocolBridgeCredentials.xml](#) 파일을 사용하여 파일 서버의 신임 정보 맵핑』

프로토콜 브릿지 에이전트의 기본 신임 정보 맵핑 기능을 사용하여 Managed File Transfer의 사용자 신임 정보를 파일 서버의 사용자 신임 정보에 맵핑합니다. Managed File Transfer는 신임 정보를 포함하기 위해 편집할 수 있는 XML 파일을 제공합니다.

267 페이지의 『[ProtocolBridgeProperties.xml](#) 파일을 사용하여 프로토콜 파일 서버의 특성 정의』

에이전트 구성 디렉토리에서 Managed File Transfer가 제공하는 ProtocolBridgeProperties.xml 파일을 사용하여 파일을 주고 받을 하나 이상의 프로토콜 파일 서버에 대한 특성을 정의하십시오.

관련 참조

[fteCreateBridgeAgent\(MFT 프로토콜 브릿지 에이전트 작성 및 구성\)](#)

[프로토콜 브릿지 신임 정보 파일 형식](#)

[프로토콜 브릿지 특성 파일 형식](#)

[프로토콜 브릿지의 FTPS 서버 지원](#)

개별 파일 서버에 전송할 파일 수를 제한하는 시나리오와 예제

수정된 프로토콜 브릿지 에이전트가 **maxActiveDestinationTransfers** 및 **failTransferWhenCapacityReached** 속성과 작동하는 방식 및 몇 가지 예제

maxActiveDestinationTransfers 값을 기반으로 프로토콜 브릿지 에이전트의 작동을 보여주는 시나리오

시나리오 1

프로토콜 브릿지 에이전트의 ProtocolBridgeProperties.xml 파일에는 두 개의 파일 서버 정의가 포함되어 있습니다.

- 글로벌 **maxActiveDestinationTransfers** 속성을 설정하지 않았습니다.
- fileServerA 및 FileServerB 모두에서 **maxActiveDestinationTransfers** 속성을 설정하지 않았습니다.
- 프로토콜 브릿지 에이전트 **maxDestinationTransfers** 속성을 기본값으로 설정했습니다.

프로토콜 브릿지 에이전트 **maxDestinationTransfers** 속성을 기본값인 25로 설정한 경우 다음이 수행됩니다.

- 목적지 에이전트가 fileServerA에 대한 두 개의 관리 전송 처리를 시작합니다.
- 두 전송이 모두 완료됩니다.

이 시점에서 클라이언트는 fileServerA가 실패했음을 인식하고 ProtocolBridgeProperties.xml 파일에서 fileServerA에 대해 다음 값을 설정합니다.

```
maxActiveDestinationTransfers = 0
```

```
failTransferWhenCapacityReached =true
```

- 다른 전송은 fileServerA에 도착하고 일부는 fileServerB에 도착합니다.

이전 단계에서 설정한 특성을 기반으로 fileServerA로의 관리 전송은 거부되며 실패로 표시되는 한편, fileServerB에 대한 전송은 기존 표준 플로우에서 처리합니다.

- 잠시 후 클라이언트에서 fileServerA가 다시 실행 중임을 확인하므로 클라이언트가 ProtocolBridgeProperties.xml에서 이전에 추가된 값을 제거하거나 주석 처리합니다. fileServerA에 대한 새 관리 전송이 도착하고 기존 표준 플로우에서 처리합니다.

시나리오 2

- 한 파일 서버에 대해 **maxActiveDestinationTransfers** 속성은 설정하고 **failTransferWhenCapacityReached** 속성은 설정하지 않았습니다.
- 프로토콜 브릿지 에이전트는 파일 서버에 대한 이 수의 관리 전송에 사용하는 대상 에이전트 역할을 합니다.
- **maxActiveDestinationTransfers** 속성의 값은 1만큼 감소합니다.

프로토콜 브릿지 에이전트는 동적으로 구성을 업데이트하고 여전히 활성 상태에 있는 동안 **maxActiveDestinationTransfers**를 새 값으로 설정합니다. 진행 중인 관리 전송은 이 업데이트의 영향을 받지 않으므로 완료할 수 있습니다.

시나리오 3

프로토콜 브릿지 에이전트의 ProtocolBridgeProperties.xml 파일에는 두 개의 파일 서버 정의가 포함되어 있습니다.

- 글로벌 **maxActiveDestinationTransfers** 속성을 설정하지 않았습니다.
- **failTransferWhenCapacityReached** 속성을 설정하지 않았습니다.
- fileServerA에서 **maxActiveDestinationTransfers** 를 1로 설정했습니다.
- fileServerB에서 **maxActiveDestinationTransfers** 속성을 설정하지 않았습니다.

프로토콜 브릿지 에이전트에 **maxDestinationTransfers** 속성이 5로 설정된 경우 다음이 해당됩니다.

- 프로토콜 브릿지 에이전트에서 fileServerA로의 최대 활성 대상 전송 수는 1입니다(대상 에이전트에 5개의 대상 전송 슬롯이 있어도 하나만 fileServerA로의 관리 전송에 사용할 수 있음).

이는 fileServerA가 실패하는 경우 유용합니다. fileServerA가 다시 실행되면 허용되는 대상 전송의 전체 용량을 사용할 수 있도록 **maxActiveDestinationTransfers** 값을 5로 늘릴 수 있습니다.

- 프로토콜 브릿지 에이전트에서 fileServerB로의 최대 활성 대상 전송 수는 5입니다.

이 파일 서버에는 **maxActiveDestinationTransfers**가 설정되지 않으므로 프로토콜 브릿지 에이전트에서 관리 전송을 위한 대상 전송 슬롯 5개를 모두 사용합니다.

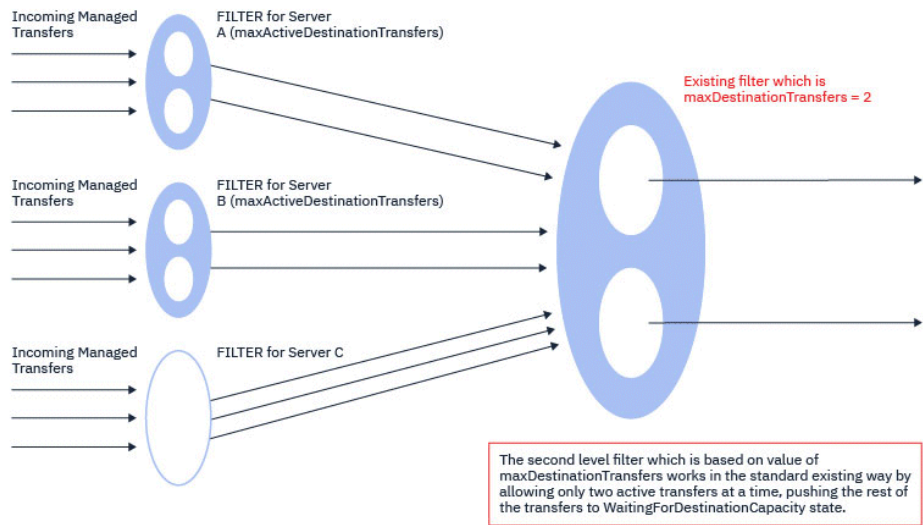
시나리오 4

다음 다이어그램에서:

- agent.properties 파일에서 **maxDestinationTransfers** 속성을 2로 설정했습니다.
- fileServerA에서 **maxActiveDestinationTransfers** 를 2로 설정했습니다.
- fileServerB에서 **maxActiveDestinationTransfers** 속성을 2로 설정했습니다.
- fileServerC에서 **maxActiveDestinationTransfers** 속성을 설정하지 않았습니다.

SCENARIO
 maxDestinationTransfers=2
 fileServerA :
 maxActiveDestinationTransfers=2
 fileServerB :
 maxDestinationTransfers=2
 fileServerC :
 maxDestinationTransfers not set

In both cases A and B, only two managed transfers are let in by the maxActiveDestinationTransfers filter. The rest of the transfers go into the WaitingForDestinationFileServerCapacity state.
 In the case of C, the filter gives everything a passthrough since no value is set.



다이어그램에 표시된 대로 **maxActiveDestinationTransfers**와 **maxDestinationTransfers** 속성은 서로 독립적입니다.

각 서버의 **maxActiveDestinationTransfers** 값이 확인됩니다. 이 값을 기반으로 전송이 진행하도록 허용되거나 **WaitingForDestinationFileServerCapacity** 상태로 푸시됩니다.

그러면 허용된 전송은 **maxDestinationTransfers**에 대한 기존의 표준 검사 플로우를 통과합니다.

시나리오 5

주의: **maxDestinationTransfers** 속성의 값을 기억해야 하므로 **maxActiveDestinationTransfers** 속성의 값을 설정할 때 주의해야 합니다.

이를 수행하지 않으면 다음 텍스트에 설명된 상황이 발생할 수 있습니다.

- 글로벌 **maxActiveDestinationTransfers** 속성의 값을 설정하지 않았습니다.
- `agent.properties` 파일에서 **maxDestinationTransfers=2** 값을 설정했습니다.
- `fileServerA`에서 **maxActiveDestinationTransfers=2** 값을 설정했습니다.
- `fileServerB`에서 **maxActiveDestinationTransfers**의 값을 설정하지 않았습니다.

다음과 같은 순서의 이벤트가 발생한다고 가정하십시오.

- 프로토콜 브릿지 에이전트가 `fileServerA`에 파일을 전송하라는 요청을 받습니다. 프로토콜 브릿지 에이전트에서 현재 아무 작업도 수행하지 않으므로 이 관리 전송 요청을 수락합니다.

이제 전송 슬롯은 다음과 같이 표시됩니다.

- 대상 전송: 1
- `fileServerA`의 대상 전송: 1
- `fileServerB`의 대상 전송: 0
- 이제 프로토콜 브릿지 에이전트가 `fileServerA`와 관련된 관리 전송에 사용되는 대상 에이전트로 작동하기 위한 또 다른 요청을 받습니다. 다시 한 번, 이 요청을 수락하므로 전송 슬롯은 다음과 같이 표시됩니다.
 - 대상 전송: 2
 - `fileServerA`의 대상 전송: 2
 - `fileServerB`의 대상 전송: 0

에이전트의 두 Destination Transfer 슬롯이 현재 사용 중이므로, fileServerA로의 전송 중 하나가 완료될 때까지 에이전트가 추가 관리 전송에 참여할 수 없습니다.

- 잠시 후에 fileServerA가 실패하므로, 두 개의 관리 전송이 복구됩니다. 이 관리 전송에서 사용 중인 Destination transfer 슬롯은 이 기간 동안 사용 중인 상태로 있습니다.
- 다음으로 프로토콜 브릿지 에이전트가 fileServerB에 파일을 전송하라는 요청을 받습니다. Destination Transfers for fileServerB 슬롯에 이 전송을 위한 공간이 있지만, 에이전트의 모든 Destination Transfer 슬롯이 사용 중이므로 나중에 재시도할 수 있도록 전송이 백로그에 배치됩니다.

결과적으로 fileServerB 로의 전송은 fileServerA 로의 전송 중 하나 이상이 완료되고 해당 Destination Transfer 슬롯을 해제할 때까지 차단됩니다.

이 상황이 발생하지 않게 하려면 다음을 수행하십시오.

- 사용 가능한 슬롯이 남도록 파일 서버에서 **maxActiveDestinationTransfers** 의 값을 **maxDestinationTransfers** 값보다 작게 설정하십시오.
- 또는 모든 엔드포인트 서버 간에 **maxActiveDestinationTransfers** 속성의 값을 균등하게 분배하십시오.

maxActiveDestinationTransfers 속성의 값을 기반으로 하는 프로토콜 브릿지 에이전트의 동작

참고: 다음 표에 나열된 모든 오류 사례에서 **maxActiveDestinationTransfers** 속성이 올바르지 않은 값으로 설정된 경우 프로토콜 브릿지 에이전트에서는 이 속성이 설정되지 않았다고 간주합니다.

maxActiveDestinationTransfers	샘플 값	설명
지정되지 않음	지정되지 않음	전송은 평소대로 진행됩니다. *ftp* 엔드포인트의 전송 수에는 한계가 없습니다.
지정됨	0	이 특정 *ftp* 엔드포인트에 허용되는 전송이 없습니다.
음수 값	-1	output0.log에 오류가 로깅됩니다. 값 -1은 음수가 아닌 정수에 올바르지 않습니다. 프로토콜 브릿지 에이전트에서는 속성이 설정되지 않은 것으로 간주합니다.
정수가 아닌 값	abc	output0.log에 오류가 로깅됩니다. 값 abc는 정수에 올바르지 않습니다. 프로토콜 브릿지 에이전트에서는 속성이 설정되지 않은 것으로 간주합니다.
비어 있음	''	maxActiveDestinationTransfers 속성의 '' 값은 음수가 아닌 정수 값에 올바르지 않습니다.
지정됨	5	이 *ftp* 엔드포인트의 경우 언제든지 5개의 활성 전송만 실행할 수 있습니다. failTransferWhenCapacityReached 속성 값을 기반으로 초과 전송은 재시도하거나 거부합니다.

maxActiveDestinationTransfers 및 failTransferWhenCapacityReached 속성 조합을 위한 프로토콜 브릿지 에이전트의 동작

failTransferWhenCapacityReached 값	maxActiveDestinationTransfers 값	결과
아니오	3	이 엔드포인트 서버에는 세 개의 활성 전송이 허용됩니다. 추가 전송은 재시도합니다.
True	3	이 엔드포인트 서버에는 세 개의 활성 전송이 허용됩니다. 추가 전송은 거부하거나 실패로 표시합니다.
지정되지 않음	3	failTransferWhenCapacityReached의 기본값인 false 가 고려됩니다. 결과적으로 이 엔드포인트 서버에 3 개의 활성 전송이 허용됩니다. 추가 전송은 재시도합니다.
Boolean 값이 아닌 값	지정됨	output.log에 오류가 로깅됩니다. failTransferWhenCapacityReached에 지정된 값이 Boolean 값이 아닙니다. failTransferWhenCapacityReached의 기본값이 고려됩니다.

maxDestinationTransfers 및 failTransferWhenCapacityReached 속성 조합을 위한 프로토콜 브릿지 에이전트의 동작

failTransferWhenCapacityReached 값	maxDestinationTransfers 값	결과
True	10	동시 활성 전송 수가 10에 도달하면 프로토콜 브릿지 에이전트에서 11 번째 관리 전송에 실패합니다.
아니오	10	기존 동작. 동시 활성 전송 수가 10에 도달하면 11 번째 관리 전송이 큐에 대기하여 슬롯이 사용 가능하게 될 때까지 기다립니다.
지정되지 않음	10	기존 동작

오류 메시지

기존 메시지:

BFGS0082I

프로토콜 브릿지 에이전트에서 이미 **maxDestinationTransfers** 속성에 정의된 최대 전송 수가 실행 중이면 프로토콜 브릿지 에이전트가 전송을 거부할 때 소스 에이전트의 output0.log 파일에 로깅됩니다.

새로운 메시지:

BFGSS0085I

프로토콜 브릿지 에이전트가 관리 전송을 거부하고 재시도하면 소스 에이전트의 output0.log 파일에 로깅됩니다.

BFGSS0086I

프로토콜 브릿지 에이전트가 관리 전송을 거부하고 재시도하며 대상 항목에 파일 서버 이름이 포함되지 않은 경우 소스 에이전트의 output0.log 파일에 로깅됩니다.

BFGSS0084E

maxActiveDestinationTransfers 속성에 지정된 최대 동시 전송 수를 초과하므로 프로토콜 브릿지 에이전트가 관리 전송을 거부하고 실패함으로 표시하면 탐색기와 audit.xml 파일에 로깅됩니다.

BFGSS0087E

maxActiveDestinationTransfers 속성에 지정된 최대 대상 전송 수를 초과하므로 프로토콜 브릿지 에이전트가 관리 전송을 거부하고 실패함으로 표시하면 탐색기와 audit.xml 파일에 로깅됩니다.

BFGSS0088W

maxActiveDestinationTransfers 속성의 값이 **maxDestinationTransfers** 속성의 값을 초과하면 output0.log에 로깅됩니다.

BFGSS0089I

IBM MQ 9.3.0 이상이 아닌 소스 에이전트에서 작업하는 경우 대상 프로토콜 브릿지 에이전트의 output0.log 파일에 로깅됩니다.

관련 개념

[266 페이지의 『프로토콜 브릿지』](#)

프로토콜 브릿지를 사용하면 Managed File Transfer(MFT) 네트워크에서 로컬 도메인 또는 원격 위치에 있는 MFT 네트워크 외부의 파일 서버에 저장된 파일에 액세스할 수 있습니다. 이 파일 서버는 FTP, FTPS 또는 SFTP 네트워크 프로토콜을 사용할 수 있습니다. 각 파일 서버에는 전용 에이전트가 적어도 하나 이상 필요합니다. 전용 에이전트는 프로토콜 브릿지 에이전트라고 합니다. 브릿지 에이전트는 여러 파일 서버와 서로 작용할 수 있습니다.

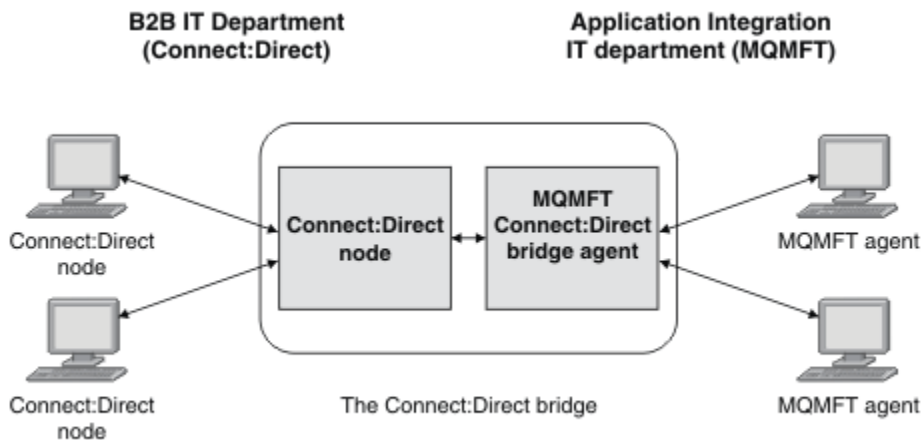
관련 태스크

[267 페이지의 『ProtocolBridgeProperties.xml 파일을 사용하여 프로토콜 파일 서버의 특성 정의』](#)

에이전트 구성 디렉토리에서 Managed File Transfer가 제공하는 ProtocolBridgeProperties.xml 파일을 사용하여 파일을 주고 받을 하나 이상의 프로토콜 파일 서버에 대한 특성을 정의하십시오.

Connect:Direct 브릿지

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.



이 다이어그램은 B2B IT 부서와 애플리케이션 통합 IT 부서라는 두 부서 간의 MFT Connect:Direct 브릿지를 보여줍니다. B2B IT 부서에서는 Connect:Direct를 사용하여 회사의 비즈니스 파트너로부터 파일을 전송합니다. Application Integration IT 부서에서는 IBM MQ를 메시징 인프라로 사용하므로 최근에 Managed File Transfer를 파일 전송 솔루션으로 선택했습니다.



MFT Connect:Direct 브릿지를 사용하여 두 부서는 B2B IT 부서의 Connect:Direct 네트워크와 Application Integration IT 부서의 MFT 네트워크 사이에서 파일을 전송할 수 있습니다. Connect:Direct 브릿지는

Connect:Direct 노드와 통신하는 MFT 에이전트를 포함하는 Managed File Transfer의 컴포넌트입니다. MFT 에이전트는 Connect:Direct 노드를 사용한 전송 전용 에이전트이며 Connect:Direct 브릿지 에이전트로 알려져 있습니다.

Connect:Direct 브릿지는 Managed File Transfer의 서비스 및 에이전트 컴포넌트의 일부로 사용 가능하며 다음 태스크에 사용할 수 있습니다.

1. Managed File Transfer 명령을 사용하여 MFT 에이전트에서 Connect:Direct 노드로 단일 파일 또는 다중 파일 전송 시작
2. Managed File Transfer 명령을 사용하여 Connect:Direct 노드에서 MFT 에이전트로 단일 파일 또는 다중 파일 전송 시작
3. Managed File Transfer 명령을 사용하여 사용자 정의된 Connect:Direct 프로세스를 시작하는 파일 전송 시작
4. Connect:Direct 프로세스를 사용하여 MFT 파일 전송 요청 제출

Connect:Direct 브릿지는 Connect:Direct 노드(부터)만 파일을 전송할 수 있습니다. Connect:Direct 브릿지는 Connect:Direct 프로세스에서 제출하는 전송의 부분으로만 해당 로컬 파일 시스템(부터) 파일을 전송할 수 있습니다.

 Connect:Direct 브릿지를 사용하여 z/OS 시스템의 Connect:Direct 노드에 있는 데이터 세트 간에 전송할 수 있습니다. Managed File Transfer 에이전트만 포함하는 데이터 세트 전송과 비교한 작동에서 일부 차이점이 있습니다. 자세한 정보는  Connect:Direct 노드 간 데이터 세트 전송을 참조하십시오.

지원 플랫폼

Connect:Direct 브릿지는 MFT Connect:Direct 브릿지 에이전트 및 Connect:Direct 노드로 구성됩니다. 에이전트는 Windows 및 Linux for x86-64에서 지원됩니다. 노드는 Windows 용 IBM Sterling Connect:Direct 및 UNIX용 IBM Sterling Connect:Direct에 대해 지원되는 플랫폼에서 지원됩니다. Connect:Direct 브릿지 에이전트를 작성하고 통신할 에이전트에 맞게 Connect:Direct 노드를 구성하는 데 대한 지시사항은 [Connect:Direct 브릿지 구성](#)을 참조하십시오.

Connect:Direct 브릿지는 Windows 용 Connect:Direct 또는 UNIX 또는 z/OS의 경우 Connect:Direct 서비스 설치용 Connect:Direct의 일부로 실행 중인 Connect:Direct 노드 간에 파일을 전송할 수 있습니다. 지원되는 Connect:Direct 버전에 대한 세부사항은 웹 페이지 [IBM MQ의 시스템 요구사항](#)을 참조하십시오.

Connect:Direct 브릿지를 구성하는 에이전트 및 노드에는 동일한 시스템에 있거나 공유 NFS 마운트와 같은 동일한 파일 시스템에 대한 액세스 권한이 있어야 합니다. 이 파일 시스템은 **cdTmpDir** 매개변수에 의해 정의된 디렉토리에 Connect:Direct 브릿지와 관련된 파일 전송 중에 파일을 임시로 저장하는 데 사용됩니다.

Connect:Direct 브릿지 에이전트와 Connect:Direct 브릿지 노드는 동일한 경로 이름을 사용하여 이 디렉토리에 액세스할 수 있어야 합니다. 예를 들어, 에이전트와 노드가 개별 Windows 시스템에 있는 경우 시스템은 동일한 드라이브 문자를 사용하여 공유 파일 시스템을 마운트해야 합니다. 다음 구성에서 에이전트와 노드가 동일한 경로 이름을 사용할 수 있습니다.

- 에이전트와 노드가 Windows 또는 Linux for x86-64를 실행 중인 동일한 시스템에 있습니다.
- 에이전트가 Linux for x86-64에 있으며 노드가 AIX에 있습니다.
- 에이전트가 하나의 Windows 시스템에 있으며 노드가 다른 Windows 시스템에 있습니다.

다음 구성에서는 에이전트와 노드가 동일한 경로 이름을 사용할 수 없습니다.

- 에이전트가 Linux for x86-64에 있으며 노드가 Windows에 있습니다.
- 에이전트가 Windows에 있으며 노드가 UNIX에 있습니다.

Connect:Direct 브릿지의 설치를 계획할 때 이 제한을 고려하십시오.

관련 개념

[295 페이지의 『Connect:Direct 노드 간 전송을 위한 복구 및 재시작』](#)

Managed File Transfer는 전송 중에 IBM Sterling Connect:Direct 노드에 연결하지 못할 수 있습니다(예를 들어, 노드가 사용 불가능하게 되는 경우). Managed File Transfer가 전송 복구를 시도하거나 전송이 실패하여 오류 메시지가 생성됩니다.

[296 페이지의 『파일 전송 요청에서 사용자 정의 Connect:Direct 프로세스 제출』](#)

파일 전송의 부분으로 사용자 정의 Connect:Direct 프로세스를 호출하는 Connect:Direct 브릿지 에이전트를 사용하는 전송에 대한 전송 요청을 제출할 수 있습니다.

300 페이지의 『[Connect:Direct 프로세스를 사용하여 Managed File Transfer 전송 요청 제출](#)』
Connect:Direct 프로세스에서 Connect:Direct 브릿지 에이전트에 전송 요청을 제출할 수 있습니다. Managed File Transfer 는 Connect:Direct 프로세스의 **RUN TASK** 문에서 호출할 수 있는 명령을 제공합니다.

관련 태스크

Connect:Direct 브릿지 구성

289 페이지의 『[Connect:Direct 노드에 파일 전송](#)』

Connect:Direct 브릿지를 사용하여 Managed File Transfer 에이전트에서 Connect:Direct 노드로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 대상 에이전트로 지정하고 목적지 파일을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송 목적지로 지정하십시오.

290 페이지의 『[Connect:Direct 노드로부터 파일 전송](#)』

Connect:Direct 브릿지를 사용하여 Connect:Direct 노드에서 Managed File Transfer Agent 로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 소스 에이전트로 지정하고 소스 스펙을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송의 소스로 지정할 수 있습니다.

292 페이지의 『[Connect:Direct 노드에 다중 파일 전송](#)』

Connect:Direct 브릿지를 사용하여 Managed File Transfer Agent 에서 Connect:Direct 노드로 여러 파일을 전송할 수 있습니다. Connect:Direct 노드를 다중 파일 전송의 목적지로 사용하려면 Connect:Direct 브릿지 에이전트를 목적지 에이전트로 지정하고 목적지 디렉토리를 `connect_direct_node_name:directory_path` 양식으로 지정하십시오.

293 페이지의 『[Transferring multiple files from a Connect:Direct node](#)』

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

294 페이지의 『[와일드카드를 사용하여 Connect:Direct에 다중 파일 전송](#)』

Managed File Transfer 에이전트에서 Connect:Direct 노드로 다중 파일을 전송하려면 Connect:Direct 브릿지를 사용하십시오. **fteCreateTransfer** 명령에 제공하는 소스 스펙에 와일드카드 문자를 사용할 수 있습니다. 와일드카드를 사용하는 모든 Managed File Transfer 전송과 마찬가지로, 파일 경로의 마지막 부분은 와일드카드 문자를 포함할 수 있습니다. 예를 들어, `/abc/def*`는 올바른 파일 경로이고 `/abc*/def`는 올바르지 않습니다.

[Connect:Direct 브릿지 문제점 해결](#)

관련 참조

[fteCreateCDAgent: Connect:Direct 브릿지 에이전트 작성](#)

[Connect:Direct 브릿지 에이전트의 제한사항](#)

Connect:Direct 노드에 파일 전송

Connect:Direct 브릿지를 사용하여 Managed File Transfer 에이전트에서 Connect:Direct 노드로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 대상 에이전트로 지정하고 목적지 파일을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송 목적지로 지정하십시오.

시작하기 전에

파일을 전송하기 전에 Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 구성해야 합니다. 자세한 정보는 [Connect:Direct 브릿지 구성](#)을 참조하십시오.

이 태스크 정보

이 예에서 Connect:Direct 브릿지 에이전트는 CD_BRIDGE입니다. 소스 에이전트는 FTE_AGENT라고 하며 WMQFTE의 모든 버전 중 하나입니다. 목적지 Connect:Direct 노드는 CD_NODE1입니다. 전송될 파일은

FTE_AGENT가 있는 시스템의 파일 경로 /home/helen/file.log에 있습니다. 이러한 파일은 CD_NODE1이 실행 중인 시스템에 있는 파일 경로 /files/data.log에 전송됩니다.

프로시저

1. fteCreateTransfer 명령을 **-df** (목적지 파일) 매개변수의 값 (`connect_direct_node_name:file_path` 양식) 및 Connect:Direct 브릿지 에이전트의 이름으로 지정된 **-da** (목적지 에이전트) 매개변수의 값과 함께 사용하십시오.

참고: `connect_direct_node_name`에 지정된 Connect:Direct 노드는 Connect:Direct 브릿지의 일부로 작동하는 Connect:Direct 노드가 아니라 파일 전송의 대상이 될 노드입니다.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
-df CD_NODE1:/files/data.log /home/helen/file.log
```

자세한 정보는 **fteCreateTransfer**: 새 파일 전송 시작을 참조하십시오.

2. 소스 에이전트 FTE_AGENT는 파일을 Connect:Direct 브릿지 에이전트 CD_BRIDGE에 전송합니다. 파일은 Connect:Direct 브릿지 에이전트가 실행 중인 시스템에서 cdTmpDir 에이전트 특성에 의해 정의된 위치에 임시로 저장됩니다. Connect:Direct 브릿지 에이전트는 파일을 Connect:Direct 노드 CD_NODE1에 전송합니다.

관련 개념

[287 페이지의 『Connect:Direct 브릿지』](#)

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.

관련 태스크

[290 페이지의 『Connect:Direct 노드로부터 파일 전송』](#)

Connect:Direct 브릿지를 사용하여 Connect:Direct 노드에서 Managed File Transfer Agent 로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 소스 에이전트로 지정하고 소스 스펙을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송의 소스로 지정할 수 있습니다.

관련 참조

[MFTagent.properties 파일](#)

Connect:Direct 노드로부터 파일 전송

Connect:Direct 브릿지를 사용하여 Connect:Direct 노드에서 Managed File Transfer Agent 로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 소스 에이전트로 지정하고 소스 스펙을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송의 소스로 지정할 수 있습니다.

시작하기 전에

파일을 전송하기 전에 Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 구성해야 합니다. [Connect:Direct 브릿지 구성](#)을 참조하십시오.

이 태스크 정보

이 예에서 Connect:Direct 브릿지 에이전트는 CD_BRIDGE입니다. 목적지 에이전트는 FTE_AGENT라고 하며 Managed File Transfer의 모든 버전 중 하나입니다. 소스 Connect:Direct 노드는 CD_NODE1입니다. 전송될 파일은 CD_NODE1이 있는 시스템의 파일 경로 /home/brian/in.file에 있습니다. FTE_AGENT가 실행 중인 시스템의 파일 경로 /files/out.file에 파일이 전송됩니다.

프로시저

fteCreateTransfer 명령을 `connect_direct_node_name:file_path` 양식의 소스 스펙 값 및 Connect:Direct 브릿지 에이전트의 이름으로 지정된 **-sa** 매개변수 값과 함께 사용하십시오.

참고: `connect_direct_node_name`에 지정된 Connect:Direct 노드는 Connect:Direct 브릿지의 일부로 작동하는 Connect:Direct 노드가 아니라 파일 전송의 소스가 될 노드입니다. 예를 들면, 다음과 같습니다.

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                 -df /files/out.file CD_NODE1:/home/brian/in.file
```

자세한 정보는 **fteCreateTransfer**: 새 파일 전송 시작을 참조하십시오.

결과

Connect:Direct 브릿지 에이전트 CD_BRIDGE는 Connect:Direct 노드 CD_NODE1으로부터 파일을 요청합니다. Connect:Direct 노드는 파일을 Connect:Direct 브릿지에 송신합니다. 파일이 Connect:Direct 노드로부터 전송 중인 동안 Connect:Direct 브릿지는 cdTmpDir 에이전트 특성에 의해 정의된 위치에 임시로 파일을 저장합니다. Connect:Direct 노드에서 Connect:Direct 브릿지로 파일 전송이 완료되면 Connect:Direct 브릿지는 파일을 목적지 에이전트 FTE_AGENT로 송신하고 임시 위치에서 파일을 삭제합니다.

관련 개념

287 페이지의 『Connect:Direct 브릿지』

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.

관련 참조

MFTagent.properties 파일

z/OS의 Connect:Direct 노드에 데이터 세트 전송

Windows 또는 Linux 시스템에 있는 Connect:Direct 브릿지를 사용하여 z/OS의 Managed File Transfer 에이전트에서 z/OS의 Connect:Direct 노드로 데이터 세트를 전송할 수 있습니다.

시작하기 전에

파일을 전송하기 전에 Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 구성해야 합니다. Connect:Direct 브릿지 구성을 참조하십시오.

이 태스크 정보

이 예에서 **-df** 매개변수는 전송의 목적지를 지정하는 데 사용됩니다. **-df** 매개변수는 버전에 상관없이 전송의 소스 에이전트가 Managed File Transfer에 있을 때 사용할 수 있습니다. 대신 **-ds** 매개변수를 사용할 수 있습니다. 소스 에이전트는 FTE_ZOS1이라고 하며 Managed File Transfer 에이전트입니다. Connect:Direct 브릿지 에이전트는 CD_BRIDGE라고 하며 Linux 시스템에 있습니다. 목적지 Connect:Direct 노드는 CD_ZOS2라고 합니다. 소스 에이전트와 목적지 Connect:Direct 노드 모두 z/OS 시스템에 있습니다. 전송될 데이터 세트는 FTE_ZOS1이 있는 시스템의 //FTEUSER.SOURCE.LIB에 있습니다. 데이터 세트는 CD_ZOS2가 있는 시스템의 데이터 세트 //CDUSER.DEST.LIB로 전송됩니다.

프로시저

1. fteCreateTransfer 명령을 `connect_direct_node_name:data_set_name;attributes` 양식의 **-df** 매개변수 값 및 Connect:Direct 브릿지 에이전트의 이름으로 지정된 **-da** (목적지 에이전트) 매개변수 값과 함께 사용하십시오.

`connect_direct_node_name`에 지정된 Connect:Direct 노드는 Connect:Direct 브릿지의 일부로 작동하는 Connect:Direct 노드가 아니라 데이터 세트가 전송될 노드입니다.

`data_set_name`에 의해 지정된 데이터 세트 이름은 상대적이 아니라 절대적이어야 합니다. Connect:Direct에서는 데이터 세트 이름에 사용자 이름을 접두부로 추가하지 않습니다.

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                 -da CD_BRIDGE -dm QM_BRIDGE
                 -df CD_ZOS2:/'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80)'
                 //'FTEUSER.SOURCE.LIB'
```

자세한 정보는 **fteCreateTransfer**: 새 파일 전송 시작을 참조하십시오.

2. 소스 에이전트 FTE_ZOS1은 데이터 세트의 데이터를 Connect:Direct 브릿지 에이전트 CD_BRIDGE로 전송합니다. 데이터는 Connect:Direct 브릿지 에이전트가 실행되는 시스템에서 cdTmpDir 에이전트 특성이 정의하는 위치에 플랫폼 파일로 임시 저장됩니다. Connect:Direct 브릿지 에이전트는 Connect:Direct 노드 CD_ZOS2로 데이터를 전송합니다. 전송이 완료되면 Connect:Direct 브릿지 에이전트가 실행되는 시스템에서 플랫폼 파일이 삭제됩니다.

관련 개념

287 페이지의 『[Connect:Direct 브릿지](#)』

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.

관련 태스크

[Connect:Direct 노드 간 데이터 세트 전송](#)

관련 참조

[MFT를 사용해서는 안 되는 BPXWDYN 특성](#)

Connect:Direct 노드에 다중 파일 전송

Connect:Direct 브릿지를 사용하여 Managed File Transfer Agent 에서 Connect:Direct 노드로 여러 파일을 전송할 수 있습니다. Connect:Direct 노드를 다중 파일 전송의 목적지로 사용하려면 Connect:Direct 브릿지 에이전트를 목적지 에이전트로 지정하고 목적지 디렉토리를 `connect_direct_node_name:directory_path` 양식으로 지정하십시오.

시작하기 전에

파일을 전송하기 전에 Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 구성해야 합니다. [Connect:Direct 브릿지 구성](#)을 참조하십시오.

이 태스크 정보

이 예에서 소스 에이전트는 FTE_AGENT입니다. Connect:Direct 브릿지 에이전트는 CD_BRIDGE입니다. 목적지 Connect:Direct 노드는 CD_NODE1입니다. 전송될 파일은 FTE_AGENT가 있는 시스템의 /home/jack/data.log, /logs/log1.txt 및 /results/latest입니다. 이러한 파일은 CD_NODE1이 실행 중인 시스템에 있는 /in/files 디렉토리에 전송됩니다.

프로시저

`connect_direct_node_name:directory_path` 형식으로 된 **-dd**(목적지 디렉토리) 매개변수의 값이 포함된 `fteCreateTransfer` 명령을 사용하십시오. **-da**(목적지 에이전트) 매개변수의 값을 Connect:Direct 브릿지 에이전트의 이름으로 지정하십시오.

참고: `connect_direct_node_name`에 지정된 Connect:Direct 노드는 Connect:Direct 브릿지의 일부로 작동하는 Connect:Direct 노드가 아니라 파일 전송의 대상이 될 노드입니다.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

자세한 정보는 [fteCreateTransfer: 새 파일 전송 시작](#)을 참조하십시오.

결과

소스 에이전트 FTE_AGENT는 첫 번째 파일을 Connect:Direct 브릿지 에이전트 CD_BRIDGE에 전송합니다. Connect:Direct 브릿지 에이전트는 cdTmpDir 특성에 정의된 위치에 파일을 임시로 저장합니다. 파일이 소스 에이전트에서 Connect:Direct 브릿지로 완전하게 전송된 경우 Connect:Direct 브릿지 에이전트는 해당 파일을 cdNode 에이전트 특성에 의해 정의된 Connect:Direct 노드에 송신합니다. 이 노드는 해당 파일을 목적지 Connect:Direct 노드 CD_NODE1에 송신합니다. Connect:Direct 브릿지 에이전트는 두 Connect:Direct 노드 간 전송이 완료되면 임시 위치에서 파일을 삭제합니다. 이 프로세스는 지정된 각 소스 파일에 대해 반복됩니다.

관련 개념

287 페이지의 『[Connect:Direct 브릿지](#)』

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.

관련 태스크

[289 페이지의 『Connect:Direct 노드에 파일 전송』](#)

Connect:Direct 브릿지를 사용하여 Managed File Transfer 에이전트에서 Connect:Direct 노드로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 대상 에이전트로 지정하고 목적지 파일을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송 목적지로 지정하십시오.

[294 페이지의 『와일드카드를 사용하여 Connect:Direct에 다중 파일 전송』](#)

Managed File Transfer 에이전트에서 Connect:Direct 노드로 다중 파일을 전송하려면 Connect:Direct 브릿지를 사용하십시오. `fteCreateTransfer` 명령에 제공하는 소스 스펙에 와일드카드 문자를 사용할 수 있습니다. 와일드카드를 사용하는 모든 Managed File Transfer 전송과 마찬가지로, 파일 경로의 마지막 부분은 와일드카드 문자를 포함할 수 있습니다. 예를 들어, `/abc/def*`는 올바른 파일 경로이고 `/abc*/def`는 올바르지 않습니다.

[290 페이지의 『Connect:Direct 노드로부터 파일 전송』](#)

Connect:Direct 브릿지를 사용하여 Connect:Direct 노드에서 Managed File Transfer Agent 로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 소스 에이전트로 지정하고 소스 스펙을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송의 소스로 지정할 수 있습니다.

[293 페이지의 『Transferring multiple files from a Connect:Direct node』](#)

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

관련 참조

[MFTagent.properties](#) 파일

Transferring multiple files from a Connect:Direct node

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Before you begin

Before transferring a file, you must configure the Connect:Direct bridge, which is a component of Managed File Transfer. See [Configuring the Connect:Direct bridge](#).

About this task

In this example, the Connect:Direct bridge agent is called CD_BRIDGE. The destination agent is called FTE_Z, and is running on a z/OS system. The source Connect:Direct node is called CD_NODE1. The files to be transferred are located at the file paths `/in/file1`, `/in/file2`, and `/in/file3` on the system where CD_NODE1 is located. The files are transferred to the partitioned data set `//OBJECT.LIB` on the system where FTE_Z is running.

Procedure

Use the `fteCreateTransfer` command with the values for the source specifications in the form `connect_direct_node_name:file_path` and the value of the `-sa` parameter specified as the name of the Connect:Direct bridge agent.

Note: The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the files to be transferred from, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp //'OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

For more information, see [fteCreateTransfer: start a new file transfer](#).

Results

The Connect:Direct bridge agent CD_BRIDGE requests the first file from the Connect:Direct node CD_NODE1. The Connect:Direct node sends the file to the Connect:Direct bridge. While the file is being transferred from the Connect:Direct node, the Connect:Direct bridge stores the file temporarily in the location defined by the `cdTmpDir` agent property. When the file has finished transferring from the Connect:Direct node to the Connect:Direct bridge, the Connect:Direct bridge sends the file to the destination agent FTE_Z and then deletes the file from the temporary location. This process is repeated for each specified source file.

Related concepts

[“Connect:Direct 브릿지” on page 287](#)

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.

Related reference

[The MFT agent.properties file](#)

와일드카드를 사용하여 Connect:Direct에 다중 파일 전송

Managed File Transfer 에이전트에서 Connect:Direct 노드로 다중 파일을 전송하려면 Connect:Direct 브릿지를 사용하십시오. `fteCreateTransfer` 명령에 제공하는 소스 스펙에 와일드카드 문자를 사용할 수 있습니다. 와일드카드를 사용하는 모든 Managed File Transfer 전송과 마찬가지로, 파일 경로의 마지막 부분은 와일드카드 문자를 포함할 수 있습니다. 예를 들어, `/abc/def*`는 올바른 파일 경로이고 `/abc*/def`는 올바르지 않습니다.

시작하기 전에

파일을 전송하기 전에 Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 구성해야 합니다. 자세한 정보는 [Connect:Direct 브릿지 구성](#)을 참조하십시오.

이 태스크 정보

이 예에서 소스 에이전트는 FTE_AGENT이고 Connect:Direct 브릿지 에이전트는 CD_BRIDGE입니다. 목적지 Connect:Direct 노드는 CD_NODE1입니다. 전송될 파일은 FTE_AGENT가 있는 시스템의 `/reports` 디렉토리에 있습니다. 이름이 `report`로 시작하고 다음에 두 개의 문자와 접미부 `.log`가 붙는 파일만 전송됩니다. 예를 들어, `/reports/report01.log` 파일은 전송되지만 `/reports/report1.log` 파일은 전송되지 않습니다. 이러한 파일은 CD_NODE1이 실행 중인 시스템에 있는 `/home/fred` 디렉토리에 전송됩니다.

프로시저

1. `connect_direct_node_name:directory_path` 형식으로 된 `-dd`(목적지 디렉토리) 매개변수의 값이 포함된 `fteCreateTransfer` 명령을 사용하십시오. `-da`(목적지 에이전트) 매개변수의 경우 Connect:Direct 브릿지 에이전트를 지정하십시오.

참고: `connect_direct_node_name`에 지정된 Connect:Direct 노드는 Connect:Direct 브릿지의 일부로 작동하는 Connect:Direct 노드가 아니라 파일 전송의 대상이 될 노드입니다.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/home/fred "/reports/report??.log"
```

자세한 정보는 [fteCreateTransfer: 새 파일 전송 시작](#)을 참조하십시오.

2. 소스 에이전트 FTE_AGENT는 /reports/report??.log 패턴과 일치하는 첫 번째 파일을 Connect:Direct 브릿지 에이전트 CD_BRIDGE에 전송합니다. Connect:Direct 브릿지 에이전트는 cdTmpDir 특성에 정의된 위치에 파일을 임시로 저장합니다. 파일이 소스 에이전트에서 Connect:Direct 브릿지로 완전하게 전송된 경우 Connect:Direct 브릿지 에이전트는 해당 파일을 cdNode 에이전트 특성에 의해 정의된 Connect:Direct 노드에 송신합니다. 이 노드는 해당 파일을 목적지 Connect:Direct 노드 CD_NODE1에 송신합니다. Connect:Direct 브릿지 에이전트는 두 Connect:Direct 노드 간 전송이 완료되면 임시 위치에서 파일을 삭제합니다. 이 프로세스는 와일드카드 패턴 /reports/report??.log와 일치하는 각 소스 파일에 대해 반복됩니다.

참고: /reports/report??.log 패턴과 일치하는 파일의 목록은 소스 에이전트 FTE_AGENT가 있는 시스템의 운영 체제에 따라 다릅니다.

- 소스 에이전트가 Windows 운영 체제를 사용하는 시스템에 있는 경우 패턴 일치에서는 대소문자를 구분하지 않습니다. 이 패턴은 /reports 디렉토리에서 대소문자 구분 없이 두 개의 문자와 접미부 .log가 뒤따르는 report 양식의 파일 이름을 가진 모든 파일과 일치합니다. 예를 들어, Report99.Log는 일치 항목입니다.
- 소스 에이전트가 Linux 또는 UNIX 운영 체제를 사용하는 시스템에 있는 경우 패턴 일치에서는 대소문자를 구분하지 않습니다. 이 패턴은 /reports 디렉토리에서 두 개의 문자와 접미부 .log가 뒤따르는 report 양식의 파일 이름을 가진 파일과만 일치합니다. 예를 들어, reportAB.log는 일치 항목이지만 reportAB.LOG 및 Report99.Log는 일치 항목이 아닙니다.

관련 개념

[287 페이지의 『Connect:Direct 브릿지』](#)

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct 간에 파일을 전송하십시오.

관련 태스크

[MFT와 함께 와일드카드 문자 사용](#)

[289 페이지의 『Connect:Direct 노드에 파일 전송』](#)

Connect:Direct 브릿지를 사용하여 Managed File Transfer 에이전트에서 Connect:Direct 노드로 파일을 전송할 수 있습니다. Connect:Direct 브릿지 에이전트를 대상 에이전트로 지정하고 목적지 파일을 `connect_direct_node_name:file_path` 양식으로 지정하여 Connect:Direct 노드를 전송 목적지로 지정하십시오.

[292 페이지의 『Connect:Direct 노드에 다중 파일 전송』](#)

Connect:Direct 브릿지를 사용하여 Managed File Transfer Agent 에서 Connect:Direct 노드로 여러 파일을 전송할 수 있습니다. Connect:Direct 노드를 다중 파일 전송의 목적지로 사용하려면 Connect:Direct 브릿지 에이전트를 목적지 에이전트로 지정하고 목적지 디렉토리를 `connect_direct_node_name:directory_path` 양식으로 지정하십시오.

[293 페이지의 『Transferring multiple files from a Connect:Direct node』](#)

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

관련 참조

[MFTagent.properties](#) 파일

Connect:Direct 노드 간 전송을 위한 복구 및 재시작

Managed File Transfer는 전송 중에 IBM Sterling Connect:Direct 노드에 연결하지 못할 수 있습니다(예를 들어, 노드가 사용 불가능하게 되는 경우). Managed File Transfer가 전송 복구를 시도하거나 전송이 실패하여 오류 메시지가 생성됩니다.

Connect:Direct 노드를 사용할 수 없게 되는 경우

네트워크 중단 또는 정전 등으로 인해 Connect:Direct 노드가 사용 불가능하게 되는 경우, Managed File Transfer가 다음 방법으로 파일 전송을 복구합니다.

- Managed File Transfer 가 이전에 이 전송 요청의 일부로 Connect:Direct 노드에 연결되지 않은 경우 **cdMaxConnectionRetries** 및 **recoverableTransferRetryInterval properties**의 값으로 판별된 기간 동안 전송이 다시 시도됩니다. 이러한 특성은 Connect:Direct 브릿지 에이전트의 `agent.properties` 파일에 지정됩니다. 실패한 시도 수가 **cdMaxConnectionRetries property**의 값에 도달하면 전송이 실패하고 오류 메시지가 생성됩니다. 기본적으로 60초 간격으로 횟수 제한 없이 전송이 시도됩니다.
- Managed File Transfer가 이전에 이 전송 요청의 일부로 Connect:Direct 노드에 연결된 경우, **cdMaxPartialWorkConnectionRetries** 및 **recoverableTransferRetryInterval** 특성의 값으로 판별된 시간 동안 전송이 다시 시도됩니다. 실패한 시도 수가 **cdMaxPartialWorkConnectionRetries** 특성의 값에 도달하면 전송이 실패하고 오류 메시지가 생성됩니다. 기본적으로 60초 간격으로 횟수 제한 없이 전송이 시도됩니다.
- 특정 유형의 Connect:Direct 노드 실패의 경우(예: 강제로 중지 중인 노드) 노드가 복구되면 Connect:Direct 프로세스가 Held Due to Error (HE) 상태로 전환됩니다. 노드가 복구된 후 Managed File Transfer 는 파일 전송과 관련된 모든 Connect:Direct 프로세스를 자동으로 재개하며 상태는 HE입니다.
- 전송이 실패하면 전송과 관련된 모든 임시 파일은 Connect:Direct 브릿지를 호스트하는 시스템에서 삭제됩니다. 이러한 임시 파일의 위치는 **cdTmpDir** 특성으로 정의됩니다.
- 전송이 Managed File Transfer에서 Connect:Direct로 진행되고 삭제의 소스 처리가 지정된 경우에는 전송이 실패해도 소스 파일이 삭제되지 않습니다.

Connect:Direct 노드 사용자 신임 정보가 올바르지 않음

노드에서 사용자 신임 정보가 거부되어 Managed File Transfer를 Connect:Direct 노드에 연결하는 데 실패하면, 전송이 실패하고 오류 메시지가 생성됩니다. 이 상황에서는 Connect:Direct 노드에 대한 올바른 사용자 신임 정보를 제공했는지 확인하십시오. 자세한 정보는 [Connect:Direct의 신임 정보 맵핑을 참조하십시오](#).

Connect:Direct 브릿지 에이전트를 사용할 수 없게 되는 경우

Connect:Direct 브릿지 에이전트를 사용할 수 없게 되는 경우, 진행 중인 파일 전송은 표준 Managed File Transfer 전송과 동일한 방식으로 복구됩니다. 자세한 정보는 302 페이지의 『MFT 복구 및 재시작』의 내용을 참조하십시오.

관련 개념

[287 페이지의 『Connect:Direct 브릿지』](#)

기존 IBM Sterling Connect:Direct 네트워크에서 또는 해당 네트워크로 파일을 전송할 수 있습니다. Managed File Transfer의 컴포넌트인 Connect:Direct 브릿지를 사용하여 MFT 및 IBM Sterling Connect:Direct간에 파일을 전송하십시오.

[302 페이지의 『MFT 복구 및 재시작』](#)

에이전트 또는 큐 관리자가 사용 불가능한 경우(예: 전원 문제 또는 네트워크 장애), Managed File Transfer는 다음과 같이 복구합니다.

관련 태스크

[Connect:Direct 브릿지 구성](#)

관련 참조

[MFTagent.properties 파일](#)

파일 전송 요청에서 사용자 정의 Connect:Direct 프로세스 제출

파일 전송의 부분으로 사용자 정의 Connect:Direct 프로세스를 호출하는 Connect:Direct 브릿지 에이전트를 사용하는 전송에 대한 전송 요청을 제출할 수 있습니다.

기본적으로, Connect:Direct 브릿지를 검토하는 전송에 대한 파일 전송 요청을 제출하는 경우 Connect:Direct 브릿지 에이전트는 원격 Connect:Direct 노드(부터)의 파일 전송에 사용되는 Connect:Direct 프로세스를 생성합니다.

그러나 `ConnectDirectProcessDefinition.xml` 파일을 사용하여 사용자 정의 Connect:Direct 프로세스를 대신 호출하도록 Connect:Direct 브릿지 에이전트를 구성할 수 있습니다.

ConnectDirectProcessDefinition.xml 파일

fteCreateCDAgent 명령은 에이전트 구성 디렉토리 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`에서 `ConnectDirectProcessDefinitions.xml` 파일을 작성합니다. Connect:Direct 브릿지 에이전트에서 사용자 정의 Connect:Direct 프로세스를 호출하려면 이 파일을 편집하여 프로세스 정의를 설정해야 합니다.

파일은 전송의 부분으로 호출되는 하나 이상의 Connect:Direct 프로세스 위치를 포함하는 하나 이상의 프로세스 세트를 정의합니다. 각 프로세스 세트는 다수의 조건을 포함합니다. 전송이 프로세스 세트의 모든 조건을 만족하는 경우, 프로세스 세트는 전송에서 호출되는 Connect:Direct 프로세스를 지정하는 데 사용됩니다. 자세한 정보는 297 페이지의 『[ConnectDirectProcessDefinition.xml 파일을 사용하여 시작하도록 Connect:Direct 프로세스 지정](#)』의 내용을 참조하십시오.

고유 기호 변수

Managed File Transfer에서 정의한 고유 기호 변수를 사용하여 값을 사용자 정의 Connect:Direct 프로세스로 대체할 수 있습니다. Connect:Direct 이름 지정 규칙을 따르기 위해 Managed File Transfer에서 사용하는 모든 고유 기호 변수는 다섯 개의 대문자 영숫자 문자가 뒤에 오는 %FTE 형식으로 되어 있습니다.

Connect:Direct 노트에서 Connect:Direct 브릿지 시스템으로 파일을 전송하는 프로세스를 작성할 때 Connect:Direct 프로세스에서 TO FILE의 값으로 고유 변수 %FTEFILE을 사용해야 합니다. Connect:Direct 브릿지 시스템에서 Connect:Direct 노트로 파일을 전송하는 프로세스를 작성할 때 Connect:Direct 프로세스에서 FROM FILE의 값으로 고유 변수 %FTEFFILE을 사용해야 합니다. 이러한 변수에는 Connect:Direct 브릿지 에이전트가 Managed File Transfer 네트워크와 전송을 주고 받는 데 사용하는 임시 파일 경로가 포함됩니다.

고유 기호 변수에 대한 자세한 정보는 Connect:Direct 제품 문서를 참조하십시오.

샘플 Connect:Direct 프로세스

Managed File Transfer는 샘플 Connect:Direct 프로세스를 제공합니다. 이러한 샘플은 `MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates` 디렉토리에 있습니다.

관련 태스크

297 페이지의 『[ConnectDirectProcessDefinition.xml 파일을 사용하여 시작하도록 Connect:Direct 프로세스 지정](#)』

Managed File Transfer 전송의 일부로 시작할 Connect:Direct 프로세스를 지정하십시오. Managed File Transfer는 프로세스 정의를 지정하기 위해 편집할 수 있는 XML 파일을 제공합니다.

298 페이지의 『[Managed File Transfer 에서 호출되는 Connect:Direct 프로세스에서 고유 기호 변수 사용](#)』
Managed File Transfer 전송에서 사용자 정의 Connect:Direct 프로세스를 호출하고 프로세스 정의에서 고유 기호 변수를 사용하여 전송에서 Connect:Direct 프로세스로 정보를 전달할 수 있습니다.

관련 참조

[Connect:Direct 프로세스 정의 파일 양식](#)

[사용자 정의 Connect:Direct 프로세스와 함께 사용할 대체 변수](#)

ConnectDirectProcessDefinition.xml 파일을 사용하여 시작하도록 Connect:Direct 프로세스 지정

Managed File Transfer 전송의 일부로 시작할 Connect:Direct 프로세스를 지정하십시오. Managed File Transfer는 프로세스 정의를 지정하기 위해 편집할 수 있는 XML 파일을 제공합니다.

이 태스크 정보

fteCreateCDAgent 명령은 에이전트 구성 디렉토리 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`에서 `ConnectDirectProcessDefinitions.xml` 파일을 작성합니다. Connect:Direct 브릿지 에이전트에서 사용자 정의 Connect:Direct 프로세스를 호출하려면 이 파일을 편집하여 프로세스 정의를 설정해야 합니다.

Connect:Direct 브릿지를 통해 전송의 부분으로 호출하도록 지정하려는 각 프로세스는 다음 단계를 수행하십시오.

프로시저

1. Connect:Direct 브릿지 에이전트가 전송의 부분으로 호출하려는 Connect:Direct 프로세스를 정의하고 파일에 프로세스 템플릿을 저장하십시오.
2. 텍스트 편집기에서 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` 파일을 여십시오.
3. <processSet> 요소를 작성하십시오.
4. <processSet> 요소 내에서 <condition> 요소를 작성하십시오.
5. <condition> 요소 내에서, 1단계에서 정의한 Connect:Direct 프로세스를 호출하기 위해 전송 요청이 일치해야 하는 조건을 정의하는 하나 이상의 요소를 작성하십시오. 이러한 요소는 <match> 요소 또는 <defined> 요소일 수 있습니다.
 - 변수의 값이 패턴과 일치해야 하도록 지정하려면 <match> 요소를 사용하십시오. 다음 속성을 사용하여 <match> 요소를 작성하십시오.
 - `variable` - 값과 비교하는 변수의 이름입니다. 변수는 고유 기호입니다. 자세한 정보는 [사용자 정의 Connect:Direct 프로세스에 사용할 대체 변수를 참조하십시오](#).
 - `value` - 지정된 변수의 값과 비교할 패턴입니다.
 - 선택사항: `pattern` - `value` 속성의 값에서 사용하는 패턴의 유형입니다. 이 패턴 유형은 wildcard 또는 regex일 수 있습니다. 이 속성은 선택사항이며 기본값은 wildcard입니다.
 - 변수에 값이 정의되어 있도록 지정하려면 <defined> 요소를 사용하십시오. 다음 속성을 사용하여 <defined> 요소를 작성하십시오.
 - `variable` - 정의된 값이 있어야 하는 변수의 이름입니다. 변수는 고유 기호입니다. 자세한 정보는 [사용자 정의 Connect:Direct 프로세스에 사용할 대체 변수를 참조하십시오](#).

<condition> 지정된 조건은 논리적 AND와 결합됩니다. Connect:Direct 브릿지 에이전트가 이 <processSet> 요소에 지정된 프로세스를 호출하려면 모든 조건이 충족되어야 합니다. <condition> 요소를 지정하지 않으면 프로세스 세트는 모든 전송과 일치합니다.
6. <processSet> 요소 내에서 <process> 요소를 작성하십시오.
7. <process> 요소 내에서 <transfer> 요소를 작성하십시오.

전송 요소는 Connect:Direct 브릿지 에이전트가 전송의 부분으로 호출하는 Connect:Direct 프로세스를 지정합니다. 다음 속성을 사용하여 <transfer> 요소를 작성하십시오.

 - `process` - 1단계에서 정의한 Connect:Direct 프로세스의 위치입니다. 이 파일의 위치는 절대 경로로 지정되거나 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 디렉토리에 상대적으로 지정됩니다.

결과

조건 일치 검색 시 Connect:Direct 브릿지 에이전트는 파일의 처음부터 파일의 끝까지 검색합니다. 발견되는 첫 번째 일치기가 사용됩니다.

관련 태스크

[Connect:Direct 브릿지 구성](#)

관련 참조

[Connect:Direct 프로세스 정의 파일 양식](#)

[fteCreateCDAgent: Connect:Direct 브릿지 에이전트 작성](#)

Managed File Transfer 에서 호출되는 **Connect:Direct** 프로세스에서 고유 기호 변수 사용

Managed File Transfer 전송에서 사용자 정의 Connect:Direct 프로세스를 호출하고 프로세스 정의에서 고유 기호 변수를 사용하여 전송에서 Connect:Direct 프로세스로 정보를 전달할 수 있습니다.

이 태스크 정보

이 예에서는 고유 기호 변수를 사용하여 정보를 Managed File Transfer 전송에서 사용자 정의 Connect:Direct 프로세스로 전달합니다. Managed File Transfer에서 사용하는 고유 기호 변수에 대한 자세한 정보는 [사용자 정의 Connect:Direct 프로세스에 사용할 대체 변수를 참조하십시오](#).

이 예에서 파일은 Managed File Transfer Agent에서 Connect:Direct 브릿지 노드로 전송됩니다. Managed File Transfer가 전송의 첫 번째 부분을 수행합니다. 사용자 정의 Connect:Direct 프로세스가 전송의 두 번째 부분을 수행합니다.

프로시저

1. 고유 기호 변수를 사용하는 Connect:Direct 프로세스를 작성하십시오.

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. 이 프로세스를 다음 위치의 텍스트 파일에 저장하십시오. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp`
3. 1단계에서 작성된 Connect:Direct 프로세스를 호출하는 규칙을 포함하도록 `ConnectDirectProcessDefinition.xml` 파일을 편집하십시오.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="Example.cdp" />
    </tns:process>
  </tns:processSet>

</tns:cdprocess>
```

이 예에서, TOBERMORY가 소스 또는 목적지 Connect:Direct 노드인 Connect:Direct 브릿지 에이전트에 전송 요청이 제출되면 `Example.cdp` Connect:Direct 프로세스가 호출됩니다.

4. 3단계에서 `ConnectDirectProcessDefinition.xml` 파일에 정의한 조건을 충족하는 파일 전송 요청을 제출하십시오.

예:

```
fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                 -sm QM_WIMBLEDON -dm QM_COMMON
                 -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                 -sd leave c:\bungo\source.txt
```

이 예에서 목적지 Connect:Direct 노드는 TOBERMORY입니다. 이 노드는 전송의 보조 노드이며 %FTESNODE의 값은 TOBERMORY로 설정됩니다. 이 명령은 `ConnectDirectProcessDefinition.xml` 파일에 설정된 조건과 일치합니다.

5. Managed File Transfer는 Connect:Direct 브릿지 에이전트와 동일한 시스템의 임시 위치로 소스 파일을 전송합니다.
6. Connect:Direct 브릿지 에이전트는 고유 기호 변수의 값을 전송 요청 및 구성 정보의 정보로부터 설정합니다. 고유 기호 변수는 다음 값으로 설정됩니다.
 - %FTEPNAME=*process_name* - 이 값은 Connect:Direct 브릿지 에이전트에서 생성한 8자의 프로세스 이름입니다.
 - %FTESNODE=TOBERMORY - 이 값은 **fteCreateTransfer** 명령의 **-df** 매개변수로부터 설정됩니다.
 - %FTEPUSER=*primary_node_user* - 이 정보는 ConnectDirectCredentials.xml 파일에서 가져옵니다.
 - %FTEPPASS=*primary_node_user_password* - 이 정보는 ConnectDirectCredentials.xml 파일에서 가져옵니다.
 - %FTESUSER=*secondary_node_user* - 이 정보는 ConnectDirectCredentials.xml 파일에서 가져옵니다.
 - %FTESPASS=*secondary_node_user_password* - 이 정보는 ConnectDirectCredentials.xml 파일에서 가져옵니다.
 - %FTEFFILE=*temporary_location* - 이 값은 Connect:Direct 브릿지 에이전트와 동일한 시스템에 있는 파일의 임시 위치입니다.
 - %FTEFDISP=leave - 이 값은 **fteCreateTransfer** 명령의 **-sd** 매개변수로부터 설정됩니다.
 - %FTETFILE=/home/bulgaria/destination.txt - 이 값은 **fteCreateTransfer** 명령의 **-df** 매개변수로부터 설정됩니다.
 - %FTETDISP=overwrite - 이 값은 **fteCreateTransfer** 명령의 **-de** 매개변수로부터 설정됩니다.
7. Connect:Direct 프로세스는 Connect:Direct 브릿지 노드에서 시작됩니다. Connect:Direct 는 Connect:Direct 브릿지 시스템의 임시 위치에서 Connect:Direct 노드 TOBERMORY가 실행 중인 시스템의 목적지 /home/bulgaria/destination.txt 로 파일을 전송합니다.

관련 개념

296 페이지의 『파일 전송 요청에서 사용자 정의 Connect:Direct 프로세스 제출』

파일 전송의 부분으로 사용자 정의 Connect:Direct 프로세스를 호출하는 Connect:Direct 브릿지 에이전트를 사용하는 전송에 대한 전송 요청을 제출할 수 있습니다.

관련 참조

[사용자 정의 Connect:Direct 프로세스와 함께 사용할 대체 변수](#)

Connect:Direct 프로세스를 사용하여 Managed File Transfer 전송 요청 제출

Connect:Direct 프로세스에서 Connect:Direct 브릿지 에이전트에 전송 요청을 제출할 수 있습니다. Managed File Transfer 는 Connect:Direct 프로세스의 **RUN TASK** 문에서 호출할 수 있는 명령을 제공합니다.

Managed File Transfer는 Connect:Direct 프로세스와 함께 사용할 수 있는 다음 명령을 제공합니다.

ftetag

전송에 필요한 감사 정보를 작성하려면 **ftebxfer** 또는 **ftecxfer** 명령에 선행하는 단계에서 이 명령을 지정하십시오. 이 명령은 전송의 소스 스펙을 매개변수로 취합니다. 소스 스펙의 형식에 대한 정보는 **fteCreateTransfer**: 새 파일 전송 시작을 참조하십시오.

ftebxfer

전송 요청이 제출되는 큐 관리자가 명령을 제출하는 Connect:Direct 노드와 동일한 시스템에 있는 경우 파일 전송 요청을 작성하려면 이 명령을 지정하십시오. 이 명령은 **fteCreateTransfer** 명령과 동일한 매개변수를 취합니다. 이러한 매개변수에 대한 정보는 **fteCreateTransfer**: 새 파일 전송 시작을 참조하십시오. 이 명령에는 추가 매개변수도 있습니다.

-qmgrname

필수입니다. 명령을 제출할 큐 관리자의 이름입니다.

ftexfer

전송 요청이 제출되는 큐 관리자가 명령을 제출하는 Connect:Direct 노드와 다른 시스템에 있을 때 파일 전송 요청을 작성하려면 이 명령을 지정하십시오. 이 명령은 **fteCreateTransfer** 명령과 동일한 매개변수를 취합니다. 매개변수에 대한 정보는 **fteCreateTransfer: 새 파일 전송 시작**을 참조하십시오. 이 명령에는 또한 세 개의 추가 매개변수가 있습니다.

-qmgrname

필수입니다. 명령을 제출할 큐 관리자의 이름입니다.

-connname

필수입니다. IBM MQ CONNAME 형식으로 지정된, 명령을 제출할 큐 관리자의 호스트 및 포트입니다 (예: host.example.com(1337)).

-channelname

선택사항. 명령을 제출할 큐 관리자에 연결하는 데 사용하기 위한 채널의 이름입니다. 지정되지 않는 경우 기본값 SYSTEM.DEF.SVRCONN이 사용됩니다.

관련 태스크

301 페이지의 『[Connect:Direct 요청자를 사용하여 Managed File Transfer를 호출하는 Connect:Direct 프로세스 작성 및 제출](#)』

Connect:Direct 요청자는 Managed File Transfer를 호출하는 Connect:Direct 프로세스를 작성하고 제출하는 데 사용할 수 있는 그래픽 사용자 인터페이스입니다.

관련 참조

예제: [MFT 명령을 호출하는 Connect:Direct 프로세스 파일](#)

Connect:Direct 요청자를 사용하여 Managed File Transfer를 호출하는 Connect:Direct 프로세스 작성 및 제출

Connect:Direct 요청자는 Managed File Transfer를 호출하는 Connect:Direct 프로세스를 작성하고 제출하는 데 사용할 수 있는 그래픽 사용자 인터페이스입니다.

이 태스크 정보

이 태스크는 Managed File Transfer **ftexfer** 명령 또는 **ftbxfex** 명령을 호출하는 Connect:Direct 프로세스를 작성하는 방법을 설명합니다. 전송 요청이 제출되는 큐 관리자가 명령을 제출하는 Connect:Direct 노드와 다른 시스템에 있는 경우 **ftexfer** 명령을 사용하십시오. 전송 요청이 제출되는 큐 관리자가 명령을 제출하는 Connect:Direct 노드와 동일한 시스템에 있는 경우 **ftbxfex** 명령을 사용하십시오. **ftexfer** 명령은 전송의 소스 에이전트의 에이전트 큐 관리자에 대한 클라이언트 연결을 작성합니다. **ftexfer** 명령을 호출하기 전에 **ftetag** 명령을 호출하여 소스 스펙 정보에 전달해야 합니다. 그러면 Managed File Transfer에서 전송이 시작된 방법과 동일하게 프로세스를 로그하고 감사할 수 있습니다.

프로시저

1. Connect:Direct 요청자를 시작하십시오.
2. 패널의 **노드** 탭에서 프로세스의 기본 노드로 사용되는 Connect:Direct 노드를 선택하십시오.
3. **파일 > 새로 작성 > 프로세스**를 선택하십시오. **프로세스 특성** 창이 열립니다.
4. **이름:** 필드에 프로세스의 이름을 입력하십시오.
5. **Snode > 이름:** 목록에서 보조 노드를 선택하십시오.
6. **Snode > 운영 체제** 목록에서 보조 노드의 운영 체제를 선택하십시오.
7. 옵션: 이 창에서 필요한 추가 정보를 완료하십시오.
8. **확인**을 클릭하십시오. **프로세스 특성** 창이 열립니다.
9. Managed File Transfer **ftetag** 명령을 실행하는 명령문을 작성하십시오.
 - a) 종료 명령문의 **프로세스** 창에서 마우스의 오른쪽 단추를 클릭하십시오.
 - b) **삽입 > 태스크 실행**을 선택하십시오. **태스크 실행 명령문** 창이 열립니다.
 - c) **레이블:** 필드에 Tag를 입력하십시오.

- d) 선택적 매개변수 또는 명령 필드에 `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)`를 입력하십시오. `source_specification` 형식에 대한 자세한 정보는 **fteCreateTransfer**: 새 파일 전송 시작을 참조하십시오.
 - e) 확인을 클릭하십시오. **태스크 실행 명령문** 창이 닫힙니다.
10. Managed File Transfer **ftecxfer** 또는 **ftebxfer** 명령을 실행하는 명령문을 작성하십시오.
- a) 종료 명령문의 프로세스 창에서 마우스의 오른쪽 단추를 클릭하십시오.
 - b) 삽입 > 태스크 실행을 선택하십시오. **태스크 실행 명령문** 창이 열립니다.
 - c) 레이블: 필드에 Transfer를 입력하십시오.
 - d) 선택적 매개변수 또는 명령 필드에 선택한 명령에 따라 `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` 또는 `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)`를 입력하십시오. **ftecxfer** 및 **ftebxfer** 명령이 사용하는 매개변수는 **fteCreateTransfer** 명령이 사용하는 매개변수와 같으며, 이외에 **ftecxfer** 및 **ftebxfer**에 고유한 몇 가지 추가 매개변수가 있습니다. 자세한 정보는 **fteCreateTransfer**: 새 파일 전송 시작 및 300 페이지의 『Connect:Direct 프로세스를 사용하여 Managed File Transfer 전송 요청 제출』의 내용을 참조하십시오.
 - e) 확인을 클릭하십시오. **태스크 실행 명령문** 창이 닫힙니다.
11. 옵션: 필요한 추가 명령문을 작성하십시오.
12. 프로세스를 제출하십시오.
- a) 프로세스 창에서 마우스의 오른쪽 단추를 클릭하십시오.
 - b) 제출을 선택하십시오. **Connect:Direct 첨부** 창이 열립니다.
 - c) 프로세스 실행에 사용할 사용자 이름 및 비밀번호를 입력하십시오.
 - d) 확인을 클릭하십시오.

관련 개념

300 페이지의 『Connect:Direct 프로세스를 사용하여 Managed File Transfer 전송 요청 제출』
Connect:Direct 프로세스에서 Connect:Direct 브릿지 에이전트에 전송 요청을 제출할 수 있습니다. Managed File Transfer 는 Connect:Direct 프로세스의 **RUN TASK** 문에서 호출할 수 있는 명령을 제공합니다.

IBM Integration Bus 에서 MFT 에 대한 작업

FTEOutput 및 FTEInput 노드를 사용하여 IBM Integration Bus 에서 Managed File Transfer 에 대해 작업할 수 있습니다.

- FTEInput 노드에서는 Managed File Transfer를 사용하여 네트워크를 통해 파일을 전송한 후 해당 파일을 Integration Bus 플로우의 일부로 처리할 수 있습니다.
- FTEOutput 노드에서는 Integration Bus 플로우를 통해 출력된 파일을 네트워크의 다른 위치로 전송할 수 있습니다.

Managed File Transfer의 모든 레벨에서 에이전트는 브로커 에이전트와 서로 파일을 주고 받을 수 있습니다.

자세한 정보는 [IBM Integration Bus 제품 문서](#)를 참조하십시오.

MFT 복구 및 재시작

에이전트 또는 큐 관리자가 사용 불가능한 경우(예: 전원 문제 또는 네트워크 장애), Managed File Transfer는 다음과 같이 복구합니다.

- 일반적으로 파일 전송 중 문제점이 발생하면 문제점을 정정한 후 Managed File Transfer에서 해당 파일 전송을 복구하고 재시작합니다.
- 에이전트 또는 큐 관리자를 사용할 수 없는 동안 전송 중인 파일이 삭제되거나 변경되는 경우 전송은 실패하고 실패에 대한 자세한 내용을 제공하는 메시지가 전송 로그에 수신됩니다.
- 파일 전송 중에 에이전트 프로세스가 실패하면 에이전트를 재시작할 때 전송이 계속됩니다.
- 에이전트 큐 관리자와 에이전트의 연결이 끊어지면 에이전트는 대기하면서 큐 관리자에 다시 연결하기 위해 시도합니다. 에이전트가 큐 관리자와 다시 연결되면 현재 전송이 계속됩니다.

- 어떤 이유로든 에이전트가 중지되는 경우 에이전트와 연관된 모든 자원 모니터가 폴링을 중지합니다. 에이전트가 복구될 때 모니터도 재시작하며 자원 폴링도 계속됩니다.
- 소스 처리가 삭제인 파일 전송의 경우 모든 데이터가 소스 에이전트에서 목적지 에이전트로 송신된 후 복구가 발생하면 소스 파일이 삭제되기 전 잠금 해제됩니다. 이러한 잠금 해제는 소스 파일이 삭제되기 전에 해당 파일을 수정할 수 있음을 의미합니다. 따라서 소스 파일 삭제는 안전하지 않은 것으로 간주되며 다음 경고가 표시됩니다.

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

이 경우 소스 파일의 콘텐츠가 수정되지 않았는지 확인한 다음, 수동으로 소스 파일을 삭제하십시오.

IBM MQ Explorer에서 전송 상태를 확인할 수 있습니다. 전송이 Stalled로 나타나면, 에이전트 또는 전송과 관련된 두 에이전트 사이에 문제가 발생했음을 나타내므로 정정 조치를 취해야 합니다.

관련 태스크

303 페이지의 『중단된 전송의 복구에 대한 제한시간 설정』

한 소스 에이전트의 모든 전송에 적용되는 중단된 파일 전송의 전송 복구 제한시간을 설정할 수 있습니다. 개별 전송에 대한 전송 복구 제한시간을 설정할 수도 있습니다. 소스 에이전트가 중단된 파일 전송을 복구하려고 계속 시도하는 특정 시간을 초 단위로 설정하고 에이전트가 이 제한시간에 도달할 때 전송에 성공하지 못하면 전송은 실패합니다.

중단된 전송의 복구에 대한 제한시간 설정

한 소스 에이전트의 모든 전송에 적용되는 중단된 파일 전송의 전송 복구 제한시간을 설정할 수 있습니다. 개별 전송에 대한 전송 복구 제한시간을 설정할 수도 있습니다. 소스 에이전트가 중단된 파일 전송을 복구하려고 계속 시도하는 특정 시간을 초 단위로 설정하고 에이전트가 이 제한시간에 도달할 때 전송에 성공하지 못하면 전송은 실패합니다.

이 태스크 정보

전송 복구 제한시간 매개변수를 에이전트의 `agent.properties` 파일에 추가하여 소스 에이전트의 모든 전송에 적용되는 전송 복구 제한시간을 설정할 수 있습니다. 명령행에서 또는 IBM MQ Explorer를 사용하여 또는 Apache Ant 태스크를 사용하여 개별 전송의 전송 복구 제한시간을 설정할 수도 있습니다.

`agent.properties` 파일에서 설정된 전송 복구 제한시간 값이 있는 경우 개별 전송의 전송 복구 제한시간을 설정하면 `agent.properties` 파일의 값을 대체합니다.

전송 복구 제한시간에 대해 다음 세 가지 옵션이 있습니다.

- 에이전트는 성공적으로 완료될 때까지 중단된 전송을 계속해서 복구하려고 시도합니다. 전송 복구 제한시간이 설정되지 않은 경우 이는 에이전트의 기본 작동과 같습니다.
- 에이전트는 전송이 복구 상태가 되는 즉시 전송을 실패로 표시합니다.
- 전송이 실패됨으로 표시되기 전까지 에이전트가 지정된 시간 동안 중단된 전송을 계속해서 재시도합니다.

파일 전송 복구 제한시간 설정은 선택사항입니다. 이를 설정하지 않으면 전송은 기본 동작을 따르며, 여기서 에이전트는 성공할 때까지 정지된 전송을 복구하려고 계속 시도합니다.

관련 개념

302 페이지의 『MFT 복구 및 재시작』

에이전트 또는 큐 관리자가 사용 불가능한 경우(예: 전원 문제 또는 네트워크 장애), Managed File Transfer는 다음과 같이 복구합니다.

전송 복구 제한시간 개념

소스 에이전트가 중단된 파일 전송의 복구를 계속 시도하는 기간(초)을 설정할 수 있습니다. 에이전트가 재시도 간격에 대한 제한시간에 도달해도 전송에 성공하지 못한 경우, 전송이 실패합니다.

복구 제한시간 우선순위

fteCreateTransfer, **fteCreateTemplate** 또는 **fteCreateMonitor** 명령을 통해 지정되거나 IBM MQ Explorer를 사용하여 지정되거나 **fte:filespec** 중첩 요소에 지정된 개별 전송에 대한 전송 복구 제한시간 값은 소스 에이전트에 대한 `agent.properties` 파일의 **transferRecoveryTimeout** 매개변수에 지정된 값보다 우선합니다.

예를 들어, **fteCreateTransfer** 명령이 **-rt** 매개변수 및 값 쌍 없이 시작되는 경우 소스 에이전트 AGENT1은 **transferRecoveryTimeout** 값에 대해 `agent.properties` 파일을 검사하여 복구 제한시간 작동을 판별합니다.

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

`agent.properties` 파일의 **transferRecoveryTimeout** 매개변수가 설정되어 있지 않거나 **-1**로 설정된 경우 에이전트는 기본 작동을 따르며 성공할 때까지 전송을 복구하려고 시도합니다.

그러나 **fteCreateTransfer** 명령에 **-rt** 매개변수가 포함되는 경우 이 매개변수의 값이 `agent.properties` 파일의 값보다 우선하고 전송을 위한 복구 제한시간 설정으로 사용됩니다.

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

복구 제한시간 카운터

전송이 복구 상태로 전환되면 복구 제한시간 카운터가 시작됩니다. 전송 상태가 복구 중으로 변경되었음을 나타내는 `Log/agent_name/transfer_ID` 토픽 스트링 및 상태가 변경된 소스 에이전트 클럭 시간이 포함된 전송 로그 메시지가 `SYSTEM.FTE` 토픽에 공개됩니다. 설정된 재시도 간격 내에 전송이 재개되고 복구 제한시간(카운터<=복구 제한시간)에 도달하지 않은 경우 카운터가 0으로 재설정됩니다. 이 경우 전송이 복구 상태로 전환되면 다시 시작할 수 있습니다.

카운터가 복구 제한시간에 설정된 최대 값에 도달한 경우(counter==recovery timeout), 전송 복구가 중지되고 소스 에이전트는 전송 실패를 보고합니다. 복구 제한시간에 도달하여 발생하는 이 유형의 전송 실패는 메시지 코드 `RECOVERY TIMEOUT (69)`로 표시됩니다. 다른 전송 로그 메시지는 토픽 문자열 `Log/agent_name/transfer_ID`와 함께 `SYSTEM.FTE` 토픽에 발행되어 전송이 실패했음을 표시하고 메시지, 리턴 코드 및 소스 에이전트의 이벤트 로그를 포함합니다. 소스 에이전트의 이벤트 로그는 다음 이벤트 중 어느 하나가 복구 중에 발생하면 메시지와 함께 업데이트됩니다.

- 복구 제한시간 매개변수가 **-1**보다 큰 값으로 설정되는 경우 전송이 복구 상태가 됩니다. **TransferId**에 대한 복구 타이머의 시작과 소스 에이전트가 복구 제한시간 처리를 시작하기 전에 대기하는 시간을 나타내도록 에이전트의 이벤트 로그가 업데이트됩니다.
- 전송 복구가 재개되면 소스 에이전트의 이벤트 로그는 복구 상태였던 **TransferId**가 재개됨을 나타내는 새 메시지로 업데이트됩니다.
- 복구 중인 전송이 제한시간을 초과한 경우 소스 에이전트의 이벤트 로그는 복구 제한시간으로 인해 복구하는 동안 실패한 **TransferId**를 표시하기 위해 업데이트됩니다.

이 로그 메시지를 사용하여 사용자(구독자 및 로거)는 전송 복구 제한시간 때문에 실패한 전송을 식별할 수 있습니다.

복구 제한시간의 카운터는 항상 소스 에이전트에 있습니다. 그러나 대상 에이전트가 소스 에이전트로부터 적절한 시간 내에 정보를 수신하지 못할 경우 전송을 복구 상태로 전환하라는 요청을 소스 에이전트에 전송할 수 있습니다. 복구 제한시간 옵션이 설정된 전송의 경우 소스 에이전트는 대상 에이전트로부터 요청을 수신하면 복구 제한시간 카운터를 시작합니다.

복구 제한시간 옵션을 사용하지 않는데 전송이 실패하거나 부분적으로 완료된 전송의 경우 수동으로 처리해야 합니다.

여러 파일에 대해 단일 전송 요청이 발행되고 일부 파일은 성공했지만 일부는 부분적으로 완료된 전송 세트의 경우, 해당 전송은 예상대로 완료되지 않았기 때문에 계속 실패로 표시됩니다. 소스 에이전트가 부분적으로 완료된 파일을 전송하는 중에 제한시간 초과가 발생할 수도 있습니다.

대상 에이전트 및 파일 서버가 파일 전송을 승인할 수 있는 상태이고 준비되었는지 확인하십시오.

전체 세트에 대해 다시 전송 요청을 실행해야 하지만, 최초 전송 시도부터 일부 파일이 대상에 남아 있기 때문에 문제를 방지하기 위해 `overwrite if existing`(존재할 경우 덮어쓰기) 옵션을 지정하여 새 요청을 실행할 수 있습니다. 이렇게 하면 이전 전송 시도에서 완료되지 않은 파일 세트가 해당 파일이 다시 대상에 쓰여지기 전에 새 전송 중에 지워집니다.

IBM MQ 9.1.5부터는 초기 전송 시도가 실패한 후에 대상에 남겨진 부분 파일을 수동으로 제거하지 않아도 됩니다. 전송 복구 제한시간이 전송에 대해 설정된 경우 전송 복구가 제한시간을 초과하면 소스 에이전트는 전송을 `RecoveryTimedOut` 상태로 이동합니다. 전송이 다시 동기화된 후, 대상 에이전트는 전송 중에 작성된 파트 파일을 제거하고 완료 메시지를 소스 에이전트에 송신합니다.

추적 및 메시지

진단 목적을 위해 추적 지점이 포함되어 있습니다. 복구 제한시간 값, 재시도 간격 시작, 재개 기간의 시작 및 카운터 재설정 및 전송이 제한시간을 초과했는지 및 실패했는지 여부가 로깅됩니다. 문제점이나 예기치 않은 동작이 발생한 경우 소스 에이전트 출력 로그 및 추적 파일을 수집하여 IBM 지원 센터에서 요청할 경우 이를 제공하여 문제 해결을 지원할 수 있습니다.

다음의 경우 메시지가 알립니다.

- 전송이 복구(`BFGTR0081I`) 상태가 됨
- 전송이 복구 중 제한시간을 초과하여 종료되었음(`BFGSS0081E`).
- 복구 상태 후에 전송이 재개됨(`BFGTR0082I`)

관련 개념

302 페이지의 『MFT 복구 및 재시작』

에이전트 또는 큐 관리자가 사용 불가능한 경우(예: 전원 문제 또는 네트워크 장애), Managed File Transfer는 다음과 같이 복구합니다.

하나의 소스 에이전트에 대한 모든 전송의 전송 복구 제한시간 설정

`transferRecoveryTimeout` 매개변수를 `agent.properties` 파일에 추가하여 한 소스 에이전트의 모든 전송에 적용되는 전송 복구 제한시간을 설정할 수 있습니다.

이 태스크 정보

한 소스 에이전트의 모든 전송에 적용되는 전송 복구 제한시간을 설정하려면 `transferRecoveryTimeout`에 대한 매개변수 및 값 쌍을 `agent.properties` 파일에 추가합니다.

`transferRecoveryTimeout` 매개변수에 대해 다음 세 가지 옵션이 있습니다.

-1

전송이 성공할 때까지 에이전트는 중단된 전송의 복구를 계속 시도합니다. 이 옵션을 사용할 경우 특성이 설정되지 않았을 때 에이전트의 기본 동작과 같습니다.

0

에이전트는 복구를 시작하자마자 파일 전송을 중지합니다.

>0

에이전트는 지정된 양의 정수 값으로 설정된 기간(초) 동안 중단된 전송의 복구를 계속 시도합니다.

`agent.properties` 파일의 변경사항은 에이전트가 재시작 후에만 적용됩니다.

필요한 경우 개별 전송을 위해 `agent.properties` 파일에서 전송 복구 제한시간 값을 대체할 수 있습니다. 자세한 정보는 306 페이지의 『개별 전송의 전송 복구 제한시간 설정』의 내용을 참조하십시오.

프로시저

- 중단된 전송이 성공적으로 완료될 때까지 에이전트가 이 전송을 계속해서 복구 시도하도록 지정하려면 다음 예에 표시된 대로 전송 복구 제한시간 값 -1을 설정하십시오.

```
transferRecoveryTimeout=-1
```

- 전송이 복구 상태가 되는 즉시 에이전트가 이 전송을 실패함으로 표시하도록 지정하려면 다음 예에 표시된 대로 전송 복구 제한시간 값 0을 설정하십시오.

```
transferRecoveryTimeout=0
```

- 전송이 실패됨으로 표시되기 전에 에이전트가 지정된 시간 동안 중단된 전송을 계속해서 재시도하도록 지정하려면 에이전트가 계속해서 재시도하게 할 시간에 대해 전송 복구 제한시간 값(초)을 설정하십시오.

예를 들어, 전송 복구 제한시간 값을 21600으로 설정하는 것은 에이전트가 전송이 복구 상태가 된 시간부터 6시간 동안 계속해서 전송 복구를 시도함을 의미합니다.

```
transferRecoveryTimeout=21600
```

이 매개변수의 최대값은 999999999입니다.

개별 전송의 전송 복구 제한시간 설정

명령행에서 또는 IBM MQ Explorer를 사용하거나 Apache Ant 태스크를 사용하여 개별 전송의 복구 제한시간을 설정할 수 있습니다. `agent.properties` 파일에 설정된 전송 복구 제한시간 값이 있는 경우 개별 전송의 전송 복구 제한시간을 설정하면 `agent.properties` 파일에 설정된 값이 대체됩니다.

이 태스크 정보

다음과 같은 경우 개별 전송의 전송 복구 제한시간 매개변수를 설정할 수 있습니다.

- **fteCreateTransfer** 명령을 사용하거나 IBM MQ Explorer를 사용하여 전송을 작성 중인 경우.
- **fteCreateTemplate** 명령을 사용하거나 IBM MQ Explorer를 사용하여 전송 템플릿을 작성 중인 경우.
- **fteCreateMonitor** 명령을 사용하거나 IBM MQ Explorer를 사용하여 자원 모니터를 작성 중인 경우.
- `fte:filecopy` 또는 `fte:filemove` Ant 태스크를 사용하여 파일을 복사하거나 이동 중인 경우.

개별 전송의 전송 복구 제한시간 값을 설정하면 이 값이 `agent.properties` 파일에 설정된 전송 복구 제한시간 값을 대체합니다(305 페이지의 『하나의 소스 에이전트에 대한 모든 전송의 전송 복구 제한시간 설정』 참조).

프로시저

- **fteCreateTransfer** 또는 **fteCreateTemplate** 명령을 사용하여 전송 복구 제한시간을 설정하려면 **-rt** 매개변수에 적절한 옵션을 지정하십시오.
 - 1
전송이 성공할 때까지 에이전트는 중단된 전송의 복구를 계속 시도합니다. 이 옵션을 사용할 경우 특성이 설정되지 않았을 때 에이전트의 기본 동작과 같습니다.
 - 0
에이전트는 복구를 시작하자마자 파일 전송을 중지합니다.
 - >0
에이전트가 지정된 시간(초) 동안 계속 중단된 전송을 복구하려고 시도합니다.

fteCreateTransfer 명령의 예

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 0 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```


fteCreateTemplate 명령의 예

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt -1 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 21600 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

fteCreateMonitor 명령에 대한 **-rt** 매개변수가 없습니다. **-rt** 매개변수를 **fteCreateTransfer** 명령과 함께 설정하고 **-gt** 매개변수도 설정하는 경우 복구 제한시간 매개변수는 **fteCreateTransfer** 명령 실행 시 생성되는 전송 정의와 함께 XML 문서에 포함됩니다. 그러면 자원 모니터가 **fteCreateMonitor** 명령 실행 시 이 XML 문서를 사용합니다. 다음 예에서는 전송 복구 제한시간 세부사항이 task.xml 파일에 포함됩니다.

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr
"fileSize>=5MB,*.zip"
```

- IBM MQ Explorer 새 전송, 새 모니터 또는 새 템플릿 마법사 페이지를 사용하여 전송 복구 제한시간을 설정하려면 **전송 복구 제한시간(초)** 필드에서 필요한 옵션을 선택하십시오.

소스 에이전트로

소스 에이전트를 선택하는 경우 agent.properties 파일의 **transferRecoveryTimeout** 매개변수 값이 사용됩니다(설정된 경우). 그렇지 않은 경우 전송 복구 제한시간의 기본 작동이 적용됩니다.

숫자 목록 상자

숫자 목록 상자에 시간을 초 단위로 입력하면 에이전트는 지정된 시간 동안 중단된 전송을 계속해서 복구하려고 시도합니다.

없음

없음을 선택하면 전송 복구 제한시간이 설정되지 않으며 에이전트는 전송이 성공할 때까지 중단된 전송을 계속해서 복구하려고 시도합니다.

- Ant 태스크를 사용하여 복구 제한시간을 설정하려면 파일 이동 또는 복사를 위한 **fte:filecopy** 또는 **fte:filemove** 요소와 함께 **transferRecoveryTimeout** 옵션 및 값을 포함시키십시오. 예를 들면, 다음과 같습니다.

fte:filecopy의 예

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result" transferRecoveryTimeout="0">

  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

fte:filemove의 예

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="move.result" transferRecoveryTimeout="21600">

  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

관련 개념

302 페이지의 『MFT 복구 및 재시작』

에이전트 또는 큐 관리자가 사용 불가능한 경우(예: 전원 문제 또는 네트워크 장애), Managed File Transfer는 다음과 같이 복구합니다.

관련 태스크

[202 페이지의 『새 파일 전송 시작』](#)

IBM MQ Explorer 또는 명령행에서 새 파일 전송을 시작할 수 있으며 단일 파일 또는 그룹의 여러 파일을 전송하도록 선택할 수 있습니다.

[241 페이지의 『IBM MQ Explorer를 사용하여 파일 전송 템플릿 작성』](#)

IBM MQ Explorer 또는 명령행에서 파일 전송 템플릿을 작성할 수 있습니다. 그런 다음 해당 템플릿을 사용하여 템플릿 세부사항으로 새 파일 전송을 작성하거나, 템플릿을 제출하여 파일 전송을 시작할 수 있습니다.

[211 페이지의 『MFT 자원 모니터링』](#)

Managed File Transfer 자원(예: 큐 또는 디렉토리)을 모니터링할 수 있습니다. 이 자원의 조건이 충족되면 자원 모니터가 파일 전송과 같은 태스크를 시작합니다. IBM MQ Explorer용 Managed File Transfer 플러그인에서 **fteCreateMonitor** 명령 또는 **모니터** 보기를 사용하여 자원 모니터를 작성할 수 있습니다.

[210 페이지의 『전송 로그에서 파일 전송 상태 보기』](#)

IBM MQ Explorer의 **전송 로그** 를 사용하여 파일 전송의 세부사항을 볼 수 있습니다. 이는 명령행 또는 IBM MQ Explorer에서 시작된 전송일 수 있습니다. 또한 **전송 로그**에 표시되는 내용을 사용자 정의할 수 있습니다.

관련 참조

MFTagent.properties 파일

fteCreateTransfer: 새 파일 전송 시작

fteCreateTemplate: 새 파일 전송 템플릿 작성

fteCreateMonitor: MFT 자원 모니터 작성

fte:filecopy Ant 태스크

fte:filemove Ant 태스크

Windows

Linux

AIX

MQ Telemetry 관리

MQ Telemetry는 IBM MQ Explorer를 사용하거나 명령행에서 관리됩니다. 탐색기를 사용하여 텔레메트리 채널을 구성하고 텔레메트리 서비스를 제어하며 IBM MQ에 연결된 MQTT 클라이언트를 모니터링하십시오. JAAS, TLS 및 IBM MQ 오브젝트 권한 관리자를 사용하여 MQ Telemetry의 보안을 구성하십시오.

IBM MQ Explorer를 사용하여 관리

탐색기를 사용하여 텔레메트리 채널을 구성하고 텔레메트리 서비스를 제어하며 IBM MQ에 연결된 MQTT 클라이언트를 모니터링하십시오. JAAS, TLS 및 IBM MQ 오브젝트 권한 관리자를 사용하여 MQ Telemetry의 보안을 구성하십시오.

명령행을 사용하여 관리

MQ Telemetry는 **MQSC** 명령을 사용하여 명령행에서 완전히 관리할 수 있습니다.

MQ Telemetry 문서에는 IBM MQ Telemetry Transport v3 클라이언트 애플리케이션의 기본 사용법을 증명하는 샘플 스크립트가 있습니다.

사용하기 전에 [IBM MQ Telemetry Transport 샘플 프로그램](#)에 있는 샘플을 읽고 이해하십시오.

관련 개념

[MQ Telemetry](#)

관련 참조

[MQXR 특성](#)

Linux

AIX

Linux 및 AIX에서 텔레메트리에 대한 큐 관리자 구성

다음 단계에 따라 MQ Telemetry 를 수동으로 구성하십시오. 게스트 사용자 ID를 사용하는 단순 구성만 필요한 경우 대신 IBM MQ Explorer에서 MQ Telemetry 지원 마법사를 실행할 수 있습니다.

시작하기 전에

단순 구성만 필요한 경우에는 IBM MQ Explorer의 MQ Telemetry 지원 사용을 고려하십시오. 이 지원에는 마법사 및 샘플 명령 프로시저 `sampleMQM`가 포함됩니다. 이러한 자원은 게스트 사용자 ID를 사용하여 초기 구성을

설정합니다. [IBM MQ Explorer](#) 를 사용하여 [MQ Telemetry 설치 확인](#) 및 [IBM MQ Telemetry Transport 샘플 프로그래밍](#)의 내용을 참조하십시오.

다른 인증 방법을 사용하는 보다 복잡한 구성이 필요한 경우 이 태스크의 단계를 사용하십시오. 다음 초기 단계로 시작하십시오.

1. IBM MQ 및 MQ Telemetry 기능을 설치하는 방법에 대한 정보는 [MQ Telemetry의 설치 고려사항](#) 을 참조하십시오.
2. 큐 관리자를 작성하고 시작하십시오. 이 태스크에서는 큐 관리자를 *qMgr*이라고 합니다.
3. 이 태스크의 일부로 텔레메트리(MQXR) 서비스를 구성합니다. MQXR 특성 설정은 플랫폼 특정 특성 파일 (*mqxr_win.properties*)에 저장됩니다. 거의 모든 설정이 MQSC 관리 명령 또는 IBM MQ Explorer를 통해 구성될 수 있으므로 일반적으로 MQXR 특성 파일을 직접적으로 편집할 필요가 없습니다. 파일을 직접적으로 편집하기로 결심한 경우, 변경하기 전에 큐 관리자를 중지하십시오. [MQXR 특성](#)을 참조하십시오.

이 태스크 정보

다른 권한 부여 스킴을 사용하여 MQ Telemetry 를 수동으로 구성하려면 이 태스크의 단계를 따르십시오.

프로시저

1. 텔레메트리 샘플 디렉토리에서 명령 창을 여십시오.

텔레메트리 샘플 디렉토리는 `/opt/mqm/mqxr/samples`입니다.

2. 텔레메트리 전송 큐를 작성하십시오.

SYSTEM.MQTT.TRANSMIT.QUEUE 가 없으면 텔레메트리 (MQXR) 서비스가 처음 시작될 때 자동으로 작성되고 게스트 사용자 ID를 사용하도록 설정됩니다. 그러나 이 태스크는 다른 권한 부여 스킴을 사용하도록 MQ Telemetry 를 구성합니다. 이 태스크의 경우 텔레메트리 (MQXR) 서비스를 시작하기 전에 SYSTEM.MQTT.TRANSMIT.QUEUE를 작성하고 이에 대한 액세스를 구성합니다.

다음 명령을 실행하십시오.

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

3. 기본 전송 큐를 설정하십시오.

SYSTEM.MQTT.TRANSMIT.QUEUE 가 기본 전송 큐인 경우 MQTT 클라이언트에 직접 메시지를 송신하는 것이 더 쉽습니다. 그렇지 않으면 IBM MQ 메시지를 수신하는 모든 클라이언트에 대해 리모트 큐 정의를 추가해야 합니다. [313 페이지의 『클라이언트에 직접 메시지 송신』](#)의 내용을 참조하십시오. 기본 전송 큐를 변경하면 기존 구성을 방해할 수 있습니다.

텔레메트리 (MQXR) 서비스가 처음 시작되면 SYSTEM.MQTT.TRANSMIT.QUEUE 를 큐 관리자의 기본 전송 큐로 설정하지 않습니다. 이 설정을 구성하려면 기본 전송 큐 특성을 대체합니다. IBM MQ Explorer를 사용하거나 다음 명령을 실행하여 이를 수행합니다.

```
echo "ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

4. 하나 이상의 사용자 ID를 작성하려면 [315 페이지의 『MQTT 클라이언트가 IBM MQ 오브젝트에 액세스할 수 있도록 권한 부여』](#)에서 프로시저를 수행하십시오. 사용자 ID에 발행물을 MQTT 클라이언트에 발행, 구독 및 전송하기 위한 권한이 있습니다.
5. `installMQXRService_unix.mqsc` 파일을 편집하여 MQTT TLS 채널의 비밀번호 문구를 암호화하는 데 사용되는 키 파일을 구성하십시오.
 - a) `WMQ program installation directory/mqxr/samples/installMQXRService_unix.mqsc` 파일을 여십시오.
 - b) **STARTARG** 매개변수를 포함하는 행을 찾고 **-sf** 옵션을 편집하여 신임 정보 키 파일의 위치를 지정하십시오.

기본적으로 `installMQXRService_unix.mqsc` 파일은 [DEFAULT]라는 기본 키 파일을 사용합니다. 기본 키 파일은 모든 IBM MQ 설치에서 동일하므로 비밀번호 문구를 암호화할 때 설치에 고유한 키 파일을 제공해야 합니다.

310 페이지의 『SYSTEM.MQXR.SERVICE 작성』의 예제 코드도 참조하십시오.

- 다음 명령을 실행하여 텔레메트리 (MQXR) 서비스를 설치하십시오.

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc queue_manager
```

- 서비스를 시작하십시오.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE)" | runmqsc qMgr
```

텔레메트리(MQXR) 서비스는 큐 관리자가 시작될 때 자동으로 시작됩니다. 큐 관리자가 이미 실행 중이므로 이 태스크에서 수동으로 시작됩니다.

- IBM MQ Explorer를 사용하여 MQTT 클라이언트에서 연결을 승인하기 위해 텔레메트리 채널을 구성하십시오.

ID가 309 페이지의 『4』 단계에서 정의된 사용자 ID중 하나가 되도록 텔레메트리 채널을 구성해야 합니다.

`DEFINE CHANNEL(MQTT)`도 참조하십시오.

- 샘플 클라이언트를 실행하여 구성을 확인하십시오.

샘플 클라이언트가 텔레메트리 채널에서 작동하게 하려면 해당 채널이 클라이언트에게 발행, 구독 및 발행을 수신할 권한을 부여해야 합니다. 샘플 클라이언트는 기본적으로 포트 1883에서 텔레메트리 채널에 연결합니다. IBM MQ Telemetry Transport 샘플 프로그램도 참조하십시오.

SYSTEM.MQXR.SERVICE 작성

`runMQXRService` 명령을 사용하여 SYSTEM.MQXR.SERVICE를 작성하십시오.

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+" -sf "/home/keyFileLocation/keyFile.txt"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

Windows에 텔레메트리를 위한 큐 관리자 구성

다음 단계에 따라 MQ Telemetry 를 수동으로 구성하십시오. 게스트 사용자 ID를 사용하는 단순 구성만 필요한 경우 대신 IBM MQ Explorer에서 MQ Telemetry 지원 마법사를 실행할 수 있습니다.

시작하기 전에

단순 구성만 필요한 경우에는 IBM MQ Explorer의 MQ Telemetry 지원 사용을 고려하십시오. 이 지원에는 마법사 및 샘플 명령 프로시저 `sampleMQM`가 포함됩니다. 이러한 자원은 게스트 사용자 ID를 사용하여 초기 구성을 설정합니다. IBM MQ Explorer 를 사용하여 MQ Telemetry 설치 확인 및 IBM MQ Telemetry Transport 샘플 프로그램의 내용을 참조하십시오.

다른 인증 방법을 사용하는 보다 복잡한 구성이 필요한 경우 이 태스크의 단계를 사용하십시오. 다음 초기 단계로 시작하십시오.

- IBM MQ 및 MQ Telemetry 기능을 설치하는 방법에 대한 정보는 [MQ Telemetry의 설치 고려사항](#) 을 참조하십시오.
- 큐 관리자를 작성하고 시작하십시오. 이 태스크에서는 큐 관리자를 `qMgr`이라고 합니다.

- 이 태스크의 일부로 텔레메트리(MQXR) 서비스를 구성합니다. MQXR 특성 설정은 플랫폼 특정 특성 파일(mqxr_win.properties)에 저장됩니다. 거의 모든 설정이 MQSC 관리 명령 또는 IBM MQ Explorer를 통해 구성될 수 있으므로 일반적으로 MQXR 특성 파일을 직접적으로 편집할 필요가 없습니다. 파일을 직접적으로 편집하기로 결심한 경우, 변경하기 전에 큐 관리자를 중지하십시오. [MQXR 특성을 참조하십시오.](#)

이 태스크 정보

다른 권한 부여 스킴을 사용하여 MQ Telemetry 를 수동으로 구성하려면 이 태스크의 단계를 따르십시오.

프로시저

- 텔레메트리 샘플 디렉토리에서 명령 창을 여십시오.

텔레메트리 샘플 디렉토리는 *WMQ program installation directory\mqxr\samples*입니다.

- 텔레메트리 전송 큐를 작성하십시오.

SYSTEM.MQTT.TRANSMIT.QUEUE 가 없으면 텔레메트리 (MQXR) 서비스가 처음 시작될 때 자동으로 작성되고 게스트 사용자 ID를 사용하도록 설정됩니다. 그러나 이 태스크는 다른 권한 부여 스킴을 사용하도록 MQ Telemetry 를 구성합니다. 이 태스크의 경우 텔레메트리 (MQXR) 서비스를 시작하기 전에 SYSTEM.MQTT.TRANSMIT.QUEUE를 작성하고 이에 대한 액세스를 구성합니다.

다음 명령을 실행하십시오.

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

- 기본 전송 큐를 설정하십시오.

SYSTEM.MQTT.TRANSMIT.QUEUE 가 기본 전송 큐인 경우 MQTT 클라이언트에 직접 메시지를 송신하는 것이 더 쉽습니다. 그렇지 않으면 IBM MQ 메시지를 수신하는 모든 클라이언트에 대해 리모트 큐 정의를 추가해야 합니다. [313 페이지의 『클라이언트에 직접 메시지 송신』](#)의 내용을 참조하십시오. 기본 전송 큐를 변경하면 기존 구성을 방해할 수 있습니다.

텔레메트리 (MQXR) 서비스가 처음 시작되면 SYSTEM.MQTT.TRANSMIT.QUEUE 를 큐 관리자의 기본 전송 큐로 설정하지 않습니다. 이 설정을 구성하려면 기본 전송 큐 특성을 대체합니다. IBM MQ Explorer를 사용하거나 다음 명령을 실행하여 이를 수행합니다.

```
echo ALTER QMGR DEFQXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

- 하나 이상의 사용자 ID를 작성하려면 [315 페이지의 『MQTT 클라이언트가 IBM MQ 오브젝트에 액세스할 수 있도록 권한 부여』](#)에서 프로시저를 수행하십시오. 사용자 ID에 발행물을 MQTT 클라이언트에 발행, 구독 및 전송하기 위한 권한이 있습니다.
- installMQXRService_win.mqsc 파일을 편집하여 MQTT TLS 채널의 비밀번호 문구를 암호화하는 데 사용되는 키 파일을 구성하십시오.

a) WMQ program installation

directory\mqxr\samples\installMQXRService_win.mqsc 파일을 여십시오.

- STARTARG** 매개변수를 포함하는 행을 찾고 **-sf** 옵션을 편집하여 신임 정보 키 파일의 위치를 지정하십시오.

기본적으로 installMQXRService_win.mqsc 파일은 [DEFAULT]라는 기본 키 파일을 사용합니다. 기본 키 파일은 모든 IBM MQ 설치에서 동일하므로 비밀번호 문구를 암호화할 때 설치에 고유한 키 파일을 제공해야 합니다.

[312 페이지의 『작성 중 SYSTEM.MQXR.SERVICE』](#)의 예제 코드도 참조하십시오.

- 다음 명령을 실행하여 텔레메트리 (MQXR) 서비스를 설치하십시오.

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

- 서비스를 시작하십시오.


```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

텔레메트리(MQXR) 서비스는 큐 관리자가 시작될 때 자동으로 시작됩니다. 큐 관리자가 이미 실행 중이므로 이 태스크에서 수동으로 시작됩니다.

8. IBM MQ Explorer를 사용하여 MQTT 클라이언트에서 연결을 승인하기 위해 텔레메트리 채널을 구성하십시오.

ID가 [311 페이지의 『4』](#) 단계에서 정의된 사용자 ID중 하나가 되도록 텔레메트리 채널을 구성해야 합니다.

[DEFINE CHANNEL\(MQTT\)](#)도 참조하십시오.

9. 샘플 클라이언트를 실행하여 구성을 확인하십시오.

샘플 클라이언트가 텔레메트리 채널에서 작동하게 하려면 해당 채널이 클라이언트에게 발행, 구독 및 발행을 수신할 권한을 부여해야 합니다. 샘플 클라이언트는 기본적으로 포트 1883에서 텔레메트리 채널에 연결합니다. [IBM MQ Telemetry Transport 샘플 프로그램](#)도 참조하십시오.

작성 중 SYSTEM.MQXR.SERVICE

`runMQXRService` 명령을 사용하여 SYSTEM.MQXR.SERVICE를 작성하십시오.

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+\mqxr\bin\runMQXRService.bat') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\" -sf
"c:\keyFileLocation\keyFile.txt"') +
STOPCMD('+MQ_INSTALL_PATH+\mqxr\bin\endMQXRService.bat') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+\mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+\mqxr.stderr')
```

Windows

Linux

AIX

메시지를 MQTT 클라이언트로 송신하기 위한 분산 큐잉

구성

IBM MQ 애플리케이션은 클라이언트가 작성한 구독에 발행하거나 메시지를 직접적으로 보냄으로써 MQTT v3 클라이언트 메시지를 보낼 수 있습니다. 둘 중 어떤 방식을 사용하든 메시지는 SYSTEM.MQTT.TRANSMIT.QUEUE에 배치되고 텔레메트리(MQXR) 서비스를 통해 클라이언트에 보내집니다. SYSTEM.MQTT.TRANSMIT.QUEUE에 메시지를 배치하는 방법에는 여러 가지가 있습니다.

MQTT 클라이언트 구독에 대한 메시지 발행

텔레메트리(MQXR) 서비스는 MQTT 클라이언트 대신 구독을 작성합니다. 클라이언트는 클라이언트가 보낸 구독과 일치하는 발행물에 대한 목적지입니다. 텔레메트리 서비스는 일치하는 발행을 클라이언트로 다시 전달합니다.

MQTT 클라이언트는 해당 큐 관리자 이름이 `ClientIdentifier`로 설정된 큐 관리자로 IBM MQ에 연결됩니다. 클라이언트로 보낼 발행의 목적지는 SYSTEM.MQTT.TRANSMIT.QUEUE 전송 큐입니다. 텔레메트리 서비스는 특정 클라이언트에 대한 키로 대상 큐 관리자 이름을 사용하여 SYSTEM.MQTT.TRANSMIT.QUEUE의 메시지를 MQTT 클라이언트에 전달합니다.

텔레메트리(MQXR) 서비스는 `ClientIdentifier`를 큐 관리자 이름으로서 사용하여 전송 큐를 엽니다. 텔레메트리(MQXR) 서비스는 클라이언트 구독과 일치하는 발행물을 전달하기 위해 큐의 오브젝트 핸들을 MQSUB 호출로 전달합니다. 오브젝트 이름 확인에서는 `ClientIdentifier`가 리모트 큐 관리자 이름으로 작성되므로 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 해석해야 합니다. 표준 IBM MQ 오브젝트 이름 해석을 사용할 때, `ClientIdentifier`는 다음과 같이 해석됩니다. [313 페이지의 표 16](#)의 내용을 참조하십시오.

1. `ClientIdentifier`가 어떤 것과도 일치하지 않습니다.

*ClientIdentifier*는 리모트 큐 관리자 이름입니다. 로컬 큐 관리자 이름, 큐 관리자 알리어스 또는 전송 큐 이름과 일치하지 않습니다.

큐 이름이 정의되지 않습니다. 현재 텔레메트리(MQXR) 서비스에서는 SYSTEM.MQTT.PUBLICATION.QUEUE가 큐 이름으로 설정됩니다. MQTT v3 클라이언트는 큐를 지원하지 않습니다. 따라서 해석된 큐 이름이 클라이언트에 의해 무시됩니다.

발행이 SYSTEM.MQTT.TRANSMIT.QUEUE에 넣어서 클라이언트로 보내질 수 있도록 로컬 큐 관리자 특성 Default transmission queue의 이름은 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정되어야 합니다.

2. *ClientIdentifier*는 이름이 *ClientIdentifier*인 큐 관리자 알리어스와 일치합니다.

*ClientIdentifier*는 리모트 큐 관리자 이름입니다. 이는 큐 관리자 알리어스의 이름과 일치합니다.

큐 관리자 알리어스는 *ClientIdentifier*와 함께 리모트 큐 관리자 이름으로 정의해야 합니다.

큐 관리자 알리어스 정의에 전송 큐 이름을 설정하면 기본 전송을 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정할 필요가 없습니다.

	입력		Output		
	큐 관리자 이름	큐 이름	큐 관리자 이름	큐 이름	전송 큐
일치 항목 없음	<i>ClientIdentifier</i>	<i>undefined</i>	<i>ClientIdentifier</i>	<i>undefined</i>	기본 전송 큐. SYSTEM.MQTT.TRANSMIT.QUEUE
<i>ClientIdentifier</i> 라는 큐 관리자 알리어스와 일치합니다.	<i>ClientIdentifier</i>	<i>undefined</i>	<i>ClientIdentifier</i>	<i>undefined</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

이름 해석에 대한 추가 정보는 [이름 해석](#)을 참조하십시오.

IBM MQ 프로그램은 동일한 토픽으로 발행될 수 있습니다. 실행은 토픽에 구독을 하는 MQTT v3 클라이언트를 포함하여 해당 구독자에게 보내집니다.

예를 들어, 관리 토픽이 CLUSTER(*clusterName*) 속성이 있는 클러스터에서 작성되면 클러스터의 모든 애플리케이션이 클라이언트에 발행할 수 있습니다.

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

참고: SYSTEM.MQTT.TRANSMIT.QUEUE에 클러스터 속성을 부여하지 마십시오.

MQTT 클라이언트 구독자 및 발행자는 다른 큐 관리자에 연결될 수 있습니다. 구독자 및 발행자는 일부 동일한 클러스터에 포함되거나 발행/구독 계층을 통해 연결될 수 있습니다. 발행은 IBM MQ를 사용하여 발행자에서 구독자로 전달됩니다.

클라이언트에 직접 메시지 송신

구독을 작성하고 구독 토픽과 일치하는 실행을 수신하는 클라이언트에 대한 대안은 직접적으로 메시지를 MQTT v3 클라이언트에 보냅니다. MQTT v3 클라이언트 애플리케이션은 직접적으로 메시지를 보낼 수 없지만, IBM MQ 애플리케이션과 같은 다른 애플리케이션은 보낼 수 있습니다.

IBM MQ 애플리케이션은 MQTT v3 클라이언트의 *ClientIdentifier* 를 알아야 합니다. MQTT v3 클라이언트가 큐를 가지고 있지 않으므로, 대상 큐 이름은 MQTT v3 애플리케이션 클라이언트 *messageArrived* 메소드

에 토픽 이름으로 전달됩니다. 예를 들어, MQI 프로그램에서 클라이언트가 포함된 오브젝트 디스크립터를 ObjectQmgrName으로 작성하십시오.

```
MQOD.ObjectQmgrName = ClientIdentifier ;
MQOD.ObjectName = name ;
```

애플리케이션이 JMS를 사용하여 작성되면, 포인트-투-포인트 목적지를 작성하십시오. 예:

```
JM 3.0
jakarta.jms.Destination jmsDestination =
(jakarta.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

```
JMS 2.0
javax.jms.Destination jmsDestination =
(javax.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifier/name");
```

비청구 메시지를 MQTT 클라이언트에 보내려면 리모트 큐 정의를 사용하십시오. 리모트 큐 관리자 이름은 클라이언트의 ClientIdentifier로 해석되어야 합니다. 전송 큐는 SYSTEM.MQTT.TRANSMIT.QUEUE로 해석되어야 합니다. 자세한 정보는 314 페이지의 표 17의 내용을 참조하십시오. 리모트 큐 이름은 어떤 것이든 가능합니다. 클라이언트는 이 이름을 토픽 문자열로 수신합니다.

표 17. MQTT 클라이언트 리모트 큐 정의의 이름 해석				
입력		Output		
큐 이름	큐 관리자 이름	큐 이름	큐 관리자 이름	전송 큐
리모트 큐 정의의 이름	공백 또는 로컬 큐 관리자 이름	토픽 문자열로 사용되는 리모트 큐 이름	ClientIdentifier	SYSTEM.MQTT.TRANSMIT.QUEUE

클라이언트가 연결된 경우 메시지가 MQTT 클라이언트에 직접 전송되며, 이는 messageArrived 메소드를 호출합니다. messageArrived 메소드를 참조하십시오.

클라이언트가 지속적 세션과 연결이 끊어진 경우 메시지는 SYSTEM.MQTT.TRANSMIT.QUEUE 에 저장됩니다. MQTT Stateless 및 Stateful 세션을 참조하십시오. 클라이언트가 세션에 다시 연결할 때 클라이언트로 전달됩니다.

비지속 메시지를 보내는 경우 해당 메시지는 최대 한 번 서비스 품질(QoS=0)을 사용하여 클라이언트에 전송됩니다. 지속 메시지를 클라이언트에 직접 보내는 경우 해당 메시지는 기본적으로 정확히 한 번 서비스 품질(QoS=2)을 사용하여 전송됩니다. 클라이언트에 지속 메커니즘이 없을 수 있기 때문에, 클라이언트는 직접적으로 보내진 메시지를 위해 승인하는 서비스 품질(QoS)을 줄일 수 있습니다. 클라이언트로 직접 보내지는 메시지를 위해 QoS(Quality of Service)를 줄이려면 토픽 DEFAULT.QoS에 대한 구독을 작성하십시오. 클라이언트가 지원할 수 있는 최대 서비스 품질(QoS)을 지정하십시오.

Windows Linux AIX MQTT 클라이언트 식별, 권한 부여, 인증

텔레메트리 (MQXR) 서비스는 MQTT 채널을 사용하여 MQTT 클라이언트 대신 IBM MQ 토픽을 발행하거나 구독합니다. IBM MQ 관리자는 IBM MQ 권한 부여에 사용되는 MQTT 채널 ID를 구성합니다. 관리자는 채널에 대한 공통 ID를 정의하거나 채널에 연결된 클라이언트의 Username 또는 ClientIdentifier를 사용할 수 있습니다.

텔레메트리(MQXR) 서비스는 클라이언트 인증서를 사용하거나 클라이언트가 제공하는 Username을 사용하여 클라이언트를 인증할 수 있습니다. Username은 클라이언트에서 제공하는 비밀번호를 사용하여 인증됩니다.

요약: 클라이언트 ID는 클라이언트 ID의 선택입니다. 컨텍스트에 따라, 클라이언트는 ClientIdentifier, Username, 관리자에서 작성하는 공통 클라이언트 ID 또는 클라이언트 인증서에 의해 식별됩니다. 인증 검사에 사용되는 클라이언트 ID는 권한 부여에 사용되는 ID와 동일할 필요는 없습니다.

MQTT 클라이언트 프로그램은 MQTT 채널을 사용하여 서버에 전송되는 사용자 이름 및 비밀번호를 설정합니다. 이 프로그램은 연결을 인증하고 암호화하는 데 필요한 TLS 특성을 설정할 수도 있습니다. 관리자는 MQTT 채널을 인증하는지 여부와 채널을 인증하는 방법을 결정합니다.

MQTT 클라이언트에 IBM MQ 오브젝트에 액세스할 수 있는 권한을 부여하려면 클라이언트의 ClientIdentifier 또는 Username에 권한을 부여하거나 공통 클라이언트 ID에 권한을 부여하십시오. 클라이언트가 IBM MQ에 연결되도록 허용하려면 Username에 권한을 부여하거나 클라이언트 인증서를 사용하십시오. Username을 인증하려면 JAAS를 구성하고 클라이언트 인증서를 인증하려면 TLS를 구성합니다.

클라이언트에서 Password를 설정하면, VPN을 사용하여 연결을 암호화하거나 TLS를 사용할 MQTT 채널을 구성하여 비밀번호를 개인용으로 유지하십시오.

클라이언트 인증서는 관리하기 어렵습니다. 이러한 이유로, 비밀번호 인증과 연관되는 위험을 감수할 수 있는 경우, 비밀번호 인증이 종종 클라이언트를 인증하기 위해 사용됩니다.

클라이언트 인증서를 관리하고 저장하기 위한 안전한 방법이 있는 경우, 인증서 인증에 의존할 수 있습니다. 그러나, 텔레메트리가 사용되는 환경의 유형에서 인증서가 안전하게 관리될 수 있는 경우는 드뭅니다. 대신, 클라이언트 인증서를 사용하는 디바이스의 인증이 서버에서 클라이언트 비밀번호를 인증하여 보완됩니다. 클라이언트 인증서 사용은 더 복잡하므로 매우 민감한 애플리케이션에서만 사용됩니다. 두 가지 형식의 인증 사용을 두 요소 인증이라고 합니다. 한 가지 요소(예: 비밀번호)를 알고 있고 다른 요소(예: 인증서)가 있어야 합니다.

매우 민감한 애플리케이션(예: 칩 앤 핀 디바이스)에서 디바이스는 제조할 때 내부 하드웨어와 소프트웨어가 도용되지 않도록 잠겨집니다. 신뢰할 수 있는 시간 제한된 클라이언트 인증서가 디바이스로 복사됩니다. 디바이스는 사용할 수 있는 위치에 배치됩니다. 디바이스를 사용할 때마다 비밀번호 또는 스마트 카드의 다른 인증서가 사용되어 추가적인 인증이 수행됩니다.

Windows Linux AIX MQTT 클라이언트 ID 및 권한

권한 부여에 대한 공통 클라이언트 ID 및 클라이언트 ID(Username)를 사용하여 IBM MQ 오브젝트에 액세스하십시오.

IBM MQ 관리자에게는 MQTT 채널의 ID를 선택하기 위한 세 가지 선택사항이 있습니다. 관리자는 클라이언트가 사용하는 MQTT 채널을 정의하거나 수정할 때 선택합니다. ID는 IBM MQ 토픽에 액세스 권한을 부여하기 위해 사용됩니다. 다음 순서로 선택됩니다.

1. 클라이언트 ID([USECLNTID](#) 참조).
2. 관리자가 채널에 제공하는 ID(채널의 MCAUSER. [MCAUSER](#) 참조).
3. 이전 선택사항 중 어느 것도 적용되지 않는 경우, MQTT 클라이언트에서 전달된 사용자 이름 (Username은 MqttConnectOptions 클래스의 속성입니다. 클라이언트가 서비스에 연결되기 전에 설정되어야 합니다. 해당 기본값은 NULL입니다).

문제점 예방: DISPLAY CHSTATUS(MQTT)에 의한 예가 클라이언트의 MCAUSER로서 명령을 내리기 때문에 이 프로세스에 의해 선택된 ID는 그 후에 언급됩니다. 이 ID는 선택 (2)에서 참조되는 채널의 MCAUSER와 반드시 동일한 ID는 아닙니다.

IBM MQ `setmqaut` 명령을 사용하여 MQTT 채널과 연관되는 ID에서 사용하도록 권한 부여되는 오브젝트 및 조치를 선택하십시오. 예를 들어, 다음 코드는 큐 관리자 QM1의 관리자가 제공하는 채널 ID MQTTClient에 권한을 부여합니다.

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

Windows Linux AIX MQTT 클라이언트가 IBM MQ 오브젝트에 액세스할 수 있도록 권한 부여

MQTT 클라이언트에게 IBM MQ 오브젝트를 발행하고 구독할 수 있는 권한을 부여하려면 다음 단계를 수행하십시오. 단계는 네 개의 대체 액세스 제어 패턴을 따릅니다.

시작하기 전에

MQTT 클라이언트는 텔레메트리 채널에 연결할 때 ID를 지정하여 IBM MQ에서 오브젝트에 액세스하기 위한 권한이 부여됩니다. IBM MQ 관리자는 IBM MQ 탐색기를 사용하여 세 가지 유형의 ID 중 하나의 클라이언트에 제공하는 텔레메트리 채널을 구성합니다.

1. ClientIdentifier
2. 사용자 이름
3. 관리자가 채널에 지정하는 이름

사용되는 유형이 어떤 것이든 ID는 설치된 권한 서비스에 의해 프린시펄로서 IBM MQ에 정의되어야 합니다. Windows 또는 Linux에 대한 기본 권한 서비스는 QAM(Object Authority Manager)이라고 합니다. OAM을 사용 중인 경우, ID는 사용자 ID로 정의되어야 합니다.

이 ID를 사용하여 IBM MQ에 정의된 토픽으로 발행하거나 구독하기 위한 권한을 클라이언트 또는 클라이언트 콜렉션에 제공하십시오. MQTT 클라이언트가 토픽을 구독하는 경우, ID를 사용하여 결과 발행물을 수신하기 위한 권한을 제공하십시오.

각각이 개별 액세스 권한을 요청하는 수만의 MQTT 클라이언트로 시스템을 관리하는 것은 어렵습니다. 하나의 솔루션은 공용 ID를 정의하는 것이고 공용 ID 중 하나와 개별 MQTT 클라이언트를 연관시키는 것입니다. 권한의 다른 결합을 정의하기 위해 필요한 수만큼의 공용 ID를 정의하십시오. 다른 솔루션은 수천명의 사용자가 운영 체제보다 더 용이하게 처리할 수 있는 자체 권한 서비스를 작성하는 것입니다.

OAM을 사용하여, 두 가지 방법으로 MQTT 클라이언트를 공용 ID에 결합시킬 수 있습니다.

1. 관리자가 IBM MQ 탐색기를 사용하여 할당하는 다른 사용자 ID를 가진 각각의 다중 텔레메트리 채널을 정의하십시오. 다른 TCP/IP 포트 번호를 사용하여 연결하는 클라이언트는 다른 텔레메트리 채널과 연관되며 다른 ID가 지정됩니다.
2. 단일 텔레메트리 채널을 정의하지만, 각 클라이언트가 작은 사용자 ID 세트에서 Username을 선택하도록 하십시오. 관리자는 해당 ID로서 클라이언트 Username을 선택하도록 텔레메트리 채널을 구성합니다.

이 태스크에서 Telemetry 채널의 ID는 설정 방식에 관계없이 *mqttUser*라고 합니다. 클라이언트 콜렉션이 서로 다른 ID를 사용할 경우 클라이언트 콜렉션별로 하나씩 여러 *mqttUsers*를 사용하십시오. 태스크에서 OAM이 사용되는 경우 각 *mqttUser*가 사용자 ID여야 합니다.

이 태스크 정보

이 태스크에서 특정 요구사항에 따라 네 가지 액세스 제어 패턴 중 하나를 선택할 수 있습니다. 패턴은 액세스 제어의 단위에 따라 다릅니다.

- [316 페이지의 『액세스 제어 없음』](#)
- [316 페이지의 『거친 액세스 제어』](#)
- [317 페이지의 『중간 단위 액세스 제어』](#)
- [317 페이지의 『세밀한 액세스 제어』](#)

모델의 결과는 IBM MQ에 발행 및 구독할 *mqttUsers* 권한 세트를 지정하고 IBM MQ에서 발행물을 수신하는 것입니다.

액세스 제어 없음

MQTT 클라이언트는 IBM MQ 관리 권한을 받고 오브젝트에서 조치를 수행할 수 있습니다.

프로시저

1. 모든 MQTT 클라이언트의 ID로 사용할 사용자 ID *mqttUser* 를 작성하십시오.
2. *mqttUser*를 *mqm* 그룹에 추가하십시오. [Windows에서 그룹에 사용자 추가](#) 또는 [Linux에서 그룹 작성 및 관리를 참조](#)하십시오.

거친 액세스 제어

MQTT 클라이언트에 메시지를 발행 및 구독하고 이를 MQTT 클라이언트에 보내기 위한 권한이 있습니다. 기타 조치를 수행하거나 기타 오브젝트에 액세스하기 위한 권한이 없습니다.

프로시저

1. 모든 MQTT 클라이언트의 ID로 사용할 사용자 ID *mqttUser* 를 작성하십시오.
2. 모든 토픽을 발행하고 구독하며 MQTT 클라이언트에 발행물을 송신할 수 있도록 *mqttUser* 에 권한을 부여하십시오.

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

중간 단위 액세스 제어

MQTT 클라이언트는 토픽의 다른 세트에 발행 및 구독하고 메시지를 MQTT 클라이언트에 보내기 위해 다양한 그룹으로 나뉘어집니다.

프로시저

1. 발행/구독 토픽 트리에서 다중 사용자 ID(*mqttUsers*)와 다중 관리 토픽을 작성하십시오.
2. 다양한 *mqttUsers*에게 서로 다른 토픽에 대한 권한을 부여하십시오.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. 그룹 *mqtt*를 작성한 후 해당 그룹에 모든 *mqttUsers*를 추가하십시오.
4. MQTT 클라이언트에 토픽을 보낼 수 있도록 *mqtt* 에 권한을 부여하십시오.

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

세밀한 액세스 제어

MQTT 클라이언트는 오브젝트에서 조치를 수행하기 위한 그룹에 권한을 부여하는 액세스 제어의 기존 시스템으로 통합됩니다.

이 태스크 정보

사용자 ID가 필요한 권한 부여에 따라 하나 이상의 운영 체제 그룹으로 지정됩니다. IBM MQ 애플리케이션이 MQTT 클라이언트와 동일한 토픽 공간을 발행하고 구독하는 경우, 이 모델을 사용하십시오. 그룹은 Publish X, Subscribe Y, mqtt로 참조됩니다.

Publish X

Publish X 그룹의 멤버가 *topicX*에 발행될 수 있습니다.

Subscribe Y

Subscribe Y 그룹의 멤버는 *topicY*를 구독할 수 있습니다.

mqtt

mqtt 그룹의 멤버는 MQTT 클라이언트에 발행물을 보낼 수 있습니다.

프로시저

1. 발행/구독 토픽 트리에서 다중 관리 토픽에 할당된 다중 그룹, Publish X 및 Subscribe Y 를 작성하십시오.
2. 그룹 *mqtt*를 작성하십시오.
3. 부여되는 권한에 따라 여러 사용자 ID *mqttUsers*를 작성하고 해당 사용자를 임의의 그룹에 추가하십시오.
4. 다른 Publish X 및 Subscribe X 그룹에 다른 토픽에 대한 권한을 부여하고 MQTT 클라이언트에 메시지를 전송할 수 있도록 *mqtt* 그룹에 권한을 부여하십시오.

```
setmqaut -m qMgr -t topic -n topic1 -p Publish X -all +pub
```

```
setmqaut -m qMgr -t topic -n topic1 -p Subscribe X -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Windows

Linux

AIX

비밀번호를 사용하여 MQTT 클라이언트 인증

클라이언트 비밀번호를 사용하여 Username을 인증하십시오. 클라이언트에 권한을 부여하기 위해 사용되는 다른 ID를 사용하여 토픽에 발행하고 구독하는 클라이언트를 인증할 수 있습니다.

텔레메트리(MQXR) 서비스는 JAAS를 사용하여 클라이언트 Username을 인증합니다. JAAS 는 MQTT 클라이언트에서 제공하는 비밀번호를 사용합니다.

IBM MQ 관리자는 클라이언트가 연결되는 MQTT 채널을 구성하여 사용자 이름을 인증할지 또는 인증하지 않을지 여부를 결정합니다. 클라이언트는 다른 채널에 지정될 수 있고 각 채널은 다른 방법으로 해당 클라이언트를 인증하기 위해 구성될 수 있습니다. JAAS를 사용하여 클라이언트를 인증해야 하는 메소드 및 클라이언트를 선택적으로 인증할 수 있는 메소드를 구성할 수 있습니다.

인증을 위한 ID 선택은 권한 부여 ID 선택에 영향을 미치지 않습니다. 관리상의 편의를 위해 권한 부여를 위한 공통 ID를 설정할 수 있지만 해당 ID를 사용하도록 사용자별로 인증하십시오. 다음 프로시저는 개별 사용자가 공용 ID를 사용하도록 인증하는 단계를 파악합니다.

1. IBM MQ 관리자는 IBM MQ 탐색기를 사용하여 MQTT 채널 ID를 임의의 이름 (예: MQTTClientUser) 으로 설정합니다.
2. IBM MQ 관리자는 토픽을 발행하고 구독하는 권한을 MQTTClient에 부여합니다.

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. MQTT 클라이언트 애플리케이션 개발자는 서버에 연결하기 전에 MqttConnectOptions 오브젝트를 작성하고 사용자 이름 및 비밀번호를 설정합니다.
4. 보안 개발자는 비밀번호를 가지는 사용자 이름에 권한을 부여하기 위한 JAAS LoginModule을 작성하고 JAAS 구성 파일에 이를 포함시킵니다.
5. IBM MQ 관리자는 JAAS를 사용하여 클라이언트의 UserName 을 인증하도록 MQTT 채널을 구성합니다.

Windows

Linux

AIX

TLS를 사용하여 MQTT 클라이언트 인증

MQTT 클라이언트와 큐 관리자 사이의 연결은 항상 MQTT 클라이언트가 시작합니다. MQTT 클라이언트는 항상 SSL 클라이언트입니다. 서버의 클라이언트 인증 및 MQTT 클라이언트의 서버 인증은 모두 선택사항입니다.

클라이언트에 개인 서명 디지털 인증서를 제공하면 MQTT 클라이언트를 인증하여 다음을 수행할 수 있습니다. IBM MQ . IBM MQ 관리자는 MQTT 클라이언트가 TLS를 사용하여 큐 관리자에 대해 자신을 인증하도록 강제 실행할 수 있습니다. 클라이언트를 상호 인증의 부분으로만 요청할 수 있습니다.

SSL 사용의 대안으로서 IPsec 등과 같은 일종의 가상 사설망(VPN)은 TCP/IP 연결의 엔드 포인트를 인증합니다. VPN은 네트워크를 통해 플로우는 각 IP 패킷을 암호화합니다. 이러한 VPN 연결이 설정되면 신뢰할 수 있는 네트워크가 설정된 것입니다. VPN 네트워크를 통해 TCP/IP를 사용하여 MQTT 클라이언트를 임시 채널에 연결할 수 있습니다.

TLS를 사용하는 클라이언트 인증은 본인확인정보가 있는 클라이언트에 의존합니다. 본인확인정보는 자체 서명 인증서의 경우에 클라이언트의 개인 키이거나 인증 기관이 제공하는 키입니다. 키는 클라이언트의 디지털 인증서에 서명하는 데 사용됩니다. 해당하는 공개 키를 가진 사용자는 누구나 디지털 인증서를 확인할 수 있습니다. 인증서는 신뢰할 수 있거나, 체인 연결된 경우 인증서 체인을 통해 신뢰되는 루트 인증서로 추적될 수 있습니다. 클라이언트 확인은 클라이언트가 제공한 인증서 체인에 있는 모든 인증서를 서버로 보냅니다. 서버는 신뢰하는 인증서를 찾을 때까지 인증서 체인을 검사합니다. 신뢰되는 인증서는 자체 서명 인증서에서 생성된 공용 인증서이거나 일반적으로 인증 기관이 발행한 루트 인증서입니다. 마지막으로, 신뢰되는 인증서를 "라이브" 인증서 패키지 목록과 비교할 수 있는 선택적 단계가 있습니다.

신뢰되는 인증서는 인증 기관이 발행하거나 JRE 인증서 저장소에 이미 포함되었을 수 있습니다. 이는 자체 서명 인증서이거나 신뢰되는 인증서로서 텔레메트리 채널 키 저장소에 추가된 인증서일 수도 있습니다.

참고: 텔레메트리 채널에는 하나 이상의 텔레메트리 채널 및 클라이언트를 인증하는 데 필요한 모든 공용 인증서들 모두를 보유하는 결합된 키 저장소/신뢰 저장소가 있습니다. SSL 채널에는 키 저장소가 있어야 하고 채널 신뢰

저장소와 동일한 파일이므로 JRE 인증서 저장소는 절대 참조되지 않습니다. 클라이언트의 인증에 CA 루트 인증서가 필요한 경우 CA 루트 인증서가 이미 JRE 인증서 저장소에 있더라도 채널의 키 저장소에 루트 인증서를 배치해야 한다는 의미입니다. JRE 인증서 저장소는 절대 참조되지 않습니다.

클라이언트 인증이 카운터하려는 위협과 클라이언트 및 서버가 위협 카운터링 시에 수행하는 역할에 대해 생각하십시오. 클라이언트 인증서만을 인증하는 것은 시스템에 비인가 액세스를 방지하기에는 충분하지 않습니다. 다른 누군가가 클라이언트 디바이스를 보유한 경우 클라이언트 디바이스는 인증서 홀더의 권한을 사용하여 수행할 필요가 없습니다. 원치 않는 공격에 대응하는 방법으로 한 가지 방어책에만 의존하지 마십시오. 적어도 두 요소 인증 방법을 사용하고 개인용 정보에 대한 지식이 있는 인증서를 보조로 소유하십시오. 예를 들어, JAAS를 사용하여 서버가 발행한 비밀번호를 사용하여 클라이언트를 인증하십시오.

클라이언트 인증서에 대한 1차 위협은 이것이 올바르지 않은 사용자의 수중에 들어간다는 점입니다. 인증서는 클라이언트에서 비밀번호로 보호된 키 저장소에 보관됩니다. 이를 키 저장소에 배치하는 방법은 무엇입니까? MQTT 클라이언트가 키 저장소에 비밀번호를 가져오는 방법은 무엇입니까? 비밀번호 보호는 얼마나 안전합니까? 텔레메트리 디바이스는 종종 제거하기 쉬우므로 개인적으로 해킹될 수 있습니다. 디바이스 하드웨어가 변경할 수 없도록 설정되어야 합니까? 클라이언트측 인증서를 분배하고 보호하는 일은 인식하기가 어렵습니다. 이는 키 관리 문제점이라고 불립니다.

2차 위협은 디바이스가 의도하지 않은 방법으로 서버에 액세스하기 위해 잘못 사용되는 것입니다. 예를 들어, MQTT 애플리케이션이 변조되는 경우, 인증된 클라이언트 ID를 사용하여 서버 구성에서 약점을 이용할 수 있습니다.

SSL을 사용하여 MQTT 클라이언트를 인증하려면 텔레메트리 채널을 구성한 다음 클라이언트를 구성하십시오.

관련 개념

319 페이지의 『TLS를 사용하는 MQTT 클라이언트 인증의 텔레메트리 채널 구성』

그만큼 IBM MQ 관리자는 서버에서 원격 측정 채널을 구성합니다. 각 채널은 서로 다른 포트 번호에서 TCP/IP 연결을 승인하도록 구성됩니다. TLS 채널은 키 파일에 대한 액세스를 비밀번호 문구로 보호하여 구성됩니다. 비밀번호 문구 또는 키 파일 없이 TLS 채널이 정의된 경우 이 채널은 TLS 연결을 승인하지 않습니다.

[TLS를 사용하여 클라이언트 인증을 위한 MQTT 클라이언트 구성](#)

TLS를 사용하는 MQTT 클라이언트 인증의 텔레메트리 채널 구성

그만큼 IBM MQ 관리자는 서버에서 원격 측정 채널을 구성합니다. 각 채널은 서로 다른 포트 번호에서 TCP/IP 연결을 승인하도록 구성됩니다. TLS 채널은 키 파일에 대한 액세스를 비밀번호 문구로 보호하여 구성됩니다. 비밀번호 문구 또는 키 파일 없이 TLS 채널이 정의된 경우 이 채널은 TLS 연결을 승인하지 않습니다.

TLS 텔레메트리 채널의 `com.ibm.mq.MQTT.ClientAuth` 특성을 `REQUIRED(으)`로 설정하여 해당 채널에 연결하는 모든 클라이언트가 디지털 인증서를 검증했다는 증거를 제공하도록 강제 실행하십시오. 클라이언트 인증서는 인증 기관의 인증서를 사용하여 인증되며 신뢰되는 루트 인증서로 안내됩니다. 클라이언트 인증서가 자체 서명되었거나 인증 기관에서 발급한 인증서에 의해 서명된 경우 공개적으로 서명된 클라이언트 인증서나 인증 기관이 서버에 안전하게 보관되어야 합니다.

텔레메트리 채널 키 저장소에 공용으로 서명된 클라이언트 인증서 또는 인증 기관의 인증서를 배치하십시오. 서버에서 공개적으로 서명된 인증서는 별도의 신뢰 저장소가 아니라 개인적으로 서명된 인증서로 동일한 키 파일에 저장됩니다.

서버는 모든 공용 인증서와 서버에서 보유한 암호 스위트를 사용하여 전송되는 모든 클라이언트 인증서의 서명을 확인합니다. 서버는 키 체인을 확인합니다. 큐 관리자는 인증서 폐기 목록(CRL)에 대응하는 인증서를 테스트하도록 구성될 수 있습니다. 큐 관리자 폐기 이름 목록 특성은 `SSLCRLNL`입니다.

클라이언트가 보낸 모든 인증서가 서버 키 저장소에 있는 인증서에 의해 확인되면 클라이언트가 인증됩니다.

IBM MQ 관리자는 JAAS를 사용하여 클라이언트 Password로 클라이언트의 Username 또는 ClientIdentifier를 확인하기 위해 동일한 텔레메트리 채널을 구성할 수 있습니다.

다중 텔레메트리 채널에 동일한 키 저장소를 사용할 수 있습니다.

디바이스에서 비밀번호로 보호되는 클라이언트 키 저장소에 있는 최소 하나의 디지털 인증서를 확인하면 클라이언트가 서버에 대해 인증됩니다. 디지털 인증서는 IBM MQ에 의한 인증에만 사용됩니다. 클라이언트의 TCP/IP 주소를 검증하거나, 권한 부여 또는 회계를 위한 클라이언트 ID를 설정하는 데에는 사용되지 않습니다. 서버에

의해 채택된 클라이언트의 ID는 Username 또는 클라이언트의 ClientIdentifier 또는 IBM MQ 관리자에 의해 작성된 ID입니다.

클라이언트 인증에 TLS 암호 스위트를 사용할 수도 있습니다. SHA-2 암호 스위트를 사용하려는 경우 323 페이지의 『MQTT 채널과 함께 SHA-2 암호 스위트를 사용하기 위한 시스템 요구사항』의 내용을 참조하십시오.

관련 개념

320 페이지의 『TLS를 사용하는 채널 인증의 텔레메트리 채널 구성』

그만큼 IBM MQ 관리자는 서버에서 원격 측정 채널을 구성합니다. 각 채널은 서로 다른 포트 번호에서 TCP/IP 연결을 승인하도록 구성됩니다. TLS 채널은 키 파일에 대한 액세스를 비밀번호 문구로 보호하여 구성됩니다. 비밀번호 문구 또는 키 파일 없이 TLS 채널이 정의된 경우 이 채널은 TLS 연결을 승인하지 않습니다.

CipherSpecs 및 CipherSuites

관련 참조

DEFINE CHANNEL(MQTT)

ALTER CHANNEL(MQTT)

Windows

Linux

AIX

TLS를 사용하여 텔레메트리 채널 인증

MQTT 클라이언트와 큐 관리자 사이의 연결은 항상 MQTT 클라이언트가 시작합니다. MQTT 클라이언트는 항상 SSL 클라이언트입니다. 서버의 클라이언트 인증 및 MQTT 클라이언트의 서버 인증은 모두 선택사항입니다.

클라이언트가 익명 연결을 지원하는 CipherSpec을 사용하도록 구성되지 않는 한 클라이언트는 항상 서버를 인증하려고 시도합니다. 인증에 실패하면 연결이 설정되지 않습니다.

SSL 사용의 대안으로서 IPsec 등과 같은 일종의 가상 사설망(VPN)은 TCP/IP 연결의 엔드 포인트를 인증합니다. VPN은 네트워크를 통해 플로우하는 각 IP 패킷을 암호화합니다. 이러한 VPN 연결이 설정되면 신뢰할 수 있는 네트워크가 설정된 것입니다. VPN 네트워크를 통해 TCP/IP를 사용하여 MQTT 클라이언트를 임시 채널에 연결할 수 있습니다.

SSL을 사용한 서버 인증을 통해 기밀 정보를 보낼 서버를 인증합니다. 클라이언트는 서버에서 전송된 인증서와 신뢰 저장소 또는 JRE에 있는 인증서가 일치하는지 확인합니다.cacerts 가계.

JRE 인증서 저장소는 JKS 파일입니다.cacerts. 이 파일은 JRE InstallPath\lib\security\에 있습니다. 이 파일은 changeit라는 기본 비밀번호로 설치됩니다. JRE 인증서 저장소에서 신뢰할 수 있는 인증서나 클라이언트 신뢰 저장소에서 신뢰할 수 있는 인증서 중 하나를 저장할 수 있습니다. 두 저장소를 모두 사용할 수는 없습니다. 클라이언트가 신뢰하는 공개 인증서를 다른 인증서와 별도로 유지하려면 클라이언트 신뢰 저장소를 사용하십시오.Java 응용 프로그램 사용. 모든 항목에 대해 공통 인증서 저장소를 사용하려면 JRE 인증서 저장소를 사용하십시오.Java 클라이언트에서 실행되는 애플리케이션. JRE 인증서 저장소를 사용하기로 결정하면 해당 저장소에 속한 인증서를 검토하여 이러한 인증서를 신뢰할 수 있는지 확인합니다.

다른 신뢰 제공자를 통해 JSSE 구성을 수정할 수 있습니다. 신뢰 제공자를 사용자 정의하여 인증서에 대해 다른 검사를 수행할 수도 있습니다. MQTT 클라이언트를 사용한 일부 OGSi 환경에서 환경은 다른 신뢰 제공자를 제공합니다.

TLS를 사용하여 텔레메트리 채널을 인증하려면 서버를 구성한 다음 클라이언트를 구성하십시오.

Windows

Linux



AIX

TLS를 사용하는 채널 인증의 텔레메트리 채널 구성

그만큼 IBM MQ 관리자는 서버에서 원격 측정 채널을 구성합니다. 각 채널은 서로 다른 포트 번호에서 TCP/IP 연결을 승인하도록 구성됩니다. TLS 채널은 키 파일에 대한 액세스를 비밀번호 문구로 보호하여 구성됩니다. 비밀번호 문구 또는 키 파일 없이 TLS 채널이 정의된 경우 이 채널은 TLS 연결을 승인하지 않습니다.

텔레메트리 채널이 서버에 사용할 키 저장소에서 해당 개인 키로 서명되는 서버의 디지털 인증서를 저장하십시오. 키 체인을 클라이언트로 전송하려면 키 스토어의 키 체인에 인증서를 저장하십시오. TLS를 사용하려면 IBM MQ 탐색기를 사용하여 텔레메트리 채널을 구성하십시오. 이 채널은 키 저장소에 대한 경로 및 키 저장소에 액세스하는 데 필요한 비밀번호 문구와 함께 제공됩니다. 채널의 TCP/IP 포트 번호를 설정하지 않는 경우, TLS 텔레메트리 채널 포트 번호의 기본값은 8883입니다.

채널 인증에 TLS 암호 스위트를 사용할 수도 있습니다. SHA-2 암호 스위트를 사용하려는 경우 323 페이지의 『MQTT 채널과 함께 SHA-2 암호 스위트를 사용하기 위한 시스템 요구사항』의 내용을 참조하십시오.

중요사항:   IBM MQ 9.4.0부터 CMS 키 저장소 및 스택 파일은 SSL/TLS를 사용하는 AMQP 및 MQTT 채널에서 지원되지 않습니다. PKCS #12 키 저장소를 사용하고 대신 IBM MQ 비밀번호 보호 시스템을 사용하여 키 저장소 비밀번호를 보호하십시오. 다음 명령을 사용하여 PKCS #12 키 저장소를 작성할 수 있습니다.

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

이 명령은 지정된 비밀번호로 보안되는 `filename.p12` 라는 PKCS #12 키 저장소 파일을 작성합니다.

관련 개념

319 페이지의 『TLS를 사용하는 MQTT 클라이언트 인증의 텔레메트리 채널 구성』

그만큼 IBM MQ 관리자는 서버에서 원격 측정 채널을 구성합니다. 각 채널은 서로 다른 포트 번호에서 TCP/IP 연결을 승인하도록 구성됩니다. TLS 채널은 키 파일에 대한 액세스를 비밀번호 문구로 보호하여 구성됩니다. 비밀번호 문구 또는 키 파일 없이 TLS 채널이 정의된 경우 이 채널은 TLS 연결을 승인하지 않습니다.

[CipherSpecs 및 CipherSuites](#)

관련 참조

[DEFINE CHANNEL\(MQTT\)](#)

[ALTER CHANNEL\(MQTT\)](#)



TLS 인증을 사용하는 MQTT 채널 구성 예

이 예에서는 TLS 인증을 사용하는 MQTT 채널을 작성합니다. 합한거나 합습니다.

이 예제나 십시오 있가 필요기 때문에 MQTT 및 모기 간 채널을 구성합니다.

이 예제에서는 Red Hat Enterprise Linux 의 IBM MQ 및 CentOS의 모기용 Docker 컨테이너를 사용하지만 모든 유형의 서버에 적용됩니다. (레지스트리 인타이틀먼트로 인해 CentOS 가 모기에 사용되었습니다.)

단방향 TLS를 위한 IBM MQ 키 저장소 및 채널 구성

중요사항:   IBM MQ 9.4.0부터 CMS 키 저장소 및 스택 파일은 SSL/TLS를 사용하는 AMQP 및 MQTT 채널에서 지원되지 않습니다. PKCS #12 키 저장소를 사용하고 대신 IBM MQ 비밀번호 보호 시스템을 사용하여 키 저장소 비밀번호를 보호하십시오.

다음 단계를 완료하십시오.

1.   IBM MQ 키 저장소를 작성하십시오.



```
runmqakm -keydb -create -db mqtt.p12 -pw "passw0rd" -type p12
```

2.   개인 인증서를 작성하십시오.

```
runmqakm -cert -create -db mqtt.p12 -pw "passw0rd" -size 2048 -dn "CN= mqm, OU=MQTest, O=MQSuppor, C=US" -sig_alg SHA256_WITH_RSA -label ibmwebspheremqmqm
```

다음 명령을 사용하여 인증서 작성을 확인할 수 있습니다.

```
runmqakm -cert -list -v -db mqtt.p12 -pw "passw0rd"
```

3.   runmqsc 프롬프트에서 다음 명령을 입력하여 MQTT 채널을 작성하십시오.

```
DEFINE CHANNEL(MQTTDEMO) CHLTYPE(MQTT) BACKLOG(4096) PORT(8883) MCAUSER('mqm')  
PROTOCOL(MQTTV311,MQTTV3,HTTP) SSLCAUTH(OPTIONAL) SSLCIPH('SSL_RSA_WITH_AES_256_CBC_SHA256')  
SSLKEYR('/var/mqm/mqtt/mqtt.p12') SSLKEYP('passw0rd') TRPTYPE(TCP)
```

채널은 Java 암호 매핑을 사용합니다. [JMS용 IBM MQ 클래스의 TLS CipherSpecs 및 CipherSuites](#)의 내용을 참조하십시오.

4. 인증서를 추출하십시오.

```
runmqakm -cert -extract -db mqtt.kdb -stashed -label ibmwebspheremqmqm -target serverCert.pem
```

Docker 컨테이너의 CentOS 에 Mrto 설치

다음 단계를 완료하여 CentOS에서 실행 중인 모기가 있는 Docker 컨테이너를 작성하십시오.

1. `docker pull centos`
2. `docker run -it centos /bin/bash`
3. `yum -y install epel-release`
4. `yum -y install mosquito`

서명자 인증서를 모기로 이동

IBM MQ 에서 작성한 인증서를 모기로 이동하려면 다음 단계를 완료하십시오. 이러한 단계는 Docker 호스트 머신에서 실행됩니다.

1. Docker에서 컨테이너 ID를 보십시오.

```
docker container ls
```

2. Docker 컨테이너에서 로컬 시스템 Docker로 파일 복사

```
cp MQ_Container_ID:/var/mqm/mqtt/serverCert.pem serverCert.pem
```

3. 로컬 시스템에서 CentOS 시스템의 루트 디렉토리로 파일을 복사하십시오.

```
docker cp serverCert.pem CentOS_ContainerID:/serverCert.pem
```

모기를 사용하여 공개

다음 명령을 사용하여 모기에 테스트 메시지를 공개하십시오.

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d
```

명령 인수의 의미는 다음과 같습니다.

-h

Red Hat Enterprise Linux 호스트 IP 주소 (**nslookup**를 사용하여 찾을 수 있음).

-- cafile

서명자 인증서가 포함된 파일입니다.

-- 비보안

이 옵션은 예제에서 자체 서명된 인증서를 사용하기 때문에 지정됩니다. 실제 CA 인증서를 사용할 때 이 옵션을 사용하지 마십시오.

-p

포트 번호입니다.

-i

클라이언트 ID입니다.

-t

발행되는 토픽입니다.

-m

발행되는 메시지입니다.

-d

디버그 메시지를 사용 가능하게 하십시오

상호 TLS 인증을 위해 MQTT 채널 구성

다음 명령을 입력하여 MQTT 채널을 SSLCAUTH (REQUIRED) 로 재구성하십시오.

```
ALTER CHANNEL (MQTTDEMO) CHLTYPE (MQTT) SSLCAUTH (REQUIRED)
```

모기 서버에서 키/인증서 쌍을 작성하고 IBM MQ 에 추가하십시오.

다음 명령을 입력하여 모기에 키/인증서 쌍을 작성하십시오.

1. **openssl** 를 사용하여 다음과 같이 모기용 키/인증서 쌍을 작성하십시오.

```
openssl req -x509 -newkey rsa:4096 -keyout mosquitoKey.pem -out mosquitoCert.pem -subj "/"  
CN=Mosquitto"
```

2. 컨테이너의 컨테이너 ID를 나열하십시오.

```
docker container ls
```

3. 다음과 같이 모기를 인증서에 로컬 시스템 Docker에 복사하십시오.

```
docker cp CentOS_ContainerID:mosquittoCert.pem .
```

4. IBM MQ에 모기를 인증서에 복사하십시오.

```
docker cp mosquitoCert.pem MQ_Container_ID:/var/mqm/mqtt
```

5. IBM MQ 키 저장소에 인증서를 추가하십시오.

```
runmqakm -cert -add -db mqtt.kdb -stashed -file mosquitoCert.pem
```

6. MQTT 채널을 재시작하십시오.

모기를 사용하여 공개 및 상호 인증

상호 인증을 사용하여 모기와 함께 공개하려면 다음 단계를 완료하십시오.

1. 다음 명령은 테스트 메시지를 공개해야 합니다.

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t  
test -m 'test message' -d --cert mosquitoCert.pem --key mosquitoKey.pem
```

2. 다음 명령은 테스트 메시지를 공개하는 데 실패하고 오류 메시지를 생성해야 합니다. 이 명령은 개인 인증서를 모기에서 전송하지 않기 때문입니다.

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t  
test -m 'test message' -d /var/mqm/qmgrs/mqttDemoQM/errors/ mqxr_0.log
```

관련 정보

[키 및 인증서 관리](#)

Windows Linux AIX MQTT 채널과 함께 SHA-2 암호 스위트를 사용하기 위한 시스템 요구사항

SHA-2 암호 스위트를 지원하는 Java의 버전을 사용할 경우, 이러한 스위트를 사용하여 MQTT(텔레메트리) 채널 및 클라이언트 앱을 보안할 수 있습니다.

텔레메트리 (MQXR) 서비스를 포함하는 IBM MQ 8.0 의 경우 최소 Java 버전은 IBM 의 Java 7, SR6입니다. SHA-2 암호 스위트는 기본적으로 IBM, SR4 부터 Java 7에서 지원됩니다. 따라서 텔레메트리(MQXR) 서비스와 함께 SHA-2 암호 스위트를 사용하여 MQTT(텔레메트리) 채널을 보안할 수 있습니다.

다른 JRE와 함께 MQTT 클라이언트를 실행 중인 경우, SHA-2 암호 스위트 또한 지원되는지 확인해야 합니다.

관련 개념

텔레메트리(MQXR) 서비스

320 페이지의 『TLS를 사용하는 채널 인증의 텔레메트리 채널 구성』

그만큼 IBM MQ 관리자는 서버에서 원격 측정 채널을 구성합니다. 각 채널은 서로 다른 포트 번호에서 TCP/IP 연결을 승인하도록 구성됩니다. TLS 채널은 키 파일에 대한 액세스를 비밀번호 문구로 보호하여 구성됩니다. 비밀번호 문구 또는 키 파일 없이 TLS 채널이 정의된 경우 이 채널은 TLS 연결을 승인하지 않습니다.

관련 참조

DEFINE CHANNEL(MQTT)

ALTER CHANNEL(MQTT)

Windows

Linux

AIX

텔레메트리 채널의 발행물 개인정보 보호

텔레메트리 채널 전체에 걸쳐 각각 방향으로 보내진 MQTT 발행물의 개인정보는 연결 중 전송을 암호화하기 위해 TLS를 사용하여 보안 설정됩니다.

텔레메트리 채널에 연결되는 MQTT 클라이언트는 TLS를 사용하여 대칭 키 암호화를 사용하는 채널에서 전송된 발행물의 개인정보를 보호합니다. 엔드 포인트가 인증되지 않으므로 채널 암호화만 신뢰할 수는 없습니다. 개인정보 보호와 서버 또는 상호 인증을 결합하십시오.

SSL 사용의 대안으로서 IPsec 등과 같은 일종의 가상 사설망(VPN)은 TCP/IP 연결의 엔드 포인트를 인증합니다. VPN은 네트워크를 통해 플로우는 각 IP 패킷을 암호화합니다. 이러한 VPN 연결이 설정되면 신뢰할 수 있는 네트워크가 설정된 것입니다. VPN 네트워크를 통해 TCP/IP를 사용하여 MQTT 클라이언트를 임시 채널에 연결할 수 있습니다.

채널을 암호화하고 서버를 인증하는 일반 구성의 경우, 320 페이지의 『TLS를 사용하여 텔레메트리 채널 인증』의 내용을 참조하십시오.

서버를 인증하지 않고 TLS 연결을 암호화하면 연결이 중간자 공격(man-in-the-middle attack)에 노출됩니다. 교환하는 정보가 도청에 대해 보호되는 경우에도 누구와 교환하는지 알지 못합니다. 네트워크를 제어하지 않는 경우, IP 전송을 인터셉트하고 엔드 포인트로 위장하는 사람에게 노출됩니다.

서버를 인증하지 않고 익명 TLS을 지원하는 Diffie-Hellman 키 교환 CipherSpec을 사용하여 암호화된 TLS로 연결할 수 있습니다. 클라이언트와 서버 사이에 공유되고 TLS 전송을 암호화하는 데 사용되는 마스터 보안은 개인적으로 서명된 서버 인증서를 교환하지 않고 설정됩니다.

익명 연결은 안전하지 않으므로 대부분의 TLS 구현은 기본적으로 익명 CipherSpec을 사용하지 않습니다. 텔레메트리 채널에서 TLS 연결에 대한 클라이언트 요청이 승인되는 경우, 채널에는 비밀번호 문구로 보호되는 키 저장소가 있어야 합니다. 기본적으로 TLS 구현은 익명 CipherSpec을 사용하지 않으므로 키 저장소에는 클라이언트가 인증할 수 있는 개인적으로 서명된 인증서가 포함되어 있어야 합니다.

익명 CipherSpec을 사용하는 경우 서버 키 저장소가 있어야 하지만 개인적으로 서명된 인증서를 포함할 필요는 없습니다.

암호화된 연결을 설정하는 다른 방법은 클라이언트에서 신뢰 제공자를 사용자 고유의 구현으로 바꾸는 것입니다. 사용자의 신뢰 제공자는 서버 인증서를 인증하지 않지만 연결이 암호화됩니다.



주의: MQTT 에서 TLS를 사용하는 경우 대형 메시지를 사용할 수 있지만 이를 수행할 때 성능에 영향을 줄 수 있습니다. MQTT 는 작은 메시지 처리에 최적화되어 있습니다 (일반적으로 크기가 1KB - 1MB).

Windows

Linux

AIX

MQTT Java 클라이언트 및 텔레메트리 채널의 TLS 구성

성

MQTT Java 클라이언트와 텔레메트리 채널을 인증하기 위해 TLS를 구성하고 그 사이의 메시지 전송을 암호화하십시오. MQTT Java 클라이언트는 JSSE (Java Secure Socket Extension) 를 사용하여 TLS를 통해 텔레메트리 채널을 연결합니다. SSL 사용의 대안으로서 IPsec 등과 같은 일종의 가상 사설망(VPN)은 TCP/IP 연결의 엔드 포인트를 인증합니다. VPN은 네트워크를 통해 플로우는 각 IP 패킷을 암호화합니다. 이러한 VPN 연결이 설정되면 신뢰할 수 있는 네트워크가 설정된 것입니다. VPN 네트워크를 통해 TCP/IP를 사용하여 MQTT 클라이언트를 임시 채널에 연결할 수 있습니다.

TCP/IP를 통해 TLS 프로토콜을 사용하도록 Java MQTT 클라이언트와 텔레메트리 채널 사이의 연결을 구성할 수 있습니다. 보안이 유지되는 항목은 JSSE를 사용하도록 TLS를 구성하는 방법에 따라 다릅니다. 가장 안전한 구성에서 시작하여 세 가지 다른 보안 레벨을 구성할 수 있습니다.

1. 신뢰받는 MQTT 클라이언트만 연결되도록 하십시오. 신뢰받는 텔레메트리 채널에만 MQTT 클라이언트를 연결시키십시오. 클라이언트 및 큐 관리자 사이의 메시지를 암호화하십시오(318 페이지의 『TLS를 사용하여 MQTT 클라이언트 인증』 참조).
2. 신뢰받는 텔레메트리 채널에만 MQTT 클라이언트를 연결시키십시오. 클라이언트와 큐 관리자 사이에 메시지를 암호화하십시오. 320 페이지의 『TLS를 사용하여 텔레메트리 채널 인증』의 내용을 참조하십시오.
3. 클라이언트와 큐 관리자 사이에 메시지를 암호화하십시오. 324 페이지의 『텔레메트리 채널의 발행물 개인 정보 보호』의 내용을 참조하십시오.

JSSE 구성 매개변수

TLS 연결 구성 방법을 변경하려면 JSSE 매개변수를 수정하십시오. JSSE 구성 매개변수는 세 가지 세트로 배열됩니다.

1. [MQ Telemetry 채널](#)
2. [MQTT Java 클라이언트](#)
3. [JRE](#)

IBM MQ 탐색기를 사용하여 텔레메트리 채널 매개변수를 구성하십시오.

MqttConnectionOptions.SSLProperties 속성에서 MQTT Java 클라이언트 매개변수를 설정하십시오. 클라이언트 및 서버 모두의 JRE 보안 디렉토리에서 파일을 편집하여 JRE 보안 매개변수를 수정하십시오.

MQ Telemetry 채널

IBM MQ 탐색기를 사용하여 모든 텔레메트리 채널 TLS 매개변수를 설정하십시오.

ChannelName

ChannelName은 모든 채널의 필수 매개변수입니다.

채널 이름은 특정 포트 번호와 연관된 채널을 식별합니다. 사용자를 돕기 위한 이름 채널은 MQTT 클라이언트 세트를 관리합니다.

PortNumber

PortNumber는 모든 채널에서 선택적 매개변수입니다. TCP 채널의 경우 1883으로 기본 설정되며, TLS 채널의 경우 8883으로 기본 설정됩니다.

이 채널과 연관된 TCP/IP 포트 번호입니다. MQTT 클라이언트는 채널에 대해 정의된 포트를 지정하여 채널에 연결됩니다. 채널에 TLS 특성이 있는 경우, 클라이언트는 TLS 프로토콜을 사용하여 연결해야 합니다. 예를 들어, 다음과 같습니다.

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId1");
mqttClient.connect();
```

KeyFileName

KeyFileName은 TLS 채널의 필수 매개변수입니다. TCP 채널의 경우에는 생략되어야 합니다.

KeyFileName은 제공하는 디지털 인증서를 포함하는 Java 키 저장소에 대한 경로입니다. 서버에서 키 저장소의 유형으로 JKS, JCEKS 또는 PKCS12를 사용하십시오.

다음 파일 확장자 중 하나를 사용하여 키 저장소 유형을 식별하십시오.

- .jks
- .jceks
- .p12
- .pkcs12

다른 파일 확장자를 사용하는 키 저장소를 JKS 키 저장소라고 가정합니다.

서버의 키 저장소 유형과 클라이언트의 다른 키 저장소 유형을 결합할 수 있습니다.

키 저장소에 서버의 개인용 인증서를 저장하십시오. 이 인증서를 서버 인증서라고 합니다. 인증서는 자체 서명되거나 서명 기관에서 서명한 인증서 체인의 일부가 될 수 있습니다.

인증서 체인을 사용하는 경우에는 연관된 인증서를 서버 키 저장소에 저장하십시오.

서버 인증서 및 해당 인증서 체인의 모든 인증서는 클라이언트로 보내져 서버 ID를 인증합니다.

ClientAuth 를 Required로 설정한 경우 키 저장소에는 클라이언트를 인증하는 데 필요한 인증서가 포함되어 있어야 합니다. 클라이언트가 자체 서명 인증서 또는 인증서 체인을 송신하면 이를 키 저장소의 인증서와 대조 확인한 후 클라이언트가 인증됩니다. 인증서 체인을 사용하면 클라이언트가 다양한 클라이언트 인증서를 사용하여 실행되더라도 하나의 인증서로 많은 클라이언트를 확인할 수 있습니다.

PassPhrase

PassPhrase는 TLS 채널의 필수 매개변수입니다. TCP 채널의 경우에는 생략되어야 합니다.

비밀번호 문구는 키 저장소를 보호하기 위해 사용됩니다.

ClientAuth

ClientAuth는 선택적 TLS 매개변수입니다. 클라이언트 인증으로 기본값이 설정됩니다. TCP 채널의 경우에는 생략되어야 합니다.

텔레메트리(MQXR) 서비스에서 클라이언트를 인증하도록 하려면 ClientAuth를 설정한 후 클라이언트가 텔레메트리 채널에 연결하도록 허용하십시오.

ClientAuth를 설정하면 클라이언트는 TLS를 사용하여 서버에 연결하고 서버를 인증해야 합니다. ClientAuth 설정에 대한 응답으로 클라이언트는 디지털 인증서 및 키 저장소에 있는 다른 모든 인증서를 서버로 송신합니다. 이 디지털 인증서는 클라이언트 인증서라고 합니다. 이러한 인증서는 채널 키 저장소 및 JRE cacerts 저장소에 있는 인증서에 대응하여 인증됩니다.

CipherSuite

CipherSuite는 선택적 TLS 매개변수입니다. 기본값은 사용 가능한 모든 CipherSpec을 시도하는 것입니다. TCP 채널의 경우에는 생략되어야 합니다.

특정 CipherSpec를 사용하려면 CipherSuite를 TLS 연결을 설정하는 데 사용해야 하는 CipherSpec의 이름으로 설정하십시오.

텔레메트리 서비스와 MQTT 클라이언트는 각 마지막에 사용으로 설정되는 모든 CipherSpec으로부터 공동 CipherSpec을 협상합니다. 연결의 한 끝 또는 양 끝에 특정 CipherSpec 지정된 경우에는 반대쪽 끝에 있는 CipherSpec과 일치시켜야 합니다.

JSSE에 추가 제공자를 추가하여 추가 암호를 설치하십시오.

FIPS(Federal Information Processing Standard)

FIPS는 선택적 설정입니다. 기본적으로 설정되지 않습니다.

큐 관리자의 특성 패널 또는 **runmqsc**를 사용하여 SSLFIPS를 설정하십시오. SSLFIPS는 FIPS 인증 알고리즘만 사용되는지 여부를 지정합니다.

폐기 이름 목록

폐기 이름 목록은 선택적 설정입니다. 기본적으로 설정되지 않습니다.

큐 관리자의 특성 패널 또는 **runmqsc**를 사용하여 SSLCRLNL을 설정하십시오. SSLCRLNL은 인증서 폐기 위치를 제공하는 데 사용되는 인증 정보 오브젝트의 이름 목록을 지정합니다.

TLS 특성을 설정하는 다른 큐 관리자 매개변수는 사용되지 않습니다.

MQTT Java 클라이언트

MqttConnectionOptions.SSLProperties 에서 Java 클라이언트에 대한 TLS 특성을 설정하십시오. 예:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
```

```
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

특정 특성의 이름 및 값은 MqttConnectOptions 클래스에 설명되어 있습니다. MQTT 클라이언트 라이브러리의 클라이언트 API 문서에 대한 링크는 [MQTT 클라이언트 프로그래밍 참조](#)를 참조하십시오.

프로토콜

Protocol은 선택사항입니다.

프로토콜은 텔레메트리 서버와 협상할 때 선택됩니다. 특정 프로토콜이 필요한 경우에는 선택할 수 있습니다. 텔레메트리 서버가 해당 프로토콜을 지원하지 않는 경우, 연결이 실패합니다.

ContextProvider

ContextProvider는 선택사항입니다.

KeyStore

KeyStore는 선택사항입니다. 서버에 ClientAuth가 설정되어 클라이언트 인증을 강제 실행하는 경우 이를 구성하십시오.

개인 키를 사용하여 서명된 클라이언트의 디지털 인증서를 키 저장소에 저장하십시오. 키 저장소 경로와 비밀번호를 지정하십시오. 유형 및 제공자는 선택사항입니다. JKS는 기본 유형이며 IBMJCE는 기본 제공자입니다.

새 키 저장소 제공자를 추가하는 클래스를 참조하려면 다른 키 저장소 제공자를 지정하십시오. 키 관리자 이름을 설정하여 KeyManagerFactory를 인스턴스화하려면 키 저장소 제공자가 사용하는 알고리즘 이름을 전달하십시오.

TrustStore

TrustStore는 선택사항입니다. 신뢰하는 모든 인증서를 JRE cacerts 저장소에 배치할 수 있습니다.

클라이언트에 다른 신뢰 저장소를 사용하려면 신뢰 저장소를 구성하십시오. 서버가 cacerts에 루트 인증서가 이미 저장되어 있는 잘 알려진 CA에서 발행한 인증서를 사용 중인 경우 신뢰 저장소를 구성할 필요가 없습니다.

공용으로 서명된 서버 인증서 또는 루트 인증서를 신뢰 저장소에 추가하고 신뢰 저장소의 경로와 비밀번호를 지정하십시오. JKS는 기본 유형이며 IBMJCE는 기본 제공자입니다.

새 신뢰 저장소 제공자를 추가하는 클래스를 참조하려면 다른 신뢰 저장소 제공자를 지정하십시오. 신뢰 관리자 이름을 설정하여 TrustManagerFactory를 인스턴스화하려면 키 저장소 제공자가 사용하는 알고리즘 이름을 전달하십시오.

JRE

클라이언트 및 서버의 TLS 동작에 영향을 주는 다른 측면의 Java 보안은 JRE에서 구성됩니다. Windows의 구성 파일은 *Java Installation Directory*\jre\lib\security에 있습니다. 다음 표에서는 IBM MQ와 함께 제공되는 JRE를 사용하는 경우의 경로를 보여줍니다.

표 18. 플랫폼별 JRE TLS 구성 파일용 파일 경로	
플랫폼	파일 경로
Windows	<i>WMQ Installation Directory</i> \java\jre\lib\security
AIX and Linux 플랫폼	<i>WMQ Installation Directory</i> /java/jre64/jre/lib/security

잘 알려진 인증 기관

cacerts 파일에는 잘 알려진 인증 기관의 루트 인증서가 포함되어 있습니다. 신뢰 저장소를 지정하지 않은 경우 기본적으로 cacerts가 사용됩니다. cacerts 저장소를 사용하거나 신뢰 저장소를 제공하지 않은 경우 cacerts의 서명자 목록을 검토하여 보안 요구사항을 충족하도록 편집해야 합니다.

V 9.4.0 ▶ **V 9.4.0** `runmqktool` 명령을 사용하여 cacerts 인증서 파일을 관리할 수 있습니다. cacerts 는 JKS 파일입니다. `runmqktool` 명령을 사용하여 인증서 파일을 관리하는 경우 **-storetype jks** 매개변수를 지정하십시오.

V 9.4.0 ▶ **V 9.4.0** cacerts 파일의 기본 비밀번호는 changeit입니다. 파일을 보호하기 위해 `runmqktool -storepasswd` 명령을 사용하여 비밀번호를 변경하십시오.

보안 클래스 구성

java.security 파일을 사용하여 추가 보안 제공자 및 기타 기본 보안 특성을 등록하십시오.

권한

java.policy 파일을 사용하여 자원에 부여된 권한을 수정하십시오. javaws.policy는 javaws.jar에 권한을 부여합니다.

암호화 강도

일부 JRE는 암호화 강도가 약해진 상태로 배송됩니다. 키를 키 저장소로 가져올 수 없는 경우, 약해진 암호화 강도가 원인일 수 있습니다. 필요한 경우 [IBM 개발자 키트](#), 보안 정보에서 강력하지만 제한된 관할 구역 파일을 다운로드하십시오.

중요사항: 비밀번호화 소프트웨어의 원산 국가는 다른 나라로 가져오기, 소유권, 사용 또는 다시 내보내기를 수행하는 데 제한사항이 있습니다. 제한 없는 정책 파일을 다운로드하거나 사용하려면 사용자 국가의 법을 확인하십시오. 암호화 소프트웨어의 수입, 소유, 사용 및 재수출과 관련된 규정 및 정책을 조사하여 허용되는지 판별하십시오.

클라이언트가 서버에 연결할 수 있도록 신뢰 제공자 수정

예는 신뢰 제공자를 추가하고 MQTT 클라이언트 코드로부터 그것에 참조 사항을 다는 방법을 설명합니다. 예는 클라이언트 또는 서버의 인증을 수행하지 않습니다. 그 결과 TLS 연결은 인증 없이 암호화됩니다.

328 페이지의 [그림 16](#)의 코드 스니펫은 MQTT 클라이언트에 대한 AcceptAllProviders 신뢰 제공자 및 신뢰 관리자를 설정합니다.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

그림 16. MQTT 클라이언트 코드 스니펫

```
package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
private static final long serialVersionUID = 1L;
public AcceptAllProvider() {
super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
put("TrustManagerFactory.TrustAllCertificates",
AcceptAllTrustManagerFactory.class.getName());
}
}
```

그림 17. AcceptAllProvider.java

```
protected static class AcceptAllTrustManagerFactory extends
javax.net.ssl.TrustManagerFactorySpi {
public AcceptAllTrustManagerFactory() {}
protected void engineInit(java.security.KeyStore keystore) {}
protected void engineInit(
javax.net.ssl.ManagerFactoryParameters parameters) {}
protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
}
}
```

그림 18. AcceptAllTrustManagerFactory.java

```

protected static class AcceptAllX509TrustManager implements
javax.net.ssl.X509TrustManager {
public void checkClientTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Client authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public void checkServerTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Server authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[0];
}
private static void report(String string) {
System.out.println(string);
}
}
}

```

그림 19. AcceptAllX509TrustManager.java

Windows Linux AIX 텔레메트리 채널 JAAS 구성

클라이언트에서 보내는 Username을 인증하기 위해 JAAS를 구성하십시오.

IBM MQ 관리자는 JAAS를 사용하여 클라이언트 인증을 요청하는 MQTT 채널을 구성합니다. JAAS 인증을 수행할 각 채널에 대한 JAAS 구성 이름을 지정하십시오. 채널은 동일한 JAAS 구성을 모두 사용할 수 있거나 다른 JAAS 구성을 사용할 수 있습니다. 구성은 *WMQData directory\qmgrs\qMgrName\mqxr\jaas.config*에 정의됩니다.

jaas.config 파일은 JAAS 구성 이름으로 구성됩니다. 로그인 구성의 목록은 각 구성 이름 아래에 있습니다. 330 페이지의 『샘플 jaas.config 파일』의 내용을 참조하십시오.

JAAS에서는 네 개의 표준 로그인 모듈을 제공합니다. 표준 NT 및 UNIX 로그인 모듈은 제한된 값입니다.

JndiLoginModule

JNDI(Java Naming and Directory Interface)에 구성된 디렉토리 서비스에 대응하여 인증합니다.

Krb5LoginModule

Kerberos 프로토콜을 사용하여 인증합니다.

NTLoginModule

현재 사용자에게 대한 NT 보안 정보를 사용하여 인증합니다.

UnixLoginModule

현재 사용자에게 대한 UNIX 보안 정보를 사용하여 인증합니다.

NTLoginModule 또는 UnixLoginModule 사용의 문제점은 텔레메트리 (MQXR) 서비스가 MQTT 채널의 ID가 아닌 mqm ID로 실행되는 것입니다. mqm은 클라이언트의 ID가 아닌 인증을 위해 NTLoginModule 또는 UnixLoginModule로 전달되는 ID입니다.

이 문제점을 해결하려면 자체 로그인 모듈을 작성하거나 기타 표준 로그인 모듈을 사용하십시오. 샘플 JAASLoginModule.java가 MQ Telemetry로 제공됩니다. 이는 javax.security.auth.spi.LoginModule 인터페이스의 구현입니다. 이를 사용하여 자체 인증 메소드를 개발하십시오.

사용자가 제공하는 새 LoginModule 클래스는 텔레메트리(MQXR) 서비스의 클래스 경로에 있어야 합니다. 클래스 경로에 있는 IBM MQ 디렉토리에 클래스를 배치하지 마십시오. 자체 디렉토리를 작성하고 텔레메트리(MQXR) 서비스에 대한 전체 클래스 경로를 정의하십시오.

service.env 파일에서 클래스 경로를 설정하여 텔레메트리(MQXR) 서비스가 사용하는 클래스를 보강할 수 있습니다. CLASSPATH는 대문자여야 하고 클래스 경로 명령문에는 리터럴만 포함될 수 있습니다. CLASSPATH에

는 변수를 사용할 수 없습니다. 예를 들어, CLASSPATH=%CLASSPATH%는 올바르지 않습니다. 텔레메트리 (MQXR) 서비스는 자체 CLASSPATH를 설정합니다. service.env에 정의된 CLASSPATH가 해당 파일에 추가 됩니다.

텔레메트리 (MQXR) 서비스는 MQTT 채널에 연결된 클라이언트에 대해 Username 및 Password 를 리턴하는 두 개의 콜백을 제공합니다. 사용자 이름 및 비밀번호 는 MqttConnectOptions 오브젝트에서 설정됩니다. Username 및 Password에 액세스하는 방법에 대한 예제는 330 페이지의 『샘플 JAASLoginModule.Login() 메소드』의 내용을 참조하십시오.

샘플 jaas.config 파일

이름 지정된 구성 MQXRConfig가 있는 JAAS 구성 파일의 예입니다.

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//      principal=principal@your_realm
//      useDefaultCcache=TRUE
//      renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//      useTicketCache="true"
//      ticketCache="${user.home}/${tickets}";
};
```

샘플 JAASLoginModule.Login() 메소드

MQTT 클라이언트에서 제공하는 사용자 이름 및 비밀번호 를 수신하도록 코딩된 JAAS 로그인 모듈의 예제입니다.

```
public boolean login()
throws javax.security.auth.login.LoginException {
    javax.security.auth.callback.Callback[] callbacks =
    new javax.security.auth.callback.Callback[2];
    callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
    callbacks[1] = new javax.security.auth.callback.PasswordCallback(
    "PasswordCallback", false);
    try {
        callbackHandler.handle(callbacks);
        String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
        .getName();
        char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
        .getPassword();
        // Accept everything.
        if (true) {
            loggedIn = true;
        } else
        {
            throw new javax.security.auth.login.FailedLoginException("Login failed");
        }
        principal= new JAASPrincipal(username);
    } catch (java.io.IOException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    } catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    }
    return loggedIn;
}
```

관련 태스크

[문제점 해결: 텔레메트리 서비스가 JAAS 로그인 모듈을 호출하지 않음](#)

관련 참조

[AuthCallback MQXR 클래스](#)

AMQP 클라이언트 관리

IBM MQ Explorer을(를) 사용하거나 명령행에서 AMQP 클라이언트를 관리할 수 있습니다. 탐색기를 사용하여 채널을 구성하고 IBM MQ에 연결된 AMQP 클라이언트를 모니터링하십시오. TLS 및 JAAS를 사용하여 AMQP 클라이언트의 보안을 구성하십시오.

시작하기 전에

플랫폼에 AMQP 설치에 대한 정보는 [설치 항목 선택](#)을 참조하십시오.

IBM MQ Explorer를 사용하여 관리

탐색기를 사용하여 AMQP 채널을 구성하고 IBM MQ에 연결된 AMQP 클라이언트를 모니터링하십시오. TLS 및 JAAS를 사용하여 AMQP 클라이언트의 보안을 구성할 수 있습니다.

명령행을 사용하여 관리

명령행에서 [MQSC 명령](#)을 사용하여 AMQP 클라이언트를 관리할 수 있습니다.

Windows Linux AIX V9.4.0 V9.4.0 AMQP 서비스가 큐 관리자 시동 시 자동으로 시작되지 않음

IBM MQ 9.4.0부터 AMQP 서비스를 시작하기 위한 **CONTROL** 속성 설정의 기본 동작이 변경되었습니다. 새 큐 관리자를 작성하고 시작할 때 AMQP 서비스는 큐 관리자 시동 프로세스의 일부로 자동 시작되지 않습니다.

IBM MQ 9.0.4 와 IBM MQ 9.4.0사이에서 AMQP 서비스를 시작하기 위한 **CONTROL** 속성 설정의 기본 동작은 QMGR입니다.

AMQP 구성요소가 설치된 경우, AMQP 서비스는 이용되지 않는 경우에도 자동으로 시작됩니다. AMQP JVM (Java Virtual Machine)의 기본 시작을 방지하기 위해 두 가지 옵션이 있습니다.

- AMQP 구성요소를 설치하지 않음 또는
- 큐 관리자가 시작된 후 AMQP 서비스 **CONTROL** 속성을 MANUAL 로 변경합니다.

IBM MQ 9.4.0부터 새로 작성된 큐 관리자가 SYSTEM.AMQP.SERVICE 의 **CONTROL** 속성 설정을 IBM MQ 9.0.4 이전의 기본 설정인 수동로 되돌렸습니다.

마이그레이션된 큐 관리자는 AMQP를 활용하는 경우 큐 관리자 시동 중에 계속해서 서비스를 자동 시작합니다. AMQP가 사용되었는지 판별하기 위해 다음을 확인합니다.

- 기존 AMQP 채널
- AMQP 오류 로그의 채널 시작 메시지.



주의:

- 이는 한 번만 발생합니다. 업그레이드 후 처음으로 큐 관리자가 시작될 때 발생합니다.
- 마이그레이션 중에 **CONTROL** 속성이 QMGR 에서 MANUAL로 변경되면 정보 메시지가 IBM MQ 오류 로그에 로그되어 변경사항을 표시합니다. 자세한 정보는 AMQP 로그, 오류 로그 및 구성 파일의 위치를 참조하십시오.

AMQP 서비스가 자동 시작되도록 하려면 서비스 **CONTROL** 속성을 QMGR 로 변경하고 큐 관리자를 재시작하십시오. 큐 관리자의 후속 재시작은 AMQP 서비스를 시작합니다.

AMQP 클라이언트에서 사용 중인 IBM MQ 오브젝트 보기

AMQP 클라이언트가 사용 중인 다른 IBM MQ 자원을 볼 수 있습니다(예: 연결 및 구독).

연결

AMQP 서비스가 시작되면 새 Hconn이 작성되어 큐 관리자에 연결됩니다. 이 Hconns 풀은 AMQP 클라이언트가 메시지를 발행할 때 사용됩니다. **DISPLAY CONN** 명령을 사용하여 Hconn을 볼 수 있습니다. 예를 들면, 다음과 같습니다.

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM MQ Advanced Message Queuing Protocol*')
```

이 명령은 또한 클라이언트별 Hconn을 표시합니다. 공백 클라이언트 ID 속성을 갖는 Hconn은 풀에서 사용되는 Hconn입니다.

AMQP 클라이언트가 AMQP 채널에 연결하면 새 Hconn이 큐 관리자에 연결됩니다. 이 Hconn은 AMQP 클라이언트가 작성한 구독에 대해 비동기식으로 메시지를 소비하는 데 사용됩니다. **DISPLAY CONN** 명령을 사용하여 특정 AMQP 클라이언트가 사용하는 Hconn을 볼 수 있습니다. 예를 들면, 다음과 같습니다.

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

클라이언트가 작성한 구독

AMQP 클라이언트가 토픽을 구독하면 새 IBM MQ 구독이 작성됩니다. 구독 이름에는 다음 정보가 포함됩니다.

- 클라이언트의 이름. 클라이언트가 공유 구독을 결합한 경우에는 공유 이름이 사용됩니다.
- 클라이언트가 구독한 토픽 패턴
- 접두부. 클라이언트가 비공유 구독을 작성한 경우 접두부는 `private`이고 클라이언트가 공유 구독에 결합된 경우 접두부는 `share`입니다.

특정 AMQP 클라이언트에서 사용 중인 구독을 보려면 **DISPLAY SUB** 명령을 실행하고 `private` 접두부를 필터링하십시오.

```
DISPLAY SUB('/:private:*')
```

여러 클라이언트가 사용 중인 공유 구독을 보려면 **DISPLAY SUB** 명령을 실행하고 `share` 접두부를 필터링하십시오.

```
DISPLAY SUB('/:share:*')
```

여러 AMQP 클라이언트가 공유 구독을 사용할 수 있으므로 공유 구독에서 현재 메시지를 소비하는 클라이언트를 볼 수 있습니다. 이 작업을 수행하려면 현재 구독 큐에서 핸들이 열려 있는 Hconn을 나열하면 됩니다. 현재 공유를 사용하는 클라이언트를 보려면 다음 단계를 완료하십시오.

1. 공유 구독이 목적지로 사용하는 큐 이름을 찾으십시오. 예를 들면, 다음과 같습니다.

```
DISPLAY SUB('/:private:recv_e298452:public') DEST
5 : DISPLAY SUB('/:private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D312020202020202020707E0A565C2D0020)
SUB('/:private:recv_e298452:public')
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```

2. **DISPLAY CONN** 명령을 실행하여 해당 큐에서 열려 있는 핸들을 찾으십시오.

```
DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME
EQ SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
21 : DISPLAY CONN(*) TYPE(HANDLE) WHERE(OBJNAME EQ
SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(HANDLE)
```

```
OBJNAME(SYSTEM.BASE.TOPIC)      OBJTYPE(TOPIC)
OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(Queue)
```

3. 핸들마다 핸들이 열려 있는 AMQP 클라이언트 ID를 확인하십시오.

```
DISPLAY CONN(707E0A56642B0020) CLIENTID
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
24 : DISPLAY CONN(707E0A565F290020) CLIENTID
AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)
```

AMQP 클라이언트 식별, 권한 부여 및 인증

다른 IBM MQ 클라이언트 애플리케이션과 마찬가지로 AMQP 연결도 여러 가지 방법으로 보안을 설정할 수 있습니다.

다음 보안 기능을 사용하여 IBM MQ에 대한 AMQP 연결에 보안을 설정할 수 있습니다.

- [채널 인증 레코드](#)
- [연결 인증](#)
- 채널 MCA 사용자 구성
- IBM MQ 권한 정의
- [TLS 연결성](#)

보안 관점에서 연결 설정은 다음 두 단계로 구성됩니다.

- 연결을 계속해야 하는지 여부 결정
- 애플리케이션이 나중에 권한 검사를 수행하기 위해 가정하는 IBM MQ ID 결정

다음 정보는 AMQP 클라이언트가 연결 작성을 시도할 때 수행하는 단계와 다양한 IBM MQ 구성을 개략적으로 설명합니다. IBM MQ 구성에 따라 설명된 일부 단계를 사용하지 않을 수 있습니다. 예를 들어, 일부 구성은 회사 방화벽 내부 연결을 위해 TLS를 사용하지 않고 일부 구성은 TLS를 사용하지만 인증을 위해 클라이언트 인증서를 사용하지 않습니다. 사용자 정의 JAAS 모듈을 사용하지 않는 환경도 많습니다.

연결 설정

다음 단계는 AMQP 클라이언트가 연결을 설정할 때 그 결과를 설명합니다. 이 단계는 연결이 계속되는지 여부와 애플리케이션이 권한 검사를 위해 가정하는 IBM MQ ID를 결정합니다.

1. 클라이언트가 IBM MQ에 대한 TLS 연결을 열고 인증서를 제공하는 경우, 큐 관리자는 클라이언트 인증서의 유효성을 검증하려고 합니다.
2. 클라이언트가 사용자 이름 및 비밀번호 신임 정보를 제공하면 큐 관리자가 AMQP SASL 프레임을 수신하고 MQ CONNAUTH 구성을 확인합니다.
3. MQ 채널 인증 규칙을 검사합니다(예를 들어, IP 주소와 TLS 인증 DN이 올바른지 여부).
4. 채널 인증 규칙이 다르게 판별하지 않는 한 채널 MCAUSER를 확인합니다.
5. JAAS 모듈이 구성된 경우 모듈이 호출됩니다.
6. 결과 MQ 사용자 ID에 MQ CONNECT 권한 검사가 적용되지 않습니다.
7. 가정된 IBM MQ ID로 연결이 설정됩니다.

메시지 발행

다음 단계는 AMQP 클라이언트가 메시지를 발행할 때 그 결과를 설명합니다. 이 단계는 연결이 계속되는지 여부와 애플리케이션이 권한 검사를 위해 가정하는 IBM MQ ID를 결정합니다.

1. AMQP 링크 첨부 프레임이 큐 관리자에 도착합니다. 연결 중에 설정된 MQ 사용자 ID에 대해 지정된 토픽 문자열에 대한 IBM MQ 발행 권한을 검사합니다.
2. 지정된 토픽 문자열에 메시지가 발행됩니다.

토픽 패턴 구독

다음 단계는 AMQP 클라이언트가 토픽 패턴을 구독할 때 그 결과를 설명합니다. 이 단계는 연결이 계속되는지 여부와 애플리케이션이 권한 검사를 위해 가정하는 IBM MQ ID를 결정합니다.

1. AMQP 링크 첨부 프레임이 큐 관리자에 도착합니다. 연결 중에 설정된 MQ 사용자 ID에 대해 지정된 토픽 패턴에 대한 IBM MQ 구독 권한을 검사합니다.
2. 구독이 작성됩니다.

AMQP 클라이언트 ID 및 권한 부여

IBM MQ 오브젝트에 액세스하는 권한을 부여하려면 AMQP 클라이언트 ID, AMQP 사용자 이름 또는 채널이나 채널 인증 규칙에 정의된 고통 클라이언트 ID를 사용하십시오.

관리자는 AMQP 채널을 정의하거나 수정할 때 큐 관리자 CONNAUTH 설정을 구성하거나 채널 인증 규칙을 정의하여 선택합니다. ID는 IBM MQ 토픽에 액세스 권한을 부여하기 위해 사용됩니다. 선택 시 다음 항목을 기반으로 합니다.

1. 채널 USECLNTID 속성
2. 큐 관리자 CONNAUTH 규칙의 ADOPTCTX 속성
3. 채널에 정의된 MCAUSER 속성
4. 일치 채널 인증 규칙의 USERSRC 속성

문제점 예방: 이 프로세스에서 선택한 ID는 예를 들어, DISPLAY CHSTATUS (AMQP) 명령에 의해 클라이언트의 MCAUSER로 참조됩니다. 이 ID는 선택 (2)에서 참조되는 채널의 MCAUSER와 반드시 동일한 ID는 아닙니다.

IBM MQ `setmqaut` 명령을 사용하여 AMQP 채널과 연관된 ID가 사용할 수 있도록 권한이 부여된 오브젝트와 조치를 선택할 수 있습니다. 예를 들어, 다음 명령은 큐 관리자 QM1의 관리자가 제공한 채널 ID AMQPClient에 권한을 부여합니다.

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPClient -all +pub +sub
```

및

```
setmqaut -m QM1 -t qmgr -p AMQPClient -all +connect
```

비밀번호를 사용하여 AMQP 클라이언트 인증

클라이언트 비밀번호를 사용하여 AMQP 클라이언트 사용자 이름을 인증하십시오. 클라이언트가 토픽을 발행하고 구독하도록 권한 부여된 ID와 다른 ID를 사용하여 클라이언트를 인증할 수 있습니다.

AMQP 서비스는 MQ CONNAUTH 또는 JAAS를 사용하여 클라이언트 사용자 이름을 인증할 수 있습니다. 이들 중 하나가 구성된 경우, 클라이언트가 제공하는 비밀번호를 MQ CONNAUTH 구성 또는 JAAS 모듈로 확인합니다.

다음 프로시저에서는 로컬 OS 사용자 및 비밀번호에 대해 개별 사용자를 인증하고 성공하는 경우 공통 ID AMQPUser(를) 채택하는 예제 단계를 요약합니다.

1. IBM MQ 관리자는 IBM MQ Explorer를 사용하여 AMQP 채널 MCAUSER ID를 임의 이름(예: AMQPUser)으로 설정합니다.

2. IBM MQ 관리자는 토픽을 발행하고 구독하는 권한을 AMQPUser에 부여합니다.

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. IBM MQ 관리자는 클라이언트가 제공하는 사용자 이름과 비밀번호를 확인하는 IDPWOS CONNAUTH 규칙을 구성합니다. CONNAUTH 규칙은 CHCKCLNT(REQUIRED) 및 ADOPTCTX(NO)를 설정해야 합니다.

참고: 큐 관리자에 대한 연결을 보다 강력하게 제어하려면 채널 인증 규칙을 사용하고 MCAUSER 채널 속성을 권한이 없는 사용자로 설정하는 것이 좋습니다.

채널에서 발행물 개인정보 보호

AMQP 채널 전체에 걸쳐 각각 방향으로 보내진 AMQP 발행물의 개인정보 보호정책은 연결을 통해 전송을 암호화하기 위해 TLS를 사용하여 보호됩니다.

AMQP 채널에 연결되는 AMQP 클라이언트는 대칭 키 암호화를 사용하여 채널로 전송된 발행물의 개인정보를 보호를 위해 TLS를 사용합니다. 엔드 포인트가 인증되지 않으므로 채널 암호화만 신뢰할 수는 없습니다. 개인정보 보호와 서버 또는 상호 인증을 결합하십시오.

TLS를 사용하는 대신 IPsec와 같은 일부 유형의 가상 사설 네트워크(VPN)는 TCP/IP 연결의 엔드 포인트를 인증합니다. VPN은 네트워크를 통해 플로우는 각 IP 패킷을 암호화합니다. 이러한 VPN 연결이 설정되면 신뢰할 수 있는 네트워크가 설정된 것입니다. VPN 네트워크에서 TCP/IP를 사용하여 AMQP 클라이언트를 AMQP 채널에 연결할 수 있습니다.

서버를 인증하지 않고 TLS 연결을 암호화하면 연결이 중간자 공격(man-in-the-middle attack)에 노출됩니다. 교환하는 정보가 도청에 대해 보호되는 경우에도 누구와 교환하는지 알지 못합니다. 네트워크를 제어하지 않는 경우, IP 전송을 인터셉트하고 엔드 포인트로 위장하는 사람에게 노출됩니다.

서버를 인증하지 않고 익명 TLS를 지원하는 Diffie-Hellman 키 교환 CipherSpec을 사용하여 암호화된 TLS로 연결할 수 있습니다. 클라이언트와 서버 사이에 공유되고 TLS 전송을 암호화하는 데 사용되는 마스터 보안은 개인적으로 서명된 서버 인증서를 교환하지 않고 설정됩니다.

익명 연결은 안전하지 않으므로 대부분의 TLS 구현은 기본적으로 익명 CipherSpec을 사용하지 않습니다. AMQP 채널에서 TLS 연결에 대한 요청이 승인되는 경우, 채널에는 암호화로 보호되는 키 저장소가 있어야 합니다. 기본적으로 TLS 구현은 익명 CipherSpec을 사용하지 않으므로 키 저장소에는 클라이언트가 인증할 수 있는 개인적으로 서명된 인증서가 포함되어 있어야 합니다.

익명 CipherSpec을 사용하는 경우 서버 키 저장소가 있어야 하지만 개인적으로 서명된 인증서를 포함할 필요는 없습니다.

암호화된 연결을 설정하는 다른 방법은 클라이언트에서 신뢰 제공자를 사용자 고유의 구현으로 바꾸는 것입니다. 사용자의 신뢰 제공자는 서버 인증서를 인증하지 않지만 연결이 암호화됩니다.

TLS로 AMQP 클라이언트 구성

네트워크를 통해 이동하는 데이터를 보호하고 클라이언트가 연결하는 큐 관리자의 ID를 인증하기 위해 TLS를 사용하도록 AMQP 클라이언트를 구성할 수 있습니다.

AMQP 클라이언트에서 AMQP 채널로 연결하기 위해 TLS를 사용하려면, 큐 관리자가 TLS로 구성되었는지를 확인해야 합니다. 큐 관리자에서 TLS 구성은 큐 관리자가 TLS 인증서를 읽는 키 저장소를 구성하는 방법을 설명합니다.

큐 관리자가 키 저장소와 함께 구성된 경우에는 클라이언트가 연결될 AMQP 채널에서 TLS 속성을 구성해야 합니다. AMQP 채널은 TLS 구성과 관련된 다음 네 가지 속성을 갖습니다.

SSLCAUTH

SSLCAUTH 속성은 큐 관리자에서 AMQP 클라이언트가 ID를 확인하기 위해 클라이언트 인증서를 제공해야 하는지 지정하는 데 사용됩니다.

SSLCIPH

SSLCIPH 속성은 채널이 TLS 플로우에서 데이터를 인코드하기 위해 사용해야 하는 암호를 지정합니다.

V 9.4.0 IBM MQ 9.4.0부터 AMQP 채널은 ANY* 일반 CipherSpecs를 지원합니다. CipherSpec에 대한 자세한 정보는 CipherSpec 사용을 참조하십시오.

SSLPEER

SSLPEER 속성은 연결이 허용되려면 클라이언트 인증서와 일치해야 하는 식별 이름(DN)을 지정하는 데 사용됩니다.

CERTLABL

CERTLABL은 큐 관리자가 클라이언트에게 제공해야 하는 인증서를 지정합니다. 큐 관리자의 키 저장소에는 여러 인증서가 포함될 수 있습니다. 이 속성을 사용하면 이 채널에 대한 연결에 사용할 인증서를 지정할 수 있습니다. CERTLABL이 지정되지 않은 경우, 큐 관리자 CERTLABL 속성에 해당하는 레이블이 있는 큐 관리자 키 저장소의 인증서가 사용됩니다.

TLS 속성으로 AMQP 채널을 구성한 경우 다음 명령을 사용하여 AMQP 서비스를 재시작해야 합니다.

```
STOP SERVICE(SYSTEM.AMQP.SERVICE) START SERVICE(SYSTEM.AMQP.SERVICE)
```

AMQP 클라이언트가 TLS를 통해 보호되는 AMQP 채널에 연결할 때, 클라이언트는 큐 관리자가 제공하는 인증서의 ID를 검증합니다. 이를 수행하려면 큐 관리자의 인증서가 포함된 신뢰 저장소를 사용하여 AMQP 클라이언트를 구성해야 합니다. 이 작업을 수행하는 단계는 사용 중인 AMQP 클라이언트에 따라 다릅니다. 다양한 AMQP 클라이언트와 API에 대한 정보는 각 AMQP 클라이언트 문서를 참조하십시오.

관련 참조

[DEFINE CHANNEL\(새 채널 정의\)](#)

[멀티플랫폼의 STOP SERVICE\(서비스 중지\)](#)

[멀티플랫폼의 START SERVICE\(서비스 시작\)](#)

큐 관리자에서 AMQP 클라이언트 연결 끊기

큐 관리자에서 AMQP 클라이언트의 연결을 끊으려면 PURGE CHANNEL 명령을 실행하거나 AMQP 클라이언트에 대한 연결을 중지하십시오.

- **PURGE CHANNEL** 명령을 실행하십시오. 예를 들면, 다음과 같습니다.

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- 또는 다음 단계를 완료하여 AMQP 클라이언트가 클라이언트의 연결을 끊기 위해 사용하는 연결을 중지하십시오.

1. **DISPLAY CONN** 명령을 실행하여 클라이언트가 사용 중인 연결을 찾으십시오. 예를 들면, 다음과 같습니다.

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

명령 출력은 다음과 같습니다.

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_28dbb7e)
```

2. 연결을 중지하십시오. 예를 들면, 다음과 같습니다.

```
STOP CONN(707E0A565F2D0020)
```

멀티캐스트 관리

이 정보를 사용하여 멀티캐스트 메시지의 크기를 줄이고 데이터 변환을 사용 가능하도록 설정하는 것과 같은 IBM MQ 멀티캐스트 관리 태스크에 대해 학습하십시오.

멀티캐스트 시작하기

이 정보를 사용하여 IBM MQ 멀티캐스트 토픽 및 통신 정보 오브젝트를 시작하십시오.

이 태스크 정보

IBM MQ 멀티캐스트 메시징은 토픽을 그룹 주소로 매핑하여 메시지를 전달하도록 네트워크를 사용합니다. 다음 태스크를 수행하면 멀티캐스트 메시징을 위한 필수 IP 주소 및 포트가 올바르게 구성되는지의 여부를 빠르게 테스트할 수 있습니다.

멀티캐스트에 대한 **COMMINFO** 오브젝트 작성

통신 정보(COMMINFO) 오브젝트에는 멀티캐스트 전송과 연관되는 속성이 포함됩니다. COMMINFO 오브젝트 매개변수에 대한 자세한 정보는 [DEFINE COMMINFO](#)를 참조하십시오.

다음 명령행 예를 사용하여 멀티캐스트에 대한 COMMINFO 오브젝트를 정의하십시오.

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

여기서 *MC1*은 COMMINFO 오브젝트의 이름이고, *group address*는 그룹 멀티캐스트 IP 주소 또는 DNS 이름이며, *port number*는 전송할 포트(기본값은 1414임)입니다.

*MC1*이라는 새 COMMINFO 오브젝트가 작성됩니다. 이 이름은 다음 예에서 TOPIC 오브젝트를 정의할 때 지정해야 하는 이름입니다.

멀티캐스트에 대한 **TOPIC** 오브젝트 작성

토픽은 발행/구독 메시지로 발행된 정보의 주제이며, 이러한 토픽은 TOPIC 오브젝트를 작성하여 정의합니다. TOPIC 오브젝트에는 멀티캐스트와 함께 사용될 수 있을지 여부를 정의하는 두 개의 매개변수가 있습니다. 이러한 매개변수는 **COMMINFO** 및 **MCAST**입니다.

- **COMMINFO** 이 매개변수는 멀티캐스트 통신 정보 오브젝트의 이름을 지정합니다. COMMINFO 오브젝트 매개변수에 대한 자세한 정보는 [DEFINE COMMINFO](#)를 참조하십시오.
- **MCAST** 이 매개변수는 토픽 트리의 이 지점에서 멀티캐스트를 허용할 수 있는지의 여부를 지정합니다.

다음 명령행 예를 사용하여 멀티캐스트에 대해 TOPIC 오브젝트를 정의하십시오.

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

*ALLSPORTS*라는 새 TOPIC 오브젝트가 작성됩니다. 이 오브젝트에는 토픽 문자열 *Sports*가 있고, 관련된 통신 정보 오브젝트는 *MC1*이라고 하며(이전 예에서 COMMINFO 오브젝트를 정의할 때 지정된 이름임) 멀티캐스트가 사용으로 설정됩니다.

멀티캐스트 발행/구독 테스트

TOPIC 및 COMMINFO 오브젝트가 작성된 후 `amqspubc` 샘플 및 `amqssubc` 샘플을 사용하여 이러한 오브젝트를 테스트할 수 있습니다. 이러한 샘플에 대한 자세한 정보는 [발행/구독 샘플 프로그램을 참조하십시오](#).

1. 두 개의 명령행 창을 여십시오. 첫 번째 명령행은 `amqspubc` 발행 샘플에 대한 것이고 두 번째 명령행은 `amqssubc` 구독 샘플에 대한 것입니다.
2. 명령행 1에서 다음 명령을 입력하십시오.

```
amqspubc Sports QM1
```

여기서 *Sports*는 이전 예에서 정의된 TOPIC 오브젝트의 토픽 문자열이고 *QM1*은 큐 관리자의 이름입니다.

3. 명령행 2에 다음 명령을 입력하십시오.

```
amqssubc Sports QM1
```

여기서 *Sports* 및 *QM1*은 337 페이지의 『2』 단계에서 사용되는 것과 동일합니다.

4. 명령행 1에서 Hello world 를 입력하십시오. COMMINFO 오브젝트에 지정된 포트 및 IP 주소가 올바르게 구성된 경우, 지정된 주소의 발행물에 대해 포트에서 대기하는 amqssubc 샘플은 명령행 2에서 Hello world 를 출력합니다.

IBM MQ 멀티캐스트 토픽 토폴로지

이 예를 사용하여 IBM MQ 멀티캐스트 토픽 토폴로지를 이해하십시오.

IBM MQ 멀티캐스트 지원에서는 전체 계층 내의 각 서브트리에 자체 멀티캐스트 그룹 및 데이터 스트림이 있어야 합니다.

클래스풀 네트워크 IP 주소 지정 설계에서는 멀티캐스트 주소를 위한 주소 공간을 지정합니다. IP 주소의 전체 멀티캐스트 범위는 224.0.0.0 - 239.255.255.255이지만, 이러한 주소 중 일부는 예약되어 있습니다. 예약된 주소 목록은 시스템 관리자에게 문의하거나 <https://www.iana.org/assignments/multicast-addresses>에서 자세한 정보를 참조하십시오. 239.0.0.0 - 239.255.255.255범위에서 로컬로 범위가 지정된 멀티캐스트 주소를 사용하는 것이 좋습니다.

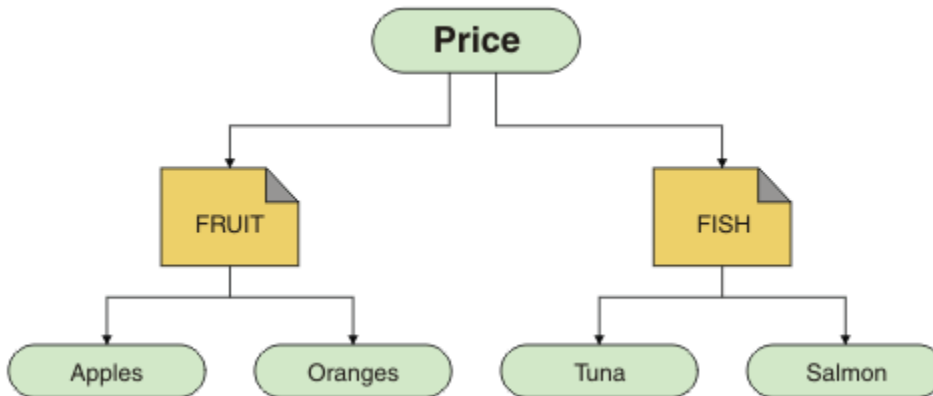
다음 다이어그램에는 사용 가능한 두 개의 멀티캐스트 데이터 스트림이 있습니다.

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX)
)
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

여기서 239.XXX.XXX.XXX 및 239.YYY.YYY.YYY는 올바른 멀티캐스트 주소입니다.

이러한 토픽 정의를 사용하여 다음 다이어그램에 표시된 대로 토픽 트리를 작성할 수 있습니다.

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



각 멀티캐스트 통신 정보(COMMINFO) 오브젝트는 해당 그룹 주소가 다르므로 다른 데이터 스트림을 표시합니다. 이 예에서 FRUIT 토픽은 COMMINFO 오브젝트 MC1을 사용하기 위해 정의되고, FISH 토픽은 COMMINFO 오브젝트 MC2를 사용하기 위해 정의되며, Price 노드에는 멀티캐스트 정의가 없습니다.

IBM MQ 멀티캐스트는 토픽 문자열에 대해 255자 제한을 가집니다. 이 제한사항은 트리 내에 있는 노드 및 리프 노드의 이름에 주의해야 함을 의미합니다. 노드 및 리프 노드의 이름이 지나치게 길면 토픽 문자열이 255자를 초과하여 2425(0979)(RC2425): MQRC_TOPIC_STRING_ERROR 이유 코드를 리턴하기 때문입니다. 토픽 문자열이 길면 성능에 좋지 않은 영향을 주므로 가능한 한 토픽 문자열을 짧게 작성하는 것이 좋습니다.

멀티캐스트 메시지의 크기 조절

IBM MQ 메시지 유형에 대해 학습하기 위해 이 정보를 사용하고 IBM MQ 메시지의 크기를 줄이십시오.

IBM MQ 메시지에는 메시지 디스크립터에 포함되는 것과 연관되는 수많은 속성을 가지고 있습니다. 작은 메시지의 경우, 이러한 속성은 대부분의 데이터 트래픽을 나타내고, 전송 속도에 대한 중요한 악영향을 가질 수 있습니다.

다. IBM MQ 멀티캐스트는 사용자가 있다해도 이러한 속성 중에 어느 것이 메시지와 함께 전송되는지 구성할 수 있게 합니다.

토픽 문자열 이외의 메시지 속성 존재는 속성이 보내져야 하는지 여부에 대해 COMMINFO 오브젝트가 언급하는 지 여부에 따라 다릅니다. 속성이 전송되지 않으면, 수신 애플리케이션은 기본값을 적용합니다. 기본 MQMD 값이 MQMD_DEFAULT 값과 반드시 동일한 것은 아니며 339 페이지의 표 19에 설명되어 있습니다.

COMMINFO 오브젝트에는 메시지로 플로우하는 MQMD 필드 및 사용자 특성 수를 제어하는 MCPROP 속성이 포함됩니다. 적절한 레벨에 이 속성의 값을 설정하여, IBM MQ 멀티캐스트 메시지의 크기를 제어할 수 있습니다.

MCPROP

멀티캐스트 특성은 메시지와 함께 플로우되는 MQMD 특성 및 사용자 특성 수를 제어합니다.

모두

모든 사용자 특성 및 MQMD의 모든 필드는 전송됩니다.

REPLY

메시지에 대한 응답을 처리하는 MQMD 필드 및 사용자 특성만 전송됩니다. 이러한 특성은 다음과 같습니다.

- MessageType
- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

사용자

사용자 특성만 전송됩니다.

NONE

사용자 특성 또는 MQMD 필드는 전송되지 않습니다.

COMPAT

이 값은 메시지의 전송이 RMM에 호환 모드에서 행해지게 하며, 현재 XMS 애플리케이션과 IBM Integration Bus RMM 애플리케이션으로 작동 사이 일부를 허용합니다.

 XMS .NET 멀티캐스트 메시징(사용RMM)는 더 이상 사용되지 않습니다. IBM MQ 9.2 그리고 에서 제거되었습니다 IBM MQ 9.3 .

멀티캐스트 메시지 속성

메시지 속성은 MQMD, MQRFH2의 필드 및 메시지 속성과 같은 다양한 장소에서 발생할 수 있습니다.

다음 표에는 속성이 보내질 때 사용되는 기본값 및 MCPROP의 값(이 섹션에서 이전에 설명됨)에 영향을 받는 메시지가 전송될 때 발생하는 것이 표시됩니다.

표 19. 메시징 속성 및 멀티캐스트와의 관련 방법		
속성	멀티캐스트를 사용할 때 조치	전송되지 않는 경우 기본값
TopicString	항상 포함됨	적용할 수 없음
MQMQ StrucId	전송되지 않음	적용할 수 없음
MQMD 버전	전송되지 않음	적용할 수 없음
보고서	기본값이 아닌 경우 포함됨	0
MessageType	기본값이 아닌 경우 포함됨	MQMT_DATAGRAM
만기	기본값이 아닌 경우 포함됨	0
Feedback	기본값이 아닌 경우 포함됨	0
Encoding	기본값이 아닌 경우 포함됨	MQENC_NORMAL(equiv)

표 19. 메시징 속성 및 멀티캐스트와의 관련 방법 (계속)

속성	멀티캐스트를 사용할 때 조치	전송되지 않는 경우 기본값
CodedCharSetId	기본값이 아닌 경우 포함됨	1208
형식	기본값이 아닌 경우 포함됨	MQRFH2
Priority	기본값이 아닌 경우 포함됨	4
Persistence	기본값이 아닌 경우 포함됨	MQPER_NOT_PERSISTENT
MsgId	기본값이 아닌 경우 포함됨	Null
CorrelId	기본값이 아닌 경우 포함됨	Null
BackoutCount	기본값이 아닌 경우 포함됨	0
ReplyToQ	기본값이 아닌 경우 포함됨	공백
ReplyToQMgr	기본값이 아닌 경우 포함됨	공백
UserIdentifier	기본값이 아닌 경우 포함됨	공백
AccountingToken	기본값이 아닌 경우 포함됨	Null
PutAppIType	기본값이 아닌 경우 포함됨	MQAT_JAVA
PutAppIName	기본값이 아닌 경우 포함됨	공백
PutDate	기본값이 아닌 경우 포함됨	공백
PutTime	기본값이 아닌 경우 포함됨	공백
ApplOriginData	기본값이 아닌 경우 포함됨	공백
GroupID	제외됨	적용할 수 없음
MsgSeqNumber	제외됨	적용할 수 없음
오프셋	제외됨	적용할 수 없음
MsgFlags	제외됨	적용할 수 없음
OriginalLength	제외됨	적용할 수 없음
UserProperties	포함됨	적용할 수 없음

관련 참조

Multi ALTER COMMINFO
 DEFINE COMMINFO

멀티캐스트 메시징에 대한 데이터 변환 사용 가능

이 정보를 사용하여 IBM MQ Multicast 메시징에서 데이터 변환이 작동하는 방식을 이해하십시오.

IBM MQ Multicast는 연결되지 않은 공유 프로토콜이므로, 각 클라이언트가 특정하게 데이터 변환을 요청할 수 없습니다. 동일한 멀티캐스트 스트림을 구독하는 모든 클라이언트가 동일한 2진 데이터를 수신합니다. 따라서 IBM MQ 데이터 변환이 필요한 경우 각 클라이언트에서 로컬로 변환을 수행합니다.

혼합 플랫폼 설치에서 전송 애플리케이션의 고유 형식이 아닌 형식에서 대부분의 클라이언트에 데이터가 필요할 수도 있습니다. 이 상황에서 멀티캐스트 COMMINFO 오브젝트의 **CCSID** 및 **ENCODING** 값은 효율성을 위해 메시지 전송의 인코딩을 정의하기 위해 사용될 수 있습니다.

IBM MQ 멀티캐스트는 다음 형식으로 빌드된 것에 대한 메시지 페이로드의 데이터 변환을 지원합니다.

- MQADMIN
- MQEVENT

- MQPCF
- MQRFH
- MQRFH2
- MQSTR

이러한 형식뿐만 아니라 자체 형식을 정의하고 MQDXP - 데이터-변환 엑시트 매개변수 데이터 변환 엑시트를 사용할 수도 있습니다.

데이터 변환 프로그래밍에 대한 정보는 멀티캐스트 메시징을 위한 MQI의 데이터 변환을 참조하십시오.

데이터 변환에 대한 자세한 정보는 데이터 변환을 참조하십시오.

데이터 변환 엑시트와 ClientExitPath에 대한 자세한 정보는 클라이언트 구성 파일의 ClientExitPath 스탠자를 참조하십시오.

멀티캐스트 애플리케이션 모니터링

이 정보를 사용하여 IBM MQ 멀티캐스트 관리 및 모니터링에 대해 학습하십시오.

멀티캐스트 트래픽에 대한 현재 발행자 및 구독자의 상태(예: 송신되고 수신되는 메시지 수 또는 유실된 메시지 수)는 클라이언트에서 서버로 주기적으로 전송됩니다. 상태가 수신될 때, **COMMINFO** 오브젝트의 **COMMEV** 속성은 큐 메시지가 **SYSTEM.ADMIN.PUBSUB.EVENT**에 이벤트 메시지를 배치할지 여부를 지정합니다. 이벤트 메시지는 수신된 상태 정보가 포함됩니다. 이 정보는 문제점의 소스를 찾는 데 있어서 매우 귀중한 진단 보조 프로그램입니다.

MQSC 명령 **DISPLAY CONN**을 사용하면 큐 관리자에 연결된 애플리케이션에 대한 연결 정보를 표시할 수 있습니다. **DISPLAY CONN** 명령에 대한 자세한 정보는 DISPLAY CONN을 참조하십시오.

MQSC 명령 **DISPLAY TPSTATUS**를 사용하면 발행자 및 구독자의 상태를 표시할 수 있습니다. **DISPLAY TPSTATUS** 명령에 대한 자세한 정보는 DISPLAY TPSTATUS를 참조하십시오.

COMMEV 및 멀티캐스트 메시지 신뢰도 표시기

COMMINFO 오브젝트의 **COMMEV** 속성과 함께 사용되는 신뢰도 표시기는 IBM MQ 멀티캐스트 발행자 및 구독자 모니터링의 핵심 요소입니다. 신뢰도 표시기(발행 또는 구독 상태 명령에서 리턴되는 **MSGREL** 필드)는 Sometimes 메시지가 전송 오류로 인해 재전송되어야 하는 오류가 없는 전송 백분율을 설명하는 IBM MQ 표시기입니다. 이는 **MSGREL**의 값에 반영됩니다. 전송 오류의 잠재적 원인에는 느린 구독자, 사용량이 많은 네트워크 및 네트워크 가동 중단이 포함됩니다. **COMMEV**는 **COMMINFO** 오브젝트를 사용하여 작성되고 세 개의 가능한 값 중 하나로 설정되는 멀티캐스트 핸들을 위해 이벤트 메시지가 생성되는지 여부를 제어합니다.

DISABLED

이벤트 메시지가 작성되지 않습니다.

ENABLED

이벤트 메시지는 **COMMINFO MONINT** 매개변수로 정의된 빈도로 항상 작성됩니다.

EXCEPTION

메시지 신뢰성이 신뢰성 임계값보다 낮을 경우 이벤트 메시지가 기록됩니다. 90% 이하의 메시지 신뢰도 레벨은 네트워크 구성을 가진 문제점일 수도 있고 또는 하나 이상의 발행/구독 애플리케이션이 너무 느리게 실행된다는 것을 표시합니다.

- **MSGREL (100, 100)**의 값은 단기간 또는 장기간 시간 범위에서 실행되지 않는지 표시합니다.
- **MSGREL (80, 60)**의 값은 메시지의 20%가 현재 문제점을 가지고 있지만, 60의 장기간 값에 개선이 있다는 것을 표시합니다.

클라이언트는 큐 관리자에 대한 유니캐스트 연결이 중단될 때 멀티캐스트 트래픽을 계속 전송하고 수신할 수 있습니다. 따라서, 데이터는 시대에 뒤떨어질 수 있습니다.

멀티캐스트 메시지 신뢰성

이 정보를 사용하여 IBM MQ 멀티캐스트 구독 및 메시지 실행 기록을 설정하는 방법에 대해 학습하십시오.

멀티캐스트를 가지는 전송 실패 극복의 핵심 요소는 IBM MQ에 의해 전송된 데이터의 버퍼링(링크의 전송 종료 시 유지될 메시지의 실행 기록)입니다. 이 프로세스는 IBM MQ에서 신뢰성을 제공하므로 넣기 애플리케이션 프로세스에 필요한 메시지의 버퍼링이 아니라는 것을 의미합니다. 다음 정보에 설명된 대로 통신 정보 (COMMINFO) 오브젝트를 통해 이 실행 기록의 크기가 구성됩니다. 더 큰 전송 버퍼는 필요한 경우 재전송될 추가 전송 실행 기록이 있지만, 멀티캐스트의 성질로 인해 100% 보장 전달이 지원될 수 없다는 것을 의미합니다.

IBM MQ 멀티캐스트 메시지 히스토리는 **MSGHIST** 속성에 의해 통신 정보(COMMINFO) 오브젝트에서 제어됩니다.

MSGHIST

이 값은 NACK(부정적 수신확인)의 경우에 재전송을 핸들링하기 위해 시스템에서 보관하는 메시지 실행 기록의 양(KB)입니다.

0 값은 최소 레벨의 신뢰도를 제공합니다. 기본값은 100KB입니다.

IBM MQ 멀티캐스트 새 구독 실행 기록은 **NSUBHIST** 속성에 의해 통신 정보 (COMMINFO) 오브젝트에서 제어됩니다.

NSUBHIST

새 구독자 실행 기록은 발행 스트림을 조인하는 구독자가 현재 사용 가능한 만큼의 데이터를 수신하는지 또는 구독 시 작성된 발행물만 수신하는지를 제어합니다.

NONE

NONE 값은 전송기가 등록 시간에서 작성된 발행물만 전송하도록 합니다. NONE은 기본값입니다.

모두

ALL 값을 사용하면 전송자가 알려진 것처럼 주제의 실행기록이 다시 전송됩니다. 어떤 경우, 이 상황은 보유된 발행물에 유사한 작동을 제공할 수 있습니다.

참고: ALL 값을 사용하면 모든 토픽 실행기록이 재전송되기 때문에 큰 토픽 실행기록이 있는 경우 성능에 해로운 영향을 줄 수 있습니다.

관련 참조

[DEFINE COMMINFO](#)

 [ALTER COMMINFO](#)

고급 멀티캐스트 태스크

이 정보를 사용하여 IBM MQ LLM과의 상호 운용성 및 .ini 파일 구성과 같은 고급 IBM MQ 멀티캐스트 관리 태스크에 대해 학습하십시오.

멀티캐스트 설치의 보안에 대한 고려사항의 경우, 멀티캐스트 보안을 참조하십시오.

멀티캐스트 및 비멀티캐스트 발행/구독 도메인 사이의 브릿징

이 정보를 사용하여 비멀티캐스트 발행자가 IBM MQ 멀티캐스트 사용 토픽에 발행할 때 발생하는 것에 대해 이해하십시오.

비멀티캐스트 발행자가 **MCAST** 사용 및 **BRIDGE** 사용으로 정의되는 토픽에 발행하는 경우, 큐 관리자는 멀티캐스트를 통해 대기할 수 있는 구독자에게 직접적으로 메시지를 전송합니다. 멀티캐스트 발행자는 멀티캐스트가 사용되지 않는 토픽에 발행할 수 없습니다.

기존 토픽은 토픽 오브젝트의 **MCAST** 및 **COMMINFO** 매개변수를 설정하여 사용 가능하게 되는 멀티캐스트가 될 수 있습니다. 이러한 매개변수에 대한 자세한 정보는 [초기 멀티캐스트 개념](#)을 참조하십시오.

COMMINFO 오브젝트 **BRIDGE** 속성은 멀티캐스트를 사용하고 있지 않는 애플리케이션으로부터 발행을 제어합니다. **BRIDGE**이 ENABLED로 설정되고 토픽의 **MCAST** 매개변수도 ENABLED로 설정되면, 멀티캐스트를 사용하고 있지 않는 애플리케이션으로부터의 발행물은 애플리케이션과 브릿지됩니다. **BRIDGE** 매개변수에 대한 자세한 정보는 [DEFINE COMMINFO](#)의 내용을 참조하십시오.

멀티캐스트용 .ini 파일 구성

이 정보를 사용하여 .ini 파일의 IBM MQ 멀티캐스트 필드를 이해할 수 있습니다.

추가 IBM MQ 멀티캐스트 구성은 ini 파일에서 작성할 수 있습니다. 사용해야 하는 특정 ini 파일은 애플리케이션 유형에 따라 다릅니다.

- 클라이언트: `MQ_DATA_PATH/mqclient.ini` 파일을 구성하십시오.
- 큐 관리자: `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini` 파일을 구성하십시오.

여기서, `MQ_DATA_PATH`은(는) IBM MQ 데이터 디렉토리(`/var/mqm/mqclient.ini`)의 위치이고, `QMNAME`은 `.ini` 파일이 적용되는 큐 관리자의 이름입니다.

`.ini` 파일에는 IBM MQ 멀티캐스트의 작동을 미세 조정하는 데 사용되는 필드가 포함되어 있습니다.

```
Multicast:
Protocol      = IP | UDP
IPVersion     = IPv4 | IPv6 | ANY | BOTH
LimitTransRate = DISABLED | STATIC | DYNAMIC
TransRateLimit = 100000
SocketTTL     = 1
Batch         = NO
Loop          = 1
Interface     = <IPAddress>
FeedbackMode  = ACK | NACK | WAIT1
HeartbeatTimeout = 20000
HeartbeatInterval = 2000
```

프로토콜

UDP

이 모드에서, 패킷은 UDP 프로토콜을 사용하여 보내집니다. 그러나, IP 모드에서 수행한 것처럼 네트워크 요소는 멀티캐스트 분산에서 보조를 제공할 수 없습니다. 패킷 형식은 PGM으로 호환된 채로 남아 있습니다. 이는 기본값입니다.

IP

이 모드에서, 전송자는 원시 IP 패킷을 보냅니다. PGM 지원을 가지는 네트워크 요소는 신뢰할 수 있는 멀티캐스트 패킷 분배를 지원합니다. 이 모드는 PGM 표준과 완전히 호환 가능합니다.

IPVersion

IPv4

IPv4 프로토콜만 사용하여 통신하십시오. 이는 기본값입니다.

IPv6

IPv6 프로토콜만 사용하여 통신하십시오.

ANY

사용 가능한 프로토콜에 따라 IPv4, IPv6 또는 둘 모두를 사용하여 통신하십시오.

BOTH

IPv4 및 IPv6 모두를 사용하는 통신을 지원합니다.

LimitTransRate

DISABLED

전송을 제어가 없습니다. 이는 기본값입니다.

정적

정적 전송을 제어를 구현합니다. 전송자는 `TransRateLimit` 매개변수에서 지정하는 비율을 초과하는 비율에서 전송하지 않습니다.

DYNAMIC

전송자는 수신자로부터 얻는 피드백에 따라 해당 전송 속도를 채택합니다. 이 경우에 전송 속도 한계는 `TransRateLimit` 매개변수에 의해 지정된 값 보다 더 많을 수 없습니다. 전송자는 최적 전송 속도에 도달하려고 합니다.

TransRateLimit

전송 비율 한계(Kbp).

SocketTTL

`SocketTTL`의 값은 멀티캐스트 트래픽이 라우터 또는 통과할 수 있는 라우터 수를 통해 전달될 수 있을지 여부를 판별합니다.

배치

메시지가 배치되거나 즉시 보내지는지 여부를 제어합니다. 가능한 값은 두 개입니다.

- NO 메시지가 배치되지 않고 바로 보내집니다.
- YES 메시지가 배치됩니다.

루프

멀티캐스트 루프를 사용으로 설정하기 위해 값을 1로 설정하십시오. 멀티캐스트 루프는 보내진 데이터가 호스트로 다시 루프되는지 여부를 정의합니다.

인터페이스

멀티캐스트 트래픽이 플로우하는 인터페이스의 IP 주소, 자세한 정보 및 문제점 해결은 멀티캐스트가 아닌 네트워크에서 멀티캐스트 애플리케이션 테스트 및 멀티캐스트 트래픽에 대해 적절한 네트워크 설정을 참조하십시오.

FeedbackMode

NACK

부정적인 수신확인에 의한 피드백. 이는 기본값입니다.

ACK

긍정적인 수신확인에 의한 피드백.

WAIT1

전송자가 임의의 수신자로부터 1 ACK만 대기하는 긍정적인 수신확인에 의한 피드백.

HeartbeatTimeout

하트비트 제한시간(밀리초). 0의 값은 하트비트 제한시간 이벤트가 수신자 또는 토픽의 수신자에 의해 제기되지 않는지 확인합니다. 기본값은 20000입니다.

HeartbeatInterval

하트비트 간격(밀리초). 0의 값은 보내진 하트비트가 없다는 것을 표시합니다. 하트비트 제한시간 초과 이벤트를 방지하려면 하트비트 간격이 **HeartbeatTimeout** 값보다 상당히 작아야 합니다. 기본값은 2000입니다.

IBM MQ LLM(Low Latency Messaging)을 가지는 멀티캐스트 상호 운용성

이 정보를 사용하여 IBM MQ 멀티캐스트 및 IBM MQ LLM(Low Latency Messaging) 사이의 상호 운용성에 대해 이해하십시오.

기본 페이로드 전송은 다른 애플리케이션이 양쪽 지시사항으로 메시지를 교환하기 위해 멀티캐스트를 사용하면, LLM을 사용하는 애플리케이션을 위해 가능합니다. 멀티캐스트가 LLM 기술을 사용하지만, LLM 제품 자체는 임베드되지 않습니다. 따라서 LLM과 IBM MQ 멀티캐스트 모두 설치할 수 있고, 개별적으로 두 개의 제품을 작동시키고 서비스를 제공할 수 있습니다.

멀티캐스트와 통신하는 LLM 애플리케이션은 메시지 특성을 보내고 수신해야 할 수도 있습니다. IBM MQ 메시지 특성과 MQMD 필드는 다음 표에 나타난 바와 같이 특정 LLM 메시지 특성 코드를 가진 LLM 메시지 특성으로 전송됩니다.

IBM MQ 특성	IBM MQ LLM 특성 유형	LLM 특성 종류	LLM 특성 코드
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008

표 20. IBM MQ LLM 특성 맵핑에 대한 IBM MQ 메시지 특성 (계속)			
IBM MQ 특성	IBM MQ LLM 특성 유형	LLM 특성 종류	LLM 특성 코드
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1013
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_String	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

LLM에 대한 자세한 정보는 LLM 제품 문서(IBM MQ LLM(Low Latency Messaging))를 참조하십시오.

IBM i IBM MQ for IBM i 관리

CL 명령은 IBM i에서 IBM MQ 를 관리하기 위해 선호되는 방법입니다. MQSC 명령, PCF 명령, 제어 명령 및 원격 관리를 사용할 수도 있습니다.

이 태스크 정보

관리 태스크에는 클러스터, 프로세스 및 IBM MQ 오브젝트(큐 관리자, 큐, 이름 목록, 프로세스 정의, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트)의 작성, 시작, 변경, 보기, 중지 및 삭제가 포함됩니다.

IBM MQ for IBM i관리 방법에 대한 세부사항은 다음 링크를 참조하십시오.

관련 개념

[IBM MQ for IBM i 큐 관리자 라이브러리 이름 이해](#)

[IBM i의 설치 가능 서비스 및 컴포넌트](#)

관련 태스크

[멀티플랫폼에서 IBM MQ 구성 정보 변경](#)

[IBM i에서 보안 설정](#)

[154 페이지의 『IBM i에서 데드-레터 큐 핸들러 호출』](#)

IBM MQ for IBM i에서는 **STRMQMDLQ** 명령을 설정하여 DLQ 핸들러를 호출합니다.

[IBM MQ for IBM i 애플리케이션으로 문제점 판별](#)

관련 참조

[시스템 및 기본 오브젝트](#)

IBM i CL 명령을 사용하여 IBM MQ for IBM i 관리

이 정보를 사용하여 IBM MQ IBM i 명령을 이해하십시오.

큐 관리자, 큐, 토픽, 채널, 이름 목록, 프로세스 정의 및 인증 정보 오브젝트와 연관되는 것을 포함하는 대부분의 IBM MQ 명령 그룹이 관련 **WRK*** 명령을 사용하여 액세스될 수 있습니다.

세트의 프린시플 명령은 **WRKMQM**입니다. 예를 들어, 이 명령을 사용하면 상태 정보와 함께 시스템에서 모든 큐 관리자 목록을 표시할 수 있습니다. 또는 각 입력 항목에 대해 다양한 옵션을 사용하여 모든 큐 관리자 특정 명령을 처리할 수 있습니다.

WRKMQM 명령에서 각 큐 관리자의 특정 영역(예: 채널, 토픽 또는 큐에 대한 작업)을 선택할 수 있고 거기에서 개별 오브젝트를 선택하십시오.

IBM MQ 애플리케이션 정의 레코딩

IBM MQ 애플리케이션을 작성하거나 사용자 정의를 할 때, 이는 IBM MQ 정의를 작성되도록 하는 데 유용합니다. 이 레코드는 다음에 사용될 수 있습니다.

- 복구 목적
- 유지보수
- IBM MQ 애플리케이션 롤 아웃

두 가지 방법 중 하나로 IBM MQ 애플리케이션 정의를 레코드할 수 있습니다.

1. 서버에 대한 IBM MQ 정의를 생성하기 위한 CL 프로그램 작성.
2. 크로스 플랫폼 IBM MQ 명령 언어를 사용하여 IBM MQ 정의를 생성하기 위해 SRC 멤버로 MQSC 텍스트 파일 작성.

큐 오브젝트를 정의하는 것에 대한 추가 세부사항은 [11 페이지의 『MQSC 명령을 사용하여 IBM MQ 관리』](#) 및 [25 페이지의 『IBM MQ PCF\(Programmable Command Format\) 사용』](#)의 내용을 참조하십시오.

관련 참조

[IBM MQ for IBM i CL 명령 참조](#)

IBM i 시작하기 전에 CL 명령을 사용하여 IBM MQ for IBM i 사용

이 정보를 사용하여 IBM MQ 서브시스템을 시작하고 로컬 큐 관리자를 작성하십시오.

시작하기 전에

IBM MQ 서브시스템이 실행 중이고(명령 STRSBS QMQM/QMQM 사용) 서브시스템과 연관되는 작업 큐가 유지되지 않는다는 것을 확인하십시오. 기본적으로, IBM MQ 서브시스템 및 작업 큐는 모두 라이브러리 QMQM에서 QMQM으로 이름 지정됩니다.

이 태스크 정보

IBM i 명령행을 사용하여 큐 관리자 시작

프로시저

1. IBM i 명령행에서 CRTMQM 명령을 실행하여 로컬 큐 관리자를 작성하십시오.
큐 관리자를 작성할 때, 해당 큐 관리자가 기본 큐 관리자를 작성하는 옵션을 가지고 있습니다. 기본 큐 관리자(유일하게 하나일 수 있음)는 큐 관리자 이름 매개변수(MQMNAME)이 생략되는 경우 CL 명령이 적용되는 큐 관리자입니다.
2. IBM i 명령행에서 STRMQM 명령을 실행하여 로컬 큐 관리자를 시작하십시오.
큐 관리자 시동에 몇 초 이상 걸리는 경우 IBM MQ에서는 시동 진행 상황을 자세히 보여주는 상태 메시지를 간헐적으로 표시합니다. 이러한 메시지에 대한 자세한 정보는 [메시지 및 이유 코드를 참조하십시오](#).

다음에 수행할 작업

IBM i 명령행에서 ENDMQM 명령을 실행하여 큐 관리자를 중지하고 IBM i 명령행에서 다른 IBM MQ 명령을 실행하여 큐 관리자를 제어할 수 있습니다.

리모트 큐 관리자는 원격으로 시작될 수 없지만 로컬 연산자로서 해당 시스템에서 작성되고 시작되어야 합니다. 이에 대한 예외는 오퍼레이팅 기능(IBM MQ for IBM i 외부)이 그런 조작을 사용 가능하게 하도록 존재하는 위치입니다.

로컬 큐 관리자는 리모트 큐 관리자를 중지할 수 없습니다.

참고: IBM MQ 시스템을 정지하는 것의 일부로서 활성 큐 관리자를 일시정지해야 합니다. 이 프로그램은 [410 페이지의 『IBM MQ for IBM i 일시정지』](#)에서 설명합니다.

이 정보를 사용하여 IBM i용 IBM MQ 오브젝트를 작성하는 방법을 이해하십시오.

시작하기 전에

다음 태스크는 명령행에서 IBM MQ for IBM i를 사용할 수 있는 다양한 방법을 제안합니다.

이 태스크 정보

IBM MQ 오브젝트를 작성하기 위한 두 가지 온라인 방법이 있으며, 다음과 같습니다.

프로시저

1. 작성 명령 사용 (예: **Create MQM Queue** 명령: **CRTMQMQ**)
2. F6이 뒤에 오는 MQM 오브젝트에 대한 작업 명령 사용 (예: **Work with MQM Queues** 명령: **WRKMQMQ**)

다음에 수행할 작업

모든 명령의 목록의 경우, [IBM MQ for IBM i CL 명령](#)의 내용을 참조하십시오.

참고: 모든 MQM 명령은 메시지 큐 관리자 명령 메뉴에서 제출될 수 있습니다. 이 메뉴를 표시하려면 명령행에 **GO CMDMQM**을 입력하고 **Enter** 키를 누르십시오.

명령을 이 메뉴에서 선택할 때 시스템은 자동으로 신속한 패널을 표시합니다. 직접적으로 명령행에 입력한 명령을 위한 신속한 패널을 표시하려면, **Enter** 키를 누르기 전에 **F4**를 누르십시오.

CRTMQMQ 명령을 사용하여 로컬 큐 작성

프로시저

1. 명령행에서 **CHGMQM**을 입력하고 **F4** 키를 누르십시오.
2. **MQM**큐 작성 패널에서 Queue name 필드에 작성할 큐의 이름을 입력하십시오. 혼합된 사건명을 지정하려면, 이름을 아포스트로피로 묶으십시오.
3. Queue type 필드에 *LCL 를 입력하십시오.
4. 기본 큐 관리자를 사용하지 않으면 큐 관리자 이름을 지정하고 **Enter** 키를 누르십시오. 값을 새 값으로 겹쳐 쓸 수 있습니다. 추가 필드를 보려면 앞으로 이동하십시오. 클러스터에 사용되는 옵션은 옵션 목록의 끝에 있습니다.
5. 값을 변경했을 때, **Enter** 키를 눌러 큐를 작성하십시오.

WRKMQMQ 명령을 사용하여 로컬 큐 작성

프로시저

1. 명령행에서 **WRKMQMQ**를 입력하십시오.
2. 큐 관리자의 이름을 입력하십시오.
3. 프롬프트 패널을 표시하려는 경우, **F4**를 누르십시오. 프롬프트 패널은 일반 큐 이름 또는 큐 유형을 지정하여 표시되는 큐의 수를 감소시키는 데 유용합니다.
4. **Enter** 를 누르면 **MQM**큐에 대한 작업 패널 이 표시됩니다. 새 값으로 값을 겹쳐 쓸 수 있습니다. 추가 필드를 보려면 앞으로 이동하십시오. 클러스터에 사용되는 옵션은 옵션 목록의 끝에 있습니다.
5. **F6**을 눌러 새 큐를 작성하십시오. 이는 **CRTMQMQ** 패널로 안내합니다. 큐 작성 방법에 대한 지시사항은 [347 페이지의 『CRTMQMQ 명령을 사용하여 로컬 큐 작성』](#)의 내용을 참조하십시오. 큐를 작성했을 때, **MQM** 큐에 대한 작업 패널이 다시 표시됩니다. **F5=Refresh**를 누를 때 새 큐가 목록에 추가됩니다.

큐 관리자 속성 대체

이 태스크 정보

CHGMQM 명령에서 지정되는 큐 관리자의 속성을 대체하려면 변경하려는 속성 및 값을 지정하십시오. 예를 들어, 다음 옵션을 사용하여 `jupiter.queue.manager`의 속성을 변경하십시오.

프로시저

명령행에서 **CHGMQM**을 입력하고 F4 키를 누르십시오.

결과

명령은 사용되는 데드-레터 큐를 변경하고 이벤트 억제를 사용합니다.

IBM i IBM i에서 로컬 큐에 대한 작업

이 섹션에는 로컬 큐를 관리하기 위해 사용할 수 있는 일부 명령에 대한 예가 포함됩니다. 표시되는 모든 명령은 **WRKMQMQ** 명령 패널에서 옵션을 사용하여 사용 가능합니다.

로컬 큐 정의

애플리케이션의 경우, 로컬 큐 관리자는 애플리케이션이 연결되는 큐 관리자입니다. 로컬 큐 관리자에 의해 관리되는 큐는 해당 큐 관리자에 대해 로컬이라고 합니다.

CRTMQMQ QTYPE *LCL 명령을 사용하여 로컬 큐의 정의를 작성하고 큐라고 하는 데이터 구조도 작성하십시오. 기본 로컬 큐의 특성에서 큐 특성을 수정할 수도 있습니다.

이 예에서 정의하는 큐(`orange.local.queue`)가 해당 특성으로 지정됩니다.

- 가져오기(GET)에 사용되고, 넣기(PUT)에 사용 불가능하고 FIFO(First-In-First-Out) 기본에서 실행됩니다.
- 이는 원래 큐입니다. 즉, 이는 이니시에이션 큐 또는 전송 큐가 아니고 트리거 메시지를 생성하지 않습니다.
- 최대 큐 용량은 1000개의 메시지입니다. 최대 메시지 길이는 2000바이트입니다.

다음 명령은 기본 큐 관리자에서 이를 수행합니다.

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL)
TEXT('Queue for messages from other systems')
PUTENBL(*NO)
GETENBL(*YES)
TRGENBL(*NO)
MSGDLYSEQ(*FIFO)
MAXDEPTH(1000)
MAXMSGLEN(2000)
USAGE(*NORMAL)
```

참고:

1. USAGE *NORMAL에서는 이 큐가 전송 큐가 아니라는 것을 표시합니다.
2. 동일한 큐 관리자에서 `orange.local.queue` 이름을 가지는 로컬 큐가 이미 있는 경우, 이 명령에 실패합니다. 큐의 기존 정의를 덮어쓰려는 경우 REPLACE *YES 속성을 사용하고 [349 페이지의 『로컬 큐 속성 변경』](#)의 내용도 참조하십시오.

데드-레터 큐 정의

각 큐 관리자는 올바른 목적지로 전달될 수 없는 메시지가 나중에 검색을 위해 저장될 수 있도록 로컬 큐가 데드-레터 큐로서 사용되도록 해야 합니다. 데드-레터 큐에 대한 큐 관리자를 명확하게 구별해야 합니다. **CRTMQMQ** 명령에서 데드-레터 큐를 지정하여 이를 수행할 수 있거나 **CHGMQM** 명령을 사용하여 나중에 하나를 지정할 수 있습니다. 사용될 수 있으려면 데드-레터 큐도 정의해야 합니다.

SYSTEM.DEAD.LETTER.QUEUE라는 샘플 데드-레터 큐가 제품과 함께 제공됩니다. 이 큐는 큐 관리자를 작성할 때 자동으로 작성됩니다. 필요한 경우 이 정의를 수정할 수 있습니다. 사용자가 좋아하는 경우 사용할 수 있을 지라도 이름을 바꿀 필요는 없습니다.

데드-레터 큐에는 다음을 제외하고 특별한 요구사항이 없습니다.

- 이는 로컬 큐여야 합니다.
- MAXMSGL(최대 메시지 길이) 속성은 큐가 데드-레터 헤더(MQDLH)의 크기와 함께 큐 관리자가 핸들링해야 하는 가장 큰 메시지를 수용할 수 있도록 설정해야 합니다.

IBM MQ에서는 데드-레터 큐에서 발견되는 메시지가 처리되거나 제거되는 방법을 지정할 수 있도록 데드-레터 큐 핸들러를 제공합니다. 추가 정보는 [154 페이지의 『IBM i에서 데드-레터 큐 핸들러 호출』](#)의 내용을 참조하십시오.

기본 오브젝트 속성 표시

IBM MQ 오브젝트를 정의할 때, 기본 오브젝트에서 지정하지 않는 속성을 사용합니다. 예를 들어, 로컬 큐를 정의할 때, 큐는 SYSTEM.DEFAULT.LOCAL.QUEUE라는 기본 로컬 큐의 정의에서 생략하는 속성을 상속합니다. 이러한 속성의 개념을 정확하게 보려면 다음 명령을 참조하십시오.

```
DSPMQMQ QNAME(SYSTEM.DEFAULT.LOCAL.QUEUE) MQMNAME(MYQUEUEMANAGER)
```

로컬 큐 정의 복사

CPYMQMQ 명령을 사용하여 큐 정의를 복사할 수 있습니다. 예를 들면, 다음과 같습니다.

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

이 명령은 시스템 기본 로컬 큐의 속성 보다 원래 큐 orange.local.queue와 동일한 속성을 가진 큐를 작성합니다.

CPYMQMQ 명령을 사용하여 큐 정의를 복사할 수도 있지만, 원래의 속성에 대한 하나 이상의 변경사항 대체를 사용할 수도 있습니다. 예를 들면, 다음과 같습니다.

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)  
MAXMSGLEN(1024)
```

이 명령은 큐 orange.local.queue의 속성을 큐 third.queue로 복사하지만, 새 큐의 최대 메시지 길이가 2000바이트라기 보다는 1024바이트가 되도록 지정합니다.

참고: **CPYMQMQ** 명령을 사용할 때, 큐의 메시지가 아니라 큐 속성만 복사합니다.

로컬 큐 속성 변경

REPLACE *YES 속성으로 **CHGMQMQ** 명령 또는 **CPYMQMQ** 명령을 사용하여, 두 가지 방법으로 큐 속성을 변경할 수 있습니다. [348 페이지의 『로컬 큐 정의』](#)에서 orange.local.queue를 정의했습니다. 예를 들면, 10,000 바이트에 이 큐의 최대 메시지 길이를 증가시킬 필요가 있는 경우.

- **CHGMQMQ** 명령 사용:

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

이 명령은 단일 속성, 즉 최대 메시지 길이의 속성을 변경합니다. 기타 모든 속성이 동일하게 유지됩니다.

- REPLACE *YES 옵션으로 **CRTMQMQ** 명령 사용. 예:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)  
MAXMSGLEN(10000) REPLACE(*YES)
```

이 명령은 최대 메시지 길이가 아닌 기타 모든 속성을 변경시키고, 기본값이 제공됩니다. 이전에는 큐를 넣을 (put) 수 없었지만, 이제는 넣을(put) 수 있습니다. 변경하지 않으면 큐 SYSTEM.DEFAULT.LOCAL.QUEUE에 의해 지정된 대로 가능한 넣기가 기본값이 됩니다.

기존 큐의 최대 메시지 길이를 **줄일 경우** 기존 메시지는 영향을 받지 않습니다. 그러나, 새 메시지는 새 기준을 충족해야 합니다.

로컬 큐 지우기

magenta.queue라는 로컬 큐에서 모든 메시지를 삭제하려면 다음 명령을 사용하십시오.

```
CLRMQM QNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

다음의 경우 큐를 지울 수 없습니다.

- 동기점 아래에 큐에 걸린 미확약된 메시지가 있습니다.
- 애플리케이션이 현재 큐를 열어 두었습니다.

로컬 큐 삭제

DLTMQM 명령을 사용하여 로컬 큐를 삭제하십시오.

큐에 미확약된 메시지가 있는 경우 또는 사용 중인 경우 큐는 삭제될 수 없습니다.

큰 큐를 사용으로 설정

IBM MQ에서는 2GB보다 큰 큐를 지원합니다. IBM i가 큰 파일을 지원할 수 있게 하는 방법에 대한 정보는 운영 체제 문서를 참조하십시오.

IBM i 제품 정보는 [IBM Documentation](#)에서 찾을 수 있습니다.

일부 유틸리티는 2GB 보다 큰 파일을 대체할 수 없습니다. 큰 파일 지원을 사용으로 설정하기 전에, 그와 같은 지원에 대한 제한에 대한 정보는 사용자의 운영 체제 문서를 확인하십시오.

IBM i IBM i에서 알리어스 큐에 대한 작업

이 절에는 알리어스 큐를 관리하기 위해 사용할 수 있는 일부 명령의 예가 포함됩니다. 표시되는 모든 명령은 **WRKMQM** 명령 패널에서 옵션을 사용하여 사용 가능합니다.

알리어스 큐(가끔 큐 알리어스로 알려짐)에서는 MQI 호출의 경로를 재지정하는 메소드를 제공합니다. 알리어스 큐는 실제 큐가 아니라 실제 큐로 해석되는 정의입니다. 알리어스 큐 정의에는 대상 큐 이름이 포함됩니다. 이는 TGTQNAME 속성에 의해 지정됩니다.

애플리케이션이 MQI 호출에서 알리어스 큐를 지정할 때, 큐 관리자는 런타임 시 실제 큐 이름을 해석합니다.

예를 들어, 애플리케이션은 my.alias.queue라고 하는 큐에 메시지를 배치하기 위해 개발됩니다. **MQOPEN** 요청을 작성할 때와 간접적으로 이 큐에 메시지를 배치하는 경우 이 큐의 이름을 지정합니다. 애플리케이션은 큐가 알리어스 큐라는 것을 인지하지 않습니다. 이 알리어스를 사용하는 각 MQI 호출의 경우, 큐 관리자는 실제 큐 이름을 해석합니다. 이는 이 큐 관리자에 정의되는 로컬 큐 또는 리모트 큐가 될 수 있습니다.

TGTQNAME 속성의 값을 변경하여, 다른 큐 관리자에서 사용 가능한 다른 큐로 MQI 호출의 경로를 재지정할 수 있습니다. 이는 유지보수, 마이그레이션 및 로드 밸런싱에 유용합니다.

알리어스 큐 정의

다음 명령은 알리어스 큐를 작성합니다.

```
CRTMQM QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')  
MQMNAME(MYQUEUEMANAGER)
```

이 명령은 `my.alias.queue`를 큐 `yellow.queue`로 지정하는 MQI 호출의 경로를 재지정합니다. 명령은 대상 큐를 작성하지 않습니다. MQI 호출은 `yellow.queue`가 런타임 시 존재하지 않는 경우 실패합니다.

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

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

이 명령은 MQI 호출의 경로를 다른 큐 `magenta.queue`로 재지정합니다.

알리어스 큐를 사용하여 단일 큐(대상 큐)가 다른 애플리케이션에 대해 다른 속성을 갖는 것처럼 보이도록 할 수 있습니다. 각 애플리케이션에 하나씩 두 개의 알리어스를 정의하여 이를 수행합니다. 두 개의 애플리케이션이 있다고 가정하십시오.

- 애플리케이션 ALPHA는 `yellow.queue`에 메시지를 배치할 수 있지만, 여기에서 메시지를 가져올 수 없습니다.
 - 애플리케이션 BETA는 `yellow.queue`에서 메시지를 가져올 수 있지만, 여기에 메시지를 배치할 수 없습니다.
- 다음 명령을 사용하여 이를 수행할 수 있습니다.

```
/* This alias is put enabled and get disabled for application ALPHA */
CRTMQM QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQM QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEMANAGER)
```

ALPHA는 MQI 호출에서 큐 이름 `alphas.alias.queue`를 사용합니다. BETA는 큐 이름 `betas.alias.queue`를 사용합니다. 이는 모두 동일한 큐에 액세스하지만, 다른 방법으로 액세스합니다. 로컬 큐로 사용하는 동일한 방법으로 알리어스 큐를 정의할 때 `REPLACE *YES` 속성을 사용할 수 있습니다.

알리어스 큐로 기타 명령 사용

적절한 명령을 사용하여 알리어스 큐 속성을 표시하거나 변경할 수 있습니다. 예를 들면, 다음과 같습니다.

```
* Display the alias queue's attributes */
DSPMQM QNAME('alphas.alias.queue') MQMNAME(MYQUEUEMANAGER)

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

CHQMOM QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEMANAGER)
```

IBM i IBM i에서 모델 큐에 대한 작업

이 섹션에는 모델 큐를 관리하기 위해 사용할 수 있는 일부 명령의 예가 포함됩니다. 표시되는 모든 명령은 **WRKMQM 명령 패널**에서 옵션을 사용하여 사용 가능합니다.

큐 관리자는 모델 큐로서 정의되는 큐 이름을 지정하여 애플리케이션에서 MQI를 수신하는 경우 동적 큐를 작성합니다. 새 동적 큐의 이름은 큐가 작성될 때 큐 관리자에서 생성됩니다. 모델 큐는 작성되는 동적 큐의 속성을 지정하는 템플릿입니다.

모델 큐에서는 요청될 때 큐를 작성하기 위한 애플리케이션에 편리한 메소드를 제공합니다.

모델 큐 정의

로컬 큐를 정의하는 동일한 방법으로 속성 세트와 함께 모델 큐를 정의합니다. 모델 큐 및 로컬 큐에는 작성되는 동적 큐가 임시 또는 영구적인지 여부를 지정할 수 있는 모델 큐의 속성을 제외하고 동일한 속성 세트가 있습니다.

다. (영구적 큐는 큐 관리자 재시작을 통해 유지보수되고 임시 큐는 그렇지 않습니다). 예를 들면, 다음과 같습니다.

```
CRTMQMQ QNAME('green.model.queue') QTYPE(*MDL) DFNTYPE(*PERMDYN)
```

이 명령은 모델 큐 정의를 작성합니다. DFNTYPE 속성에서 이 템플릿에서 작성된 실제 큐는 영구적 동적 큐입니다. 지정되지 않은 속성은 SYSYSTEM.DEFAULT.MODEL.QUEUE 기본 큐에서 자동적으로 복사됩니다.

로컬 큐로 사용하는 동일한 방법으로 모델 큐를 정의할 때 REPLACE *YES 속성을 사용할 수 있습니다.

모델 큐로 기타 명령 사용

적절한 명령을 사용하여 모델 큐의 속성을 표시하거나 대체할 수 있습니다. 예를 들면, 다음과 같습니다.

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

IBM i IBM i에서 트리거링에 대한 작업

이 정보를 사용하여 트리거에 대해 학습하고 정의를 처리하십시오.

IBM MQ에서는 큐에 대한 특정 조건이 충족될 때 자동으로 애플리케이션을 시작하기 위한 기능을 제공합니다. 조건의 한 예는 큐에 대한 메시지 수가 지정된 수에 도달할 때입니다. 이 기능은 트리거라고 하고 [트리거 채널](#)에서 상세하게 설명됩니다.

트리거 개념

큐 관리자는 트리거 이벤트를 구성할 때 특정 조건을 정의합니다. 트리거가 큐에 사용되도록 설정되고 트리거 이벤트가 발생하는 경우, 큐 관리자는 트리거 메시지를 이니시에이션 큐라고 하는 큐에 보냅니다. 이니시에이션 큐에 있는 트리거 메시지는 트리거 이벤트가 발생했음을 표시합니다.

큐 관리자에서 생성된 트리거 메시지는 지속적이지 않습니다. 이는 로깅을 줄이고(성능 개선) 재시작 동안 복제를 최소화하는 효과가 있으므로 재시작 시간을 향상시키는 효과가 있습니다.

트리거 모니터의 개념

이니시에이션 큐를 처리하는 프로그램은 트리거 모니터 애플리케이션이라고 하고 해당 기능은 트리거 메시지에 포함된 정보를 기반으로 트리거 메시지를 읽고 적절한 조치를 수행하는 것입니다. 일반적으로 이 조치는 트리거 메시지가 생성되도록 하는 큐를 처리하기 위해 일부 기타 애플리케이션을 시작하는 것입니다. 큐 관리자의 관점에서 트리거 모니터 애플리케이션에 대해 특별한 것이 없고, 큐(이니시에이션 큐)에서 메시지를 읽는 것은 다른 애플리케이션입니다.

트리거 모니터의 작업 제출 속성 대체

STRMQMTRM 명령으로 제공된 트리거 모니터는 시스템 기본 작업 설명(QDFTJOBDD)을 사용하여 각 트리거 메시지에 대한 작업을 제출합니다. 이는 제출된 작업을 항상 QDFTJOBDD라고 하고 라이브러리 목록(*SYSVAL)을 포함하여 기본 작업 설명의 속성을 가지고 있다는 점에서 제한사항이 있습니다. IBM MQ에서는 이러한 속성을 대체하기 위한 메소드를 제공합니다. 예를 들어, 다음과 같이 더 의미있는 작업 이름을 가지고 있도록 제출된 작업을 사용자 정의할 수 있습니다.

1. 작업 설명에서 사용자가 원하는 설명을 지정하십시오(예: 로깅 값).
2. 트리거 프로세스에서 사용되는 프로세스 정의의 환경 데이터를 지정하십시오.

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

지정된 설명을 사용하여 트리거 모니터가 SBMJOB을 수행합니다.

프로세스 정의의 환경 데이터에서 적절한 키워드 및 값을 지정하여 SBMJOB의 기타 속성을 대체할 수 있습니다. 이 속성이 트리거 모니터에 의해 채워지기 때문에 이에 대한 유일한 예외는 CMD 키워드입니다. 작업 이름과 설명 모두 변경되는 프로세스 정의의 환경 데이터를 지정하기 위한 명령의 예는 다음과 같습니다.

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)
JOB(TRIGGER)')
```

트리거에 대한 애플리케이션 큐 정의

애플리케이션 큐는 MQI를 통해 메시징을 위해 애플리케이션에서 사용하는 로컬 큐입니다. 트리거는 애플리케이션 큐에 정의되는 수많은 큐 속성이 필요합니다. 트리거 자체가 TRGENBL 속성에 의해 사용됩니다.

이 예에서 트리거 이벤트는 다음과 같이 로컬 큐(`motor.insurance.queue`)에 우선순위가 5 이상인 메시지가 100가지가 있을 때 생성되는 것입니다.

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)
PRCNAME('motor.insurance.quote.process') MAXMSGLEN(2000)
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')
TRGENBL(*YES) TRGTYPE(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

여기서 매개변수는 다음과 같습니다.

MQMNAME(MYQUEUEMANAGER)

큐 관리자 이름.

QNAME('motor.insurance.queue')

정의되는 애플리케이션 큐의 이름.

PRCNAME('motor.insurance.quote.process')

트리거 모니터 프로그램에 의해 시작되는 애플리케이션의 이름.

MAXMSGLEN(2000)

큐의 최대 메시지 길이.

DFTMSGPST(*YES)

기본적으로 이 큐의 메시지가 지속됩니다.

INITQNAME('motor.ins.init.queue')

큐 관리자가 트리거 메시지를 사용하는 이니시에이션 큐의 이름.

TRGENBL(*YES)

트리거 속성 값.

TRGTYPE(*DEPTH)

트리거 이벤트는 필수 우선순위(**TRGMSGPTY**)의 메시지 수가 **TRGDEPTH**에 지정되는 수에 도달할 때 생성됩니다.

TRGDEPTH(100)

트리거 이벤트를 생성하기 위해 필요한 메시지 수.

TRGMSGPTY(5)

트리거 이벤트를 생성할지 여부를 결정하는 큐 관리자에 의해 계산되는 메시지의 우선순위. 우선순위 5 이상인 메시지 수만 계수됩니다.

이니시에이션 큐 정의

트리거 이벤트가 발생할 때, 큐 관리자는 애플리케이션 큐 정의에 지정되는 이니시에이션 큐에 트리거 메시지를 배치합니다. 이니시에이션 큐에 특별한 설정이 없지만, 자세한 내용을 위해 로컬 큐 `motor.ins.init.queue` 의 다음 정의를 사용할 수 있습니다.

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.ins.init.queue') QTYPE(*LCL)
GETENBL(*YES) SHARE(*NO) TRGTYPE(*NONE)
MAXMSGL(2000)
MAXDEPTH(1000)
```

프로세스 정의 작성

CRTMQMPCRC 명령을 사용하여 프로세스 정의를 작성하십시오. 프로세스 정의는 큐로부터 메시지를 처리하는 애플리케이션과 애플리케이션 큐를 연관시킵니다. 이는 애플리케이션 큐 `motor.insurance.queue` 에서 **PRCNAME** 속성을 통해 수행됩니다. 다음 명령은 이 예에서 식별되는 필수 프로세스 (`motor.insurance.quote.process`)를 작성합니다.

```
CRTMQMPCRC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')
TEXT('Insurance request message processing')
APPTYPE(*OS400) APPID(MQTEST/TESTPROG)
USRDATA('open, close, 235')
```

여기서 매개변수는 다음과 같습니다.

MQMNAME(MYQUEUEMANAGER)

큐 관리자 이름.

PRCNAME('motor.insurance.quote.process')

프로세스 정의의 이름.

TEXT('Insurance request message processing')

이 정의가 관련되는 애플리케이션 프로그램에 대한 설명. **DSPMQMPCRC** 명령을 사용할 때 이 텍스트가 표시됩니다. 이는 프로세스가 수행하는 것을 식별하는 데 도움이 될 수 있습니다. 문자열에서 공백을 사용하는 경우, 작은따옴표로 문자열을 묶어야 합니다.

APPTYPE(*OS400)

시작할 애플리케이션의 유형.

APPID(MQTEST/TESTPROG)

완전한 파일 이름으로 지정되는 애플리케이션 실행 파일의 이름.

USRDATA('open, close, 235')

애플리케이션이 사용할 수 있는 사용자 정의 데이터.

프로세스 정의 표시

DSPMQMPCRC 명령을 사용하여 정의의 결과를 조사하십시오. 예를 들면, 다음과 같습니다.

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPCRC('motor.insurance.quote.process')
```

CHGMQMPCRC 명령을 사용하여 기존 프로세스 정의를 변경하고 **DLTMQMPCRC** 명령을 사용하여 프로세스 정의를 삭제할 수도 있습니다.

IBM i IBM i의 두 IBM MQ시스템 간 통신

이 코딩 예는 두 IBM MQ for IBM i 시스템이 서로 통신하도록 CL 명령을 사용하여 설정하는 방법을 설명합니다. 시스템은 SYSTEMA 및 SYSTEMB라고 하며 사용되는 통신 프로토콜은 TCP/IP입니다.

다음 프로시저를 수행하십시오.

1. SYSTEMA에서 큐 관리자를 작성하십시오. 이를 QMGRA1이라고 합니다.

```
CRTMQM  MQMNAME(QMGRA1) TEXT('System A - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. 이 큐 관리자를 시작하십시오.

```
STRMQM  MQMNAME(QMGRA1)
```

3. SYSTEMB의 큐 관리자에 메시지를 송신해야 하는 IBM MQ 오브젝트를 SYSTEMA에 정의하십시오.

```
/* Transmission queue */
CRTMQMQ  QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +
MQMNAME(QMGRA1) TEXT('Transmission Queue +
to QMGRB1') MAXDEPTH(5000) USAGE(*TMQ)

/* Remote queue that points to a queue called TARGETB */
/* TARGETB belongs to queue manager QMGRB1 on SYSTEMB */
CRTMQMQ  QNAME(TARGETB.ON.QMGRB1) QTYPE(*RMT) +
MQMNAME(QMGRA1) TEXT('Remote Q pointing +
at Q TARGETB on QMGRB1 on Remote System +
SYSTEMB') RMTQNAME(TARGETB) +
RMTMQMNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)

/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +
MQMNAME(QMGRA1) TRPTYPE(*TCP) +
TEXT('Sender Channel From QMGRA1 on +
SYSTEMA to QMGRB1 on SYSTEMB') +
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. SYSTEMB에서 큐 관리자를 작성하십시오. 이를 QMGRB1이라고 합니다.

```
CRTMQM  MQMNAME(QMGRB1) TEXT('System B - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

5. SYSTEMB에서 큐 관리자를 시작하십시오.

```
STRMQM  MQMNAME(QMGRB1)
```

6. SYSTEMA의 큐 관리자에서 메시지를 수신하는 데 필요한 IBM MQ 오브젝트를 정의하십시오.

```
/* Local queue to receive messages on */
CRTMQMQ  QNAME(TARGETB) QTYPE(*LCL) MQMNAME(QMGRB1) +
TEXT('Sample Local Queue for QMGRB1')

/* Receiver channel of the same name as the sender channel on SYSTEMA */
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*RCVR) +
MQMNAME(QMGRB1) TRPTYPE(*TCP) +
TEXT('Receiver Channel from QMGRA1 to +
QMGRB1')
```

7. 마지막으로 채널이 시작될 수 있도록 SYSTEMB에서 TCP/IP 리스너를 시작하십시오. 이 예는 1414의 기본 포트를 사용합니다.

```
STRMQMLSR MQMNAME(QMGRB1)
```

지금 SYSTEMA 및 SYSTEMB 사이에 테스트 메시지를 보낼 준비가 됩니다. 제공된 샘플 중 하나를 사용함으로써 SYSTEMA의 리모트 큐에 일련의 메시지를 두십시오.

STRMQMCHL 명령을 사용하거나 **WRKMQMCHL** 명령을 사용하고 송신자 채널에 대한 시작 요청(옵션 14)을 입력하여 SYSTEMA에서 채널을 시작하십시오.

채널은 RUNNING 상태로 이동해야 하고 메시지는 SYSTEMB의 큐 TARGETB로 보내집니다.

명령을 실행하여 메시지를 검사하십시오.

```
WRKMQMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

IBM i IBM i에서 샘플 자원 정의

이 샘플은 AMQSAMP4 샘플 IBM i CL 프로그램을 포함합니다.

```
/* **** */
/* */
/* * Program name: AMQSAMP4 */
/* */
/* * Description: Sample CL program defining MQM queues */
/* * to use with the sample programs */
/* * Can be run, with changes as needed, after */
/* * starting the MQM */
/* */
/* * <N_OCO_COPYRIGHT> */
/* * Licensed Materials - Property of IBM */
/* */
/* * 63H9336 */
/* * (c) Copyright IBM Corp. 1993, 2024. All Rights Reserved. */
/* * US Government Users Restricted Rights - Use, duplication or */
/* * disclosure restricted by GSA ADP Schedule Contract with */
/* * IBM Corp. */
/* * <NOC_COPYRIGHT> */
/* */
/* **** */
/* * Function: */
/* */
/* * AMQSAMP4 is a sample CL program to create or reset the */
/* * MQI resources to use with the sample programs. */
/* */
/* * This program, or a similar one, can be run when the MQM */
/* * is started - it creates the objects if missing, or resets */
/* * their attributes to the prescribed values. */
/* */
/* */
/* * Exceptions signaled: none */
/* * Exceptions monitored: none */
/* */
/* * AMQSAMP4 takes a single parameter, the Queue Manager name */
/* */
/* **** */
QSYS/PGM PARM(&QMGRNAME)

/* **** */
/* * Queue Manager Name Parameter */
/* **** */
QSYS/DCL VAR(&QMGRNAME) TYPE(*CHAR)

/* **** */
/* * EXAMPLES OF DIFFERENT QUEUE TYPES */
/* */
/* * Create local, alias and remote queues */
/* */
/* * Uses system defaults for most attributes */
/* */
/* **** */
/* * Create a local queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.LOCAL') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Sample local queue') /* description */+
SHARE(*YES) /* Shareable */+
DFTMSGPST(*YES) /* Persistent messages OK */

/* * Create an alias queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ALIAS') +
MQMNAME(&QMGRNAME) +
```

```

QTYPE(*ALS) REPLACE(*YES)      +
+
TEXT('Sample alias queue')      +
DFTMSGPST(*YES) /* Persistent messages OK */+
TGTQNAME('SYSTEM.SAMPLE.LOCAL')

/* Create a remote queue - in this case, an indirect reference */
/* is made to the sample local queue on OTHER queue manager */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REMOTE') +
MQMNAME(&QMGRNAME) +
QTYPE(*RMT) REPLACE(*YES)      +
+
TEXT('Sample remote queue')/* description */+
DFTMSGPST(*YES) /* Persistent messages OK */+
RMTQNAME('SYSTEM.SAMPLE.LOCAL') +
RMTMQMNAME(OTHER) /* Queue is on OTHER */

/* Create a transmission queue for messages to queues at OTHER */
/* By default, use remote node name */
CRTMQMQ QNAME('OTHER') /* transmission queue name */+
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Transmission queue to OTHER') +
USAGE(*TMQ) /* transmission queue */

/*****
/* SPECIFIC QUEUES AND PROCESS USED BY SAMPLE PROGRAMS */
/*
/* Create local queues used by sample programs */
/* Create MQI process associated with sample initiation queue */
/*
/*****
/* General reply queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REPLY') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('General reply queue') +
DFTMSGPST(*NO) /* Not Persistent */

/* Queue used by AMQSINQ4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.INQ') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSINQ4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO) /* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSSET4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.SET') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSSET4') +
SHARE(*YES) /* Shareable */ +
DFTMSGPST(*NO)/* Not Persistent */ +
+
TRGENBL(*YES) /* Trigger control on */ +
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSECH4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ECHO') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSECH4') +
SHARE(*YES) /* Shareable */ +
DFTMSGPST(*NO)/* Not Persistent */ +
+
TRGENBL(*YES) /* Trigger control on */ +
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

```

```

/* Initiation Queue used by AMQSTRG4, sample trigger process */
CRTMQMQ QNAME('SYSTEM.SAMPLE.TRIGGER') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Trigger queue for sample programs')

/* MQI Processes associated with triggered sample programs */
/*
/***** Note - there are versions of the triggered samples *****/
/***** in different languages - set APPID for these *****/
/***** process to the variation you want to trigger *****/
/*
CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSINQ4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSINQ4') /* C */ +
/* APPID('QMOM/AMQOINQ4') /* COBOL */ +
/* APPID('QMOM/AMQ3INQ4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSSET4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSSET4') /* C */ +
/* APPID('QMOM/AMQOSET4') /* COBOL */ +
/* APPID('QMOM/AMQ3SET4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
TEXT('Trigger process for AMQSECH4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSECH4') /* C */ +
/* APPID('QMOM/AMQOECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****
/*
/* Normal return.
/*
/*****
SNDPGMMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM

/*****
/*
/* END OF AMQSAMP4
/*
/*****

```

IBM i IBM MQ for IBM i 관리의 대체 방법

IBM MQ for IBM i를 관리하는 데 선호하는 방법은 CL 명령 사용입니다. 그러나 MQSC 명령, PCF 명령, 제어 명령 및 원격 관리를 포함하여 다양한 기타 관리 방법을 사용할 수 있습니다.

이 태스크 정보

일반적으로 IBM i CL 명령을 사용하여 IBM MQ for IBM i를 관리합니다. 이러한 명령에 대한 개요는 345 페이지의 [『CL 명령을 사용하여 IBM MQ for IBM i 관리』](#)의 내용을 참조하십시오.

하위 주제에서 설명한 대로 MQSC 명령 및 PCF 명령을 사용할 수도 있고 10 페이지의 [『제어 명령을 사용하여 IBM MQ for Multiplatforms 관리』](#)에서 설명한 대로 제어 명령을 사용할 수도 있습니다.

IBM MQ 도구 이벤트를 사용하여 큐 관리자의 조작을 모니터링할 수 있습니다. IBM MQ 도구 이벤트 및 이를 사용하는 방법에 대한 정보는 [도구 이벤트](#)를 참조하십시오.

IBM i CL 명령 사용에 대한 대안으로 다음 하위 주제에서 설명하는 관리 방법을 사용하십시오.

IBM i IBM i에서 로컬 및 원격 관리

IBM MQ for IBM i 오브젝트를 로컬 또는 원격으로 관리합니다.

이 태스크 정보

로컬 관리는 로컬 시스템에 정의한 큐 관리자에서 관리 태스크를 실행한다는 것을 의미합니다. IBM MQ에서 IBM MQ 채널이 관련되지 않기 때문에 로컬 관리로서 이를 고려할 수 있습니다. 즉, 통신이 운영 체제에 의해 관리됩니다. 이 유형의 태스크를 수행하려면 원격 시스템에 로그인하여 거기서 명령을 발행하거나, 명령을 자동으로 발행할 수 있는 프로세스를 작성해야 합니다.

IBM MQ에서는 원격 관리로서 알려진 것을 통해 단일 지점에서 관리를 지원합니다. 원격 관리는 대상 큐 관리자에서 프로그래밍 가능 명령 형식(PCF) 제어 메시지를 SYSTEM.ADMIN.COMMAND.QUEUE로 보내는 것으로 구성됩니다.

PCF 메시지를 생성하는 여러 방법이 있습니다. 이에 대해서는 다음 단계에서 설명합니다.

프로시저

- PCF 메시지를 사용하여 프로그램을 작성합니다. 360 페이지의 『IBM i에서 PCF 명령을 사용하여 관리』의 내용을 참조하십시오.
- PCF 메시지를 송신하는 MQAI를 사용하여 프로그램을 작성하십시오. 35 페이지의 『MQAI를 사용하여 PCF의 사용 단순화』의 내용을 참조하십시오.
- IBM MQ for Windows에서 사용 가능한 IBM MQ 탐색기를 사용하여 그래픽 사용자 인터페이스 (GUI) 를 사용하고 올바른 PCF 메시지를 생성할 수 있습니다. 361 페이지의 『IBM MQ for IBM i와 함께 IBM MQ Explorer 사용』의 내용을 참조하십시오.
- **STRMQMQSC**를 사용하여 명령을 리모트 큐 관리자에 직접적으로 송신하십시오. 359 페이지의 『IBM i에서 MQSC 명령을 사용하여 관리』을 참조하십시오.

예를 들어, 리모트 큐 관리자에서 큐 정의를 변경하기 위한 리모트 명령을 실행할 수 있습니다.

일부 명령은 특히, 큐 관리자를 작성하거나 시작하고 명령 서버를 시작하는 방법으로 실행될 수 없습니다. 이 유형의 태스크를 수행하려면 원격 시스템에 로그인하여 거기서 명령을 발행하거나 명령을 자동으로 발행할 수 있는 프로세스를 작성해야 합니다.

IBM i IBM i에서 MQSC 명령을 사용하여 관리

IBM i에서는 스크립트 파일에 명령 목록을 작성한 후 **STRMQMQSC** 명령을 사용하여 파일을 실행합니다. 큐 관리자 자체, 큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스, 토픽 및 인증 정보 오브젝트를 포함하여 큐 관리자 오브젝트를 관리하기 위해 MQSC 명령을 사용합니다.

이 태스크 정보

IBM MQ 스크립트 (MQSC) 명령은 EBCDIC 텍스트로 사용자가 읽을 수 있는 양식으로 작성됩니다.

STRMQMQSC IBM MQ CL 명령을 사용하여 큐 관리자에 MQSC 명령을 발행합니다. 이 메소드는 서버 라이브러리 시스템의 소스 실제 파일에서 해당 입력을 가져오는 유일한 배치 메소드입니다. 이 소스 실제 파일에 대한 기본 이름은 QMQSC입니다.



주의: QTEMP 라이브러리의 사용법이 제한된 것처럼 STRMQMQSC에 소스 라이브러리로서 QTEMP 라이브러리를 사용하지 마십시오. 명령의 입력 파일로 다른 라이브러리를 사용해야 합니다.

IBM MQ 환경 사이의 이식성의 경우, MQSC 명령 파일의 행 길이를 72자로 제한하십시오. 명령이 다음 행에서 계속됨을 표시하려면 더하기 부호를 사용하십시오.

MQSC에 지정된 오브젝트 속성은 대소문자를 구분하지 않지만 이 토픽에 대문자로 표시됩니다 (예: QMQNAME).

참고:

1. MQSC 파일의 형식은 파일 시스템에서 해당 위치에 의존하지 않습니다.
2. MQSC 속성 이름은 8자로 제한됩니다.
3. MQSC 명령은 모든 IBM MQ 플랫폼에서 사용 가능합니다.

각 MQSC 명령 및 해당 구문의 설명은 [MQSC 명령](#)을 참조하십시오.

프로시저

1. QMQSC 소스 파일을 작성하십시오.

IBM MQ for IBM i는 QMQSC라는 소스 파일을 제공하지 않습니다. MQSC 명령을 처리하려면 다음 명령을 실행하여 선택한 라이브러리에 QMQSC 소스 파일을 작성해야 합니다.

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT('IBM MQ - MQSC Source')
```

2. 멤버에 대한 작업을 수행하십시오.

MQSC 소스는 이 소스 파일 내에 멤버에서 유지됩니다. 멤버와 함께 일하려면 다음 명령을 입력하십시오.

```
WRKMBRPDM MYLIB/QMQSC
```

이제 새 구성원을 추가하고 기존 구성원을 유지보수할 수 있습니다.

```

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

```

그림 20. MQSC 명령 파일 *myprog.in*에서 추출하여 해당 속성과 함께 MQSC 명령 (*DEFINE QLOCAL*)을 표시합니다.

관련 정보

[MQSC 명령을 사용하여 IBM MQ 관리](#)

IBM i IBM i에서 PCF 명령을 사용하여 관리

PCF(IBM MQ programmable command format) 명령의 목적은 관리 태스크가 관리 프로그램으로 프로그래밍될 수 있도록 하는 것입니다. 이런 방식으로 프로그램에서 큐를 작성하고 정의를 처리하며 큐 관리자를 변경할 수 있습니다.

PCF 명령은 MQSC 명령에서 제공하는 동일한 범위의 기능을 포함합니다. 그러나, MQSC 명령과 달리 PCF 명령 및 응답은 읽을 수 있는 텍스트 형식에 없습니다.

단일 노트에서 네트워크의 임의의 큐 관리자에 PCF 명령을 발행하는 프로그램을 작성할 수 있습니다. 이러한 방법으로 관리 태스크를 집중시키고 자동화할 수 있습니다.

각 PCF 명령은 IBM MQ 메시지의 애플리케이션 데이터 부분에 임베드되는 데이터 구조입니다. 각 명령은 기타 메시지와 동일한 방식으로 MQI 기능 MQPUT을 사용하여 대상 큐 관리자로 보내집니다. 메시지를 수신하는 큐 관리자의 명령 서버는 명령 메시지로서 이를 해석하고 명령을 실행합니다. 응답을 얻기 위해 애플리케이션은 MQGET 호출을 실행하고 응답 데이터가 다른 데이터 구조로 리턴됩니다. 그런 다음, 애플리케이션은 응답을 처리하고 그에 따라 수행합니다.

간략하게, 이는 애플리케이션 프로그래머가 PCF 명령 메시지를 작성하기 위해 지정해야 하는 것 중 일부입니다.

메시지 디스크립터

이는 표준 IBM MQ 메시지 디스크립터입니다.

- 메시지 유형(*MsgType*)은 MQMT_REQUEST입니다.
- 메시지 형식(*Format*)은 MQFMT_ADMIN입니다.

애플리케이션 데이터

PCF 헤더를 포함하는 PCF 메시지가 들어 있습니다.

- PCF 메시지 유형(*Type*)은 MQCFT_COMMAND를 지정합니다.
- 명령 ID는 명령을 지정합니다 (예: *Change Queue* (MQCMD_CHANGE_Q)).

PCF 나가는 메시지 텍스트 내에 MQSC 명령을 포함하는 PCF 명령입니다. PCF를 사용하여 리모트 큐 관리자로 명령을 송신할 수 있습니다. 자세한 정보는 35 페이지의 『MQAI를 사용하여 PCF의 사용 단순화』의 내용을 참조하십시오.

PCF 데이터 구조의 전체 설명 및 이를 구현하는 방법은 [명령 및 응답에 대한 구조](#)를 참조하십시오.

IBM i IBM MQ for IBM i 와 함께 IBM MQ Explorer 사용

IBM MQ Explorer를 사용하여 IBM MQ for IBM i 를 관리하려면 이 정보를 사용하십시오.

IBM MQ for Windows(x86 플랫폼) 및 IBM MQ for Linux(x86 및 x86-64 플랫폼)에서는 CL, 제어 또는 MQSC 명령 사용에 대한 대체로서 관리 태스크를 수행하기 위해 IBM MQ 탐색기라는 관리 인터페이스를 제공합니다.

IBM MQ Explorer 를 사용하면 관심있는 큐 관리자 및 클러스터에서 IBM MQ Explorer 를 지정하여 Windows (x86 플랫폼) 또는 Linux (x86 및 x86-64 플랫폼) 를 실행 중인 컴퓨터에서 네트워크의 로컬 또는 원격 관리를 수행할 수 있습니다.

IBM MQ Explorer를 사용하여 다음을 수행할 수 있습니다.

- 큐 관리자를 시작하고 중지하십시오(로컬 시스템 전용).
- 큐, 토픽 및 채널과 같은 IBM MQ 오브젝트의 정의를 정의, 표시 및 대체하십시오.
- 큐에서 메시지를 찾으십시오.
- 채널을 시작하고 중지하십시오.
- 채널에 대한 상태 정보를 보십시오.
- 클러스터에서 큐 관리자를 보십시오.
- 애플리케이션, 사용자 또는 채널이 특정 큐를 여는지 확인하기 위해 검사하십시오.
- 새 클러스터 작성 마법사를 사용하여 새 큐 관리자 클러스터를 작성하십시오.
- 클러스터에 큐 관리자 추가 마법사를 사용하여 클러스터에 큐 관리자를 추가하십시오.
- TLS(Transport Layer Security) 채널 보안과 함께 사용되는 인증 정보 오브젝트를 관리합니다.

온라인 자세한 내용을 사용하여 다음을 수행할 수 있습니다.

- 큐 관리자, 큐, 채널, 프로세스 정의, 클라이언트 연결 채널, 리스너, 토픽, 서비스, 이름 목록 및 클러스터를 포함하여 다양한 자원을 정의하고 제어하십시오.
- 큐 관리자 및 연관된 프로세스를 시작 또는 중지하십시오.
- 워크스테이션 또는 기타 워크스테이션에서 큐 관리자 및 연관된 오브젝트를 보십시오.
- 큐 관리자, 클러스터 및 채널의 상태를 검사하십시오.

IBM MQ Explorer를 사용하여 서버 시스템에서 IBM MQ를 관리하도록 시도하기 전에 다음 요구사항을 충족시켰다는 것을 확인하십시오. 다음을 확인하십시오.

1. 명령 서버는 관리되는 모든 큐 관리자에 대해 실행 중이며 CL 명령 **STRMQCSVR**에 의해 서버에서 시작됩니다.
2. 적당한 TCP/IP 리스너는 모든 리모트 큐 관리자를 위해 존재합니다. 이는 **STRMQLSR** 명령으로 시작되는 IBM MQ 리스너입니다.

3. SYSTEM.ADMIN.SVRCONN라는 서버 연결 채널은 모든 리모트 큐 관리자에 존재합니다. 이 채널을 직접 작성해야 합니다. 모든 리모트 큐 관리자가 관리되는 것은 필수입니다. 그것 없이, 원격 관리는 가능하지 않습니다.
4. SYSTEM.MQEXPLORER.REPLY.MODEL 큐가 존재한다는 것을 확인하십시오.

IBM i IBM i에서 원격 관리를 위한 명령 서버 관리

이 정보를 사용하여 IBM MQ for IBM i 명령 서버의 원격 관리에 대해 학습하십시오.

각 큐 관리자에게는 이와 연관되는 명령 서버가 있을 수 있습니다. 명령 서버는 리모트 큐 관리자에게 수신되는 명령 또는 애플리케이션에서 PCF 명령을 처리합니다. 이는 처리를 위해 큐 관리자에게 명령을 나타내고 원래 명령에 따라 완료 코드 또는 운영자 메시지를 리턴합니다.

명령 서버는 PCF, MQAI를 포함하는 모든 관리 및 원격 관리를 위해 필수적입니다.

참고: 원격 관리의 경우, 대상 큐 관리자가 실행 중인지 확인해야 합니다. 그렇지 않으면, 명령을 포함하는 메시지는 발행되는 큐 관리자를 남길 수 없습니다. 그 대신, 이러한 메시지는 리모트 큐 관리자를 제공하는 로컬 전송 큐에서 큐잉됩니다. 조금이라도 가능하다면 이 상황을 피하십시오.

명령 서버를 시작하고 중지하기 위한 분리 제어 명령이 있습니다. IBM MQ 탐색기를 사용하여 다음 섹션에서 설명된 조작을 수행할 수 있습니다.

명령 서버 시작 및 중지

명령 서버를 시작하려면 이 CL 명령을 사용하십시오.

```
STRMQMCSVR MQMNAME('saturn.queue.manager')
```

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

명령 서버를 중지하려면 다음 CL 명령 중 하나를 사용하십시오.

1. ENDMQMCSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRLD)

제어된 중지를 수행합니다. 여기서 saturn.queue.manager는 명령 서버가 중지되는 큐 관리자입니다. 이는 기본 옵션이고, 이는 OPTION(*CNTRLD)이 생략될 수 있다는 것을 의미합니다.

2. ENDMQMCSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)

즉시 중지를 수행합니다. 여기서 saturn.queue.manager는 명령 서버가 중지되는 큐 관리자입니다.

명령 서버의 상태 표시

원격 관리의 경우, 대상 큐 관리자의 명령 서버가 실행 중인지 확인하십시오. 실행 중이지 않은 경우, 원격 명령을 처리할 수 없습니다. 명령을 포함하는 메시지는 대상 큐 관리자의 명령 큐 SYSTEM.ADMIN.COMMAND.QUEUE에서 큐잉됩니다.

여기서 saturn.queue.manager라고 하는 큐 관리자에 대한 명령 서버의 상태를 표시하기 위한 CL 명령은 다음과 같습니다.

```
DSPMQMCSVR MQMNAME('saturn.queue.manager')
```

대상 시스템에서 이 명령을 실행하십시오. 명령 서버가 실행 중인 경우, 363 페이지의 그림 21에 표시된 패널이 표시됩니다.


```
Display MQM Command Server (DSPMQMCSVR)
```

```
Queue manager name . . . . . > saturn.queue.manager
```

```
MQM Command Server Status. . . . > RUNNING
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys
```

그림 21. MQM 명령 서버 패널 표시

IBM i 웹 콘솔 명령 실행

웹 콘솔 관련 Qshell 명령이 IBM MQ for IBM i에서 올바르게 실행되려면 다음 텍스트에 설명된 대로 환경을 구성해야 합니다.

이 태스크 정보

Qshell이 시작되면 작업의 CCSID를 기반으로 명령을 처리하기 위해 내부 테이블을 초기화합니다. 웹 콘솔 관련 Qshell 명령이 IBM i에서 올바르게 실행되려면 환경을 구성해야 합니다.

LANG 환경 변수를 로케일 오브젝트의 경로명으로 설정하여 로케일이 설정됩니다. 예를 들어 미국 영어의 로케일을 설정하려는 경우 LANG 환경 변수는 다음과 같이 설정됩니다.

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

Qshell에서는 모든 환경 변수를 나열하도록 명령 세트를 실행하여 설정을 확인할 수 있습니다. 일반적으로 런타임 환경의 로케일에 영향을 줄 수 있는 LANG입니다. LC_ALL도 있을 수 있습니다.

Qshell 명령을 올바르게 실행하려면 로케일 환경 설정이 작업 설정과 일치해야 합니다.

프로시저

CL 명령 DSPJOB JOB(JobNumber/USERProfile/JobName)을 사용하십시오.

- 옵션 2를 선택하여 작업 정의 속성을 표시하십시오.
- 다음 속성이 LANG 또는 LC_ALL 환경 설정과 일치해야 합니다.

- 언어 ID
- 국가 또는 지역 ID
- 코드화 문자 세트 ID

예를 들어, 다음과 같은 경우

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

작업 속성은 다음과 같아야 합니다.

- 언어 ID FRA
- 국가 또는 지역 ID FR
- 코드화된 문자 세트 ID 297

다음에 수행할 작업

자국어 지원에 대한 자세한 정보는 IBM Documentation 주제 [자국어 지원\(NLS\) 고려사항](#)을 참조하십시오.

IBM i IBM i에 대한 작업 관리

이 정보는 IBM MQ가 작업 요청을 핸들링하는 방식을 설명하고 IBM MQ와 연관되는 작업을 우선시시키고 제어하기 위해 사용할 수 있는 옵션에 대해 상세히 설명합니다.

경고

IBM i 및 IBM MQ 작업 관리의 개념을 완전히 이해하지 못한 경우에는 IBM MQ 작업 관리 오브젝트를 변경하지 마십시오.

서브시스템 및 작업 설명에 대한 추가 정보는 IBM i 제품 문서의 [작업 관리](#) 에서 찾을 수 있습니다. [작업 시작 및 배치\(Batch\) 작업](#) 절을 유의하여 살펴보십시오.

IBM MQ for IBM i는 IBM i UNIX 환경과 IBM i 스레드를 통합합니다. IFS(Integrated File System)에서 오브젝트를 변경하지 마십시오.

정상 작동 동안, IBM MQ 큐 관리자는 다른 태스크를 수행하기 위해 수많은 배치 작업을 시작합니다. 기본적으로 이러한 배치 작업은 IBM MQ가 설치될 때 작성되는 QMQM 서브시스템에서 실행됩니다.

작업 관리는 사용자의 시스템으로부터 최적의 성능을 얻고 관리를 더 단순하게 하기 위해 IBM MQ 태스크를 맞춤화하는 프로세스를 참조합니다.

예를 들어, 다음 사항을 수행할 수 있습니다.

- 하나의 큐 관리자가 다른 큐 관리자보다 더 응답하도록 작업의 실행 우선순위를 변경하십시오.
- 수많은 작업의 출력을 특정 출력 큐로 경로 재지정하십시오.
- 특정 유형의 모든 작업이 특정 서브시스템에서 실행되도록 하십시오.
- 오류를 서브시스템으로 분리하십시오.

작업 관리는 IBM MQ 작업과 연관되는 작업 설명을 작성하거나 변경하여 수행됩니다. 다음에 대한 작업 관리를 구성할 수 있습니다.

- 전체 IBM MQ 설치
- 개별 큐 관리자
- 개별 큐 관리자에 대한 개별 작업

IBM i IBM MQ 태스크 IBM i

이는 IBM MQ for IBM i 작업의 테이블 및 각각에 대한 간략한 설명입니다.

큐 관리자가 실행 중인 경우, IBM MQ 서브시스템에서 QMQM 사용자 프로파일에서 실행 중인 다음 배치 작업 중 일부 또는 전부를 참조하십시오. 작업이 [364 페이지의 표 21](#)에서 간략하게 설명됩니다.

큐 관리자에 대한 작업(WRKMQM) 패널에서 옵션 22를 사용하여 큐 관리자에 연결되는 모든 작업을 볼 수 있습니다. WRKMQLSR 명령을 사용하여 리스너를 볼 수 있습니다.

표 21. IBM MQ 태스크.	
작업 이름	Function
AMQZMUC0	유틸리티 관리자. 이 작업은 결정적 큐 관리자 유틸리티를 실행합니다(예: 저널 체인 관리자).
AMQZXMA0	큐 관리자가 시작하는 첫 번째 작업인 실행 제어기. 이는 MQCONN 요청을 핸들링하고 IBM MQ API 호출을 처리하기 위해 에이전트 프로세스를 시작합니다.
AMQZFUMA	오브젝트 권한 관리자(OAM).
AMQZLAA0	MQCNO_STANDARD_BINDING을 사용하여 큐 관리자에 연결하는 애플리케이션에 대한 대부분의 작업을 수행하는 큐 관리자 에이전트.
AMQZLSA0	큐 관리자 에이전트.
AMQZMUFO	유틸리티 관리자

표 21. IBM MQ 태스크. (계속)	
작업 이름	Function
AMQZMGRO	프로세스 제어기. 이 작업은 리스너와 서비스를 시작하고 관리하는 데 사용됩니다.
AMQZMURO	유틸리티 관리자. 이 작업은 결정적 큐 관리자 유틸리티를 실행합니다(예: 저널 체인 관리자).
AMQFQPUB	큐된 발행/구독 디먼.
AMQFCXBA	브로커 작업 프로그램 작업.
RUNMQBRK	브로커 제어 작업.
AMQRMPPA	채널 프로세스 폴링 작업.
AMQCRSTA	TCP/IP 호출 채널 응답자.
AMQCRS6B	LU62 수신자 채널 및 클라이언트 연결(노트 참조).
AMQRRMFA	클러스터를 위한 저장소 관리자.
AMQCLMAA	비스레드 TCP/IP 리스너.
AMQPCSEA	PCF를 핸들링하고 원격 관리가 요청하는 PCF 명령 프로세서.
RUNMQTRM	트리거 모니터.
RUNMQDLQ	데드 레터 큐 핸들러.
RUNMQCHI	채널 시작기
RUNMQCHL	각 송신자 채널에 대해 사용되는 송신자 채널 작업.
RUNMQLSR	스레드 TCP/IP 리스너.
AMQRCMLA	채널 MQSC 및 PCF 명령 프로세서.

참고: LU62 수신자 작업이 통신 서브시스템에서 실행되고 작업을 시작하기 위해 사용되는 통신 입력 항목 및 라우팅에서 해당 런타임 특성을 사용합니다. 자세한 정보는 [시작된 한쪽 끝\(수신자\)](#)을 참조하십시오.

IBM i IBM i에서 작업 관리 오브젝트

IBM MQ가 설치될 때, 작업 관리자를 지원하기 위해 다양한 오브젝트가 QMQM 라이브러리에 제공됩니다. 이러한 오브젝트는 자체 서브시스템에서 실행하기 위해 IBM MQ 작업에 필요한 것입니다.

두 개의 IBM MQ 배치 작업에 샘플 작업 설명이 제공됩니다. 특정 작업 설명이 IBM MQ 작업에 제공되지 않으면 기본 작업 설명 QMQMJOBDO로 실행됩니다.

IBM MQ를 설치할 때 제공되는 작업 관리 오브젝트는 365 페이지의 표 22에 나열되고 큐 관리자에 작성된 오브젝트는 366 페이지의 표 23에 나열됩니다.

참고: MQMQ 라이브러리에서 작업 관리 오브젝트를 찾을 수 있고 큐 관리자 라이브러리에서 큐 관리자 오브젝트를 찾을 수 있습니다.

표 22. 작업 관리 오브젝트		
이름	유형	설명
AMQZLAA0	*JOBDO	IBM MQ 에이전트 프로세스에서 사용하는 작업 설명
AMQZLSA0	*JOBDO	고립된 바인딩 큐 관리자 에이전트
AMQZXMA0	*JOBDO	IBM MQ 실행 제어기에서 사용하는 작업 설명
QMQM	*SBSD	모든 IBM MQ 작업이 실행되는 서브시스템
QMQM	*JOBQ	제공된 서브시스템에 첨부된 작업 큐

표 22. 작업 관리 오브젝트 (계속)		
이름	유형	설명
QMQMJOB	*JOB	작업에 대한 특정 작업 설명이 아닌 경우 사용되는 기본 IBM MQ 작업 설명
QMQMMSG	*MSGQ	IBM MQ 작업에 대한 기본 메시지 큐.
QMQMRUN20	*CLS	상위 우선순위 IBM MQ 작업에 대한 클래스 설명
QMQMRUN35	*CLS	중간 우선순위 IBM MQ 작업에 대한 클래스 설명
QMQMRUN50	*CLS	하위 우선순위 IBM MQ 작업에 대한 클래스 설명

표 23. 큐 관리자에 작성된 작업 관리 오브젝트		
이름	유형	설명
AMQA000000	*JRNRCV	로컬 저널 수신자
AMQAJRN	*JRN	로컬 저널
AMQJRNINF	*USRSPC	큐 관리자의 시동 및 매체 복원에 필요한 최근 저널 수신자로 업데이트되는 사용자 공간. 이 사용자 공간은 어느 저널 수신자가 파일 보관을 요구하고 어느 것이 안전하게 삭제될 수 있는지 판별하기 위해 애플리케이션에 의해 질문받을 수 있습니다.
AMQAJRNMSG	*MSGQ	로컬 저널 메시지 큐
AMQCRC6B	*PGM	LU6.2 연결을 시작하기 위한 프로그램
AMQRFOLD	*FILE	마이그레이션된 큐 관리자 채널 정의 파일
QMQMMSG	*MSGQ	큐 관리자 메시지 큐

IBM i IBM MQ가 IBM i에서 작업 관리 오브젝트를 사용하는 방법

이 정보는 IBM MQ가 작업 관리 오브젝트를 사용하고 구성 예제를 제공하는 방법에 대해 설명합니다.



주의: 우선순위에 따라 서브시스템에서 허용되는 작업 수를 제한하기 위해 QMQM 서브시스템에서 작업 큐 입력 항목 설정을 대체하지 마십시오. 이를 수행하려고 시도하는 경우, 제출되고 큐 관리자 시동을 실패하게 한 후 본질적 IBM MQ 작업이 실행되는 것을 방지할 수 있습니다.

작업 관리자를 구성하는 방법에 대해 이해하려면 IBM MQ가 작업 설명을 사용하는 방법에 대해 먼저 이해해야 합니다.

작업을 시작하기 위해 사용되는 작업 설명은 많은 작업 속성을 제어합니다. 예를 들면, 다음과 같습니다.

- 작업이 큐잉되는 작업 큐 및 작업이 실행되는 서브시스템의 작업 큐.
- 작업이 해당 런타임 매개변수를 위해 사용하는 클래스 및 작업을 시작하기 위해 사용되는 라우팅 데이터.
- 작업이 인쇄 파일에 사용하는 출력 큐.

IBM MQ 작업을 시작하는 프로세스는 세 가지 단계로 구성될 수 있습니다.

1. IBM MQ는 작업 설명을 선택합니다.

IBM MQ는 배치 작업에 사용할 작업 설명을 판별하기 위해 다음 기술을 사용합니다.

- 작업과 동일한 이름을 가지는 작업 설명에 대해 큐 관리자 라이브러리에서 찾으십시오. 큐 관리자 라이브러리에 대한 추가 세부사항은 [IBM MQ for IBM i 큐 관리자 라이브러리 이름 이해](#)를 참조하십시오.
- 기본 작업 설명 QMQMJOB에 대해 큐 관리자 라이브러리에서 찾으십시오.
- 작업과 동일한 이름을 가지는 작업 설명에 대해 QMQM 라이브러리에서 찾으십시오.
- QMQM 라이브러리에서 기본 작업 설명(QMQMJOB)을 사용하십시오.

2. 작업이 작업 큐에 제출됩니다.

IBM MQ와 함께 제공되는 작업 설명은 기본적으로 라이브러리 QMQM의 작업 큐 QMQM에 작업을 배치시키기 위해 설정됩니다. QMQM 작업 큐는 제공된 QMQM 서브시스템에 첨부됩니다. 따라서 기본적으로 작업은 QMQM 서브시스템에서 실행을 시작됩니다.

3. 작업은 서브시스템을 입력하고 라우팅 단계를 통해 이동합니다.

작업이 서브시스템을 입력할 때, 작업 설명에 지정된 라우팅 데이터는 작업에 대한 라우팅 입력 항목을 찾기 위해 사용됩니다.

라우팅 데이터는 QMQM 서브시스템에 정의된 라우팅 입력 항목 중 하나와 일치하고 이는 작업에 사용하는 제공된 클래스(QMQMRUN20, QMQMRUN35 또는 QMQMRUN50)를 정의합니다.

참고: IBM MQ 작업이 시작되는 것으로 표시되지 않지만, 서브시스템이 실행 중이고 작업 큐가 없다는 것을 확인하십시오.

IBM MQ 작업 관리 오브젝트를 수정한 경우, 모든 것이 올바르게 연관된다는 것을 확인하십시오. 예를 들어, 작업 설명에서 QMQM/QMQM 이외의 작업 큐를 지정하는 경우, ADDJOBQE가 서브시스템, 즉 QMQM에 대해 실행된다는 것을 확인하십시오.

예로서 다음 워크시트를 사용하여 [364 페이지의 표 21](#)에 문서화된 각 작업에 대해 작업 설명을 작성할 수 있습니다.

```

What is the queue manager library name? _____
Does job description AMQZXMA0 exist in the queue manager library? Yes No
Does job description QMQMJOB0 exist in the queue manager library? Yes No
Does job description AMQZXMA0 exist in the QMQM library? Yes No
Does job description QMQMJOB0 exist in the QMQM library? Yes No

```

이러한 모든 질문에 No로 대답하는 경우, QMQM 라이브러리에서 글로벌 작업 설명 QMQMJOB0를 작성하십시오.

IBM MQ 메시지 큐

IBM MQ 메시지 큐(QMQMMSG)가 각 큐 관리자 라이브러리에서 작성됩니다. 운영 체제 메시지는 큐 관리자 작업이 종료되고 IBM MQ가 큐로 메시지를 보낼 때 이 큐로 보내집니다. 예: 시동 시 필요한 저널 수신자를 보고하기 위해. 모니터링하는 것을 보다 용이하게 하기 위한 관리 가능한 크기에 메시지의 수를 이 메시지 큐에 보존하십시오.

IBM i IBM i에 대한 기본 시스템 예

이러한 예는 수정되지 않은 IBM MQ 설치가 일부 표준 작업이 큐 관리자 시동 시간에 제출될 때 작동하는 방법을 표시합니다.

먼저, AMQZXMA0 실행 제어기 작업이 시작됩니다.

1. 큐 관리자 TESTQM에 대한 **STRMQM** 명령을 실행하십시오.
2. IBM MQ는 먼저 작업 설명 AMQZXMA0 및 작업 설명 QMQMJOB0에 대해 큐 관리자 라이브러리 QMTESTQM을 검색합니다.

이러한 작업 설명 중 어느 쪽도 존재하지 않아서 IBM MQ는 제품 라이브러리 QMQM에서 작업 설명 AMQZXMA0를 검색합니다. 이 작업 설명이 존재합니다. 따라서 작업을 제출하기 위해 사용됩니다.

3. 작업 설명은 IBM MQ 기본 작업 큐를 사용합니다. 따라서 작업이 작업 큐 QMQM/QMQM에 제출됩니다.
4. AMQZXMA0 작업 설명에 대한 라우팅 데이터는 QMQMRUN20입니다. 따라서 시스템은 해당 데이터와 일치하는 것에 대한 서브시스템 라우팅 입력 항목을 검색합니다.

기본적으로, 순서 번호 9900을 가지는 라우팅 입력 항목에는 QMQMRUN20과 일치하는 비교 데이터가 있습니다. 따라서 작업은 QMQMRUN20이라고도 하는 해당 라우팅 입력 항목에 정의되는 클래스와 함께 시작됩니다.

5. QMQM/QMQMRUN20 클래스는 20으로 설정된 우선순위로 실행되었습니다. 따라서, AMQZXMA0 작업은 시스템에서 가장 대화식 작업과 동일한 우선순위로 서브시스템 QMQM에서 실행됩니다.

IBM i IBM i에서 작업 관리 구성 예

이 정보를 사용하여 IBM MQ 작업의 런타임 속성을 변경하기 위해 IBM MQ 작업 설명을 변경하고 작성할 수 있는 방법에 대해 학습하십시오.

IBM MQ 작업 관리의 유연성의 키는 IBM MQ이 작업 설명을 찾는 2단계 방법에 있습니다.

- 큐 관리자 라이브러리에서 작업 설명을 작성 또는 변경하는 경우 해당 변경사항은 QMQM에서 글로벌 작업 설명을 대체하지만 이 변경사항은 로컬이므로 특정 큐 관리자에만 영향을 줍니다.
- QMQM 라이브러리에서 글로벌 작업 설명을 작성하거나 변경하는 경우, 개별 큐 관리자에 대해 로컬로 대체되지 않으면 해당 작업 설명이 시스템에서 모든 큐 관리자에 영향을 줍니다.

1. 다음 예는 개별 큐 관리자에 대한 채널 제어 작업의 우선순위를 증가시킵니다.

저장소 관리자 및 채널 시작기 작업(AMQRRMFA 및 RUNMQCHI)을 작성하려면 큐 관리자 TESTQM에 대해 가능한 빨리 수행하고 다음 단계를 수행하십시오.

- a. 큐 관리자 라이브러리에서 제어하려는 IBM MQ 프로세스의 이름으로 QMQM/QMQMJOBID 작업 설명의 로컬 복제를 작성하십시오. 예를 들면, 다음과 같습니다.

```
CRTDUPOBJ OBJ(QMQMJOBID) FROMLIB(QMQM) OBJTYPE(*JOBID) TOLIB(QMTESTQM)
NEWOBJ(RUNMQCHI)
CRTDUPOBJ OBJ(QMQMJOBID) FROMLIB(QMQM) OBJTYPE(*JOBID) TOLIB(QMTESTQM)
NEWOBJ(AMQRRMFA)
```

- b. 작업이 QMQMRUN20 클래스를 사용한다는 것을 확인하기 위해 작업 설명에서 라우팅 데이터 매개변수를 변경하십시오.

```
CHGJOBID JOBID(QMTESTQM/RUNMQCHI) RTGDTA('QMQMRUN20')
CHGJOBID JOBID(QMTESTQM/AMQRRMFA) RTGDTA('QMQMRUN20')
```

큐 관리자 TESTQM에 대한 AMQRRMFA 및 RUNMQCHI 작업.

- 큐 관리자 라이브러리에서 새로운 로컬 작업 설명을 사용하십시오.
 - 작업이 서브시스템을 입력할 때 QMQMRUN20 클래스가 사용되므로 우선순위 20으로 실행하십시오.
2. 다음 예제는 QMQM 서브시스템에 대한 새 실행 우선순위 클래스를 정의합니다.
- a. 기타 큐 관리자가 클래스에 액세스하도록 허용하려면 다음 명령을 실행하여 QMQM 라이브러리에서 복제 클래스를 작성하십시오.

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ(QMQMRUN10)
```

- b. 다음 명령을 실행하여 새 실행 우선순위를 가지도록 클래스를 변경하십시오.

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. 다음 명령을 실행하여 서브시스템에 새 클래스 정의를 추가하십시오.

```
ADDRTGE SBSID(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)
CLS(QMQM/QMQMRUN10)
```

참고: 라우팅 순서 번호에 숫자 값을 지정할 수 있지만, 값은 순차 순서에 있어야 합니다. 이 순서 번호는 라우팅 입력 항목이 라우팅 데이터 일치 검색할 순서를 서브시스템에 알려줍니다.

- d. 다음 명령을 실행하여 새 우선순위 클래스를 사용하려면 로컬 또는 글로벌 작업 설명을 변경하십시오.

```
CHGJOBID JOBID(QMQMlibname/QMQMJOBID) RTGDTA('QMQMRUN10')
```

QMlibraryname과 연관되는 모든 큐 관리자 작업은 10의 실행 우선순위를 사용합니다.

3. 다음 예는 자체 서브시스템에서 큐 관리자를 실행합니다

큐 관리자 TESTQM에 대한 모든 작업이 QBATCH 서브시스템에서 실행되도록 하려면 다음 단계를 실행하십시오.

- a. 큐 관리자 라이브러리에서 명령을 사용하여 QMQM/QMQMJOB JOB 설명의 로컬 복제를 작성하십시오

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
```

- b. 작업이 QBATCH 작업 큐를 사용한다는 것을 확인하기 위해 작업 설명에서 작업 큐 매개변수를 변경하십시오.

```
CHGJOB JOB(QMTESTQM/QMQMJOB) JOBQ(*LIBL/QBATCH)
```

참고: 작업 큐는 서브시스템 설명과 연관됩니다. 작업이 작업 큐에 남아 있다는 것을 알면, 작업 큐 정의가 SBSDB에서 정의되는지 확인하십시오. 서브시스템에 대해 DSPSBSDB 명령을 사용하고 옵션 6(작업 큐 입력 항목)을 사용하십시오.

큐 관리자 TESTQM에 대한 모든 작업:

- 큐 관리자 라이브러리에서 새 로컬 기본 작업 설명을 사용하십시오
- 작업 큐 QBATCH로 제출됩니다.

작업이 라우팅되고 올바르게 우선순위 지정되는지 확인하려면 다음을 수행하십시오.

- 서브시스템 QBATCH에서 IBM MQ 작업에 대한 라우팅 입력 항목을 작성하거나
- 라우팅 데이터가 사용되는 것과 상관 없이 QCMD를 호출하는 광범위한 라우팅 입력 항목을 신뢰하십시오.

작업 큐 QBATCH에 대한 최대 활성 작업 옵션이 *NOMAX로 설정되는 경우에만 이 옵션이 작동합니다. 시스템 기본값은 1입니다.

4. 다음 예는 다른 IBM MQ 서브시스템을 작성합니다

- a. 다음 명령을 실행하여 QMQM 라이브러리에서 복제 서브시스템을 작성하십시오.

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBS) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. 다음 명령을 실행하여 QMQM 작업 큐를 제거하십시오.

```
RMVJOBQE SBS(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. 다음 명령을 실행하여 서브시스템에 대한 새 작업 큐를 작성하십시오.

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. 다음 명령을 실행하여 작업 큐 입력 항목을 서브시스템에 추가하십시오.

```
ADDJOBQE SBS(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. 다음 명령을 실행하여 큐 관리자 라이브러리에서 복제 QMQMJOB를 작성하십시오.

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMlibraryname)
```

- f. 다음 명령을 실행하여 새 작업 큐를 사용하려면 작업 설명을 변경하십시오.

```
CHGJOB JOB(QMlibraryname/QMQMJOB) JOBQ(QMQM/QMQM2)
```

- g. 다음 명령을 실행하여 서브시스템을 시작하십시오.

```
STRSBS SBS(QMQM/QMQM2)
```


참고:

- 라이브러리에서 서브시스템을 지정할 수 있습니다. 어떤 이유로도 제품이 재설치되거나 QMQM 라이브러리가 대체되는 경우, 사용자가 작성한 변경사항이 제거됩니다.
- QMlibraryname과 연관되는 모든 큐 관리자 작업은 서브시스템 QMQM2에서 실행됩니다.

IBM i 가용성, 백업, 복구 및 재시작(IBM i)

이 정보를 사용하여 IBM MQ for IBM i가 해당 백업 및 복원 전략을 돕기 위한 IBM i 저널링 지원을 사용하는 방법에 대해 이해하십시오.

이 섹션을 읽기 전에 표준 IBM i 백업 및 복구 메소드와 IBM i에서 연관된 저널 및 저널의 사용에 익숙해야 합니다. 이러한 주제에 대한 정보는 백업 및 복구를 참조하십시오.

백업 및 복구 전략에 대해 이해하려면 먼저 IBM MQ for IBM i가 IBM i 파일 시스템 및 IFS(Integrated File System)에서 해당 데이터를 조직하는 방법에 대해 이해해야 합니다.

IBM MQ for IBM i는 IFS 파일 시스템의 스트림 파일 및 각 큐 관리자 인스턴스에 대한 개별 라이브러리에 해당 데이터를 보유하고 있습니다.

큐 관리자 특정 라이브러리에는 큐 관리자의 작업 관리를 제어하기 위해 필요한 저널, 저널 수신자 및 오브젝트가 포함됩니다. IFS 디렉토리 및 파일에는 IBM MQ 구성 파일, IBM MQ 오브젝트의 설명 및 포함하는 데이터가 포함됩니다.

시스템 실패에 대한 복구 가능한 이러한 오브젝트에 대한 변경은 적절한 오브젝트에 적용되기 전에 저널에 기록됩니다. 여기에는 저널에 기록된 정보를 재실행하여 그러한 변경사항이 복구될 수 있다는 효과가 있습니다.

서버 또는 큐 관리자 실패의 경우에 복구의 속도를 높이고 증가된 큐 관리자 가용성을 제공하기 위해 다른 서버에서 여러 큐 관리자 인스턴스를 사용하여 IBM MQ for IBM i를 구성할 수 있습니다.

IBM i IBM i의 큐 관리자 저널

이 정보를 사용하여 IBM MQ for IBM i가 로컬 오브젝트에 대한 업데이트를 제어하기 위해 해당 조작에서 저널을 사용하는 방법에 대해 이해하십시오.

각 큐 관리자 라이브러리에는 해당 큐 관리자에 대한 저널이 포함되어 있으며 저널의 이름은 QM GRLIB/AMQ A JRN입니다. 여기서 QM GRLIB은 큐 관리자 라이브러리의 이름이고 A는 큐 관리자 인스턴스에 고유한 문자(단일 인스턴스 큐 관리자의 경우 A)입니다.

QM GRLIB은 QM 이름을 사용하고, 고유한 양식으로 큐 관리자의 이름이 뒤에 옵니다. 예를 들어, TEST라고 이름 지정된 큐 관리자에는 QMTEST로 이름 지정된 큐 관리자 라이브러리가 있습니다. CRTMQM 명령을 사용하여 큐 관리자를 작성할 때 큐 관리자 라이브러리가 지정될 수 있습니다.

저널은 저널링되는 정보를 포함하는 저널 수신자를 연관시킵니다. 수신자는 추가될 수 있고 결국 채워지는 정보에 대한 오브젝트입니다.

저널 수신자는 구식 정보로 가치 있는 디스크 공간을 다 사용합니다. 그러나, 이 문제를 최소화하기 위해 영구적 인 스토리지에서 정보를 대체할 수 있습니다. 특정 시간에 하나의 저널 수신자가 저널에 첨부됩니다. 저널 수신자가 사전정의된 임계값 크기에 도달하면, 새 저널 수신자에 의해 분리되고 대체됩니다. CRTMQM 및 THRESHOLD 매개변수를 사용하여 큐 관리자를 작성할 때 저널 수신자의 임계값을 지정할 수 있습니다.

로컬 IBM MQ for IBM i 저널과 연관된 저널 수신자는 각 큐 관리자 라이브러리에 존재하며 다음과 같이 이름 지정 규칙을 채택합니다.

```
AMQ Arnnnnn
```

여기서,

A

는 문자 A-Z입니다. 이는 단일 인스턴스 큐 관리자에 대한 A입니다. 이는 여러 인스턴스 큐 관리자의 다른 인스턴스에 따라 다양합니다.

nnnnn

시퀀스의 다음 저널에 대해 1씩 증분되는 10진수 00000 to 99999입니다.

r

10진수 0 to 9입니다. 즉, 수신자가 복원될 때마다 1씩 증가합니다.

저널 순서는 날짜를 기반으로 합니다. 그러나, 다음 저널의 이름 지정 규칙은 다음 규칙을 기반으로 합니다.

1. AMQArnnnnn은 AMQAr(nnnnn+1)로 이동하고 nnnnn은 99999에 도달할 때 줄 바꾸기를 합니다. 예를 들어, AMQA099999는 AMQA000000으로 이동하고 AMQA999999는 AMQA900000으로 이동합니다.
2. 규칙 1이 생성한 이름을 가지는 저널이 이미 존재하는 경우, CPI70E3이 QSYSOPR 메시지 큐로 보내지고 자동 수신자 전환이 중지됩니다.
현재 첨부된 수신자는 문제점을 조사하고, 수동으로 새 수신자를 첨부시킬 때까지 계속 사용됩니다.
3. 어떤 새 이름도 시퀀스에 이용할 수 없으면(즉, 가능한 모든 저널 이름이 시스템에 있음), 다음 둘 다를 수행해야 합니다.
 - a. 더 이상 필요하지 않는 저널을 삭제하십시오(375 페이지의 『IBM i에서 저널 관리』 참조).
 - b. 사용 중인 최근 저널 수신자로 변경한 저널을 기록하고(RCDMQMIMG) 이전 단계를 반복하십시오. 이는 오래된 저널 수신자 이름이 재사용될 수 있게 허용합니다.

AMQAJRN 저널은 MNGRCV(*SYSTEM) 옵션을 사용하여 임계값에 도달될 때 운영 체제가 자동으로 저널 수신자를 변경할 수 있도록 합니다. 시스템이 수신자를 관리하는 방법에 대한 자세한 정보는 IBM i 백업 및 복구를 참조하십시오.

저널 수신자의 기본 임계값은 100,000KB입니다. 이를 큐 관리자를 작성할 때 더 큰 값으로 설정할 수 있습니다. LogReceiverSize 속성의 초기값은 mqs.ini 파일의 LogDefaults 스탠자에 작성됩니다.

저널 수신자가 지정된 임계값을 넘어 연장될 때, 이전 수신자에서 속성을 상속하여 수신자가 분리되고 새 저널 수신자가 작성됩니다. 큐 관리자가 작성한 이후 LogReceiverSize 또는 LogASP 속성에 대한 변경사항은 시스템이 새 저널 수신자를 자동으로 첨부할 때 무시됩니다

시스템 구성에 대한 자세한 내용은 [멀티플랫폼에서 IBM MQ 구성 정보 변경](#) 을 참조하십시오.

큐 관리자가 작성한 이후 저널 수신자의 크기를 변경해야 하는 경우, 새 저널 수신자를 작성하고 다음 명령을 사용하여 QMQM에 소유자를 설정하십시오.

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +
TEXT('MQM LOCAL JOURNAL RECEIVER')
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

여기서,

QMGRLIB

큐 관리자 라이브러리의 이름입니다.

A

인스턴스 ID(일반적으로 A)입니다.

rrnnnnn

이전에 설명된 이름 지정 순서에 있는 다음 저널 수신자입니다.

xxxxxx

새 수신자 임계값입니다(KB).

참고: 수신자의 최대 크기는 운영 체제에 의해 통제됩니다. 이 값을 확인하려면 **CRTJRNRCV** 명령에서 THRESHOLD 키워드를 보십시오.

새 수신자를 명령으로 AMQAJRN 저널에 첨부하십시오.

```
CHGJRN JRN(QMGRLIB/AMQ A JRN) JRNRCV(QMGRLIB/AMQ Annnnnn)
```

이러한 저널 수신자를 관리하는 방법에 대한 세부사항은 375 페이지의 『IBM i에서 저널 관리』의 내용을 참조하십시오.

IBM i IBM i에서 큐 관리자 저널 사용

이 정보를 사용하여 IBM MQ for IBM i가 로컬 오브젝트에 대한 업데이트를 제어하기 위해 해당 조작에서 저널을 사용하는 방법에 대해 이해하십시오.

메시지 큐에 대한 지속 업데이트는 두 단계로 발생합니다. 업데이트를 나타내는 레코드는 저널에 먼저 작성되고, 그런 다음 큐 파일이 업데이트됩니다.

따라서 저널 수신자는 큐 파일보다 최신 데이터가 될 수 있습니다. 일관성 지점에서 재시작 처리가 시작되도록 하기 위해 IBM MQ는 체크포인트를 사용합니다.

체크포인트는 저널에 설명된 레코드가 큐의 레코드와 동일할 때입니다. 체크포인트 자체는 큐 관리자를 재시작해야 하는 일련의 저널 레코드로 구성됩니다. 예를 들어, 모든 트랜잭션의 상태(즉, 작업 단위)는 체크포인트 당시에 활성화됩니다.

체크포인트는 IBM MQ에서 자동으로 생성됩니다. 큐 관리자가 시작되고 시스템이 종료될 때 사용되고 특정 수의 조작이 로그된 이후에 사용됩니다.

큐 관리자에서 모든 오브젝트에 대한 RCDMQMIMG 명령을 실행하고 결과를 표시하여 다음과 같이 큐 관리자가 체크포인트를 가져오도록 강제실행할 수 있습니다.

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNTA(*YES)
```

큐는 추가 메시지를 핸들링하기 때문에 체크포인트 레코드는 큐의 현재 상태와 불일치하게 됩니다.

IBM MQ가 재시작될 때, 로그에 최신 체크포인트 레코드를 배치합니다. 이 정보는 모든 체크포인트의 끝에서 업데이트되는 체크포인트 파일에 있습니다. 체크포인트 레코드는 로그와 데이터 사이의 최신 일관성 지점을 나타냅니다. 이 체크포인트로부터의 데이터는 체크포인트 시간에 존재하는 것처럼 큐를 다시 빌드하기 위해 사용됩니다. 큐가 재작성될 때, 로그는 시스템 장애 또는 닫기 전에 있는 상태로 큐를 다시 되돌리기 위해 전달됩니다.

IBM MQ가 저널을 사용하는 방법에 대해 이해하려면 큐 관리자 TEST에서 TESTQ라는 로컬 큐의 경우를 고려하십시오. 이는 다음 IFS 파일에 의해 표시됩니다.

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

지정된 메시지가 이 큐에 배치되고 큐에서 검색되는 경우, 취해지는 조치가 [372 페이지의 그림 22](#)에서 표시됩니다.

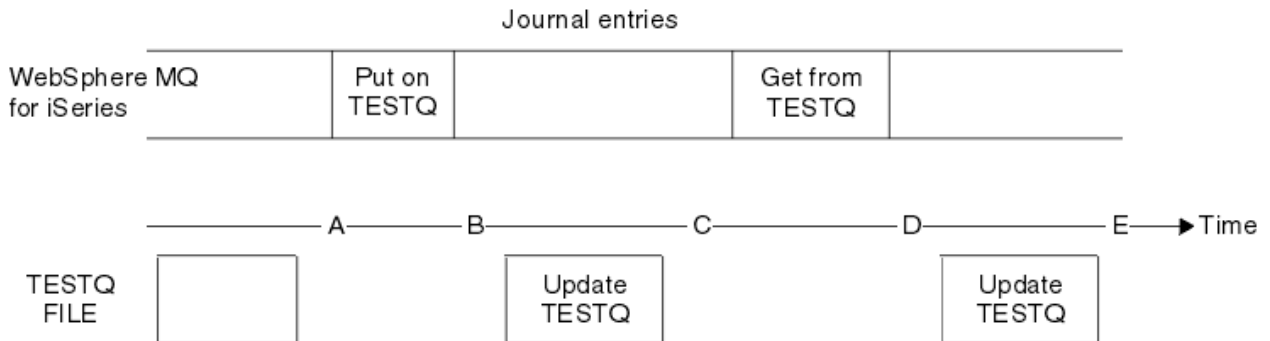


그림 22. MQM 오브젝트를 업데이트할 때 이벤트의 순서

다이어그램에 표시된 다섯 개의 지점(A-E)은 다음 상태를 정의하는 시간을 표시합니다.

- A** 큐의 IFS 파일 표현이 저널에 포함되는 정보와 일치합니다.
- B** 저널 입력 항목은 큐에 Put 조작을 정의하여 저널에 작성됩니다.
- C** 큐가 적절히 업데이트됩니다.
- D** 저널 입력 항목은 큐에서 Get 조작을 정의하여 저널에 작성됩니다.

E

큐가 적절히 업데이트됩니다.

IBM MQ for IBM i의 복구 기능에 대한 키는 사용자가 시간 A 당시의 TESTQ의 IFS 파일 표현을 저장할 수 있고, 계속해서 저장된 오브젝트를 복원하고 시간 A로부터 저널에 있는 입력 항목을 재실행하여 시간 E 당시의 TESTQ의 IFS 파일 표현을 복구할 수 있다는 것입니다.

이 전략은 시스템 실패 이후 지속 메시지를 복구하기 위해 IBM MQ for IBM i에서 사용됩니다. IBM MQ는 저널 수신자의 특정 입력 항목을 기억하고, 시동 시 이 지점으로부터 저널에 있는 입력 항목을 재실행하도록 합니다. 이 시동 입력 항목은 IBM MQ가 다음 시동 시 최소 필수 재실행만 수행하면 되도록 주기적으로 재계산됩니다.

IBM MQ에서는 오브젝트의 개별 복구가 제공됩니다. 오브젝트와 관련되는 모든 지속 정보는 로컬 IBM MQ for IBM i 저널에 기록됩니다. 손상되는 모든 IBM MQ 오브젝트는 저널에 보유한 정보로부터 완전하게 다시 빌드될 수 있습니다.

시스템이 수신자를 관리하는 방법에 대한 자세한 정보는 370 페이지의 『가용성, 백업, 복구 및 재시작(IBM i)』의 내용을 참조하십시오.

IBM i 매체 이미지(IBM i)

IBM i에서 매체 이미지는 저널에 기록된 IBM MQ 오브젝트의 완전한 사본입니다. 일부 손상된 오브젝트는 해당 매체 이미지에서 자동으로 복구될 수 있습니다.

긴 지속 기간의 IBM MQ 오브젝트는 작성된 지점으로 돌아가는 많은 수의 저널 입력 항목을 나타낼 수 있습니다. 이 문제를 방지하기 위해 IBM MQ for IBM i에서는 오브젝트의 매체 이미지 개념을 사용합니다.

이 매체 이미지는 저널에 기록된 IBM MQ 오브젝트의 완전한 사본입니다. 오브젝트의 이미지가 사용되는 경우, 오브젝트는 계속 이 이미지로부터 저널 입력 항목을 재실행하여 다시 빌드될 수 있습니다. 각 IBM MQ 오브젝트에 대한 재실행 지점을 나타내는 저널의 입력 항목은 해당 매체 복원 입력 항목으로 참조됩니다. IBM MQ는 트랙을 유지합니다.

- 각 큐 관리자 오브젝트에 대한 매체 복원 입력 항목.
- 이 세트로부터의 가장 오래된 항목(자세한 내용은 375 페이지의 『IBM i에서 저널 관리』의 오류 메시지 AMQ7462를 참조).

*CTLG 오브젝트 및 *MQM 오브젝트의 이미지는 이러한 오브젝트가 큐 관리자 재시작에 결정적이므로 규칙적으로 사용됩니다.

편리할 때 기타 오브젝트의 이미지가 사용됩니다. 기본적으로, 모든 오브젝트 이미지는 **ENDMQM** 명령을 **ENDCCTJOB(*YES)** 매개변수와 함께 사용하여 큐 관리자가 종료될 때 사용됩니다. 이 조작은 매우 큰 큐 관리자에 대해 상당한 시간이 걸릴 수 있습니다. 빠르게 시스템 종료해야 하는 경우, **ENDCCTJOB(*YES)**으로 **RCDMQMIMG(*NO)** 매개변수를 지정하십시오. 그러한 경우, 다음 명령을 사용하여 큐 관리자가 재시작된 후 저널에서 완전한 매체 이미지를 기록하는 것이 좋습니다.

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ는 오브젝트가 저널에서 작은 입력 항목에 의해 설명될 수 있는 편리한 위치를 발견하는 경우 오브젝트의 이미지를 자동으로 기록합니다. 그러나, 이는 일부 오브젝트(예: 많은 수의 메시지를 지속적으로 포함하는 큐)에는 절대 발생할 수 없습니다.

가장 오래된 매체 복원 항목의 날짜가 불필요하게 긴 기간 동안 계속될 수 있도록 허용하는 대신 IBM MQ 명령 **RCDMQMIMG**를 사용하여 선택한 오브젝트의 이미지를 수동으로 작성할 수 있습니다.

매체 이미지로부터의 복구

IBM MQ는 손상되었다는 것이 발견되면 해당 매체 이미지로부터 일부 오브젝트를 자동으로 복구합니다. 특히, 이는 정상 큐 관리자 시동의 일부로서 특수 *MQM 및 *CTLG 오브젝트에 적용됩니다. 동기점 트랜잭션이 큐 관리자의 최근 시스템 종료에서 미완료된 경우, 영향을 받은 큐도 시동 조작을 완료하기 위해 자동으로 복구됩니다.

IBM MQ 명령 **RCRMQOBJ**를 사용하여 기타 오브젝트를 수동으로 복구해야 합니다. 이 명령은 IBM MQ 오브젝트를 다시 작성하기 위해 저널에서 입력 항목을 재실행합니다. IBM MQ 오브젝트가 손상되면, 유일하게 올바른 조치는 이를 삭제하고 이 방법을 통해 이를 다시 작성하는 것입니다. 그러나, 비지속 메시지가 이 방식으로 복구될 수 없다는 것에 주목하십시오.

▶ IBM i IBM MQ for IBM i에서 체크포인트

체크포인트는 복구를 위해 알려진 지속적 시작점을 제공하기 위해 여러 번 수행됩니다.

프로세스 AMQZMUC0 내의 체크포인트 스레드는 다음 지점에서 체크포인트를 수행할 책임이 있습니다.

- 큐 관리자 시동(STRMQM).
- 큐 관리자 시스템 종료(ENDMQM).
- 마지막 체크포인트(기본 기간은 30분임) 이후 일정 기간 경과 및 이전 체크포인트 이후 최소 수(기본값은 100임)의 로그 레코드가 작성된 후.
- 많은 로그 레코드가 작성된 이후. 기본값은 10,000입니다.
- 저널 임계값 크기가 초과되고 새 저널 수신자가 자동으로 작성된 후.
- 전체 매체 이미지가 작성될 때:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

▶ IBM i IBM MQ for IBM i 데이터의 백업

이 정보를 사용하여 각 큐 관리자에 대한 IBM MQ 백업의 두 가지 유형을 이해하십시오.

각 큐 관리자의 경우, 고려할 두 가지 유형의 IBM MQ 백업이 있습니다.

- 데이터 및 저널 백업.

두 세트의 데이터 모두 지속적인지 확인하려면 큐 관리자를 종료한 후에만 이를 수행하십시오.

- 저널 백업.

큐 관리자가 활성인 동안 이를 수행할 수 있습니다.

두 방법 모두의 경우, 큐 관리자 IFS 디렉토리 및 큐 관리자 라이브러리의 이름을 찾아야 합니다. IBM MQ 구성 파일(mqs.ini)에서 이를 찾을 수 있습니다. 자세한 정보는 [멀티플랫폼에서 IBM MQ 구성 정보 변경을 참조하십시오](#).

다음 프로시저를 사용하여 두 가지 유형 모두의 백업을 수행하십시오.

특정 큐 관리자의 데이터 및 저널 백업

참고: 큐 관리자가 실행 중일 때 **활동 중 보관(save-while-active)** 요청을 사용하지 마십시오. 그러한 요청은 **보류 중인 변경 사항**을 가지는 모든 커밋 정의가 커밋되거나 롤백되지 않는 경우 완료될 수 없습니다. 큐 관리자가 활성일 때 이 명령이 사용되는 경우, 채널 연결이 정상적으로 종료될 수 없습니다. 항상 다음 프로시저를 사용하십시오.

1. 다음 명령을 사용하여 비어 있는 저널 수신자를 작성하십시오.

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. **RCDMQMIMG** 명령을 사용하여 모든 IBM MQ 오브젝트에 대한 MQM 이미지를 기록한 후 다음 명령을 사용하여 체크포인트를 강제 실행하십시오.

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. 채널을 종료하고 큐 관리자가 실행 중이지 않은지 확인하십시오. 큐 관리자가 실행 중인 경우, **ENDMQM** 명령으로 중지하십시오.
4. 다음 명령을 실행하여 큐 관리자 라이브러리를 백업하십시오.

```
SAVLIB LIB(QMTEST)
```

5. 다음 명령을 실행하여 큐 관리자 IFS 디렉토리를 백업하십시오.

```
SAV DEV(...) OBJ('/QIBM/UserData/mqm/qmgrs/test')
```

특정 큐 관리자의 저널 백업

모든 관련 정보가 저널에 유지되므로, 언젠가 전체 저장을 수행하는 한 부분 백업이 저널 수신자를 저장하여 수행될 수 있습니다. 이는 전체 백업의 시간 이후 모든 변경 사항을 레코딩하고 다음 명령을 실행하여 수행됩니다.

1. 다음 명령을 사용하여 비어 있는 저널 수신자를 작성하십시오.

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. **RCDMQMIMG** 명령을 사용하여 모든 IBM MQ 오브젝트에 대한 MQM 이미지를 기록한 후 다음 명령을 사용하여 체크포인트를 강제 실행하십시오.

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. 다음 명령을 사용하여 저널 수신자를 저장하십시오.

```
SAVOBJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

단순 백업 전략은 매주 IBM MQ 라이브러리의 전체 백업을 수행하는 것이고 매일 저널 백업을 수행하는 것입니다. 물론, 이는 엔터프라이즈에 대한 백업 전략을 설정하는 방법에 따라 다릅니다.

IBM i **IBM i에서 저널 관리**

사용자의 백업 전략의 일부로, 사용자의 저널 수신자에 주의하십시오. 다양한 이유로 IBM MQ 라이브러리에서 저널 수신자를 제거하는 것이 유용합니다.

- 공간 해제(이는 모든 저널 수신자에 적용됨)
- 시작할 때 성능 개선(STRMQM)
- 오브젝트 재작성 성능 개선(RCRMQMOBJ)

저널 수신자를 삭제하기 전에 백업 사본을 가지고 있고 더 이상 저널 수신자가 필요하지 않다는 것에 주의해야 합니다.

저널 수신자는 복구 조작에 필요한 경우 복원에 사용 가능하면 저널에서 분리되고 저장된 후 큐 관리자 라이브러리에서 제거될 수 있습니다.

저널 관리의 개념이 [376 페이지의 그림 23](#)에 표시됩니다.

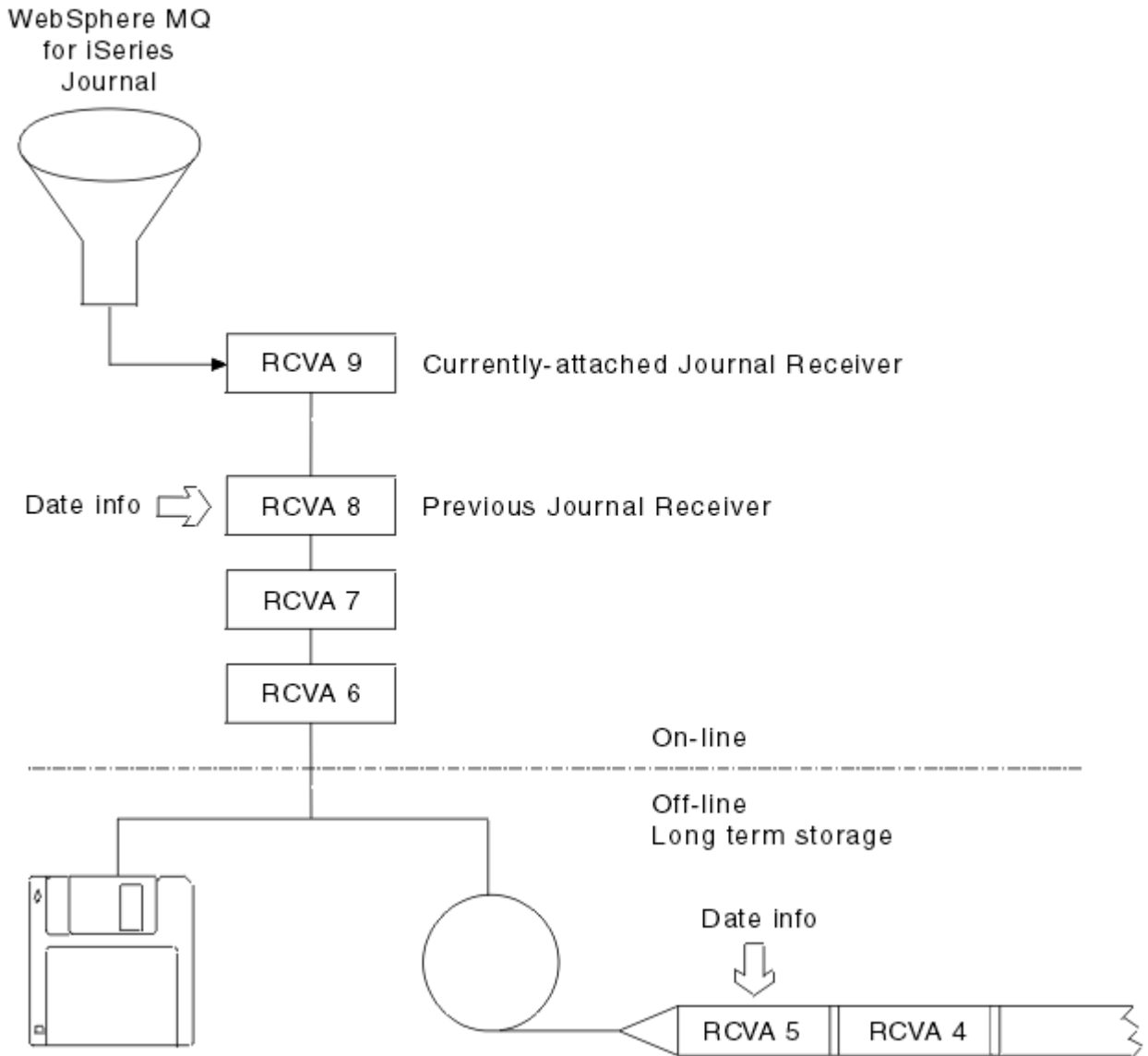


그림 23. IBM i에서 저널링

백업된 저널 수신자가 큐 관리자 라이브러리에서 제거될 수 있을 때와 백업 자체가 제거될 수 있을 때를 판별하기 위해 IBM MQ가 저널에서 다시 뒤로 이동할 수 있는 범위를 아는 것이 중요합니다.

IBM MQ는 이 시간을 판별하는 데 도움을 주기 위해 큐 관리자 메시지 큐(큐 관리자 라이브러리의 QMQMMSG)에 두 개의 메시지를 발행합니다. 이러한 메시지는 시작될 때, 로컬 저널 수신자를 변경할 때, RCDMQIMG를 사용하여 체크포인트를 강제 실행할 때 발행됩니다. 두 개의 메시지는 다음과 같습니다.

AMQ7460

시동 복구 포인트. 이 메시지는 IBM MQ가 시동 복구 단계 이벤트에서 저널을 재실행하는 시동 입력 항목의 날짜 및 시간을 정의합니다. 이 레코드를 포함하는 저널 수신자를 IBM MQ 라이브러리에서 사용할 수 있는 경우, 이 메시지에는 레코드를 포함하는 저널 수신자의 이름도 포함됩니다.

AMQ7462

가장 오래된 매체 복원 입력 항목. 이 메시지는 해당 매체 이미지에서 오브젝트를 다시 작성하기 위해 사용할 가장 오래된 항목의 날짜 및 시간을 정의합니다.

식별된 저널 수신자는 요청되는 가장 오래된 것입니다. 더 오래된 작성 날짜를 가지는 다른 어떤 IBM MQ 저널 수신자도 더 이상 요구되지 않습니다. 별표(*)만 표시되면, 가장 오래된 저널 수신자를 판별하기 위해 표시되는 날짜에서 백업을 복원해야 합니다.

- 다음 명령을 실행하여 전체 백업 시 해당 상태로 IBM MQ 데이터 라이브러리를 복원하기 위해 전체 백업에서 RSTLIB 조작을 수행하십시오.

```
RSTLIB LIB(QMQRLIB1) .....
RSTLIB LIB(QMQRLIB2) .....
```

저널 수신자가 부분적으로 하나의 저널 백업에 저장되고 후속 백업에 완전히 저장되는 경우, 완전히 저장된 것만 복원하십시오. 시간 순서로 저널을 개별적으로 복원하십시오.

- 다음 명령을 사용하여 IBM MQ IFS 디렉토리를 IFS 파일 시스템으로 복원하기 위해 RST 조작을 수행하십시오.

```
RST DEV(...) OBJ('/QIBM/UserData/mqm/qmgrs/testqm') ...
```

- 메시지 큐 관리자를 시작하십시오. 이는 전체 백업 이후 작성되는 모든 저널 레코드를 재실행하고 저널 백업 시 지속적 상태로 모든 IBM MQ 오브젝트를 복원합니다.

다른 시스템에서 전체 큐 관리자를 복원하려면 다음 프로시저를 사용하여 큐 관리자 라이브러리에서 모든 것을 복원하십시오. (샘플 큐 관리자 이름으로 TEST를 사용합니다.)

- CRTMQM TEST
- DLTLIB LIB(QMTEST)
- RSTLIB SAVLIB(QMTEST) DEV(*SAVF) SAVF(QMGRLIBSAV)
- 다음 IFS 파일을 삭제하십시오.

```
/QIBM/UserData/mqm/qmgrs/TEST/QMQMCHKPT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMQMOBJCAT
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMANAGER
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.AUTH.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CHANNEL.INITQ/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.COMMAND.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.REPOSITORY.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.TRANSMIT.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.PENDING.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.ADMIN.COMMAND.QUEUE/q
```

- STRMQM TEST
- RCRMQMOBJ OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(TEST)

IBM i IBM i에서 특정 큐 관리자에 대한 저널 수신자 복원

저널 수신자를 복원하기 위한 다른 방법을 이해하기 위해 이 정보를 사용하십시오.

제거된 수신자가 후속 복구 기능에 다시 필요하면, 가장 공통적인 조치는 백업 저널 수신자를 큐 관리자 라이브러리로 복원하는 것입니다.

이는 단순 태스크이고, 저널 수신자가 표준 IBM i RSTOBJ 명령을 사용하여 복원되도록 요구합니다.

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNRV) .....
```

일련의 저널 수신자는 단일 수신자 보다 복원될 필요가 있을 수도 있습니다. 예를 들어, AMQA000007은 IBM MQ 라이브러리에 있는 가장 오래된 수신자이고 AMQA000005와 AMQA000006 모두 복원되어야 합니다.

이 경우, 역시간 순서로 수신자를 개별적으로 복원하십시오. 이는 항상 필요하지만, 우수 사례는 아닙니다. 심각한 상황에서 복원된 저널 수신자를 저널과 연관시키기 위해 IBM i 명령 WRKJRNA를 사용해야 할 수도 있습니다.

저널을 복원할 때, 시스템은 저널 수신자 순서로 새 이름을 가지는 첨부된 저널 수신자를 자동으로 작성합니다. 그러나, 생성된 새 이름은 복원해야 하는 저널 수신자와 동일할 수도 있습니다. 매뉴얼 개입은 이 문제를 극복할 필요가 있습니다. 저널 수신자를 복원하기 전에 시퀀스와 새 저널에서 새 이름 저널 수신자를 작성합니다.

예를 들어, 저장된 저널 AMQAJRN과 다음 저널 수신자의 문제를 고려하십시오.

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

저널 AMQAJRN을 큐 관리자 라이브러리로 복구할 때, 시스템은 자동으로 저널 수신자 AMQA000000을 작성합니다. 이는 복원하려는 기존 저널 수신자(AMQA000000) 중 하나와의 수신자 충돌을 자동으로 생성하며, 이는 복원할 수 없습니다.

솔루션을 다음과 같습니다.

1. 다음 저널 수신자를 수동으로 작성하십시오(370 페이지의 『IBM i의 큐 관리자 저널』 참조).

```
CRTJRNRCV JRNRCV(QMQRLIB/AMQA9000001) THRESHOLD(XXXXX)
```

2. 저널 수신자로 저널을 수동으로 작성하십시오.

```
CRTJRN JRN(QMGRLIB/AMQAJRN) MNGRCV(*SYSTEM) +  
JRNRCV(QMGRLIB/AMQA9000001) MSGQ(QMGRLIB/AMQAJRNMSG)
```

3. 로컬 저널 수신자 AMQA000000을 AMQA900000으로 복원하십시오.

IBM i IBM i의 다중 인스턴스 큐 관리자

다중 인스턴스 큐 관리자는 활성 서버가 실패하면 자동으로 대기 서버로 전환하여 가용성을 향상시킵니다. 활성 서버와 대기 서버는 동일한 큐 관리자의 다중 인스턴스로 동일한 큐 관리자 데이터를 공유합니다. 활성 인스턴스가 실패하면 큐 관리자가 해당 큐를 다시 빌드할 수 있도록 인계받는 대기 인스턴스로 해당 저널을 전송해야 합니다.

활성 큐 관리자 인스턴스가 실패하면 사용하는 저널을 인계받는 대기 인스턴스가 사용할 수 있도록 다중 인스턴스 큐 관리자를 실행하는 IBM i 시스템을 구성합니다. 활성 인스턴스의 저널을 인계받는 인스턴스가 사용할 수 있도록 사용자 고유의 구성 및 관리 태스크를 설계할 수 있습니다. 메시지를 잃지 않으려면 실패 시점에 대기 저널이 활성 저널과 일치하도록 설계해야 합니다. 일관성을 유지하는 후속 토픽의 예로 설명되는 두 가지 구성 중 하나에서 설계를 수정할 수 있습니다.

1. 활성 큐 관리자 인스턴스를 실행하는 시스템에서 대기 인스턴스를 실행하는 시스템으로 저널을 미러링합니다.
2. 활성 인스턴스를 실행하는 시스템에서 대기 인스턴스로 전송 가능한 독립 보조 기억장치 풀(IASP)에 저널을 배치합니다.

첫 번째 솔루션은 기본 ASP를 사용하므로 추가 하드웨어 또는 소프트웨어가 필요하지 않습니다. 두 번째 솔루션에서는 별도 가격의 IBM i 라이선스 제품 5761-SS1 옵션 41로 제공되는 IBM i 클러스터링 지원이 필요한 전환 가능 IASP가 필요합니다.

IBM i 신뢰성 및 가용성(IBM i)

다중 인스턴스 큐 관리자의 목표는 애플리케이션 가용성 향상입니다. 기술 및 물리적 제한조건은 큐 관리자와 연속 조작을 백업하여 재해 복구 요구를 충족시키기 위해 다른 솔루션이 필요함을 의미합니다.

신뢰성과 가용성을 고려하여 구성하는 경우 많은 요인의 균형을 유지해야 하므로 다음과 같은 네 가지 설계점이 나타납니다.

재해 복구

모든 로컬 자산을 영구 삭제하는 중대 재해 후 복구를 위해 최적화됩니다.

IBM i의 재해 복구는 종종 IASP의 지리적 미러링을 기반으로 합니다.

백업

일반적으로 사용자 오류나 예측하지 못한 일부 기술적 문제점에 해당하는 국지적 실패 후 복구를 위해 최적화됩니다.

IBM MQ는 큐 관리자를 주기적으로 백업하기 위해 백업 큐 관리자를 제공합니다. 큐 관리자 저널의 비동기 복제를 사용하여 백업의 통화를 향상시킬 수도 있습니다.

가용성

서버 또는 디스크 고장과 같은 예측 불가능한 기술적 장애 이후 거의 중단되지 않는 서비스를 빠르게 제공할 수 있게 해주는 조작 복원을 위해 최적화됩니다.

복구는 일반적으로 분 단위로 측정되며 때때로 감지 시간이 복구 프로세스보다 오래 걸립니다. 다중 인스턴스 큐 관리자는 가용성을 고려한 구성을 지원합니다.

연속 조작

중단되지 않는 서비스 제공을 위해 최적화됩니다.

연속 조작 솔루션은 감지 문제점을 해결해야 하므로 거의 모든 경우 여러 시스템을 통해 동일한 작업을 제출하고 첫 번째 결과를 사용하거나 최소 두 가지 결과물을 비교(주요 고려 사항이 정확성인 경우)하는 작업이 포함됩니다.

다중 인스턴스 큐 관리자는 가용성을 고려한 구성을 지원합니다. 큐 관리자의 한 가지 인스턴스는 한 번에 하나씩 활성화됩니다. 대기 인스턴스로 전환하는 데는 시스템 구성, 로드, 조정 방법에 따라 10초에서 15초 이상이 소요됩니다.

다중 인스턴스 큐 관리자는 애플리케이션 프로그램이 큐 관리자 중단을 인식하지 않고도 계속 처리할 수 있는 다시 연결 가능한 IBM MQ MQI clients와 함께 사용되는 경우 거의 중단되지 않는 서비스로 보일 수 있습니다([자동 클라이언트 다시 연결](#) 주제 참조).

IBM i 고가용성 솔루션의 컴포넌트(IBM i)

강력한 큐 관리자 데이터용 네트워크 스토리지, 저널 복제 또는 강력한 큐 관리자 저널용 IASP 스토리지를 제공하여 다중 인스턴스 큐 관리자를 사용하거나 재시작 가능한 큐 관리자 서비스로 구성되는 애플리케이션의 다시 연결 가능한 클라이언트를 사용하여 고가용성 솔루션을 구성합니다.

다중 인스턴스 큐 관리자가 다른 서버에서 다른 큐 관리자 인스턴스 시작을 재개하여 큐 관리자 실패 감지에 반응합니다. 시작을 완료하려면 네트워크 스토리지의 공유 큐 관리자 데이터와 로컬 큐 관리자 저널의 해당 사본에 대한 액세스 권한이 인스턴스에 필요합니다.

고가용성 솔루션을 작성하려면 큐 관리자 데이터의 가용성, 로컬 큐 관리자 저널의 통화를 관리해야 하며 다시 연결 가능한 클라이언트 애플리케이션을 빌드하거나 애플리케이션을 큐 관리자가 재개될 때 자동으로 다시 시작되는 큐 관리자 서비스로 배치해야 합니다. 자동 클라이언트 다시 연결은 IBM MQ classes for Java에서 지원되지 않습니다.

큐 관리자 데이터

RAID 레벨 1 이상 디스크를 사용하여 공유되고 신뢰할 수 있는 고가용성 네트워크 스토리지에 큐 관리자 데이터를 배치합니다. 파일 시스템은 다중 인스턴스 큐 관리자의 공유 파일 시스템에 대한 요구 사항을 충족시켜야 합니다. 공유 파일 시스템 관련 요구 사항에 대한 자세한 정보는 [공유 파일 시스템 요구 사항](#)을 참조하십시오. 네트워크 파일 시스템 4(NFS4)는 이러한 요구 사항을 충족시키는 프로토콜입니다.

큐 관리자 저널

또한 대기 인스턴스가 해당 큐 관리자 데이터를 일치 상태로 복원할 수 있도록 큐 관리자 인스턴스가 사용하는 IBM i 저널을 구성해야 합니다. 중단되지 않는 서비스의 경우 이는 저널을 활성화 인스턴스가 실패한 시점의 상태로 복원해야 함을 의미합니다. 백업 또는 재해 복구 솔루션과 달리 이전 체크포인트로 저널을 복원하는 것으로는 충분하지 않습니다.

네트워크 스토리지의 여러 IBM i 시스템 간에 물리적으로 저널을 공유할 수 없습니다. 큐 관리자 저널을 실패 시점의 일치 상태로 복원하려면 실패 시점에 활성 큐 관리자 인스턴스와 로컬 상태였던 물리적 저널을 활성화된 새 인스턴스로 전송하거나 실행 중인 대기 인스턴스에서 저널 미러를 유지해야 합니다. 미러링된 저널은 실패 인스턴스에 속하는 로컬 저널과 정확히 일치하도록 유지된 원격 저널 복제본입니다.

다음 세 가지 구성은 다중 인스턴스 큐 관리자에 대한 저널 관리 방법을 설계하기 위한 시작점입니다.

1. 활성 인스턴스 ASP에서 대기 인스턴스 ASP로의 동기화된 저널 복제(저널 미러링) 사용
2. 큐 관리자 저널을 유지하도록 구성된 IASP를 활성 인스턴스에서 활성 인스턴스로 인계되는 대기 인스턴스로 전송
3. 동기화된 보조 IASP 미러 사용

IBM MQ IBM i CRTMQM 명령에서 iASP에 큐 관리자 데이터를 넣는 방법에 대한 자세한 정보는 [ASP 옵션](#)을 참조하십시오.

또한 IBM Documentation의 IBM i 정보에서 [고가용성](#) 을 참조하십시오.

애플리케이션

대기 큐 관리자가 재개될 때 큐 관리자에 자동으로 다시 연결되는 클라이언트를 빌드하려면 MQCONNX를 사용하여 큐 관리자로 애플리케이션을 연결하고 MQCNO 옵션 필드에서 MQCNO_RECONNECT_Q_MGR을 지정하십시오. 다시 연결 가능한 클라이언트를 사용하는 세 가지 샘플 프로그램을 보려면 [고가용성 샘플 프로그램](#)을, 복구용 클라이언트 애플리케이션 설계에 대한 정보는 [애플리케이션 복구](#)를 참조하십시오.

IBM i IBM i에서 NetServer를 사용하여 큐 관리자 데이터의 네트워크 공유 작성
IBM i 서버에서 큐 관리자 데이터를 저장하기 위한 네트워크 공유를 작성합니다. 큐 관리자 인스턴스를 호스트하는 두 서버에서 네트워크 공유에 액세스하기 위한 연결을 설정합니다.

시작하기 전에

- 이 태스크에는 세 가지 IBM i 서버가 필요합니다. 네트워크 공유는 그 중 한 가지 서버인 GAMMA에 정의됩니다. 다른 두 서버인 ALPHA와 BETA는 GAMMA에 연결됩니다.
- 세 가지 서버에 모두 IBM MQ를 설치합니다.
- System i® Navigator를 설치합니다([System i Navigator](#) 참조).

이 태스크 정보

- GAMMA에서 큐 관리자 디렉토리를 작성하고 사용자 프로파일 QMQM 및 QMQMADM에 올바른 소유권과 권한을 설정합니다. 디렉토리 및 권한은 GAMMA에 IBM MQ를 설치하여 쉽게 작성됩니다.
- System i Navigator를 사용하여 GAMMA의 큐 관리자 데이터 디렉토리에 대한 공유를 작성합니다.
- ALPHA 및 BETA에서 공유를 가리키는 디렉토리를 작성합니다.

프로시저

1. GAMMA에서 QMQM 사용자 프로파일을 소유자로 지정하고 QMQMADM을 기본 그룹으로 지정하여 큐 관리자 데이터를 호스트할 디렉토리를 작성하십시오.

팁:

올바른 권한으로 디렉토리를 빠르고 안전하게 작성할 수 있는 방법은 GAMMA에 IBM MQ를 설치하는 것입니다.

나중에 GAMMA에서 IBM MQ를 실행하지 않으려면 IBM MQ를 설치 제거하십시오. 설치 제거 후에는 소유자 QMQM 사용자 프로필 및 QMQMADM 기본 그룹과 함께 /QIBM/UserData/mqm/qmgrs 디렉토리가 GAMMA에 남아 있습니다.

이 태스크는 GAMMA의 /QIBM/UserData/mqm/qmgrs 디렉토리를 공유용으로 사용합니다.

2. System i Navigator **연결 추가** 마법사를 시작하고 GAMMA 시스템에 연결하십시오.
 - a) Windows 데스크탑에서 **System i Navigator** 아이콘을 두 번 클릭하십시오.

- b) 예를 클릭하여 연결을 작성하십시오.
 - c) **연결 추가** 마법사의 지시사항에 따라 IBM i 시스템에서 GAMMA로의 연결을 작성하십시오.
GAMMA에 대한 연결이 **내 연결**에 추가됩니다.
3. GAMMA에 새 파일 공유를 추가하십시오.
- a) **System i Navigator** 창에서 My Connections/GAMMA/File Systems의 File Shares 폴더를 클릭하십시오.
 - b) **내 태스크** 창에서 **IBM i NetServer 공유 관리**를 클릭하십시오.
새 창, **IBM i NetServer - GAMMA**가 데스크탑에서 열리고 공유 오브젝트가 표시됩니다.
 - c) Shared Objects 폴더 > **파일** > **새로 작성** > **파일**을 마우스 오른쪽 단추로 클릭하십시오.
새 창, **IBM i NetServer 파일 공유 - GAMMA**가 열립니다.
 - d) 예를 들어, 공유 이름으로 WMQ를 지정하십시오.
 - e) 액세스 제어를 Read/Write로 설정하십시오.
 - f) 이전에 작성한 /QIBM/UserData/mqm/qmgrs 디렉토리로 이동하여 **경로 이름**을 선택하고 **확인**을 클릭하십시오.
- IBM i NetServer 파일 공유 - GAMMA** 창이 닫히고 공유 오브젝트 창에 WMQ가 나열됩니다.
4. 공유 오브젝트 창에서 **WMQ**를 마우스 오른쪽 단추로 클릭하십시오. **파일** > **권한**을 클릭하십시오.
/QIBM/UserData/mqm/qmgrs 오브젝트에 대한 **Qmgrs 권한 - GAMMA** 창이 열립니다.
- a) 아직 설정되지 않은 경우 QMQM에 대한 다음 권한을 확인하십시오.
 - Read
 - Write
 - Execute
 - Management
 - Existence
 - Alter
 - Reference
 - b) 아직 설정되지 않은 경우 QMQMADM에 대한 다음 권한을 확인하십시오.
 - Read
 - Write
 - Execute
 - Reference
 - c) /QIBM/UserData/mqm/qmgrs에 대한 권한을 부여하려는 다른 사용자 프로파일을 추가하십시오.
예를 들어, /QIBM/UserData/mqm/qmgrs에 대한 기본 사용자 프로파일(공용) Read 및 Execute 권한을 부여할 수 있습니다.
5. GAMMA의 /QIBM/UserData/mqm/qmgrs에 대한 액세스 권한이 부여된 모든 사용자 프로파일이 GAMMA에 액세스하는 서버에서와 동일한 비밀번호를 사용하는지 확인하십시오.
특히 공유에 액세스하려는 다른 서버의 QMQM 사용자 프로파일의 비밀번호가 GAMMA의 QMQM 사용자 프로파일 비밀번호와 같은지 확인하십시오.
- 팁:** 비밀번호를 설정하려면 System i Navigator에서 My Connections/GAMMA/Users and Groups 폴더를 클릭하십시오. 또는 **CHFUSRPRF** 및 **CHGPWD** 명령을 사용하십시오.

결과

공유를 사용하는 다른 서버에서 GAMMA에 액세스할 수 있는지 확인하십시오. 다른 태스크를 수행하는 경우 /QNTC/GAMMA/WMQ 경로를 사용하여 ALPHA 및 BETA에서 GAMMA에 액세스할 수 있는지 확인하십시오. /

QNTC/GAMMA 디렉토리가 ALPHA 또는 BETA에 없는 경우에는 이 디렉토리를 작성해야 합니다. NetServer 도메인에 따라, 디렉토리를 작성하기 전에 ALPHA 또는 BETA에 IPL을 수행해야 할 수 있습니다.

```
CRTDIR DIR('/QNTC/GAMMA')
```

ALPHA 또는 BETA에서 /QNTC/GAMMA/WMQ에 액세스할 수 있는 것으로 확인한 경우 CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ') 명령을 실행하면 GAMMA에서 /QIBM/ UserData/mqm/qmgrs/QM1이 작성됩니다.

다음에 수행할 작업

392 페이지의 『IBM i에서 저널 미러링 및 NetServer를 사용하여 다중 인스턴스 큐 관리자 작성』 또는 396 페이지의 『IBM i에서 NetServer 및 저널 미러링을 사용하여 단일 인스턴스 큐 관리자를 다중 인스턴스 큐 관리자로 변환』 태스크의 단계에 따라 다중 인스턴스 큐 관리자를 작성하십시오.

IBM i IBM i에서 장애 복구 성능

큐 관리자 인스턴스가 실패한 사실을 감지하고 대기 인스턴스에서 처리를 재개하는 데 소요되는 시간은 구성에 따라 수십 초에서 15분 이상까지 다양합니다.고가용성 솔루션을 설계하고 테스트할 때는 성능을 중점적으로 고려해야 합니다.

다중 인스턴스 큐 관리자가 저널 복제를 사용하거나 IASP를 사용하도록 구성할지 여부를 결정하는 데 있어 고려해야 할 장점과 단점이 있습니다. 미러링을 위해서는 큐 관리자가 원격 저널에 동기식으로 쓰기를 수행해야 합니다. 하드웨어 관점에서는 성능에 영향을 주지 않아도 되지만 소프트웨어 관점에서는 로컬 저널에만 쓰는 것보다 원격 저널에 쓰는 경우의 경로 길이가 더 길므로 실행 큐 관리자의 성능이 어느 정도 저하되는 것으로 예상할 수 있습니다. 그러나 대기 큐 관리자가 인계하면, 실패 전에 활성 인스턴스가 유지하는 원격 저널로부터의 로컬 저널 동기화 지연이 IBM i가 IASP를 감지하여 큐 관리자의 대기 인스턴스를 실행하는 서버로 전송하는 데 소요되는 시간보다 일반적으로 더 짧습니다. IASP 전송 시간은 몇 초 내에 완료되지 않고 최대 10분에서 15분입니다. IASP 전송 시간은 IASP가 대기 시스템으로 전송될 때 상이해야 하는 오브젝트의 수, 액세스 경로의 크기 또는 병합되어야 하는 색인에 따라 다릅니다.

대기 큐 관리자가 인계하면, 실패 전에 활성 인스턴스가 유지하는 원격 저널로부터의 로컬 저널 동기화 지연이 IBM i가 독립 ASP를 감지하여 큐 관리자의 대기 인스턴스를 실행하는 서버로 전송하는 데 소요되는 시간보다 일반적으로 더 짧습니다. 독립 APS 전송 시간은 몇 초 내에 완료되지 않고 최대 10분에서 15분입니다. 독립 APS 전송 시간은 독립 APS가 대기 시스템으로 전송될 때 상이해야 하는 오브젝트의 수, 액세스 경로의 크기 또는 병합되어야 하는 색인에 따라 다릅니다.

그러나 저널 전송이 대기 인스턴스가 완전히 재개되는 데 소요되는 시간에 영향을 주는 유일한 요인은 아닙니다. 네트워크 파일 시스템이 대기 인스턴스가 시동을 계속 시도하도록 신호를 보내는 큐 관리자 데이터에 대한 잠금을 해제하는 데 소요되는 시간과 인스턴스가 메시지 처리를 다시 시작할 수 있도록 저널에서 큐를 복구하는 데 소요되는 시간 또한 고려해야 합니다. 이러한 기타 지연 요소가 모두 대기 인스턴스를 시작하는 데 소요되는 시간에 추가됩니다. 총 전환 시간은 다음 컴포넌트로 구성됩니다.

실패 감지 시간

NFS가 큐 관리자 데이터에 대한 잠금을 해제하고 대기 인스턴스가 시동 프로세스를 계속 진행하는 데 소요되는 시간

전송 시간

HA 클러스터의 경우 IBM i가 활성 인스턴스를 호스트하는 시스템에서 대기 인스턴스로 IASP를 전송하는 데 소요되는 시간과, 저널 복제의 경우 대기 인스턴스의 로컬 저널을 원격 복제본의 데이터로 업데이트하는 데 소요되는 시간

재시작 시간

새로 활성화된 큐 관리자 인스턴스가 복원된 저널의 최신 체크포인트에서 큐를 재빌드하고 메시지 처리를 재개하는 데 소요되는 시간

참고:

인계된 대기 인스턴스가 이전 활성 인스턴스에 동시에 복제되도록 구성되는 경우에는 시동이 지연될 수 있습니다. 원격 저널이 이전 활성 인스턴스를 호스트하는 서버에 있고 해당 서버가 실패한 경우에는 새로 활성화된 인스턴스가 원격 저널에 복제되지 못할 수 있습니다.

동기 응답을 대기하는 기본 시간은 1분입니다. 복제 제한시간이 초과되기 전에 최대 지연을 구성할 수 있습니다. 또는 실패한 활성 인스턴스에 대한 비동기 복제를 사용하여 대기 인스턴스가 시작되도록 구성할 수 있습니다. 실패한 인스턴스가 대기 인스턴스에서 다시 실행될 때 나중에 동기 복제로 전환합니다. 동기 독립 ASP 미러링을 사용하는 경우에도 동일한 고려사항이 적용됩니다.

전체 장애 복구 시간을 평가하는 데 도움을 줄 수 있도록, 또한 사용할 구성 접근 방식을 결정하는 데 영향을 줄 수 있도록 이 컴포넌트의 기준선을 별도로 측정할 수 있습니다. 가장 적합한 구성을 결정하려면 또한 동일한 서버에서 다른 애플리케이션이 장애를 복구하는 방법과 이미 IASP를 사용하는 백업 또는 재해 복구 프로세스가 있는지 여부를 고려해야 합니다.

다음과 같이 클러스터 구성을 조정하여 IASP 전송 시간을 줄일 수 있습니다.

1. 가변 프로세스에서 UID 및 GID를 변경하지 않아도 되도록 클러스터 내 시스템의 사용자 프로파일이 동일한 GID 및 UID를 갖습니다.
2. 디스크 풀 그룹에 대한 교차 참조 테이블을 작성하기 위해 병합되어야 하므로 시스템 및 기본 사용자 디스크 풀에서 데이터베이스 오브젝트 수를 최소화합니다.
3. 추가 성능 팁은 IBM Redbook, *PowerHA® for IBM i* 구현, SG24-7405를 참조하십시오.

기본 ASP, 저널 미러링, 소규모 구성을 사용하는 구성은 수십초 단위로 전환되어야 합니다.

IBM i IBM i 클러스팅 기능과 IBM MQ 클러스팅 결합의 개요

IBM i에서 IBM MQ를 실행하고 IBM i 클러스터링 기능을 이용하면 IBM MQ 클러스터링만 사용하는 것보다 더 포괄적인 고가용성 솔루션을 제공할 수 있습니다.

이 기능을 사용하려면 다음을 설정해야 합니다.

1. IBM i 시스템의 클러스터. 384 페이지의 『IBM i 클러스터』의 내용을 참조
2. 독립 보조 스토리지 풀(IASP). 여기로 큐 관리자를 이동시키십시오. 384 페이지의 『독립 보조 스토리지 풀 (IASP)』의 내용을 참조
3. 클러스터 자원 그룹(CRG). 385 페이지의 『디바이스 클러스터 자원 그룹』의 내용 참조. 여기에서 다음을 정의할 수 있습니다.
 - 복구 도메인
 - IASP
 - 엑시트 프로그램. 385 페이지의 『디바이스 CRG 엑시트 프로그램』의 내용 참조

IBM i 클러스터

IBM i 클러스터는 로컬에서 서로 링크된 IBM i 컴퓨터 또는 파티션인 인스턴스의 콜렉션입니다.

이 그룹화의 목적은 각 인스턴스를 백업할 수 있게 하여 단일 장애점을 제거하고 애플리케이션과 데이터의 탄력성을 높이기 위한 것입니다. 클러스터를 작성한 경우, 클러스터의 애플리케이션, 데이터 및 디바이스를 관리하도록 다양한 클러스터 자원 그룹(CRG) 유형을 구성할 수 있습니다.

자세한 정보는 [클러스터 작성 및 클러스터 작성\(CRTCLU\) 명령](#)을 참조하십시오.

독립 보조 스토리지 풀(IASP)

IASP는 단일 레벨 스토리지의 확장으로 기능하는 사용자 ASP의 유형입니다. 시스템 스토리지에서 독립되어 있으므로 시스템 IPL 없이 쉽게 조작할 수 있는 스토리지 부분입니다.

IASP는 다른 운영 체제 인스턴스로 쉽게 전환되거나 다른 운영 체제 인스턴스의 대상 IASP로 복제될 수 있습니다. 두 방법 모두 인스턴스 간 IASP를 전환하는 데 사용할 수 있습니다.

- 첫 번째 방법에는 클러스터의 모든 컴퓨터 및 IASP를 포함하는 전환 가능 디스크 타워가 고속 링크(HSL) 루프를 사용하여 연결되어야 합니다.
- 두 번째 방법에는 입출력 프로세서(IOS)가 두 파티션 사이에 전환될 수 있는 IBM i 컴퓨터에서 운영 체제 인스턴스가 파티셔닝되어야 합니다. IASP를 복제하는 데 특별한 하드웨어가 필요하지 않습니다. 이 복제는 네트워크에서 TCP/IP를 통해 수행됩니다.

자세한 정보는 [디바이스 ASP 구성\(CFGDEVASP\) 명령](#)을 참조하십시오.

디바이스 클러스터 자원 그룹

여러 개의 클러스터 자원 그룹(CRG) 유형이 있습니다. 사용 가능한 여러 CRG 유형에 대한 자세한 정보는 [클러스터 자원 그룹을 참조하십시오](#).

이 주제는 디바이스 CRG를 집중 조명합니다. 디바이스 CRG는 다음과 같습니다.

- 독립 보조 스토리지 풀(IASP)과 같은 디바이스 자원을 기술하고 관리합니다.
- 클러스터 노드의 복구 도메인을 정의합니다.
- 디바이스를 지정합니다. 그리고
- 클러스터 이벤트를 처리할 엑시트 프로그램을 지정합니다.

복구 도메인은 어떤 클러스터 노드를 기본 노드로 간주할지를 나타냅니다. 나머지 노드는 백업 노드로 간주됩니다. 백업 노드는 또한 복구 도메인에서 순서가 지정되는데 복구 도메인에 얼마나 많은 노드가 있는지에 따라 첫 번째 백업, 두 번째 백업 노드 등으로 지정됩니다.

기본 노드에 장애가 발생하면 엑시트 프로그램이 복구 도메인의 모든 노드에서 실행됩니다. 첫 번째 백업 노드에서 실행되는 엑시트 프로그램은 이 노드를 새로운 기본 노드로 만들기 위해 필요한 초기화를 작성할 수 있습니다.

자세한 정보는 [디바이스 CRG 작성 및 클러스터 자원 그룹 작성\(CRTCRCG\) 명령을 참조하십시오](#).

디바이스 CRG 엑시트 프로그램

장애 복구 또는 전환 이벤트와 같이 복구 도메인이 정의한 노드 중 하나에서 이벤트가 발생하면 운영 체제 클러스터 자원 서비스가 디바이스 CRG 엑시트 프로그램을 호출합니다.

장애 복구 이벤트는 클러스터 기본 노드가 실패하고 CRG가 관리하는 모든 자원으로 전환된 경우 발생하며 특정 CRG가 기본 노드에서 백업 노드로 수동으로 전환될 때 전환 이벤트가 발생합니다.

어느 쪽이든 엑시트 프로그램은 이전의 기본 노드에서 실행 중이었던 모든 프로그램의 초기화와 시작을 담당하며 이는 첫 번째 백업 노드를 새 기본 노드로 변환합니다.

예를 들어 IBM MQ에서 엑시트 프로그램은 IBM MQ 서브시스템(QMQM) 및 큐 관리자 시작을 담당해야 합니다. 큐 관리자는 트리거 모니터 같은 자동으로 리스너와 서비스를 시작하도록 구성되어야 합니다.

샘플 엑시트 프로그램 AMQSCR4는 IBM i에서 사용 가능합니다.

전환 가능한 IASP 구성

IBM MQ는 IBM i의 클러스터링 기능을 활용하도록 설정될 수 있습니다. 이를 수행하려면:

1. 데이터 센터 시스템 간에 IBM i 클러스터를 작성하십시오.
2. IASP로 큐 관리자를 이동하십시오.

[386 페이지의 『독립 보조 스토리지 풀로 큐 관리자 이동 및 풀에서 큐 관리자 제거』](#)에는 이 조작을 수행하는 데 도움이 되는 일부 샘플 코드가 포함되어 있습니다.


3. 복구 도메인, IASP, 엑시트 프로그램을 정의하는 CRG를 작성해야 합니다.

[385 페이지의 『디바이스 클러스터 자원 그룹 구성』](#)에는 이 조작을 수행하는 데 도움이 되는 일부 샘플 코드가 포함되어 있습니다.

관련 개념

[405 페이지의 『독립 ASP 및 고가용성』](#)

독립 ASP를 사용하여 서버 간에 애플리케이션과 데이터를 이동할 수 있습니다. 독립 ASP의 유연성은 ASP가 일부 IBM i 고가용성 솔루션의 기반임을 의미합니다. 큐 관리자 저널에 ASP 또는 독립 ASP를 사용할지 여부를 고려할 때는 독립 ASP를 기반으로 하는 다른 고가용성 구성을 고려해야 합니다.

 디바이스 클러스터 자원 그룹 구성
디바이스 클러스터 자원 그룹(CRG)을 설정하는 예제 프로그램입니다.

이 태스크 정보

다음 예에서 다음 사항에 유의하십시오.

- [PRIMARY SITE NAME] 및 [BACKUP SITE NAME]은 8자 이하의 두 개별 문자열이 될 수 있습니다.
- [PRIMARY IP] 및 [BACKUP IP]는 미러링에 사용될 수 있는 IP입니다.

프로시저

1. 클러스터의 이름을 식별하십시오.
2. CRG 엑시트 프로그램 이름 및 라이브러리를 식별하십시오.
3. 이 CRG로 정의할 기본 노드 및 백업 노드의 이름을 판별하십시오.
4. 이 CRG로 관리할 IASP를 식별하고 기본 노드 아래에서 작성되었는지 확인하십시오.
5. 다음 명령을 사용하여 백업 노드에 디바이스 설명을 작성하십시오.

```
CRTDEVASP DEVD([IASP NAME]) RSRNAME([IASP NAME])
```

6. 다음 명령을 사용하여 인계 IP 주소를 모든 노드에 추가하십시오.

```
ADDTCPIFC INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. 다음 명령을 사용하여 기본 노드에서만 인계 IP 주소를 시작하십시오.

```
STRTCPIFC INTNETADR(' [TAKEOVER IP]')
```

8. 옵션: IASP가 전환 가능하면 다음 명령을 호출하십시오.

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG( [CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT  
NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY) ([BACKUP NAME] *BACKUP))  
EXITPGMFMFMT(EXTP0200) CFGOBJ(([IAPS NAME] *DEVD *ONLINE ' [TAKEOVER IP]')
```

9. 옵션: IASP를 미러링하려면 다음 명령을 호출하십시오.

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(([PRIMARY NODE] *PRIMARY *LAST [PRIMARY SITE NAME] (' [PRIMARY  
IP] '))  
[BACKUP NAME] *BACKUP *LAST [BACKUP SITE NAME] (' [BACKUP IP] ')) EXITPGMFMFMT(EXTP0200)  
CFGOBJ(([IAPS NAME] *DEVD *ONLINE ' [TAKEOVER IP] '))
```

IBM i 독립 보조 스토리지 풀로 큐 관리자 이동 및 풀에서 큐 관리자 제거
큐 관리자를 독립 보조 스토리지 풀(IASP)로 이동하기 위한 예제 프로그램 및 IASP에서 큐 관리자를 제거하는 명령입니다.

이 태스크 정보

다음 예에서 다음 사항에 유의하십시오.

- [MANAGER NAME]은 큐 관리자의 이름입니다.
- [IASP NAME]은 IASP의 이름입니다.
- [MANAGER LIBRARY]는 큐 관리자 라이브러리의 이름입니다.
- [MANAGER DIRECTORY]는 큐 관리자 디렉토리의 이름입니다.

프로시저

1. 기본 노드 및 백업 노드를 식별하십시오.
2. 기본 노드에서 다음 프로시저를 수행하십시오.
 - a) 큐 관리자가 종료되었는지 확인하십시오.

b) 다음 명령을 사용하여 IASP가 vary on인지 확인하십시오.

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

c) IASP 아래에 큐 관리자 디렉토리를 작성하십시오.

다음과 같이 IASP 이름의 루트 아래에 있는 디렉토리일 수 있습니다.

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

d) 다음 명령을 사용하여 방금 IASP 아래에 작성된 큐 관리자 디렉토리로 관리자의 IFS 오브젝트를 이동하십시오.

```
QSH CMD('mv /QIBM/UserData/mqm/qmgrs/[MANAGER NAME]  
/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

e) 다음 명령을 사용하여 MGRLIB 이름의 임시 저장 파일을 작성하십시오.

```
CRTSAVF QGPL/MGRLIB
```

f) 다음 명령을 사용하여 큐 관리자 라이브러리를 MGRLIB 저장 파일로 저장하십시오.

```
SAVLIB LIB([MANGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
```

g) 다음 명령을 사용하여 큐 관리자 라이브러리를 삭제하고 모든 조회 메시지를 무시하십시오.

```
DLTLIB [MANAGER LIBRARY]
```

h) 다음 명령을 사용하여 큐 관리자 라이브러리를 IASP로 복원하십시오.

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)  
RSTASPDEV([IASP NAME])
```

i) 다음 명령을 사용하여 임시 저장 파일을 삭제하십시오.

```
DLTF FILE(QGPL/MGRLIB)
```

j) 다음 명령을 사용하여 IASP 아래에 큐 관리자 IFS 오브젝트에 대한 기호 링크를 작성하십시오.

```
ADDLNK OBJ('/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

k) 다음 명령을 사용하여 IASP에 첨부하십시오.

```
SETASPGRP [IASP NAME]
```

l) 다음 명령을 사용하여 큐 관리자를 시작하십시오.

```
STRMQM [MANAGER NAME]
```

3. 백업 노드에서 다음 프로시저를 수행하십시오.

a) 다음 명령을 사용하여 임시 큐 관리자 디렉토리를 작성하십시오.

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

b) 다음 명령을 사용하여 큐 관리자 임시 디렉토리에 대한 기호 링크를 작성하십시오.

```
ADDLNK OBJ('/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

c) 다음 명령을 사용하여 임시 디렉토리를 삭제하십시오.

```
QSH CMD('rm -r /[IASP_NAME]')
```

d) /QIBM/UserData/mqm/mqs.ini 파일의 끝에 다음을 추가하십시오.

```
QueueManager:  
Name=[MANAGER_NAME]  
Prefix=/QIBM/UserData/mqm  
Library=[MANAGER_LIBRARY]  
Directory=[MANAGER_DIRECTORY]
```

4. IASP에서 큐 관리자를 제거하려면 다음 명령을 실행하십시오.

- a) VRYCFG CFGOBJ([IASP_NAME]) CFGTYPE(*DEV) STATUS(*ON)
- b) SETASPGRP [IASP_NAME]
- c) ENDMQM [MANAGER_NAME]
- d) DLTMQM [MANAGER_NAME]

IBM i IBM i에서 ASP에 대한 미러링된 저널 구성

미러링된 저널 간 동기 복제를 사용하여 강력한 다중 인스턴스 큐 관리자를 구성합니다.

미러링된 큐 관리자 구성은 기본 또는 독립 보조 기억장치 풀(ASP)에서 작성되는 저널을 사용합니다.

IBM i의 경우에는 큐 관리자 데이터가 저널과 파일 시스템에 기록됩니다. 저널에는 큐 관리자 데이터의 마스터 사본이 포함됩니다. 저널은 동기 또는 비동기 저널 복제를 사용하여 시스템 간에 공유됩니다. 큐 관리자 인스턴스를 재시작하려면 로컬 저널과 원격 저널의 믹스가 필요합니다. 큐 관리자 재시작은 서버의 로컬 및 원격 저널 믹스의 저널 레코드와 공유 네트워크 파일 시스템의 큐 관리자 데이터를 읽습니다. 파일 시스템의 데이터로 인해 큐 관리자가 빨리 재시작됩니다. 파일 시스템과 저널 사이의 재동기화 지점을 표시하는 체크포인트가 파일 시스템에 저장됩니다. 일반 큐 관리자 재시작에는 체크포인트 이전에 저장된 저널 레코드가 필요하지 않습니다. 그러나 파일 시스템의 데이터가 최신 상태가 아닐 수 있으므로 체크포인트 이후 저널 레코드를 사용하여 큐 관리자 재시작을 완료합니다. 인스턴스에 첨부된 저널의 데이터는 최신 상태로 유지되므로 재시작이 성공적으로 완료될 수 있습니다.

그러나 대기 서버의 원격 저널이 비동기식으로 복제되고 동기화되기 전에 실패한 경우에는 저널 레코드도 최신 상태가 아닐 수 있습니다. 동기화되지 않은 원격 저널을 사용하여 큐 관리자를 재시작하려는 경우, 대기 큐 관리자 인스턴스가 활성 인스턴스가 실패하기 전에 삭제된 메시지를 재처리하거나 활성 인스턴스가 실패하기 전에 수신된 메시지를 처리하지 않을 수 있습니다.

드물지만 또 다른 경우 파일 시스템에는 최신 체크포인트 레코드가 포함되고 대기 서버에서 동기화되지 않은 원격 저널에는 포함되지 않는 경우입니다. 이러한 경우 큐 관리자가 자동으로 재시작되지 않습니다. 원격 저널이 동기화될 때까지 대기하거나 파일 시스템에서 대기 큐 관리자를 콜드 스타트할 수 있습니다. 이 경우 파일 시스템에 원격 저널보다 최신 상태의 큐 관리자 데이터 체크포인트가 포함되더라도 활성 인스턴스가 실패하기 전에 처리된 모든 메시지가 포함되지 않을 수 있습니다. 저널과 동기화되지 않은 콜드 리스타트 이후에 일부 메시지는 재처리되고 일부는 처리되지 않을 수 있습니다.

다중 인스턴스 큐 관리자를 사용하는 경우 활성 상태의 큐 관리자 인스턴스와 대기 인스턴스를 제어하기 위해 파일 시스템을 사용할 수도 있습니다. 활성 인스턴스는 큐 관리자 데이터에 대한 잠금을 획득합니다. 대기 인스턴스는 잠금 획득을 대기하며 획득 시 활성 인스턴스가 됩니다. 활성 인스턴스가 정상적으로 종료되면 잠금을 해제합니다. 파일 시스템이 활성 인스턴스가 실패했거나 파일 시스템에 액세스할 수 없는 것으로 감지하면 파일 시스템이 잠금을 해제합니다. 파일 시스템은 실패 감지를 위한 요구사항을 충족시켜야 합니다. 공유 파일 시스템에 대한 요구사항을 참조하십시오.

IBM i에서 다중 인스턴스 큐 관리자의 아키텍처는 서버 또는 큐 관리자 실패 후 자동 재시작을 제공합니다. 또한 큐 관리자 데이터가 저장되는 파일 시스템이 실패한 후 큐 관리자 데이터 복원을 지원합니다.

389 페이지의 그림 24에서 ALPHA가 실패하는 경우 미러링된 저널을 사용하여 BETA에서 QM1을 수동으로 재시작할 수 있습니다. QM1에 다중 인스턴스 큐 관리자 기능을 추가하면 ALPHA의 활성 인스턴스가 실패하는 경우 QM1의 대기 인스턴스가 BETA에서 자동으로 재개됩니다. QM1은 또한 QM1의 활성 인스턴스뿐 아니라 실패한 ALPHA 서버인 경우 자동으로 재개될 수 있습니다. BETA가 활성 큐 관리자 인스턴스의 호스트가 되면 대기 인스턴스가 ALPHA에서 시작될 수 있습니다.

389 페이지의 그림 24은 NetServer를 사용하여 큐 관리자 데이터를 저장하는 큐 관리자의 두 인스턴스 간에 저널을 미러링하는 구성을 보여줍니다. 보다 많은 저널과 그에 따라 보다 많은 인스턴스를 포함하도록 패턴을 확장할 수 있습니다. 370 페이지의 『IBM i의 큐 관리자 저널』 주제에서 설명하는 저널 이름 지정 규칙을 따르십시오. 현재 큐 관리자의 실행 인스턴스 수는 두 개(하나의 활성 인스턴스와 하나의 대기 인스턴스)로 제한됩니다.

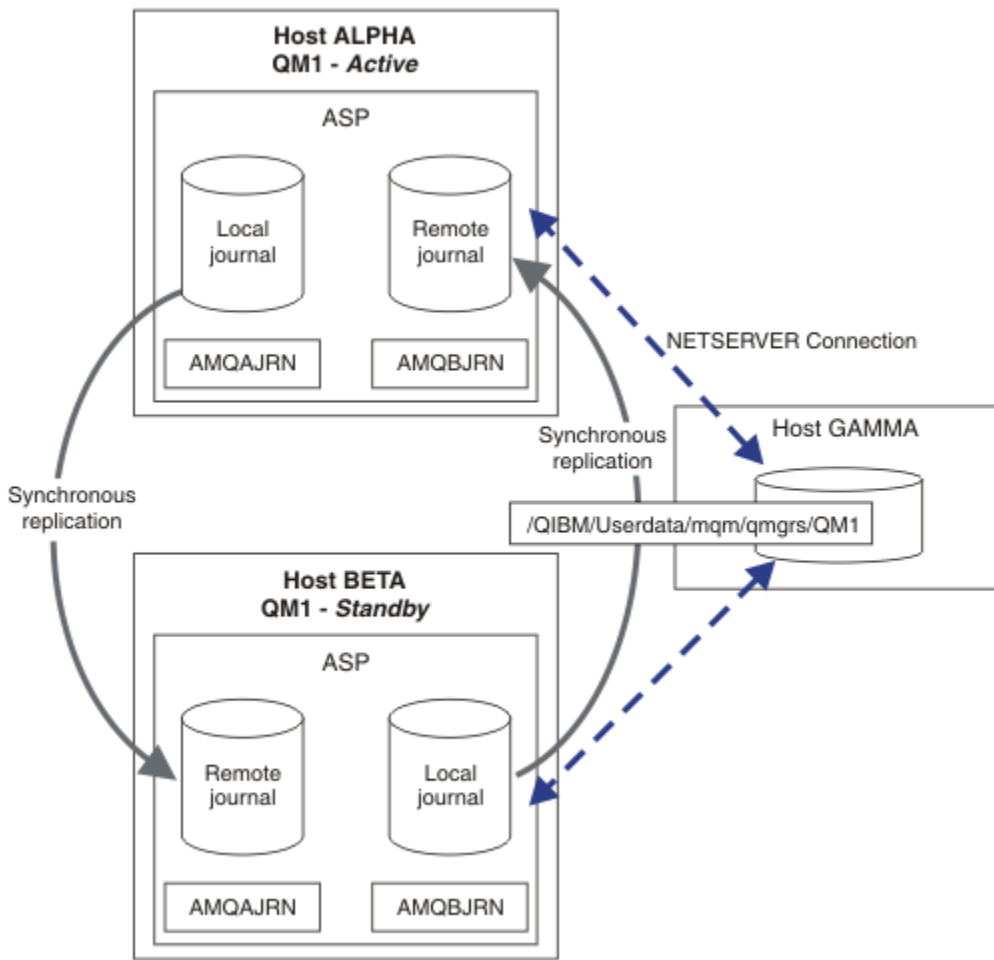


그림 24. 큐 관리자 저널 미러링

호스트 ALPHA의 QM1에 대한 로컬 저널을 AMQAJRN(보다 정확하게는 QMQM1/AMQAJRN)이라고 하며 BETA의 저널은 QMQM1/AMQBJRN입니다. 각 로컬 저널은 큐 관리자의 다른 모든 인스턴스에 있는 원격 저널에 복제됩니다. 큐 관리자가 두 개 인스턴스로 구성되면 로컬 저널이 하나의 원격 저널에 복제됩니다.

*SYNC 또는 *ASync 원격 저널 복제

IBM i 저널은 동기(*SYNC) 또는 비동기(*ASync) 저널링을 사용하여 미러링됩니다. [원격 저널 관리](#)를 참조하십시오.

389 페이지의 그림 24의 복제 모드는 *ASync가 아니라 *SYNC입니다. *ASync가 더 빠르지만 원격 저널 상태가 *ASyncPEND일 때 장애가 발생하면 로컬 및 원격 저널이 일치하지 않습니다. 원격 저널이 로컬 저널을 따라 잡아야 합니다. *SYNC를 선택하면 완료된 쓰기가 필요한 호출로부터 리턴되기 전에 로컬 시스템이 원격 저널을 대기합니다. 로컬 및 원격 저널은 일반적으로 서로 일치 상태를 유지합니다. *SYNC 조작이 지정된 시간보다 오래 걸리는 경우에만¹원격 저널링이 비활성화되면 저널이 동기화되지 않습니다. 저널 메시지 큐와 QSYSOPR에 오류가 로깅됩니다. 큐 관리자가 이 메시지를 감지하면 큐 관리자 오류 로그에 오류를 기록하고 큐 관리자 저널의 원격 복제를 비활성화합니다. 활성 큐 관리자 인스턴스는 이 저널에 대한 원격 저널링 없이 재개됩니다. 원격 서버를 다시 사용할 수 없게 되면 동기 원격 저널 복제를 수동으로 재활성화해야 합니다. 그러면 저널이 재동기화됩니다.

389 페이지의 그림 24에 나와 있는 *SYNC / *ASync 구성 관련 문제점은 BETA의 대기 큐 관리자 인스턴스가 제어할 수 있는 방식입니다. BETA의 큐 관리자 인스턴스가 첫 번째 지속 메시지를 쓰자마자 ALPHA에서 원격 저널 업데이트를 시도합니다. ALPHA에서 BETA로 전달되는 제어의 원인이 ALPHA 실패이고 ALPHA가 계속 작동

¹ 지정된 시간은 IBM i 5의 경우 60초이고 IBM i 6.1 이상의 경우 1 - 3600초 범위입니다.

중지 상태이면 ALPHA에 대한 원격 저널링이 실패합니다. BETA는 ALPHA가 응답할 때까지 대기한 다음 원격 저널링을 비활성화하며 로컬 저널링만으로 메시지 처리를 재개합니다. BETA는 ALPHA가 작동 중지 상태임을 감지하는 동안 대기해야 하므로 일정 기간 동안 비활성화됩니다.

원격 저널링을 *SYNC 또는 *ASync로 설정할지 선택하는 것은 상호 보완적입니다. 390 페이지의 표 24에는 한 쌍의 큐 관리자 간에 *SYNC 및 *ASync 저널링을 사용하는 데 따른 상호 보완성이 요약 설명되어 있습니다.

표 24. 원격 저널링 옵션			
활성	대기	*SYNC	*ASync
*SYNC		<ol style="list-style-type: none"> 1. 지속적 전환 및 장애 복구 2. 대기 인스턴스가 장애 복구 직후 재개되지 않습니다. 3. 원격 저널링은 항상 사용 가능해야 합니다. 4. 큐 관리자 성능은 원격 저널링에 따라 다릅니다. 	<ol style="list-style-type: none"> 1. 지속적 전환 및 장애 복구 2. 원격 저널링은 대기 서버를 사용할 수 있을 때 *SYNC로 전환되어야 합니다. 3. 재시작된 후에는 원격 저널링을 계속 사용할 수 있어야 합니다. 4. 큐 관리자 성능은 원격 저널링에 따라 다릅니다.
*ASync		<ol style="list-style-type: none"> 1. 바람직한 조합이 아님 	<ol style="list-style-type: none"> 1. 장애 복구 또는 전환 후 일부 메시지가 손실 또는 복제될 수 있습니다. 2. 대기 인스턴스가 항상 사용 가능한 상태가 아니어도 활성 인스턴스가 지연 없이 계속 진행됩니다. 3. 성능이 원격 저널링에 따라 영향을 받지 않습니다.

*SYNC / *ASync

활성 큐 관리자 인스턴스는 *SYNC 저널링을 사용하므로 대기 큐 관리자 인스턴스가 시작되면 즉시 *SYNC 저널링을 사용하려고 시도합니다.

1. 원격 저널은 트랜잭션 관점에서 활성 큐 관리자의 로컬 저널과 일치합니다. 큐 관리자가 대기 인스턴스로 전환되면 즉시 재개될 수 있습니다. 대기 인스턴스는 일반적으로 메시지 손실 또는 복제 없이 재개됩니다. 메시지는 마지막 체크포인트 이후 원격 저널링이 실패한 경우에만 손실 또는 복제되므로 이전에 활성화된 큐 관리자는 재시작할 수 없습니다.
2. 큐 관리자가 대기 인스턴스로 장애를 복구하면 즉시 시작하지 못할 수 있습니다. 대기 큐 관리자 인스턴스는 *SYNC 저널링으로 활성화됩니다. 장애 복구의 원인으로 인해 대기 인스턴스를 호스팅하는 서버에 대한 원격 저널링을 방해할 수 있습니다. 큐 관리자는 문제점이 감지될 때까지 대기한 후 지속 메시지를 처리합니다. 저널 메시지 큐와 QSYSOPR에 오류가 로깅됩니다. 큐 관리자가 이 메시지를 감지하면 큐 관리자 오류 로그에 오류를 기록하고 큐 관리자 저널의 원격 복제를 비활성화합니다. 활성 큐 관리자 인스턴스는 이 저널에 대한 원격 저널링 없이 재개됩니다. 원격 서버를 다시 사용할 수 없게 되면 동기 원격 저널 복제를 수동으로 재활성화해야 합니다. 그러면 저널이 재동기화됩니다.
3. 원격 저널을 유지하려면 원격 저널이 복제되는 서버가 항상 사용 가능해야 합니다. 원격 저널은 일반적으로 대기 큐 관리자를 호스팅하는 동일한 서버에 복제됩니다. 서버가 사용 불가능하게 될 수 있습니다. 저널 메시지 큐와 QSYSOPR에 오류가 로깅됩니다. 큐 관리자가 이 메시지를 감지하면 큐 관리자 오류 로그에 오류를 기록하고 큐 관리자 저널의 원격 복제를 비활성화합니다. 활성 큐 관리자 인스턴스는 이 저널에 대한 원격 저널링 없이 재개됩니다. 원격 서버를 다시 사용할 수 없게 되면 동기 원격 저널 복제를 수동으로 재활성화해야 합니다. 그러면 저널이 재동기화됩니다.
4. 원격 저널링은 로컬 저널링보다 느리며 서버가 먼 거리에 분산되어 있는 경우에는 속도가 더 느립니다. 큐 관리자는 원격 저널링을 대기해야 하므로 큐 관리자 성능이 저하됩니다.

한 쌍의 서버 간 *SYNC / *ASync 구성은 장애 복구 후 대기 인스턴스 재개가 지연되는 단점이 있습니다.

*SYNC / *ASync 구성에는 이러한 문제점이 없습니다.

*SYNC / *ASync는 원격 저널을 사용할 수 있는 한 전환 또는 장애 복구 후 메시지가 손실되지 않는 것으로 보장하지 않습니다. 장애 복구 또는 전환 후 메시지 손실 위험을 줄이려면 두 가지 방법 중 하나를 선택할 수 있

습니다. 원격 저널이 비활성화되는 경우 활성 인스턴스를 중지하거나 여러 서버에서 원격 저널을 작성하면 됩니다.

***SYNC / *ASync**

활성 큐 관리자 인스턴스는 *SYNC 저널링을 사용하므로 대기 큐 관리자 인스턴스가 시작되면 *ASync 저널링을 사용합니다. 새 대기 인스턴스를 호스팅하는 서버를 사용할 수 있게 된 직후 시스템 운영자가 활성 인스턴스의 원격 저널을 *SYNC로 전환해야 합니다. 운영자가 원격 저널링을 *ASync에서 *SYNC로 전환하면 원격 저널 상태가 *ASyncPEND인 경우 활성 인스턴스가 일시정지됩니다. 활성 큐 관리자 인스턴스는 나머지 저널 항목이 원격 저널로 전송될 때까지 대기합니다. 원격 저널이 로컬 저널과 동기화되면 새 대기 인스턴스가 트랜잭션 관점에서 새 활성 인스턴스와 다시 일치하게 됩니다. 다중 인스턴스 큐 관리자 관리 관점에서 볼 때 *SYNC / *ASync 구성에서는 IBM i 시스템 운영자에게 추가 태스크가 있습니다. 운영자는 실패한 큐 관리자 인스턴스를 재시작하는 것뿐 아니라 원격 저널링을 *SYNC로 전환해야 합니다.

1. 원격 저널은 트랜잭션 관점에서 활성 큐 관리자의 로컬 저널과 일치합니다. 활성 큐 관리자 인스턴스가 대기 인스턴스로 전환되거나 장애 복구되면 대기 인스턴스가 즉시 재개될 수 있습니다. 대기 인스턴스는 일반적으로 메시지 손실 또는 복제 없이 재개됩니다. 메시지는 마지막 체크포인트 이후 원격 저널링이 실패한 경우에만 손실 또는 복제되므로 이전에 활성화된 큐 관리자는 재시작할 수 없습니다.
2. 시스템 운영자는 활성 인스턴스를 호스팅하는 시스템을 다시 사용할 수 있게 되는 즉시 원격 저널을 *ASync에서 *SYNC로 전환해야 합니다. 운영자는 원격 저널을 *SYNC로 전환하기 전에 원격 저널이 따라잡을 때까지 대기할 수 있습니다. 또는 운영자는 원격 인스턴스를 *SYNC로 즉시 전환하고 대기 인스턴스 저널이 따라잡을 때까지 활성 인스턴스가 대기하도록 강제 실행할 수 있습니다. 원격 저널링이 *SYNC로 설정되면 일반적으로 대기 인스턴스가 활성 인스턴스와 트랜잭션 관점에서 일치하게 됩니다. 메시지는 마지막 체크포인트 이후 원격 저널링이 실패한 경우에만 손실 또는 복제되므로 이전에 활성화된 큐 관리자는 재시작할 수 없습니다.
3. 전환 또는 장애 복구에서 구성이 복원되면 원격 저널을 호스팅하는 서버를 항상 사용할 수 있어야 합니다.

장애 복구 직후 대기 큐 관리자를 재개하려면 *SYNC / *ASync를 선택하십시오. 새 활성 인스턴스에서 원격 저널 설정을 *SYNC로 수동 복원해야 합니다. *SYNC / *ASync 구성은 한 쌍의 다중 인스턴스 큐 관리자를 관리하는 정상 패턴과 일치합니다. 하나의 인스턴스가 실패하면 일정 시간 후 대기 인스턴스가 재시작되며 이 시간 동안에는 활성 인스턴스가 장애를 복구할 수 없습니다.

***ASync / *ASync**

활성 큐 관리자와 대기 큐 관리자를 호스팅하는 두 서버 모두 *ASync 원격 저널링을 사용하도록 구성됩니다.

1. 전환 또는 장애 복구가 수행되면 큐 관리자가 새 서버의 저널을 계속 처리합니다. 전환 또는 장애 복구가 수행될 때 저널이 동기화되지 않을 수 있습니다. 결과적으로 메시지가 손실 또는 복제될 수 있습니다.
2. 대기 큐 관리자를 호스팅하는 서버를 사용할 수 없는 경우라도 활성 인스턴스는 실행됩니다. 로컬 저널은 사용 가능한 경우 대기 서버와 비동기식으로 복제됩니다.
3. 로컬 큐 관리자의 성능은 원격 저널링으로 영향을 받지 않습니다.

성능이 주된 요구사항이고 장애 복구 또는 전환 후 일부 메시지가 손실 또는 복제되어도 무방한 경우에는 *ASync / *ASync를 선택하십시오.

***ASync / *SYNC**

이 옵션 조합은 사용하지 않아도 됩니다.

원격 저널에서 큐 관리자 활성화

저널은 동기 또는 비동기식으로 복제됩니다. 원격 저널은 활성화되지 않거나 로컬 저널을 따라잡을 수 있습니다. 원격 저널은 최근에 활성화되지 않았을 수 있으므로 동기식으로 복제되더라도 따라잡을 수 있습니다. 큐 관리자가 시작 시 사용하는 원격 저널의 상태에 적용하는 규칙은 다음과 같습니다.

1. 대기가 대기의 원격 저널에서 재생해야 하고 저널 상태가 *FAILED 또는 *INACTPEND인 경우 대기 시작이 실패합니다.
2. 대기 활성화가 시작되면 대기의 원격 저널 상태가 *ACTIVE 또는 *INACTIVE여야 합니다. 상태가 *INACTIVE이고 일부 저널 데이터가 복제되지 않은 경우에는 활성화가 실패할 수 있습니다.

네트워크 파일 시스템의 큐 관리자 데이터에 원격 저널에 있는 것보다 최신 체크포인트 레코드가 있으면 실패가 발생합니다. 원격 저널이 체크포인트 사이 기본 최대 간격인 30분 이내에 활성화되면 실패가 발생할 가능성이 낮습니다. 대기 큐 관리자가 파일 시스템에서 최신 체크포인트 레코드를 읽으면 시작되지 않습니다.

활성 서버의 로컬 저널을 복원할 수 있을 때까지 기다리거나 대기 큐 관리자를 콜드 스타트할 수 있습니다. 콜드 스타트를 선택하면 큐 관리자가 저널 데이터 없이 시작되며 파일 시스템에서 큐 관리자 데이터의 일관성과 완전성에 의존합니다.

참고: 큐 관리자를 콜드 스타트하는 경우 마지막 체크포인트 후 메시지 손실 또는 복제의 위험이 있습니다. 메시지 트랜잭션은 저널에 기록되었지만 일부 트랜잭션이 파일 시스템에서 큐 관리자 데이터에 기록되지 않을 수 있습니다. 큐 관리자를 콜드 스타트하는 경우 새 저널이 시작되며 파일 시스템에서 큐 관리자 데이터에 기록되지 않은 트랜잭션이 손실됩니다.

3. 대기 큐 관리자 활성화는 대기의 원격 저널 상태가 *ASYNCPEND 또는 *SYNCPEND에서 *ASYNC 또는 *SYNC로 변경될 때까지 대기합니다. 실행 제어기의 작업 로그에 주기적으로 메시지가 기록됩니다.

참고: 이 경우 활성화되는 대기 큐 관리자에 로컬인 원격 저널에서 활성화가 대기합니다. 큐 관리자는 또한 원격 저널 없이 일정 시간 대기한 후 계속 진행합니다. 하나 이상의 원격 저널에 동기식 쓰기를 시도하고 저널을 사용할 수 없을 때 대기합니다.

4. 저널 상태가 *FAILED 또는 *INACTPEND로 변경되면 활성화가 중지됩니다.

활성화에 사용될 로컬 및 원격 저널의 이름과 상태는 큐 관리자 오류 로그에 기록됩니다.

IBM i IBM i에서 저널 미러링 및 NetServer를 사용하여 다중 인스턴스 큐 관리자 작성
두 개의 IBM i 서버에서 실행될 다중 인스턴스 큐 관리자를 작성합니다. 큐 관리자 데이터는 NetServer를 사용하는 세 번째 IBM i 서버에 저장됩니다. 큐 관리자 저널은 원격 저널링을 사용하여 두 서버 간에 미러링됩니다. **ADDQMQRN** 명령은 원격 저널 작성을 단순화하는 데 사용됩니다.

시작하기 전에

1. 이 태스크에는 IBM i 서버 세 개가 필요합니다. 이 중 두 개 서버(예에서 ALPHA 및 BETA)에 IBM MQ를 설치하십시오. 제품은 IBM WebSphere MQ 7.0.1 Fix Pack 1 이상이어야 합니다.
2. 세 번째 서버는 IBM i 서버로, NetServer가 ALPHA와 BETA에 연결합니다. 이 서버는 큐 관리자 데이터를 공유하는 데 사용됩니다. IBM MQ 설치하는 필요하지 않습니다. 큐 관리자 디렉토리 및 권한을 설정하기 위해 임시 조치로 서버에 IBM MQ를 설치하는 것이 좋습니다.
3. QMQM 사용자 프로필의 비밀번호가 세 서버에서 모두 동일한지 확인하십시오.
4. IBM i NetServer를 설치하십시오([i5/OS NetServer](#) 참조).

이 태스크 정보

395 페이지의 그림 25에 표시된 구성을 작성하려면 다음 단계를 수행하십시오. 큐 관리자 데이터는 IBM i NetServer를 사용하여 연결됩니다.

- ALPHA 및 BETA에서 큐 관리자 데이터를 저장할 GAMMA의 디렉토리 공유로의 연결을 작성하십시오. 이 태스크는 또한 필요한 권한, 사용자 프로파일, 비밀번호를 설정합니다.
- 큐 관리자 인스턴스를 실행할 IBM i 시스템에 관계형 데이터베이스 항목(RDBE)을 추가하십시오. RDBE 항목은 원격 저널링에 사용되는 IBM i 시스템에 연결하는 데 사용됩니다.
- IBM i 서버 ALPHA에서 큐 관리자 QM1 를 작성하십시오.
- 다른 IBM i 서버 BETA에서 QM1 에 대한 큐 관리자 제어 정보를 추가하십시오.
- 두 IBM i 서버에서 두 가지 큐 관리자 인스턴스 모두에 대한 원격 저널을 작성하십시오. 각 큐 관리자는 로컬 저널에 쓰기를 수행합니다. 로컬 저널은 원격 저널에 복제됩니다. **ADDQMQRN** 명령은 저널 및 연결 추가를 단순화합니다.
- 대시 인스턴스를 허용하여 큐 관리자를 시작하십시오.

프로시저

1. 381 페이지의 『IBM i에서 NetServer를 사용하여 큐 관리자 데이터의 네트워크 공유 작성』 태스크를 수행하십시오.

결과적으로 ALPHA 및 BETA에는 GAMMA의 /QIBM/UserData/mqm/qmgrs를 가리키는 공유 /QNTC/GAMMA/WMQ가 있습니다. 사용자 프로파일 QMQM 및 QMQMADM은 필요한 권한을 갖고 QMQM은 세 개 시스템 모두에서 일치하는 비밀번호를 갖습니다.

2. 큐 관리자 인스턴스를 호스트할 IBM i 시스템에 관계형 데이터베이스 항목(RDBE)을 추가하십시오.
 - a) ALPHA에서 BETA에 대한 연결을 작성하십시오.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) BETA에서 ALPHA에 대한 연결을 작성하십시오.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. GAMMA에 큐 관리자 데이터를 저장하여 ALPHA에서 큐 관리자 QM1을 작성하십시오.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ')
```

경로는 NetServer 를 사용하여 큐 관리자 데이터를 작성합니다.

4. ALPHA에서 실행하십시오. 이 명령은 BETA에서 원격 저널을 추가합니다.

```
ADDQMQRN MQMNAME(QM1) RMTJRNRDB(BETA)
```

활성 인스턴스가 ALPHA에 있을 때 ALPHA의 로컬 저널에 저널 항목을 작성합니다. ALPHA의 로컬 저널은 BETA의 원격 저널에 복제됩니다.

5. ALPHA에서 작성된 IBM MQ 구성 데이터를 검사하려면, 명령을 사용하십시오.

이 정보는 다음 단계에서 필요합니다.

이 예에서 다음 구성은 ALPHA에서 작성됩니다.

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMQM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. 다음 명령을 사용하여 BETA에서 QM1의 큐 관리자 인스턴스를 작성하십시오. BETA에서 다음 명령을 실행하여 BETA에서 큐 관리자 제어 정보를 수정하십시오.

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMQM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

팁: 구성 정보를 복사하여 붙여넣으십시오. 큐 관리자 스탠자는 ALPHA와 BETA에서 동일합니다.

7. BETA에서 실행하십시오. 이 명령은 BETA의 로컬 저널과 ALPHA의 리모트 저널을 추가합니다.

```
ADDQMQRN MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

활성 인스턴스가 BETA에 있는 경우 BETA의 로컬 저널에 저널 항목을 작성합니다. BETA의 로컬 저널은 ALPHA의 원격 저널에 복제됩니다.

참고: 또는 비동기 저널링을 사용하여 BETA에서 ALPHA로의 원격 저널링을 설정할 수 있습니다.

단계 393 페이지의 『7』의 명령 대신 이 명령을 사용하여 BETA에서 ALPHA로의 비동기 저널링을 설정하십시오.

```
ADDQMQRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNC)
```

ALPHA의 서버 또는 저널링이 실패의 소스인 경우에는 새 저널 항목이 ALPHA에 복제될 때까지 대기하지 않고 BETA가 시작됩니다.

ALPHA가 다시 온라인 상태가 되면 명령을 사용하여 복제 모드를 *SYNC로 전환하십시오.

388 페이지의 『IBM i에서 ASP에 대한 미러링된 저널 구성』의 정보를 사용하여 저널을 동기, 비동기 또는 두 가지 방식으로 모두 미러링되는지 여부를 결정하십시오. 기본값은 원격 저널로부터의 응답을 60초 동안 대기하고 동기식으로 복제하는 것입니다.

8. ALPHA 및 BETA의 저널이 사용으로 설정되어 있고 원격 저널 복제의 상태가 사용으로 설정되어 있는지 확인하십시오.

a) ALPHA:

```
WRKMQRN MQMNAME(QM1)
```

b) BETA:

```
WRKMQRN MQMNAME(QM1)
```

9. ALPHA와 BETA에서 큐 관리자 인스턴스를 시작하십시오.

a) ALPHA에서 첫 번째 인스턴스를 시작하고 활성화하십시오. 대기 인스턴스로의 전환을 사용하십시오.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) BETA에서 두 번째 인스턴스를 시작하고 활성화하십시오.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

결과

큐 관리자 상태를 확인하는 데 사용합니다.

1. ALPHA에서 큐 관리자 인스턴스의 상태는 다음과 같아야 합니다.
2. BETA의 큐 관리자 인스턴스 상태는 다음과 같아야 합니다.

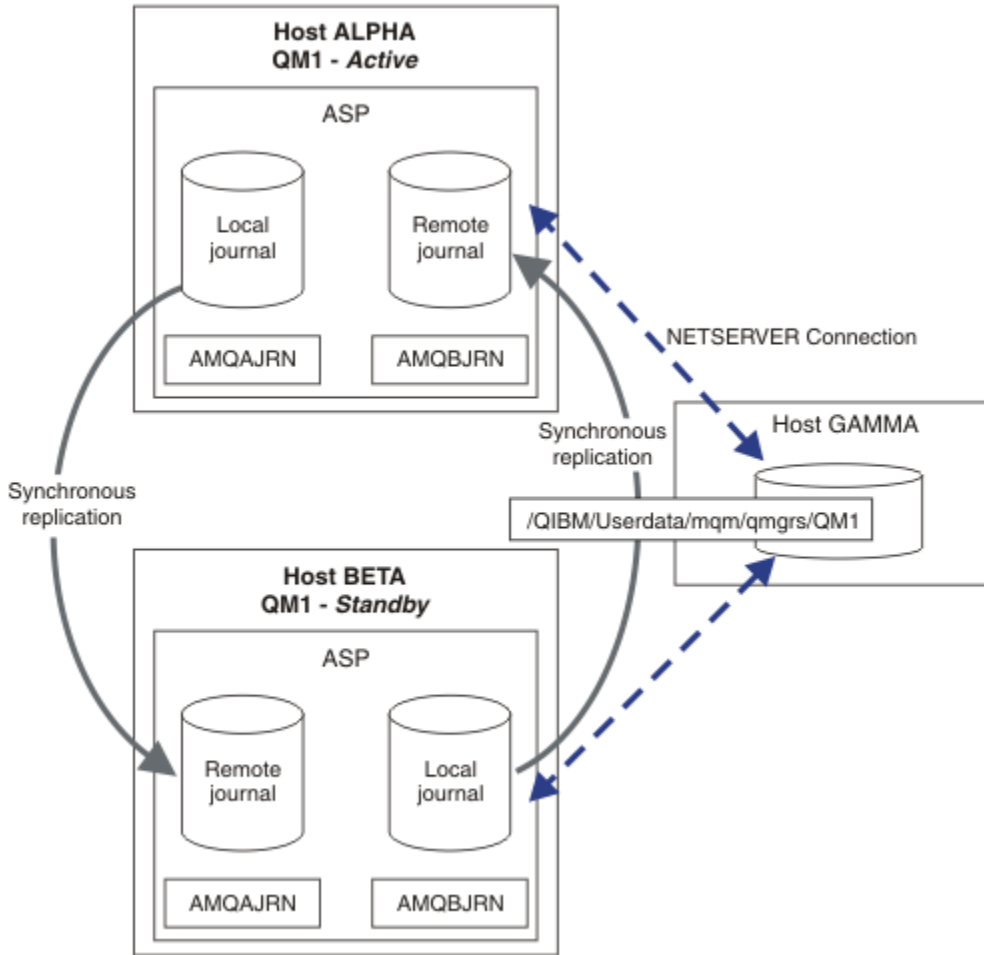


그림 25. 미러링된 저널 구성

다음에 수행할 작업

- 활성 및 대기 인스턴스가 자동으로 전환되는지 확인하십시오. 샘플 고가용성 샘플 프로그램을 실행하여 전환을 테스트할 수 있습니다([고가용성 샘플 프로그램 참조](#)). 샘플 프로그램은 'C' 클라이언트입니다. 이 프로그램은 Windows 또는 Unix 플랫폼에서 실행할 수 있습니다.

1. 고가용성 샘플 프로그램을 시작하십시오.
2. ALPHA에서 전환을 요청하는 큐 관리자를 종료하십시오.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. 베타의 인스턴스가 활성 상태인지 확인하십시오.
4. ALPHA에서 재시작

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- 대체 고가용성 구성을 확인하십시오.
 1. NetServer를 사용하여 Windows Server에 큐 관리자 데이터를 배치하십시오.

2. 원격 저널링을 사용하여 큐 관리자 저널을 미러링하는 대신 독립 ASP에 저널을 저장하십시오. 독립 ASP를 ALPHA에서 BETA로 전송하려면 IBM i 클러스터링을 사용하십시오.

IBM i IBM i에서 NetServer 및 저널 미러링을 사용하여 단일 인스턴스 큐 관리자를 다중 인스턴스 큐 관리자로 변환

단일 인스턴스 큐 관리자를 다중 인스턴스 큐 관리자로 변환합니다. 큐 관리자 데이터를 NetServer가 연결한 네트워크 공유로 이동합니다. 원격 저널링을 사용하여 큐 관리자 저널을 두 번째 IBM i 서버로 미러링합니다.

시작하기 전에

1. 이 태스크에는 IBM i 서버 세 개가 필요합니다. 예의 ALPHA 서버에서 기존 IBM MQ 설치에 IBM WebSphere MQ 7.0.1 Fix Pack 1 이상이어야 합니다. 예에서 ALPHA는 큐 관리자 QM1을 실행합니다.
2. IBM MQ를 두 번째 IBM i 서버(예의 BETA)에 설치하십시오.
3. 세 번째 서버는 IBM i 서버로, NetServer가 ALPHA와 BETA에 연결합니다. 이 서버는 큐 관리자 데이터를 공유하는 데 사용됩니다. IBM MQ 설치는 필요하지 않습니다. 큐 관리자 디렉토리 및 권한을 설정하기 위해 임시 조치로 서버에 IBM MQ를 설치하는 것이 좋습니다.
4. QMQM 사용자 프로필의 비밀번호가 세 서버에서 모두 동일한지 확인하십시오.
5. IBM i NetServer를 설치하십시오([i5/OS NetServer](#) 참조).

이 태스크 정보

단일 인스턴스 큐 관리자를 399 페이지의 그림 26에 표시된 다중 인스턴스 큐 관리자로 변환하려면 다음 단계를 수행하십시오. 단일 인스턴스 큐 관리자가 태스크에서 삭제된 다음 재작성되며, 큐 관리자 데이터는 NetServer가 연결하는 네트워크 공유에 저장됩니다. 이 프로시저는 **CPY** 명령을 사용하여 큐 관리자 디렉토리 및 파일을 네트워크 공유로 이동하는 것보다 신뢰할 수 있는 방법입니다.

- ALPHA 및 BETA에서 큐 관리자 데이터를 저장할 GAMMA의 디렉토리 공유로의 연결을 작성하십시오. 이 태스크는 또한 필요한 권한, 사용자 프로파일, 비밀번호를 설정합니다.
- 큐 관리자 인스턴스를 실행할 IBM i 시스템에 관계형 데이터베이스 항목(RDBE)을 추가하십시오. RDBE 항목은 원격 저널링에 사용되는 IBM i 시스템에 연결하는 데 사용됩니다.
- 큐 관리자 로그와 정의를 저장하고 큐 관리자를 중지한 후 삭제하십시오.
- 큐 관리자를 재작성하고 GAMMA의 네트워크 공유에 큐 관리자 데이터를 저장하십시오.
- 큐 관리자의 두 번째 인스턴스를 다른 서버에 추가하십시오.
- 두 IBM i 서버에서 두 가지 큐 관리자 인스턴스 모두에 대한 원격 저널링을 작성하십시오. 각 큐 관리자는 로컬 저널에 쓰기를 수행합니다. 로컬 저널은 원격 저널에 복제됩니다. **ADDQMJRN** 명령은 저널 및 연결 추가를 단순화합니다.
- 대시 인스턴스를 허용하여 큐 관리자를 시작하십시오.

참고:

태스크의 단계 397 페이지의 『4』에서 단일 인스턴스 큐 관리자, QM1을 삭제합니다. 큐 관리자를 삭제하면 큐의 모든 지속 메시지가 삭제됩니다. 따라서 큐 관리자로 변환하기 전에 큐 관리자가 저장한 모든 메시지 처리를 완료하십시오. 모든 메시지 처리가 불가능한 경우에는 단계 397 페이지의 『4』 이전에 큐 관리자 라이브러리를 백업하십시오. 단계 397 페이지의 『5』 이후 큐 관리자 라이브러리를 복원하십시오.

참고:

이 태스크의 397 페이지의 『5』 단계에서 QM1을 재작성하십시오. 큐 관리자의 이름은 같더라도 큐 관리자 ID는 다릅니다. 큐 관리자 클러스터링은 큐 관리자 ID를 사용합니다. 클러스터에서 큐 관리자를 삭제한 후 재작성하려면 먼저 클러스터에서 큐 관리자를 제거해야 합니다. [클러스터에서 큐 관리자 제거: 대체 메소드 또는 클러스터에서 큐 관리자 제거](#)를 참조하십시오. 큐 관리자를 재작성했으면 클러스터에 추가하십시오. 이전과 이름은 같지만 클러스터의 다른 큐 관리자가 새 큐 관리자로 인식합니다.

프로시저

1. 381 페이지의 『IBM i에서 NetServer를 사용하여 큐 관리자 데이터의 네트워크 공유 작성』 태스크를 수행하십시오.

결과적으로 ALPHA 및 BETA에는 GAMMA의 /QIBM/UserData/mqm/qmgrs를 가리키는 공유 /QNTC/GAMMA/WMQ가 있습니다. 사용자 프로파일 QMQM 및 QMQMADM은 필요한 권한을 갖고 QMQM은 세 개 시스템 모두에서 일치하는 비밀번호를 갖습니다.

2. 큐 관리자 인스턴스를 호스트할 IBM i 시스템에 관계형 데이터베이스 항목(RDBE)을 추가하십시오.
 - a) ALPHA에서 BETA에 대한 연결을 작성하십시오.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) BETA에서 ALPHA에 대한 연결을 작성하십시오.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. 큐 관리자 오브젝트를 재작성하는 스크립트를 작성하십시오.

```
QSAVEQMGR LCLQMGRNAM(QM1) FILENAME('*CURLIB/QMQSC(QM1)')  
OUTPUT(*REPLACE) MAKEAUTH(*YES) AUTHFN('*CURLIB/QMAUT(QM1)')
```

4. 큐 관리자를 중지하고 삭제하십시오.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMING(*YES) TIMEOUT(15)  
DLTMQM MQMNAME(QM1)
```

5. GAMMA에 큐 관리자 데이터를 저장하여 ALPHA에서 큐 관리자 QM1을 작성하십시오.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP('/QNTC/GAMMA/WMQ')
```

경로는 NetServer 를 사용하여 큐 관리자 데이터를 작성합니다.

6. 저장된 정의에서 QM1의 큐 관리자 오브젝트를 재작성하십시오.

```
STRMQMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. 저장된 정보에서 권한을 적용하십시오.

- a) 저장된 권한 부여 프로그램 컴파일하십시오.

```
CRTCLPGM PGM(*CURLIB/QM1) SRCFILE(*CURLIB/QMAUT)  
SRCMBR(QM1) REPLACE(*YES)
```

- b) 프로그램을 실행하여 권한을 적용하십시오.

```
CALL PGM(*CURLIB/QM1)
```

- c) QM1에 대한 보안 정보를 새로 고치십시오.

```
RFRMQMAUT MQMNAME(QM1)
```

8. ALPHA에서 실행하십시오. 이 명령은 BETA에서 원격 저널을 추가합니다.

```
ADDMQMJRN MQMNAME(QM1) RMTJNRDB(BETA)
```


활성 인스턴스가 ALPHA에 있을 때 ALPHA의 로컬 저널에 저널 항목을 작성합니다. ALPHA의 로컬 저널은 BETA의 원격 저널에 복제됩니다.

9. ALPHA에서 작성된 IBM MQ 구성 데이터를 검사하려면, 명령을 사용하십시오.

이 정보는 다음 단계에서 필요합니다.

이 예에서 다음 구성은 ALPHA에서 작성됩니다.

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. 다음 명령을 사용하여 BETA에서 QM1의 큐 관리자 인스턴스를 작성하십시오. BETA에서 다음 명령을 실행하여 BETA에서 큐 관리자 제어 정보를 수정하십시오.

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

팁: 구성 정보를 복사하여 붙여넣으십시오. 큐 관리자 스탠자는 ALPHA와 BETA에서 동일합니다.

11. BETA에서 실행하십시오. 이 명령은 BETA의 로컬 저널과 ALPHA의 리모트 저널을 추가합니다.

```
ADDQMJRN MQMNAME(QM1) RMTJRN RDB(ALPHA)
```

활성 인스턴스가 BETA에 있는 경우 BETA의 로컬 저널에 저널 항목을 작성합니다. BETA의 로컬 저널은 ALPHA의 원격 저널에 복제됩니다.

참고: 또는 비동기 저널링을 사용하여 BETA에서 ALPHA로의 원격 저널링을 설정할 수 있습니다.

단계 393 페이지의 『7』의 명령 대신 이 명령을 사용하여 BETA에서 ALPHA로의 비동기 저널링을 설정하십시오.

```
ADDQMJRN MQMNAME(QM1) RMTJRN RDB(ALPHA) RMTJRN DLV(*ASYNC)
```

ALPHA의 서버 또는 저널링이 실패의 소스인 경우에는 새 저널 항목이 ALPHA에 복제될 때까지 대기하지 않고 BETA가 시작됩니다.

ALPHA가 다시 온라인 상태가 되면 명령을 사용하여 복제 모드를 *SYNC로 전환하십시오.

388 페이지의 『IBM i에서 ASP에 대한 미러링된 저널 구성』의 정보를 사용하여 저널을 동기, 비동기 또는 두 가지 방식으로 모두 미러링되는지 여부를 결정하십시오. 기본값은 원격 저널로부터의 응답을 60초 동안 대기하고 동기식으로 복제하는 것입니다.

12. ALPHA 및 BETA의 저널이 사용으로 설정되어 있고 원격 저널 복제의 상태가 사용으로 설정되어 있는지 확인하십시오.

a) ALPHA:

```
WRKMQM JRN MQMNAME(QM1)
```

b) BETA:

```
WRKMQM JRN MQMNAME(QM1)
```

13. ALPHA와 BETA에서 큐 관리자 인스턴스를 시작하십시오.

a) ALPHA에서 첫 번째 인스턴스를 시작하고 활성화하십시오. 대기 인스턴스로의 전환을 사용하십시오.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) BETA에서 두 번째 인스턴스를 시작하고 활성화하십시오.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

결과

큐 관리자 상태를 확인하는 데 사용합니다.

1. ALPHA에서 큐 관리자 인스턴스의 상태는 다음과 같아야 합니다.
2. BETA의 큐 관리자 인스턴스 상태는 다음과 같아야 합니다.

예

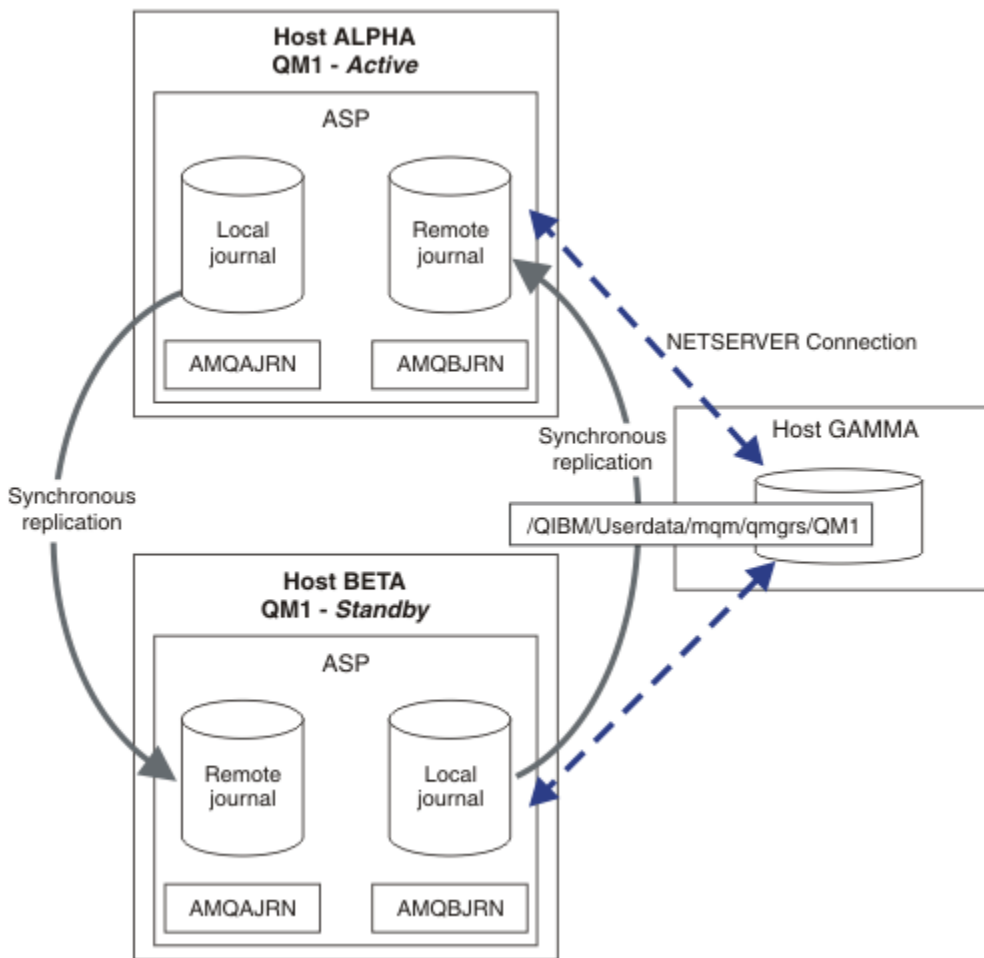


그림 26. 미러링된 저널 구성

다음에 수행할 작업

- 활성화 및 대기 인스턴스가 자동으로 전환되는지 확인하십시오. 샘플 고가용성 샘플 프로그램을 실행하여 전환을 테스트할 수 있습니다([고가용성 샘플 프로그램 참조](#)). 샘플 프로그램은 'C' 클라이언트입니다. 이 프로그램은 Windows 또는 Unix 플랫폼에서 실행할 수 있습니다.

1. 고가용성 샘플 프로그램을 시작하십시오.
2. ALPHA에서 전환을 요청하는 큐 관리자를 종료하십시오.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. 베타의 인스턴스가 활성 상태인지 확인하십시오.
4. ALPHA에서 재시작

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- 대체 고가용성 구성을 확인하십시오.

1. NetServer를 사용하여 Windows Server에 큐 관리자 데이터를 배치하십시오.
2. 원격 저널링을 사용하여 큐 관리자 저널을 미러링하는 대신 독립 ASP에 저널을 저장하십시오. 독립 ASP를 ALPHA에서 BETA로 전송하려면 IBM i 클러스터링을 사용하십시오.

IBM i IBM i에서 전환된 독립 ASP 저널 구성

다중 인스턴스 큐 관리자 구성을 작성하기 위해 독립 ASP 저널을 복제하지 않아도 됩니다. 활성 큐 관리자에서 대기 큐 관리자로 독립 ASP를 전송하기 위한 수단을 자동화해야 합니다. 독립 ASP를 사용하여 가능한 대체 고가용성 솔루션이 있으며 일부 솔루션은 다중 인스턴스 큐 관리자를 사용하지 않아도 됩니다.

독립 ASP를 사용하는 경우 큐 관리자 저널을 미러링하지 않아도 됩니다. 클러스터 관리를 설치하고 큐 관리자 인스턴스를 호스트하는 서버가 동일한 클러스터 자원 그룹에 있는 경우 활성 인스턴스를 실행하는 호스트가 실패하면 큐 관리자 저널이 근거리에서 있는 활성 서버 내 다른 서버에 자동으로 전송될 수 있습니다. 계획된 전환의 일부로 저널을 수동으로 전송하거나 독립 ASP를 프로그래밍 방식으로 전송하는 명령 프로시저를 작성할 수 있습니다.

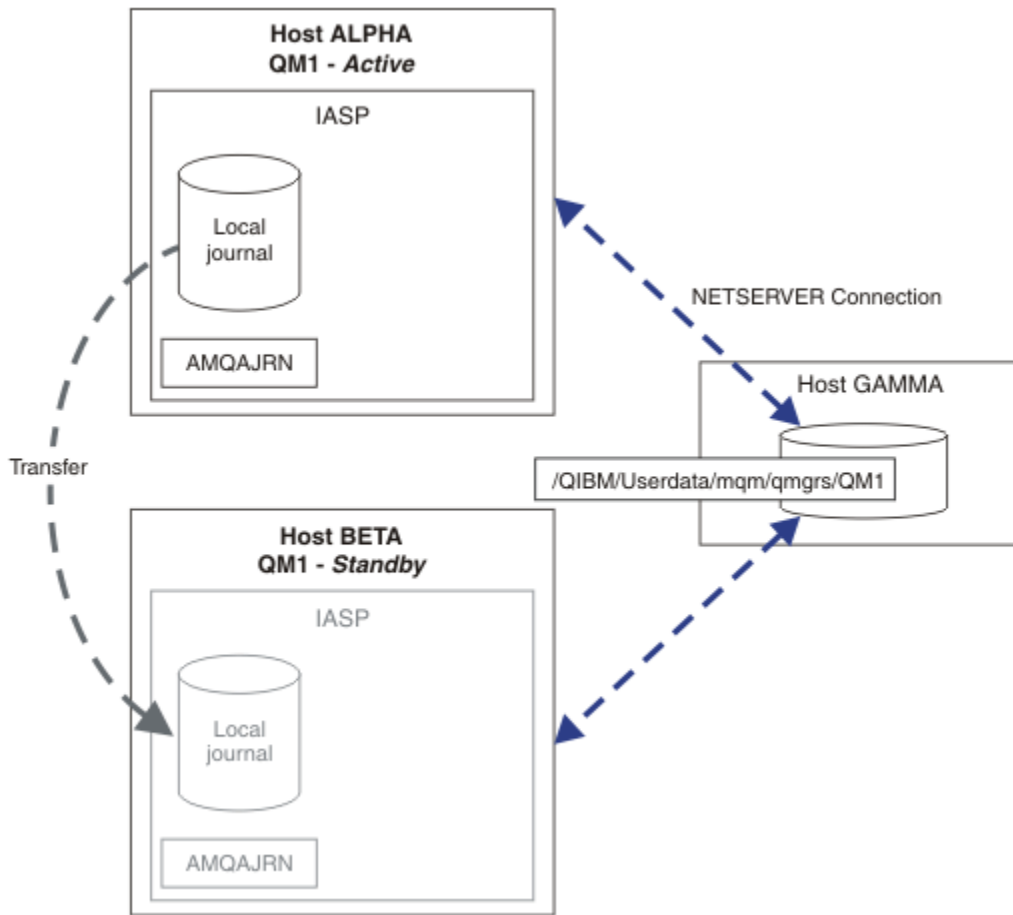


그림 27. 독립 ASP를 사용하여 큐 관리자 저널 전송

다중 인스턴스 큐 관리자 조작의 경우, 큐 관리자 데이터를 공유 파일 시스템에 저장해야 합니다. 파일 시스템은 다양한 플랫폼에서 호스트할 수 있습니다. 다중 인스턴스 큐 관리자 데이터는 ASP 또는 독립 ASP에 저장할 수 없습니다.

공유 파일 시스템은 구성에서 두 가지 역할을 수행합니다. 큐 관리자의 모든 인스턴스 간에 동일한 큐 관리자 데이터를 공유합니다. 파일 시스템에는 시작 시 큐 관리자의 인스턴스가 하나만 큐 관리자 데이터에 액세스할 수 있도록 하는 강력한 잠금 프로토콜이 필요합니다. 큐 관리자가 실패하거나 파일 서버에 대한 통신이 끊어지면 파일 시스템이 더 이상 파일 시스템과 통신하지 않는 활성 인스턴스가 보유한 큐 관리자 데이터에 대한 잠금을 해제해야 합니다. 그러면 대기 큐 관리자 인스턴스가 큐 관리자 데이터에 대한 읽기/쓰기 액세스 권한을 얻을 수 있습니다. 파일 시스템 프로토콜은 다중 인스턴스 큐 관리자와 올바르게 작업을 수행하기 위한 규칙 세트를 준수해야 합니다(380 페이지의 『고가용성 솔루션의 컴포넌트(IBM i)』 참조).

잠금 메커니즘은 큐 관리자 시작 명령을 직렬화하고 활성 상태인 큐 관리자 인스턴스를 제어합니다. 큐 관리자가 활성화되면 사용자 또는 HA 클러스터가 대기 서버로 전송한 로컬 저널로부터 해당 큐를 재빌드합니다. 동일한 큐 관리자로의 재연결을 대기하는 다시 연결 가능한 클라이언트가 다시 연결되고 인플라이트 트랜잭션은 백아웃됩니다. 큐 관리자 서비스로 시작되도록 구성된 애플리케이션이 시작됩니다.

클러스터 자원 관리자를 구성하거나 독립 ASP를 수동으로 전송하여, 독립 ASP의 실패한 활성 큐 관리자 인스턴스로부터의 로컬 저널이 새로 활성화된 대기 큐 관리자 인스턴스를 호스트하는 서버로 전송되는지 확인해야 합니다. 백업 및 재해 복구를 위해 독립 ASP를 사용하려는 경우 독립 ASP를 사용하더라도 원격 저널을 구성하고 미러링하며 다중 인스턴스 큐 관리자 구성을 위해 원격 저널 미러링을 사용합니다.

독립 ASP를 사용하도록 선택한 경우에는 다른 고가용성 구성을 고려할 수 있습니다. 이 솔루션의 배경은 405 페이지의 『독립 ASP 및 고가용성』에서 설명됩니다.

1. 다중 인스턴스 큐 관리자를 사용하는 대신, 독립 ASP에서 단일 인스턴스 큐 관리자를 설치 및 구성하고 IBM i 고가용성 서비스를 사용하여 큐 관리자 장애를 복구하십시오. 큐 관리자가 서버와 관계 없이 실패했는지 여부

를 감지하기 위해 큐 관리자 모니터로 솔루션을 강화해야 할 수 있습니다. 이는 *Supportpac MC41: 고가용성을 위한 IBM MQ for iSeries* 구성에 제공된 솔루션의 기반이 됩니다.

2. 로컬 버스의 독립 ASP를 전환하는 대신 독립 ASP와 교차 사이트 미러링(XSM)을 사용하여 독립 ASP를 미러링하십시오. 이는 장거리 로그 레코드 쓰기에 소요되는 시간이 허용하는 한 독립 ASP 솔루션의 지리적 범위를 확장합니다.

IBM i IBM i에서 독립 ASP 및 NetServer를 사용하여 다중 인스턴스 큐 관리자 작성
두 개의 IBM i 서버에서 실행될 다중 인스턴스 큐 관리자를 작성합니다. 큐 관리자 데이터는 NetServer를 사용하는 IBM i 서버에 저장됩니다. 큐 관리자 저널은 독립 ASP에 저장됩니다. 큐 관리자 저널을 포함하는 독립 ASP를 다른 IBM i 서버로 전송하려면 IBM i 클러스터링 또는 수동 프로시저를 사용하십시오.

시작하기 전에

1. 이 태스크에는 IBM i 서버 세 개가 필요합니다. 이 중 두 개 서버(예에서 ALPHA 및 BETA)에 IBM MQ를 설치하십시오. 제품은 IBM WebSphere MQ 7.0.1 Fix Pack 1 이상이어야 합니다.
2. 세 번째 서버는 IBM i 서버로, NetServer가 ALPHA와 BETA에 연결합니다. 이 서버는 큐 관리자 데이터를 공유하는 데 사용됩니다. IBM MQ 설치할 필요가 없습니다. 큐 관리자 디렉토리 권한을 설정하기 위해 임시 조치로 서버에 IBM MQ를 설치하는 것이 좋습니다.
3. QMQM 사용자 프로필의 비밀번호가 세 서버에서 모두 동일한지 확인하십시오.
4. IBM i NetServer를 설치하십시오(i5/OS NetServer 참조).
5. 실패한 큐 관리자에서 인계받는 대기 큐 관리자도 독립 ASP를 전송하는 프로시저를 작성하십시오. *SupportPac MC41: 고가용성을 위한 IBM MQ for iSeries* 구성에서 독립 ASP 전송 프로시저 설계에 유용한 일부 기법을 참조할 수 있습니다.

이 태스크 정보

404 페이지의 그림 28에 표시된 구성을 작성하려면 다음 단계를 수행하십시오. 큐 관리자 데이터는 IBM i NetServer를 사용하여 연결됩니다.

- ALPHA 및 BETA에서 큐 관리자 데이터를 저장할 GAMMA의 디렉토리 공유로의 연결을 작성하십시오. 이 태스크는 또한 필요한 권한, 사용자 프로파일, 비밀번호를 설정합니다.
- IBM i 서버 ALPHA에서 큐 관리자 QM1 를 작성하십시오.
- 다른 IBM i 서버 BETA에서 QM1 에 대한 큐 관리자 제어 정보를 추가하십시오.
- 대기 인스턴스를 허용하여 큐 관리자를 시작하십시오.

프로시저

1. 381 페이지의 『IBM i에서 NetServer를 사용하여 큐 관리자 데이터의 네트워크 공유 작성』 태스크를 수행하십시오.

결과적으로 ALPHA 및 BETA에는 GAMMA의 /QIBM/UserData/mqm/qmgrs를 가리키는 공유 /QNTC/GAMMA/WMQ가 있습니다. 사용자 프로파일 QMQM 및 QMQMADM은 필요한 권한을 갖고 QMQM은 세 개 시스템 모두에서 일치하는 비밀번호를 갖습니다.

2. GAMMA에 큐 관리자 데이터를 저장하여 ALPHA에서 큐 관리자 QM1을 작성하십시오.

```
CRTMQM MQMNAME(QM1) UDLSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

경로는 NetServer 를 사용하여 큐 관리자 데이터를 작성합니다.

3. ALPHA에서 작성된 IBM MQ 구성 데이터를 검사하려면, 명령을 사용하십시오.

이 정보는 다음 단계에서 필요합니다.

이 예에서 다음 구성은 ALPHA에서 작성됩니다.

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. 다음 명령을 사용하여 BETA에서 QM1의 큐 관리자 인스턴스를 작성하십시오. BETA에서 다음 명령을 실행하여 BETA에서 큐 관리자 제어 정보를 수정하십시오.

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

팁: 구성 정보를 복사하여 붙여넣으십시오. 큐 관리자 스탠자는 ALPHA와 BETA에서 동일합니다.

5. ALPHA와 BETA에서 큐 관리자 인스턴스를 시작하십시오.

- a) ALPHA에서 첫 번째 인스턴스를 시작하고 활성화하십시오. 대기 인스턴스로의 전환을 사용하십시오.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) BETA에서 두 번째 인스턴스를 시작하고 활성화하십시오.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

결과

큐 관리자 상태를 확인하는 데 사용합니다.

1. ALPHA에서 큐 관리자 인스턴스의 상태는 다음과 같아야 합니다.
2. BETA의 큐 관리자 인스턴스 상태는 다음과 같아야 합니다.

예

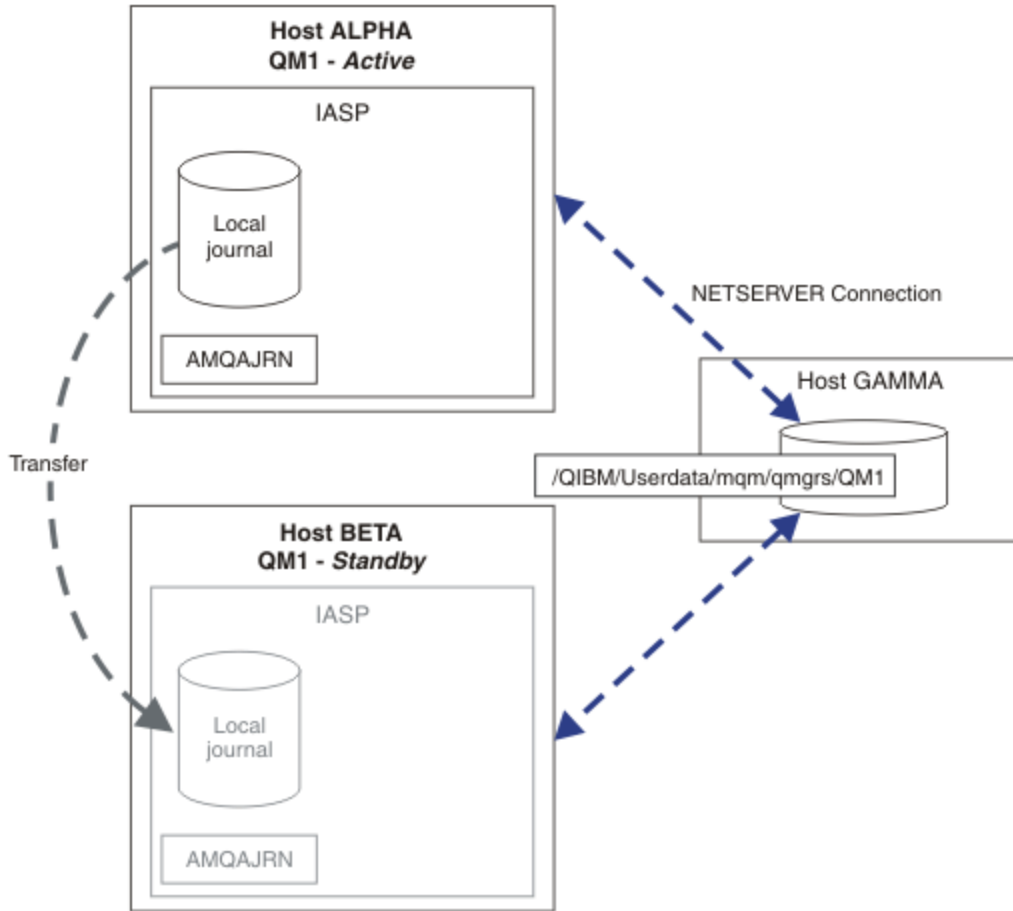


그림 28. 독립 ASP를 사용하여 큐 관리자 저널 전송

다음에 수행할 작업

- 활성 및 대기 인스턴스가 자동으로 전환되는지 확인하십시오. 샘플 고가용성 샘플 프로그램을 실행하여 전환을 테스트할 수 있습니다(고가용성 샘플 프로그램 참조). 샘플 프로그램은 'C' 클라이언트입니다. 이 프로그램은 Windows 또는 Unix 플랫폼에서 실행할 수 있습니다.

1. 고가용성 샘플 프로그램을 시작하십시오.
2. ALPHA에서 전환을 요청하는 큐 관리자를 종료하십시오.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. 베타의 인스턴스가 활성 상태인지 확인하십시오.
4. ALPHA에서 재시작

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- 대체 고가용성 구성을 확인하십시오.
 1. NetServer를 사용하여 IBM i 서버에 큐 관리자 데이터를 배치하십시오.
 2. 독립 ASP를 사용하여 대기 서버로 큐 관리자 저널을 전송하는 대신 원격 저널링을 사용하여 대기 서버에 저널을 미러링하십시오.

IBM i 독립 ASP 및 고가용성

독립 ASP를 사용하여 서버 간에 애플리케이션과 데이터를 이동할 수 있습니다. 독립 ASP의 유연성은 ASP가 일부 IBM i 고가용성 솔루션의 기반임을 의미합니다. 큐 관리자 저널에 ASP 또는 독립 ASP를 사용할지 여부를 고려할 때는 독립 ASP를 기반으로 하는 다른 고가용성 구성을 고려해야 합니다.

보조 기억장치 풀(ASP)은 IBM i 아키텍처의 빌딩 블록입니다. 디스크 장치가 함께 그룹화되어 단일 ASP를 구성합니다. 오브젝트를 여러 ASP에 배치함으로써 다른 ASP의 디스크 장애로 특정 ASP의 데이터가 영향을 받지 않도록 보호할 수 있습니다.

모든 IBM i 서버는 하나 이상의 기본 ASP(시스템 ASP라고 함)를 갖습니다. 이 ASP는 ASP1로 지정되며 *SYSBAS라고도 합니다. 애플리케이션 관점에서 시스템 ASP와 구분할 수 없는 최대 31개의 추가 기본 사용자 ASP를 구성할 수 있습니다. ASP는 동일한 네임스페이스를 공유하기 때문입니다. 여러 기본 ASP를 사용하여 많은 디스크에 애플리케이션을 분산시킴으로써 성능을 향상시키고 복구 시간을 줄일 수 있습니다. 여러 기본 ASP를 사용하는 경우 또한 디스크 장애로부터 어느 정도 격리될 수 있지만 전반적인 신뢰성은 향상되지 않습니다.

독립 ASP는 특수 유형의 ASP입니다. 이 ASP는 독립 디스크 풀이라고도 합니다. 독립 디스크 풀은 IBM i 고가용성의 핵심 구성요소입니다. 스스로를 독립 디스크 스토리지 장치에서 연결되는 현재 시스템에 독립적인 것으로 간주하는 애플리케이션과 데이터를 저장할 수 있습니다. 전환 가능 또는 전환 불가능 독립 ASP를 구성할 수 있습니다. 가용성 관점에서는 일반적으로 서버에서 서버로 자동 전송될 수 있는 전환 가능 독립 ASP만 고려합니다. 결과적으로, 독립 ASP의 애플리케이션과 데이터를 서버에서 서버로 이동할 수 있습니다.

기본 사용자 ASP와 달리 독립 ASP는 시스템 ASP와 동일한 네임스페이스를 공유하지 않습니다. 사용자 APS에 대한 작업을 수행하는 애플리케이션에는 독립 ASP에 대한 작업을 수행하기 위해 변경사항이 필요하지 않습니다. 소프트웨어와 사용하는 써드파티 소프트웨어가 독립 ASP 환경에서 작동하는지 확인해야 합니다.

독립 ASP가 다른 서버에 연결될 때 독립 ASP의 네임스페이스가 시스템 ASP의 네임스페이스와 결합되어야 합니다. 이 프로세스를 독립 ASP 변환이라고 합니다. 서버 IPL을 수행하지 않고 독립 ASP를 변환할 수 있습니다. 특정 서버에서 다른 서버로 독립 ASP를 자동으로 전송하려면 클러스터링 지원이 필요합니다.

독립 ASP로 신뢰할 수 있는 솔루션 빌드

ASP에 저널링한 후 저널 복제를 사용하지 않고 독립 ASP에 저널링하는 경우 실패한 큐 관리자 인스턴스에서 로컬 저널의 사본을 대기 큐 관리자에 제공할 수 있는 대체 방법을 제공합니다. 독립 ASP를 다른 서버에 자동으로 전송하려면 클러스터링 지원을 설치 및 구성한 상태여야 합니다. 다중 인스턴스 큐 관리자를 사용하여 결합 또는 대체할 수 있는 클러스터 지원 기반의 독립 ASP와 하위 레벨 디스크 미러링에 대한 여러 가지 고가용성 솔루션이 존재합니다.

다음 목록은 독립 ASP 기반의 신뢰할 수 있는 솔루션을 빌드하는 데 필요한 구성요소를 설명합니다.

저널링

큐 관리자와 다른 애플리케이션이 로컬 저널을 사용하여 지속 데이터를 디스크에 안전하게 기록함으로써 서버 장애로 인한 메모리에 데이터 손실을 방지합니다. 이를 특정 시점 일관성이라고도 합니다. 이는 일정 기간 수행되는 여러 업데이트의 일관성을 보장하지 않습니다.

커밋 제어

글로벌 트랜잭션을 사용함으로써 저널에 기록된 데이터가 일관성을 유지하도록 메시지 및 데이터베이스에 대한 업데이트를 조정할 수 있습니다. 2단계 커밋 프로토콜을 사용하여 일정 시간 동안 일관성을 유지할 수 있습니다.

전환된 디스크

전환된 디스크는 HA 클러스터의 디바이스 클러스터 자원 그룹(CRG)이 관리합니다. 계획되지 않은 가동 중단 시 CRG가 독립 ASP를 새 서버로 자동 전환합니다. CRG는 지리적으로 로컬 IO 버스 범위로 제한됩니다.

전환 가능 독립 ASP에 로컬 저널을 구성하여 다른 서버로 저널을 전송하고 메시지 처리를 재개할 수 있습니다. 독립 ASP가 실패하지 않으면 동기점 제어 없이 작성되거나 동기점 제어와 함께 커밋된 지속 메시지에 대한 변경사항이 손실되지 않습니다.

전환 가능 독립 ASP에 저널링과 커밋 제어를 둘 다 사용하는 경우, 데이터베이스 저널과 큐 관리자 저널을 다른 서버로 전송할 수 있으며 일관성 또는 커밋된 트랜잭션 손실 없이 트랜잭션 처리를 재개할 수 있습니다.

사이트 간 미러링(XSM)

XSM은 1차 독립 ASP를 TCP/IP 네트워크에서 지리적으로 멀리 떨어져 있는 2차 독립 ASP로 미러링하고 실패 시 제어를 자동으로 전송합니다. 동기 또는 비동기 미러를 구성하도록 선택할 수 있습니다. 동기 미러링은 프로덕션 시스템에 대한 쓰기 조작성이 완료되기 전에 데이터가 미러링되므로 큐 관리자의 성능을 저하시키지 만 2차 독립 ASP가 최신 상태로 유지되도록 보장합니다. 반면에 비동기 미러링을 사용하는 경우에는 2차 독립 ASP가 최신 상태로 유지되도록 보장할 수 없습니다. 비동기 미러링은 2차 독립 ASP의 일관성을 유지합니다.

XSM 기술은 세 가지가 있습니다.

지리적 미러링

지리적 미러링은 클러스터링의 확장으로, 넓은 영역에서 독립 ASP를 전환할 수 있습니다. 동기 모드와 비동기 모드가 둘 다 있습니다. 고가용성은 동기 모드에서만 보장할 수 있지만 독립 ASP를 분리해도 성능에 큰 영향을 주지 않습니다. 지리적 미러링과 전환된 디스크를 결합하여 로컬로는 고가용성을 제공하고 원격으로는 재해 복구를 제공할 수 있습니다.

메트로 미러링

메트로 미러링은 로컬 버스보다 먼 거리에서 빠른 로컬 동기 미러링을 제공하는 디바이스 레벨 서비스입니다. 이 서비스와 다중 인스턴스 큐 관리자를 결합하여 큐 관리자의 고가용성을 제공할 수 있으며 독립 ASP의 두 개 사본을 작성함으로써 큐 관리자 저널의 고가용성을 제공할 수 있습니다.

글로벌 미러링

글로벌 미러링은 비동기 미러링을 제공하는 디바이스 레벨 서비스이므로 장거리 백업 및 재해 복구에 적합하지만 고가용성을 위한 일반적인 선택사항은 아닙니다. 동시성이 아닌 특정 시점 일관성만 유지하기 때문입니다.

고려해야 하는 핵심 의사결정 사항은 다음과 같습니다.

ASP 또는 독립 ASP 여부

다중 인스턴스 큐 관리자를 사용하기 위해 IBM i HA 클러스터를 실행하지 않아도 됩니다. 이미 독립 ASP를 사용하고 있거나 독립 ASP가 필요한 다른 애플리케이션에 대한 가용성 요구사항이 있는 경우 독립 ASP를 선택할 수 있습니다. 큐 관리자 모니터링을 큐 관리자 실패 감지 수단으로 바꾸기 위해 독립 ASP와 다중 인스턴스 큐 관리자를 결합할 수도 있습니다.

가용성

복구 시간 목표(RTO)란 무엇입니까? 거의 중단되지 않는 동작이 나타나야 하는 경우 복구 시간이 가장 빠른 솔루션은 무엇입니까?

저널 가용성

저널을 단일 장애 지점으로서 제거하기 위한 방법. RAID 1 이상 디바이스를 사용하여 하드웨어 솔루션을 채택하거나, 복제 저널 또는 디스크 미러링을 사용하는 소프트웨어 솔루션을 사용할 수 있습니다.

거리

활성 큐 관리자 인스턴스와 대기 큐 관리자 인스턴스 사이의 거리. 사용자가 약 250미터 이상의 거리에서 동기식 복제에 따른 성능 저하를 허용할 수 있습니까?

스킬

솔루션 유지보수 및 실행과 관련된 관리 태스크를 정기적으로 자동화하기 위해 수행해야 할 작업이 있습니다. 자동화 수행을 위해 필요한 스킬은 ASP를 기반으로 하는 솔루션과 독립 ASP를 기반으로 하는 솔루션에 각각 다릅니다.

IBM i IBM i에서 다중 인스턴스 큐 관리자 삭제

다중 인스턴스 큐 관리자를 삭제하기 전에 원격 저널링을 중지하고 큐 관리자 인스턴스를 제거하십시오.

시작하기 전에

1. 이 예에서는 QM1 큐 관리자의 두 개 인스턴스가 ALPHA 및 BETA 서버에서 정의됩니다. ALPHA는 활성 인스턴스이고 BETA는 대기입니다. 큐 관리자 QM1과 연관된 큐 관리자 데이터는 NetServer를 사용하여 IBM i 서버 GAMMA에 저장됩니다. 392 페이지의 『IBM i에서 저널 미러링 및 NetServer를 사용하여 다중 인스턴스 큐 관리자 작성』의 내용을 참조하십시오.
2. 정의되는 원격 저널을 IBM MQ가 삭제할 수 있도록 ALPHA와 BETA를 연결해야 합니다.
3. 시스템 명령 **EDTF** 또는 **WRKLNK**를 사용하여 /QNTC 디렉토리 및 서버 디렉토리 파일 공유에 액세스할 수 있는지 확인하십시오.

이 태스크 정보

DLTMQM 명령을 사용하여 서버에서 다중 인스턴스 큐 관리자를 삭제하려면 **RMVMQMINF** 명령을 사용하여 다른 서버의 큐 관리자 인스턴스를 제거하십시오.

RMVMQMINF 명령을 사용하여 큐 관리자 인스턴스를 제거하면 AMQ가 앞에 오고 인스턴스와 연관된 로컬 및 원격 저널이 삭제됩니다. 서버에 로컬인 큐 관리자 인스턴스에 대한 구성 정보도 삭제됩니다.

큐 관리자의 나머지 인스턴스가 있는 서버에서는 **RMVMQMINF** 명령을 실행하지 마십시오. 명령을 실행하면 **DLTMQM**이 올바르게 작동하지 않습니다.

DLTMQM 명령을 사용하여 큐 관리자를 삭제하십시오. 네트워크 공유에서 큐 관리자 데이터가 제거됩니다. 앞에 AMQ가 표시되고 인스턴스와 연관된 로컬 및 원격 저널은 삭제됩니다. **DLTMQM**은 또한 서버에 로컬인 큐 관리자 인스턴스에 대한 구성 정보를 삭제합니다.

예에는 큐 관리자 인스턴스가 두 가지만 있습니다. IBM MQ는 하나의 활성 큐 관리자 인스턴스와 하나의 대기 인스턴스가 있는 실행 다중 인스턴스 구성을 지원합니다. 실행 구성에서 사용할 추가 큐 관리자 인스턴스를 작성한 경우, 나머지 인스턴스를 삭제하기 전에 **RMVMQMINF** 명령을 사용하여 제거하십시오.

프로시저

1. 각 서버에서 **CHGMQMJRN RMTJRNSTS (*INACTIVE)** 명령을 실행하여 큐 관리자 인스턴스 간의 원격 저널링을 비활성화하십시오.

a) ALPHA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('BETA') RMTJRNSTS(*INACTIVE)
```

b) BETA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. 활성 큐 관리자 인스턴스인 ALPHA에서 **ENDMQM** 명령을 실행하여 QM1 인스턴스를 둘 다 중지하십시오.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

3. ALPHA에서 **RMVMQMINF** 명령을 실행하여 ALPHA 및 BETA에서 인스턴스에 대한 큐 관리자 자원을 제거하십시오.

```
RMVMQMINF MQMNAME(QM1)
```

RMVMQMINF는 QM1에 대한 큐 관리자 구성 정보를 ALPHA에서 제거합니다. 저널 이름 앞에 AMQ가 있으면 QM1과 연관된 로컬 저널을 ALPHA에서 삭제합니다. 저널 이름 앞에 AMQ가 오고 원격 저널이 작성된 경우에는 BETA에서도 원격 저널을 제거합니다.

4. BETA에서 **DLTMQM** 명령을 실행하여 QM1을 삭제하십시오.

```
DLTMQM MQMNAME(QM1)
```

DLTMQM은 GAMMA의 네트워크 공유에서 큐 관리자 데이터를 삭제합니다. BETA에서 QM1에 대한 큐 관리자 구성 정보를 제거합니다. 저널 이름 앞에 AMQ가 있고 QM1과 연관된 로컬 저널을 BETA에서 삭제합니다. 저널 이름 앞에 AMQ가 오고 원격 저널이 작성된 경우에는 ALPHA에서도 원격 저널을 제거합니다.

결과

DLTMQM과 **RMVMQMINF**는 **CRTMQM**과 **ADDMQJRN**으로 작성된 로컬 및 원격 저널을 삭제합니다. 이 명령은 또한 저널 수신자를 삭제합니다. 저널과 저널 수신자는 이름이 AMQ로 시작하는 이름 지정 규칙을 따라야 합니다.

DLTMQM과 **RMVMQMINF**는 `mqs.ini`에서 큐 관리자 오브젝트, 큐 관리자 데이터, 큐 관리자 구성 정보를 제거합니다.

다음에 수행할 작업

또 다른 방법은 407 페이지의 『1』 단계에서 저널링을 비활성화한 후 및 큐 관리자 인스턴스를 종료하기 전에 다음 명령을 실행하는 것입니다. 또는 이름 지정 규칙을 따르지 않은 경우 이름별로 저널과 저널 수신자를 삭제해야 합니다.

1. ALPHA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. BETA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

저널을 삭제한 후 나머지 단계를 계속 진행하십시오.

IBM i IBM i에서 다중 인스턴스 큐 관리자 백업

이 프로시저는 로컬 서버의 큐 관리자 오브젝트와 네트워크 파일 서버의 큐 관리자 데이터를 백업하는 방법을 보여줍니다. 다른 큐 관리자의 데이터를 백업하려면 예를 수정하십시오.

시작하기 전에

이 예에서 큐 관리자 QM1과 연관된 큐 관리자 데이터는 NetServer를 사용하여 IBM i 서버 GAMMA에 저장됩니다. 392 페이지의 『IBM i에서 저널 미러링 및 NetServer를 사용하여 다중 인스턴스 큐 관리자 작성』의 내용을 참조하십시오. IBM MQ는 서버, ALPHA 및 BETA에 설치됩니다. 큐 관리자, QM1은 ALPHA와 BETA에 구성됩니다.

이 태스크 정보

IBM i는 원격 디렉토리로부터의 데이터 저장을 지원하지 않습니다. 파일 시스템 서버의 로컬 백업 프로시저를 사용하여 원격 파일 시스템에 큐 관리자 데이터를 저장하십시오. 이 태스크에서 네트워크 파일 시스템은 IBM i 서버, GAMMA에 있습니다. 큐 관리자 데이터는 GAMMA의 저장 파일에 백업됩니다.

네트워크 파일 시스템이 Windows 또는 Linux에 있는 경우에는 압축 파일에 큐 관리자 데이터를 저장한 다음 저장할 수 있습니다. Tivoli Storage Manager와 같은 백업 시스템이 있는 경우에는 해당 시스템을 사용하여 큐 관리자 데이터를 백업하십시오.

프로시저

1. QM1과 연관된 큐 관리자 라이브러리의 경우 ALPHA에 저장 파일을 작성하십시오.

큐 관리자 라이브러리 이름을 사용하여 저장 파일 이름을 지정하십시오.

```
CRTSAVF FILE(QGPL/QMQM1)
```

2. ALPHA의 저장 파일에 큐 관리자 라이브러리를 저장하십시오.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

3. GAMMA의 큐 관리자 데이터 디렉토리에 대한 저장 파일을 작성하십시오.

큐 관리자 이름을 사용하여 저장 파일 이름을 지정하십시오.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

4. GAMMA의 로컬 디렉토리에서 큐 관리자 데이터의 사본을 저장하십시오.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

IBM i 다중 인스턴스 큐 관리자를 설정하는 명령

IBM MQ에는 저널 복제 구성, 새 큐 관리자 인스턴스 추가, 큐 관리자가 독립 ASP를 사용하도록 구성 작업을 단 순화하는 명령이 있습니다.

로컬 및 원격 저널을 작성하고 관리하는 저널 명령은 다음과 같습니다.

ADDMQMJRN

이 명령을 사용하면 큐 관리자 인스턴스의 이름 지정된 로컬 및 원격 저널을 작성하고 복제가 동기 또는 비동기인지 여부, 동기 제한시간 초과한 것, 원격 저널이 즉시 활성화될 수 있는지 여부를 구성할 수 있습니다.

CHGMQMJRN

이 명령은 복제 저널에 영향을 주는 제한시간, 상태, 전달 매개변수를 수정합니다.

RMVMQMJRN

큐 관리자 인스턴스에서 이름 지정된 원격 저널을 제거합니다.

WRKMQMJRN

로컬 큐 관리자 인스턴스의 로컬 및 원격 저널 상태를 나열합니다.

mqs.ini 파일을 수정하는 다음 명령을 사용하여 추가 큐 관리자 인스턴스를 추가하고 관리하십시오.

ADDMQMINF

이 명령은 DSPMQMINF 명령으로 mqs.ini 파일에서 추출한 정보를 사용하여 다른 IBM i 서버에서 새 큐 관리자 인스턴스를 추가합니다.

RMVMQMINF

큐 관리자 인스턴스를 제거합니다. 기존 큐 관리자의 인스턴스를 제거하거나 다른 서버에서 삭제된 큐 관리자에 대한 구성 정보를 제거하려면 이 명령을 사용하십시오.

CRTMQM 명령은 다중 인스턴스 큐 관리자 구성을 지원하기 위한 세 가지 매개변수를 갖습니다.

MQMDIRP (*DFT | *directory-prefix*)

네트워크 스토리지의 큐 관리자 데이터에 맵핑되는 마운트 포인트를 선택하려면 이 매개변수를 사용하십시오.

ASP (*SYSTEM | *ASPDEV | *auxiliary-storage-pool-number*)

시스템 또는 기본 사용자 ASP에 큐 관리자 저널을 배치하려면 *SYSTEM 또는 *auxiliary-storage-pool-number*를 지정하십시오. 독립 ASP에 큐 관리자 저널을 배치하려면 *ASPDEV 옵션을 선택하고 또한 **ASPDEV** 매개변수를 사용하여 디바이스 이름을 설정하십시오.

ASPDEV (*ASP | *device-name*)

1차 또는 2차 독립 ASP 디바이스의 *device-name*를 지정하십시오. *ASP를 선택한 결과는 **ASP (*SYSTEM)**을 지정하는 경우와 같습니다.

IBM i IBM i에 대한 성능 및 디스크 장애 복구 고려사항

다양한 보조 기억장치 풀을 사용하여 성능과 신뢰성을 향상시킬 수 있습니다.

애플리케이션에서 많은 지속 메시지 또는 대형 메시지를 사용하는 경우 이러한 메시지를 디스크에 쓰는 데 소요되는 시간이 시스템 성능을 결정하는 중대 요인이 됩니다.

이러한 가능성에 대응할 수 있을 정도로 디스크가 충분히 활성화되었는지 확인하거나 큐 관리자 저널 수신자를 저장하는 별도 보조 기억장치 풀(ASP)을 고려하십시오.

CRTMQM의 ASP 매개변수를 사용하여 큐 관리자를 작성할 때 큐 관리자 라이브러리와 저널이 저장되는 ASP를 지정할 수 있습니다. 기본적으로 큐 관리자 라이브러리와 저널 및 IFS 데이터는 시스템 ASP에 저장됩니다.

ASP는 하나 이상의 특정 디스크 장치에서 오브젝트 격리를 허용합니다. 따라서 디스크 매체 고장으로 인한 데이터 손실을 줄일 수도 있습니다. 대부분의 경우, 영향을 받는 ASP의 디스크 장치에 저장된 데이터만 손실됩니다.

장애 복구를 제공하고 디스크 경합을 줄이려면 별도 사용자 ASP의 큐 관리자 라이브러리 및 저널 데이터를 루트 IFS 파일 시스템의 해당 위치에 저장하는 것이 좋습니다.

자세한 정보는 IBM i 문서의 [백업 및 복구](#) 를 참조하십시오.

IBM i SAVLIB를 사용하여 IBM i 에 IBM MQ 라이브러리 저장

SAVLIB LIB(*ALLUSR)을 사용하여 IBM MQ 라이브러리를 저장할 수 없습니다. 이 라이브러리 이름은 Q부터 시작하기 때문입니다.

SAVLIB LIB(QM*)을 사용하여 모든 큐 관리자 라이브러리를 저장하려면 *SAVF 이외의 저장 디바이스를 사용해야 합니다. DEV(*SAVF)의 경우, 시스템의 각각 모든 큐 관리자 라이브러리에 SAVLIB 명령을 사용해야 합니다.

IBM i IBM MQ for IBM i 일시정지

이 절에서는 IBM MQ for IBM i를 일시정지(우아한 종료)하는 방법을 설명합니다.

IBM MQ for IBM i를 일시정지하려면 다음을 수행하십시오.

1. 오브젝트에 액세스하지 않은 상태로 새 대화식 IBM MQ for IBM i 세션에 사인온하십시오.
2. 다음을 확인하십시오.
 - QMQM 라이브러리에 대한 오브젝트 관리 권한 또는 *ALLOBJ 권한
 - ENDSBS 명령을 사용할 수 있는 충분한 권한
3. 모든 사용자에게 IBM MQ for IBM i를 중지할 예정임을 알리십시오.
4. 다음 진행 단계는 단일 큐 관리자(다른 큐 관리자가 존재할 수 있는 경우)(410 페이지의 『IBM MQ for IBM i의 단일 큐 관리자 종료』 참조) 또는 모든 큐 관리자(412 페이지의 『IBM MQ for IBM i의 모든 큐 관리자 종료』 참조)를 종료(일시정지)하려는지에 따라 결정됩니다.
5. qshell에 다음 명령을 입력하여 mqweb 서버를 종료하십시오.

```
/QIBM/ProdData/mqm/bin/endmqweb
```

ENDMQM 매개변수 ENDCCTJOB(*YES)

ENDMQM 매개변수 ENDCCTJOB(*YES)는 이전 버전과 비교할 때 IBM MQ for IBM i V6.0 이상에서 다르게 작동합니다.

이전 버전의 경우 ENDCCTJOB(*YES)를 지정하면 MQ가 애플리케이션을 강제로 종료합니다.

IBM MQ for IBM i V6.0 이상의 경우 ENDCCTJOB(*YES)를 지정하면 애플리케이션은 종료되지 않지만 대신 큐 관리자와의 연결이 끊어집니다.

ENDCCTJOB(*YES)을 지정하고 큐 관리자가 종료 중임을 감지하기 위해 작성되지 않은 애플리케이션이 있는 경우, 다음에 새 MQI 호출이 발행되면 호출은 MQRC_CONNECTION_BROKEN(2009) 오류를 리턴합니다.

ENDCCTJOB(*YES) 사용에 대한 대안으로, ENDCCTJOB(*NO) 매개변수와 WRKMQM 옵션 22(작업 처리)를 사용하여 큐 관리자가 재시작되는 것을 방해하는 애플리케이션 작업을 수동으로 종료합니다.

IBM i IBM MQ for IBM i의 단일 큐 관리자 종료

이 정보를 통해 세 가지 종료 유형을 이해할 수 있습니다.

다음 프로시저에서는 샘플 큐 관리자 이름 QMgr1과 샘플 서브시스템 이름 SUBX를 사용합니다. 필요한 경우 이 이름을 원하는 값으로 바꿀 수 있습니다.

계획된 종료

IBM i에서의 계획된 큐 관리자 종료

1. 종료하기 전에 다음을 실행하십시오.

```
RCDCMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

2. 큐 관리자를 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

QMgr1이 종료되지 않으면 채널 또는 애플리케이션 사용량이 증가할 수 있습니다.

3. QMgr1을 즉시 종료해야 하는 경우에는 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

계획되지 않은 종료

1. 큐 관리자를 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

QMgr1이 종료되지 않으면 채널 또는 애플리케이션 사용량이 증가할 수 있습니다.

2. QMgr1을 즉시 종료해야 하는 경우에는 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

비정상적인 조건에서 종료

1. 큐 관리자를 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

다음과 같은 경우 QMgr1이 종료되지 않으면 단계 3을 계속 진행하십시오.

- QMgr1이 자체 서브시스템 내에 있습니다. 또는
 - QMgr1과 동일한 서브시스템을 공유하는 모든 큐 관리자를 종료할 수 있습니다. 그러한 모든 큐 관리자에 대해 계획되지 않은 종료 프로시저를 사용하십시오.
2. 서브시스템(예에서 SUBX)을 공유하는 모든 큐 관리자에 대해 프로시저의 모든 단계를 수행한 경우 다음을 실행하십시오.

```
ENDSBS SUBX *IMMED
```

이 명령이 완료되지 않으면 계획되지 않은 종료 프로시저를 사용하여 모든 큐 관리자를 종료하고 시스템에서 IPL을 수행하십시오.

경고: ENDJOB 또는 ENDSBS의 결과로 종료하는 데 실패하는 IBM MQ 작업에 대해 ENDJOBABN 를 사용하지 마십시오. 이후에 즉시 머신에서 IPL을 수행할 준비가 되지 않은 경우에는 사용하지 마십시오.

3. 다음을 실행하여 서브시스템을 시작하십시오.

```
STRSBS SUBX
```

4. 다음을 실행하여 큐 관리자를 즉시 종료하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(10)
```


5. 다음을 실행하여 큐 관리자를 재시작하십시오.

```
STRMQM MQMNAME(QMgr1)
```

실패하는 경우 원인은 다음과 같습니다.

- IPL을 수행하여 시스템을 다시 시작했습니다. 또는
- 단일 큐 관리자만 있습니다.

다음을 실행하여 IBM MQ 공유 메모리를 정리하십시오.

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

단계 5를 반복하기 전에

큐 관리자를 재시작하는 데 몇 초 이상이 소요되는 경우 IBM MQ가 시동 진행 상황을 설명하는 상태 메시지를 작업 로그에 간헐적으로 추가합니다.

큐 관리자 재시작 문제점이 계속 해결되지 않으면 IBM 지원 부서에 문의하십시오. 추가 조치를 수행하는 경우 큐 관리자가 손상되어 IBM MQ를 복구하지 못할 수 있습니다.

IBM i IBM MQ for IBM i의 모든 큐 관리자 종료

이 정보를 통해 세 가지 종료 유형을 이해할 수 있습니다.

이 프로시저는 단일 큐 관리자의 경우와 거의 동일하지만 가능한 경우 큐 관리자 이름 대신 *ALL을 사용하거나 각 큐 관리자 이름을 차례로 사용하여 명령을 반복적으로 사용합니다. 프로시저 전체에서 샘플 큐 관리자 이름 QMgr1과 샘플 서브시스템 이름 SUBX를 사용합니다. 이 이름은 원하는 대로 바꿀 수 있습니다.

계획된 종료

1. 종료 1시간 전에 다음을 실행하십시오.

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

종료하려는 모든 큐 관리자에 이 작업을 반복하십시오.

2. 큐 관리자를 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

종료하려는 모든 큐 관리자에 이 작업을 반복하십시오. 개별 명령을 동시에 실행할 수 있습니다.

큐 관리자가 적정 시간(예를 들어, 10분) 내에 종료되지 않으면 단계 3으로 진행하십시오.

3. 모든 큐 관리자를 즉시 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

계획되지 않은 종료

1. 큐 관리자를 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

종료하려는 모든 큐 관리자에 이 작업을 반복하십시오. 개별 명령을 동시에 실행할 수 있습니다.

큐 관리자가 종료되지 않으면 채널 또는 애플리케이션 사용량이 증가할 수 있습니다.

2. 큐 관리자를 즉시 종료해야 하는 경우에는 다음을 실행하십시오.

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

비정상적인 조건에서 종료

1. 큐 관리자를 종료하려면 다음을 실행하십시오.

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

종료하려는 모든 큐 관리자에 이 작업을 반복하십시오. 개별 명령을 동시에 실행할 수 있습니다.

2. 다음을 실행하여 서브시스템(예의 SUBX)을 종료하십시오.

```
ENDSBS SUBX *IMMED
```

종료하려는 모든 서브시스템에 이 작업을 반복하십시오. 개별 명령을 동시에 실행할 수 있습니다.

이 명령이 완료되지 않으면 시스템에서 IPL을 수행하십시오.

경고: 시스템에서 즉시 IPL을 수행할 수 있도록 준비되지 않은 경우에는 ENDJOB 또는 ENDSBS의 결과로 종료되지 않는 작업에 ENDJOBABN을 사용하지 마십시오.

3. 다음을 실행하여 서브시스템을 시작하십시오.

```
STRSBS SUBX
```

시작하려는 모든 서브시스템에 이 작업을 반복하십시오.

4. 다음을 실행하여 큐 관리자를 즉시 종료하십시오.

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. 다음을 실행하여 큐 관리자를 다시 시작하십시오.

```
STRMQM MQMNAME(QMgr1)
```

시작하려는 모든 큐 관리자에 이 작업을 반복하십시오.

큐 관리자 재시작을 위해 몇 초 이상이 소요되는 경우 IBM MQ는 시작 진행 상황을 나타내는 상태 메시지를 간헐적으로 표시합니다.

큐 관리자 재시작 문제점이 계속 해결되지 않으면 IBM 지원 부서에 문의하십시오. 추가 조치를 수행하는 경우 큐 관리자가 손상되어 MQSeries® 또는 IBM MQ를 복구하지 못할 수 있습니다.

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

For details of how to administer IBM MQ for z/OS and the different administrative tasks you might have to undertake, see the following links.

You can also administer IBM MQ for z/OS using the IBM MQ Explorer running in a Linux shell. For more information, see [“IBM MQ Explorer를 사용하여 관리” on page 112.](#)

Related concepts

[IBM MQ for z/OS concepts](#)

Related tasks

[“IBM MQ 관리” on page 7](#)

IBM MQ 큐 관리자 및 연관된 자원을 관리하려면 해당 자원을 활성화하고 관리하기 위해 사용할 수 있는 태스크 세트에서 선호하는 방법을 선택하십시오.

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

Issuing queue manager commands on z/OS

You can control most of the operational environment of IBM MQ by using control commands. You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or authorized applications.

About this task

You use MQSC commands, in batch or interactive mode, to administer queue managers directly. You use PCF commands to help you create applications that administer queue managers. MQSC commands are in human-readable text form, whereas PCF commands let applications create requests and read the replies without having to parse text strings. Like MQSC commands, applications issue PCF commands by sending them as messages to the command input queue.

The following topics describe how you issue queue manager commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications.

Not all commands can be issued from all sources. See [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS” on page 414](#).

Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

Related information

[Administering IBM MQ using MQSC commands](#)

Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

Which MQSC and PCF commands can control each IBM MQ object

Table 1 in ["Command summary for IBM MQ for z/OS"](#) maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object. See also [MQSC commands reference](#) and [“IBM MQ PCF\(Programmable Command Format\) 사용” on page 25](#).

List of sources from which commands can be issued

If you are a suitably authorized user, you can issue IBM MQ commands from the following sources:

- The z/OS console or equivalent (such as SDSF/TSO).

See also [“Using the operations and control panels on z/OS” on page 428](#).

Note: When using the z/OS console, you need to add /cpf to the start of a command, where cpf is the command prefix for the queue manager subsystem.

- The initialization input data sets CSQINP1, CSQINP2, CSQINPT and CSQINPX.

See [“Initialization commands for IBM MQ for z/OS” on page 425](#).

- The z/OS master get command routine, MGCRE (SVC 34).

- The IBM MQ batch utility programs such as CSQUTIL, which processes a list of commands in a sequential data set.

See [“Using the IBM MQ for z/OS utilities”](#) on page 436.

- Suitably authorized applications, sending commands as messages to the SYSTEM.COMMAND.INPUT queue.

The application can be any of the following:

- A batch region program
- A CICS application
- An IMS application
- A TSO application
- An application program or utility on another IBM MQ system

See [“Writing programs to administer IBM MQ for z/OS”](#) on page 444 and [Preparing sample applications for the TSO environment on z/OS](#).

Not all commands can be issued from all sources

Commands are classified according to where they can be issued from:

1

CSQINP1

2

CSQINP2

C

The z/OS console

R

The command server and command queue, by means of CSQUTIL, CSQINPT, CSQINPX, or authorized applications.

Within the command descriptions in [MQSC commands reference](#), these sources are identified by the use of the characters 1, 2, C, and R in each command description. [Table 2](#) in ["Command summary for IBM MQ for z/OS"](#) summarizes the MQSC commands and the sources from which they can be issued.

Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

Related information

[Administering IBM MQ using MQSC commands](#)

Command summary for IBM MQ for z/OS

A summary of the main MQSC and PCF commands, and of the sources from which you can run MQSC commands on IBM MQ for z/OS.

[Table 25](#) on [page 415](#) maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object.

<i>Table 25. Summary of the main MQSC and PCF commands by object type</i>				
MQSC command	ALTER	DEFINE	DISPLAY	DELETE
PCF command	Change	Create/Copy	Inquire	Delete
AUTHINFO	X	X	X	X
CFSTATUS			X	
CFSTRUCT	X	X	X	X

Table 25. Summary of the main MQSC and PCF commands by object type (continued)

MQSC command	ALTER	DEFINE	DISPLAY	DELETE
CHANNEL	X	X	X	X
CHSTATUS			X	
NAMELIST	X	X	X	X
PROCESS	X	X	X	X
QALIAS	M	M	M	M
QCLUSTER			M	
QLOCAL	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
QUEUE	P	P	X	P
QSTATUS			X	
STGCLASS	X	X	X	X

Key to table symbols:

- M = MQSC only
- P = PCF only
- X = both

There are many other MQSC and PCF commands which allow you to manage other IBM MQ resources, and carry out other actions in addition to those summarized in [Table 25 on page 415](#).

[Table 26 on page 416](#) shows every MQSC command, and where each command can be issued from.

- CSQINP1 initialization input data set
- CSQINP2 initialization input data set
- z/OS console (or equivalent)
- SYSTEM.COMMAND.INPUT queue and command server (from applications, CSQUTIL, or the CSQINPX initialization input data set)

Table 26. Sources from which to run MQSC commands

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
ALTER AUTHINFO		X	X	X
ALTER BUFFPOOL		X	X	X
ALTER CFSTRUCT		X	X	X
ALTER CHANNEL		X	X	X
ALTER NAMELIST		X	X	X
ALTER PROCESS		X	X	X
ALTER PSID			X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
ALTER QALIAS		X	X	X
ALTER QLOCAL		X	X	X
ALTER QMGR		X	X	X
ALTER QMODEL		X	X	X
ALTER QREMOTE		X	X	X
ALTER SECURITY	X	X	X	X
ALTER SMDS		X	X	X
ALTER STGCLASS		X	X	X
ALTER SUB			X	X
ALTER TOPIC		X	X	X
ALTER TRACE	X	X	X	X
ARCHIVE LOG	X	X	X	X
BACKUP CFSTRUCT			X	X
CLEAR QLOCAL		X	X	X
CLEAR TOPICSTR			X	X
DEFINE AUTHINFO		X	X	X
DEFINE BUFFPOOL	X			
DEFINE CFSTRUCT		X	X	X
DEFINE CHANNEL		X	X	X
DEFINE LOG			X	X
DEFINE MAXSMGS		X	X	X
DEFINE NAMELIST		X	X	X
DEFINE PROCESS		X	X	X
DEFINE PSID	X		X	X
DEFINE QALIAS		X	X	X
DEFINE QLOCAL		X	X	X
DEFINE QMODEL		X	X	X
DEFINE QREMOTE		X	X	X
DEFINE STGCLASS		X	X	X
DEFINE SUB			X	X
DEFINE TOPIC		X	X	X
DELETE AUTHINFO		X	X	X
DELETE BUFFPOOL		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DELETE CFSTRUCT		X	X	X
DELETE CHANNEL			X	X
DELETE NAMELIST		X	X	X
DELETE PROCESS		X	X	X
DELETE PSID			X	X
DELETE QALIAS		X	X	X
DELETE QLOCAL		X	X	X
DELETE QMODEL		X	X	X
DELETE QREMOTE		X	X	X
DELETE STGCLASS		X	X	X
DELETE SUB			X	X
DELETE TOPIC		X	X	X
DISPLAY ARCHIVE	X	X	X	X
DISPLAY AUTHINFO		X	X	X
DISPLAY CFSTATUS			X	X
DISPLAY CFSTRUCT		X	X	X
DISPLAY CHANNEL		X	X	X
DISPLAY CHINIT			X	X
DISPLAY CHLAUTH		X	X	X
DISPLAY CHSTATUS			X	X
DISPLAY CLUSQMGR			X	X
DISPLAY CMDSERV	X	X	X	X
DISPLAY CONN		X	X	X
DISPLAY GROUP		X	X	X
DISPLAY LOG	X	X	X	X
DISPLAY MAXSMSGS		X	X	X
DISPLAY NAMELIST		X	X	X
DISPLAY PROCESS		X	X	X
DISPLAY PUBSUB		X	X	X
DISPLAY QALIAS		X	X	X
DISPLAY QCLUSTER		X	X	X
DISPLAY QLOCAL		X	X	X
DISPLAY QMGR		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DISPLAY QMODEL		X	X	X
DISPLAY QREMOTE		X	X	X
DISPLAY QSTATUS		X	X	X
DISPLAY QUEUE		X	X	X
DISPLAY SBSTATUS			X	X
DISPLAY SECURITY			X	X
DISPLAY SMDS		X	X	X
DISPLAY SMDSCONN		X	X	X
DISPLAY STGCLASS		X	X	X
DISPLAY SUB			X	X
DISPLAY SYSTEM	X	X	X	X
DISPLAY TCLUSTER		X	X	X
DISPLAY THREAD		X	X	X
DISPLAY TOPIC		X	X	X
DISPLAY TPSTATUS		X	X	X
DISPLAY TRACE	X	X	X	X
DISPLAY USAGE		X	X	X
MOVE QLOCAL		X	X	X
PING CHANNEL			X	X
RECOVER BSDS			X	X
RECOVER CFSTRUCT			X	X
REFRESH CLUSTER			X	X
REFRESH QMGR		X	X	X
REFRESH SECURITY			X	X
RESET CFSTRUCT			X	X
RESET CHANNEL			X	X
RESET CLUSTER			X	X
RESET QMGR		X	X	X
RESET QSTATS		X	X	X
RESET SMDS			X	X
RESET TPIPE			X	X
RESOLVE CHANNEL			X	X
RESOLVE INDOUBT		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
RESUME QMGR			X	X
RVERIFY SECURITY		X	X	X
SET ARCHIVE	X	X	X	X
SET CHLAUTH		X	X	X
SET LOG	X	X	X	X
SET SYSTEM	X	X	X	X
START CHANNEL			X	X
START CHINIT		X	X	X
START CMDSERV	X	X	X	
START LISTENER			X	X
START QMGR			X	
START SMDSCONN		X	X	X
START TRACE	X	X	X	X
STOP CHANNEL			X	X
STOP CHINIT			X	X
STOP CMDSERV	X	X	X	
STOP LISTENER			X	X
STOP QMGR			X	X
STOP SMDSCONN		X	X	X
STOP TRACE	X	X	X	X
SUSPEND QMGR			X	X

In MQSC commands, each command description identifies the sources from which that command can be run.

Using MQSC to start and stop a queue manager on z/OS

An introduction to using control commands on IBM MQ for z/OS: After you have installed IBM MQ, use MQSC commands to start and stop a queue manager.

Before you begin

After you have installed IBM MQ, it is defined as a formal z/OS subsystem. This message appears during any initial program load (IPL) of z/OS:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

where *ssnm* is the IBM MQ subsystem name.

From now on, you can start the queue manager for that subsystem *from any z/OS console that has been authorized to issue system control commands*; that is, a z/OS SYS command group. You must issue the START command from the authorized console, you cannot issue it through JES or TSO.

If you are using queue sharing groups, you must start RRS first, and then Db2®, before you start the queue manager.

About this task

When a queue manager stops under normal conditions, its last action is to take a termination checkpoint. This checkpoint, and the logs, give the queue manager the information it needs to restart.

The following steps contain information about the START and STOP commands, and contain a brief overview of start-up after an abnormal termination has occurred.

Procedure

1. Start a queue manager

You start a queue manager by issuing a START QMGR command. However, you cannot successfully use the START command unless you have appropriate authority. See the [Setting up security on z/OS](#) for information about IBM MQ security. The following code shows examples of the START command. Note that you must prefix an MQSC command with a command prefix string (CPF).

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

See [START QMGR](#) for information about the syntax of the START QMGR command.

You cannot run the queue manager as a batch job or start it using a z/OS command START. These methods are likely to start an address space for IBM MQ that then ends abnormally. Nor can you start a queue manager from the CSQUTIL utility program or a similar user application.

You can, however, start a queue manager from an APF-authorized program by passing a START QMGR command to the z/OS MGCRC (SVC 34) service.

If you are using queue sharing groups, the associated Db2 systems and RRS must be active when you start the queue manager.

Start options

When you start a queue manager, a system parameter module is loaded. You can specify the name of the system parameter module in one of two ways:

- With the PARM parameter of the /cpf START QMGR command, for example

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- With a parameter in the startup procedure, for example, code the JCL EXEC statement as

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARM(CSQ1ZPRM)'
```

A system parameter module provides information specified when the queue manager was customized.

You can use the **QMGRPROD** option to specify the product against which the queue manager usage is to be recorded, and the **AMSPROD** option to specify the equivalent for AMS if that is used. See the MQSC [START QMGR](#) command for details of the permitted values.

An example JCL EXEC statement follows:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

See [z/OS MVS Product Management](#) for more information on measured usage and product registration.

You can also use the ENVPARM option to substitute one or more parameters in the JCL procedure for the queue manager.

For example, you can update your queue manager startup procedure, so that the **DDname** CSQINP2 is a variable. This means that you can change the CSQINP2 **DDname** without changing the startup procedure. This is useful for implementing changes, providing back-outs for operators, and queue manager operations.

Suppose your start-up procedure for queue manager CSQ1 looked like this:

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
// DD DISP=SHR,DSN=thlqual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BSDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BSDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqual.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*
```

If you then start your queue manager with the following command:

```
+CSQ1 START QMGR
```

then the CSQINP2 used is a member called CSQ1NORM.

However, suppose you are putting a new suite of programs into production so that the next time you start queue manager CSQ1, the CSQINP2 definitions are to be taken from member CSQ1NEW. To do this, you would start the queue manager with this command:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

and CSQ1NEW would be used instead of CSQ1NORM. Note: z/OS limits the KEYWORD=value specifications for symbolic parameters (as in INP2=NEW) to 255 characters.

Starting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting a queue manager after it ends abnormally is different from starting it after the STOP QMGR command has been issued. After STOP QMGR, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

However, if the queue manager ends abnormally, it terminates without being able to finish its work or take a termination checkpoint. When you restart a queue manager after an abend, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks. Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies.

User messages on start-up

When you start a queue manager successfully, the queue manager produces a set of startup messages.

2. Stop a queue manager.

Before stopping a queue manager, all IBM MQ-related write-to-operator-with-reply (WTOR) messages must receive replies, for example, getting log requests. Each of the following commands terminates a running queue manager.

```
+CSQ1 STOP QMGR
+CSQ1 STOP QMGR MODE(QUIESCE)
+CSQ1 STOP QMGR MODE(FORCE)
+CSQ1 STOP QMGR MODE(RESTART)
```

The command STOP QMGR defaults to STOP QMGR MODE(QUIESCE).

In QUIESCE mode, IBM MQ does not allow any new connection threads to be created, but allows existing threads to continue; it terminates only when all threads have ended. Applications can request to be notified in the event of the queue manager quiescing. Therefore, use the QUIESCE mode where possible so that applications that have requested notification have the opportunity to disconnect. See [What happens during termination](#) for details.

If the queue manager does not terminate in a reasonable time in response to a STOP QMGR MODE(QUIESCE) command, use the DISPLAY CONN command to determine whether any connection threads exist, and take the necessary steps to terminate the associated applications. If there are no threads, issue a STOP QMGR MODE(FORCE) command.

The STOP QMGR MODE(QUIESCE) and STOP QMGR MODE(FORCE) commands deregister IBM MQ from the MVS Automatic Restart Manager (ARM), preventing ARM from restarting the queue manager automatically. The STOP QMGR MODE(RESTART) command works in the same way as the STOP QMGR MODE(FORCE) command, except that it does not deregister IBM MQ from ARM. This means that the queue manager is eligible for immediate automatic restart.

If the IBM MQ subsystem is not registered with ARM, the STOP QMGR MODE(RESTART) command is rejected and the following message is sent to the z/OS console:

```
CSQY205I ARM element arm-element is not registered
```

If this message is not issued, the queue manager is restarted automatically. For more information about ARM, see [“Using the z/OS Automatic Restart Manager \(ARM\)”](#) on page 503.

Only cancel the queue manager address space if STOP QMGR MODE(FORCE) does not terminate the queue manager.

If a queue manager is stopped by either canceling the address space or by using the command STOP QMGR MODE(FORCE), consistency is maintained with connected CICS or IMS systems. Resynchronization of resources is started when a queue manager restarts and is completed when the connection to the CICS or IMS system is established.

Note: When you stop your queue manager, you might find message IEF352I is issued. z/OS issues this message if it detects that failing to mark the address space as unusable would lead to an integrity exposure. You can ignore this message.

Stop messages

After issuing a STOP QMGR command, you get the messages CSQY009I and CSQY002I, for example:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM
USER(userid), STOP MODE(FORCE)
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Where `userid` is the user ID that issued the STOP QMGR command, and the MODE parameter depends on that specified in the command.

When the STOP command has completed successfully, the following messages are displayed on the z/OS console:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

If you are using ARM, and you did not specify MODE(RESTART), the following message is also displayed:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type
arm-element-type successful
```

You cannot restart the queue manager until the following message has been displayed:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```

Issuing commands from a z/OS console or equivalent

You can issue IBM MQ MQSC and PCF commands from a z/OS console or its equivalent. You can also issue IBM MQ commands from anywhere where you can issue z/OS commands, such as SDSF or by a program using the MGCRC macro. When using the z/OS console, you add `/cpf` to the start of a command.

Before you begin

Not all commands can be issued by the z/OS console. Within the command description topics (the children of [MQSC commands reference](#)), each command that can be issued by the console is identified by the character 'C'. [Table 2 in "Command summary for IBM MQ for z/OS"](#) summarizes the MQSC commands and the sources from which they can be issued.

You cannot issue IBM MQ commands using the IMS/SSR command format from an IMS terminal. This function is not supported by the IMS adapter.

The input field provided by SDSF might not be long enough for some commands, particularly those commands for channels.

The maximum amount of data that can be displayed as a result of a command typed in at the console is 32 KB.

About this task

If you are a suitably authorized user, you can issue IBM MQ commands from the z/OS console or equivalent (such as SDSF/TSO).

When using the z/OS console, you need to add `/cpf` to the start of a command, where `cpf` is the command prefix for the queue manager subsystem.

The following steps refer to commands and attributes using their MQSC command names rather than their PCF names.

Procedure

- Use command prefix strings

Each IBM MQ command must be prefixed with a command prefix string (CPF).

Because more than one IBM MQ subsystem can run under z/OS, the CPF is used to indicate which IBM MQ subsystem processes the command.

For example, to start the queue manager for a subsystem called CSQ1, where CPF is ' +CSQ1 ', you issue the following command from the operator console:

```
+CSQ1 START QMGR
```

This CPF must be defined in the subsystem name table (for the subsystem CSQ1), as described in [Defining command prefix strings \(CPFs\)](#). In the examples, the string +CSQ1 is used as the command prefix.

- Use the z/OS console to issue commands

You can type simple commands from the z/OS console, for example:

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLOCAL)
```

However, for complex commands or for sets of commands that you issue frequently, the other methods of issuing commands are better.

- Receive command responses

Direct responses to commands are sent to the console that issued the command. IBM MQ supports the *Extended Console Support* (EMCS) function available in z/OS, and therefore consoles with 4 byte IDs can be used. Additionally, all commands except START QMGR and STOP QMGR support the use of Command and Response Tokens (CARTs) when the command is issued by a program using the MGCRC macro.

Related tasks

[“Using the operations and control panels on z/OS” on page 428](#)

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

[Preparing sample applications for the TSO environment on z/OS](#)

Initialization commands for IBM MQ for z/OS

Initialization commands can be used to control the queue manager startup.

Commands in the initialization input data sets are processed when IBM MQ is initialized on queue manager startup. Three types of command can be issued from the initialization input data sets:

- Commands to define IBM MQ entities that cannot be defined elsewhere, for example DEFINE BUFFPOOL.

These commands must reside in the data set identified by the DD name CSQINP1. They are processed before the restart phase of initialization. They cannot be issued through the console, operations and control panels, or an application program. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT1 statement of the started task procedure.

- Commands to define IBM MQ objects that are recoverable after restart. These definitions must be specified in the data set identified by the DD name CSQINP2. They are stored in page set zero. CSQINP2 is processed after the restart phase of initialization. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT2 statement of the started task procedure.
- Commands to manipulate IBM MQ objects. These commands must also be specified in the data set identified by the DD name CSQINP2. For example, the IBM MQ-supplied sample contains an ALTER QMGR command to specify a dead-letter queue for the subsystem. The response to these commands is written to the CSQOUT2 output data set.

Note: If IBM MQ objects are defined in CSQINP2, IBM MQ attempts to redefine them each time the queue manager is started. If the objects already exist, the attempt to define them fails. If you need to define your objects in CSQINP2, you can avoid this problem by using the REPLACE parameter of the DEFINE commands, however, this overrides any changes that were made during the previous run of the queue manager.

Sample initialization data set members are supplied with IBM MQ for z/OS. They are described in [Sample definitions supplied with IBM MQ](#).

Initialization commands for distributed queuing

You can also use the CSQINP2 initialization data set for the START CHINIT command. If you need a series of other commands to define your distributed queuing environment (for example, starting listeners), IBM MQ provides a third initialization input data set, called CSQINPX, that is processed as part of the channel initiator started task procedure.

The MQSC commands contained in the data set are executed at the end of channel initiator initialization, and output is written to the data set specified by the CSQOUTX DD statement. You might use the CSQINPX initialization data set to start listeners for example.

A sample channel initiator initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

Initialization commands for publish/Subscribe

If you need a series of commands to define your publish/subscribe environment (for example, when defining subscriptions), IBM MQ provides a fourth initialization input data set, called CSQINPT.

The MQSC commands contained in the data set are executed at the end of publish/subscribe initialization, and output is written to the data set specified by the CSQOUTT DD statement. You might use the CSQINPT initialization data set to define subscriptions for example.

A sample publish/subscribe initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

Private and global definitions on IBM MQ for z/OS

When you define an object on IBM MQ for z/OS, you can choose whether you want to share that definition with other queue managers (a *global* definition), or whether the object definition is to be used by one queue manager only (a *private* definition). This is called the object *disposition*.

Global definition

If your queue manager belongs to a queue sharing group, you can choose to share any object definitions you make with the other members of the group. This means that you have to define an object once only, reducing the total number of definitions required for the whole system.

Global object definitions are held in a *shared repository* (a Db2 shared database), and are available to all the queue managers in the queue sharing group. These objects have a disposition of GROUP.

Private definition

If you want to create an object definition that is required by one queue manager only, or if your queue manager is not a member of a queue sharing group, you can create object definitions that are not shared with other members of a queue sharing group.

Private object definitions are held on page set zero of the defining queue manager. These objects have a disposition of QMGR.

You can create private definitions for all types of IBM MQ objects except CF structures (that is, channels, namelists, process definitions, queues, queue managers, storage class definitions, and authentication information objects), and global definitions for all types of objects except queue managers.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily if you want, and IBM MQ allows you to refresh the page set copies from the repository copy if required.

IBM MQ always tries to refresh the page set copies from the repository copy at startup (for channel commands, this is done when the channel initiator restarts), or if the group object is changed.

Note: The copy of the definition is refreshed from the definition of the group, only if the definition of the group has changed after you created the copy of the definition.

This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive. The copies are refreshed by generating DEFINE REPLACE commands, therefore there are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that changes the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that deletes that queue fails.
- If a copy of a queue would require ALTER with FORCE to change it.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers.

If the queue manager is shut down and then restarted stand-alone, any local copies of objects are deleted, unless for example, the queue has associated messages.

There is a third object disposition that applies to local queues only. This allows you to create shared queues. The definition for a shared queue is held on the shared repository and is available to all the queue managers in the queue sharing group. In addition, the messages on a shared queue are also available to all the queue managers in the queue sharing group. This is described in [Shared queues and queue sharing groups](#). Shared queues have an object disposition of SHARED.

The following table summarizes the effect of the object disposition options for queue managers started stand-alone, and as a member of a queue sharing group.

Disposition	Stand-alone queue manager	Member of a queue sharing group
QMGR	Object definition held on page set zero.	Object definition held on page set zero.
GROUP	Not allowed.	Object definition held in the shared repository. Local copy held on page set zero of each queue manager in the group.
SHARED	Not allowed.	Queue definition held in the shared repository. Messages available to any queue manager in the group.

Manipulating global definitions

If you want to change the definition of an object that is held in the shared repository, you need to specify whether you want to change the version on the repository, or the local copy on page set zero. Use the object disposition as part of the command to do this.

Directing commands to different queue managers on z/OS

You can use the *command scope* to control on which queue manager the command runs.

You can choose to execute a command on the queue manager where it is entered, or on a different queue manager in the queue sharing group. You can also choose to issue a particular command in parallel on all the queue managers in a queue sharing group. This is possible for both MQSC commands and PCF commands.

This is determined by the *command scope*. The command scope is used with the object disposition to determine which version of an object you want to work with.

For example, you might want to alter some of the attributes of an object, the definition of which is held in the shared repository.

- You might want to change the version on one queue manager only, and not make any changes to the version on the repository or those in use by other queue managers.
- You might want to change the version in the shared repository for future users, but leave existing copies unchanged.
- You might want to change the version in the shared repository, but also want your changes to be reflected immediately on all the queue managers in the queue sharing group that hold a copy of the object on their page set zero.

Use the command scope to specify whether the command is executed on this queue manager, another queue manager, or all queue managers. Use the object disposition to specify whether the object you are manipulating is in the shared repository (a group object), or is a local copy on page set zero (a queue manager object).

You do not have to specify the command scope and object disposition to work with a shared queue because every queue manager in the queue sharing group handles the shared queue as a single queue.

z/OS

Using the operations and control panels on z/OS

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

Before you begin

The IBM MQ for z/OS operations and controls panels (CSQOREXX) might not support all new function and parameters added from version 7 onwards. For example, there are no panels for the direct manipulation of topic objects or subscriptions. Use one of the following supported mechanisms to administer publish/subscribe definitions and other system controls that are not directly available from other panels:

1. IBM MQ Explorer
2. z/OS console
3. Programmable Command Format (PCF) messages
4. COMMAND function of CSQUTIL
5. IBM MQ Console

Note that the generic **Command** action in the CSQOREXX panels allows you to issue any valid MQSC command, including SMDS related commands. You can use all the commands that the COMMAND function of CSQUTIL issues.

You cannot issue the IBM MQ commands directly from the command line in the panels.

To use the operations and control panels, you must have the correct security authorization; this is described in the [User IDs for command security and command resource security](#).

You cannot provide a user ID and password using CSQUTIL, or the CSQOREXX panels. Instead, if you user ID has UPDATE authority to the BATCH profile in MQCONN, you can bypass the **CHKLOCL**(REQUIRED) setting. See [Using **CHKLOCL** on locally bound applications](#) for more information.

If you are setting up or changing many objects, use the COMMAND function of the CSQUTIL utility program. See [“Using the CSQUTIL utility for IBM MQ for z/OS” on page 438](#).

About this task

The operations and control panels support the controls for the channel initiator (for example, to start a channel or a TCP/IP listener), for clustering, and for security. They also enable you to display information about threads and page set usage.

The panels work by sending MQSC type IBM MQ commands to a queue manager, through the system command input queue.

Example

This is the panel that displays when you start a panel session:

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.
Action . . . . . 1      0. List with filter  4. Manage
                        1. List or Display  5. Perform
                        2. Define like     6. Start
                        3. Alter             7. Stop
                        8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A  Q=Qmgr, C=Copy, P=Private, G=Group,
                        S=Shared, A=All
Connect name . . . . . MQ1C - local queue manager or group
Target queue manager . . . . . MQ1C
                        - connected or remote queue manager for command input
Action queue manager . . . . . MQ1C - command scope in group
Response wait time . . . . . 30 5 - 999 seconds
(C) Copyright IBM Corporation 1993, 2024. All rights reserved.
Command ==>
F1=Help      F2=Split    F3=Exit     F4=Prompt   F9=SwapNext F10=Messages
F12=Cancel
```

From this panel you can perform actions such as these:

- Choose the local queue manager you want and whether you want the commands issued on that queue manager, on a remote queue manager, or on another queue manager in the same queue sharing group as the local queue manager. Over type the queue manager name if you need to change it.
- Select the action you want to perform by typing in the appropriate number in the **Action** field.
- Specify the object type that you want to work with. Press function key F1 for help about the object types if you are not sure what they are.
- Specify the disposition of the object type that you want to work with.
- Display a list of objects of the type specified. Type in an asterisk (*) in the **Name** field and press **Enter** to display a list of objects (of the type specified) that have already been defined on the action queue manager. You can then select one or more objects to work with in sequence. All the actions are available from the list.

Note: You are recommended to make choices that result in a list of objects being displayed, and then work from that list. Use the **Display** action, because that is allowed for all object types.

Invocation and rules for the operations and control panels

You can control IBM MQ and issue control commands through the ISPF panels.

How to access the IBM MQ operations and control panels

If the ISPF/PDF primary options menu has been updated for IBM MQ, you can access the IBM MQ operations and control panels from that menu. For details about updating the menu, see the [Task 20: Set up the operations and control panels](#).

You can access the IBM MQ operations and control panels from the TSO command processor panel (typically option 6 on the ISPF/PDF primary options menu). The name of the exec that you run to do this is CSQOREXX. It has two parameters; `thlqua1` is the high-level qualifier for the IBM MQ libraries to be used, and `langletter` is the letter identifying the national language libraries to be used (for example, E for U.S. English). The parameters can be omitted if the IBM MQ libraries are permanently installed in your ISPF setup. Alternatively, you can issue CSQOREXX from the TSO command line.

These panels are designed to be used by operators and administrators with a minimum of formal training. Read these instructions with the panels running and try out the different tasks suggested.

Note: While using the panels, temporary dynamic queues with names of the form SYSTEM.CSQOREXX.* are created.

Rules for the operations and control panels

See [Rules for naming IBM MQ objects](#) about the general rules for IBM MQ character strings and names. However, there are some rules that apply only to the operations and control panels:

- Do not enclose strings, for example descriptions, in single or double quotation marks.
- If you include an apostrophe or quotation mark in a text field, you do not have to repeat it or add an escape character. The characters are saved exactly as you type them; for example:

```
This is Maria's queue
```

The panel processor doubles them for you to pass them to IBM MQ. However, if it has to truncate your data to do this, it does so.

- You can use uppercase or lowercase characters in most fields, and they are folded to uppercase characters when you press Enter. The exceptions are:
 - Storage class names and coupling facility structure names, which must start with uppercase A through Z and be followed by uppercase A through Z or numeric characters.
 - Certain fields that are not translated. These include:
 - Application ID
 - Description
 - Environment data
 - Object names (but if you use a lowercase object name, you might not be able to enter it at a z/OS console)
 - Remote system name
 - Trigger data
 - User data
- In names, leading blanks and leading underscores are ignored. Therefore, you cannot have object names beginning with blanks or underscores.
- Underscores are used to show the extent of blank fields. When you press Enter, trailing underscores are replaced by blanks.
- Many description and text fields are presented in multiple parts, each part being handled by IBM MQ independently. This means that trailing blanks are retained and the text is not contiguous.

Blank fields

When you specify the **Define** action for an IBM MQ object, each field on the define panel contains a value. See the general help (extended help) for the display panels for information about where IBM MQ gets the values. If you type over a field with blanks, and blanks are not allowed, IBM MQ puts the installation default value in the field or prompts you to enter the required value.

When you specify the **Alter** action for an IBM MQ object, each field on the alter panel contains the current value for that field. If you type over a field with blanks, and blanks are not allowed, the value of that field is unchanged.

Objects and actions on z/OS

The operations and control panels offer you many different types of object and a number of actions that you can perform on them.

The actions are listed on the initial panel and enable you to manipulate the objects and display information about them. These objects include all the IBM MQ objects, together with some extra ones. The objects fall into the following categories.

- [Queues, processes, authentication information objects, namelists, storage classes and CF structures](#)

- [Channels](#)
- [Cluster objects](#)
- [Queue manager and security](#)
- [Connections](#)
- [System](#)

Refer to [Actions](#) for a cross-reference table of the actions which can be taken with the IBM MQ objects.

Queues, processes, authentication information objects, namelists, storage classes and CF structures

These are the basic IBM MQ objects. There can be many of each type. They can be listed, listed with filter, defined, and deleted, and have attributes that can be displayed and altered, using the LIST or DISPLAY, LIST with FILTER, DEFINE LIKE, MANAGE, and ALTER actions. (Objects are deleted using the MANAGE action.)

This category consists of the following objects:

QLOCAL	Local queue
QREMOTE	Remote queue
QALIAS	Alias queue for indirect reference to a queue
QMODEL	Model queue for defining queues dynamically
QUEUE	Any type of queue
QSTATUS	Status of a local queue
PROCESS	Information about an application to be started when a trigger event occurs
AUTHINFO	Authentication information: definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers
NAMELIST	List of names, such as queues or clusters
STGCLASS	Storage class
CFSTRUCT	coupling facility (CF) structure
CFSTATUS	Status of a CF structure

Channels

Channels are used for distributed queuing. There can be many of each type, and they can be listed, listed with filter, defined, deleted, displayed, and altered. They also have other functions available using the START, STOP and PERFORM actions. PERFORM provides reset, ping, and resolve channel functions.

This category consists of the following objects:

CHANNEL	Any type of channel
SENDER	Sender channel
SERVER	Server channel
RECEIVER	Receiver channel
REQUESTER	Requester channel
CLUSRCVR	Cluster-receiver channel
CLUSSDR	Cluster-sender channel
SVRCONN	Server-connection channel
CLNTCONN	Client-connection channel
CHSTATUS	Status of a channel connection

Cluster objects

Cluster objects are created automatically for queues and channels that belong to a cluster. The base queue and channel definitions can be on another queue manager. There can be many of each type, and names can be duplicated. They can be listed, listed with filter, and displayed. PERFORM, START, and STOP are also available through the LIST actions.

This category consists of the following objects:

CLUSQ	Cluster queue, created for a queue that belongs to a cluster
CLUSCHL	Cluster channel, created for a channel that belongs to a cluster
CLUSQMGR	Cluster queue manager, the same as a cluster channel but identified by its queue manager name

Cluster channels and cluster queue managers do have the PERFORM, START and STOP actions, but only indirectly through the DISPLAY action.

Queue manager and security

Queue manager and security objects have a single instance. They can be listed, and have attributes that can be displayed and altered (using the LIST or DISPLAY, and ALTER actions), and have other functions available using the PERFORM action.

This category consists of the following objects:

MANAGER	Queue manager: the PERFORM action provides suspend and resume cluster functions
SECURITY	Security functions: the PERFORM action provides refresh and reverify functions

Connection

Connections can be listed, listed with filter and displayed.

This category consists only of the connection object, CONNECT.

System

A collection of other functions. This category consists of the following objects:

SYSTEM	System functions
CONTROL	Synonym for SYSTEM

The functions available are:

LIST or DISPLAY	Display queue sharing group, distributed queuing, page set, or data set usage information.
PERFORM	Refresh or reset clustering
START	Start the channel initiator or listeners
STOP	Stop the channel initiator or listeners

Actions

The actions that you can perform for each type of object are shown in the following table:

Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
AUTHINFO	X	X	X	X	X			

Table 27. Valid operations and control panel actions for IBM MQ objects (continued)

Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
CFSTATUS				X				
CFSTRUCT	X	X	X	X	X			
CHANNEL	X	X	X	X	X	X	X	X
CHSTATUS				X	X			
CLNTCONN	X	X	X	X	X			
CLUSCHL				X	X	X(2)	X(2)	X(2)
CLUSQ				X	X			
CLUSQMGR				X	X	X(2)	X(2)	X(2)
CLUSRCVR	X	X	X	X	X	X	X	X
CLUSDR	X	X	X	X	X	X	X	X
CONNECT				X	X			
CONTROL				X		X	X	X
MANAGER	X			X		X		
NAMELIST	X	X	X	X	X			
PROCESS	X	X	X	X	X			
QALIAS	X	X	X	X	X			
QLOCAL	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QREMOTE	X	X	X	X	X			
QSTATUS				X	X			
QUEUE	X	X	X	X	X			
RECEIVER	X	X	X	X	X	X	X	X
REQUESTER	X	X	X	X	X	X	X	X
SECURITY	X			X		X		
SENDER	X	X	X	X	X	X	X	X
SERVER	X	X	X	X	X	X	X	X
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

Note:

1. Provides Delete and other functions
2. Using the List or Display action

z/OS Object dispositions on z/OS

You can specify the *disposition* of the object with which you need to work. The disposition signifies where the object **definition** is kept, and how the object behaves.

The disposition is significant only if you are working with any of the following object types:

- queues
- channels
- processes
- namelists
- storage classes
- authentication information objects

If you are working with other object types, the disposition is disregarded.

Permitted values are:

Q

QMGR. The object definitions are on the page set of the queue manager and are accessible only by the queue manager.

C

COPY. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. They are local copies of objects defined as having a disposition of GROUP.

P

PRIVATE. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. The objects have been defined as having a disposition of QMGR or COPY.

G

GROUP. The object definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

S

SHARED. This disposition applies only to local queues. The queue definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

A

ALL. If the action queue manager is either the target queue manager, or *, objects of **all** dispositions are included; otherwise, objects of QMGR and COPY dispositions only are included. This is the default.

z/OS Selecting a queue manager, defaults, and levels using the ISPF control panel on z/OS

You can use the CSQOREXX exec in ISPF to control your queue managers.

While you are viewing the initial panel, you are not connected to any queue manager. However, as soon as you press Enter, you are connected to the queue manager, or a queue manager in the queue sharing group named in the **Connect name** field. You can leave this field blank; this means that you are using the default queue manager for batch applications. This is defined in CSQBDEFV (see [Task 19: Set up Batch, TSO, and RRS adapters](#) for information about this).

Use the **Target queue manager** field to specify the queue manager where the actions you request are to be performed. If you leave this field blank, it defaults to the queue manager specified in the **Connect name** field. You can specify a target queue manager that is not the one you connect to. In this case, you would normally specify the name of a remote queue manager object that provides a queue manager alias definition (the name is used as the *ObjectQMgrName* when opening the command input queue). To do this, you must have suitable queues and channels set up to access the remote queue manager.

The **Action queue manager** field allows you to specify a queue manager that is in the same queue sharing group as the queue manager specified in the **Target queue manager** field to be the queue manager where the actions you request are to be performed. If you specify * in this field, the actions you request are

performed on all queue managers in the queue sharing group. If you leave this field blank, it defaults to the value specified in the **Target queue manager** field. The **Action queue manager** field corresponds to using the CMDSCOPE command modifier described in [The MQSC commands](#).

Queue manager defaults

If you leave any queue manager fields blank, or choose to connect to a queue sharing group, a secondary window opens when you press **Enter**. This window confirms the names of the queue managers you will be using. Press **Enter** to continue. When you return to the initial panel after having made some requests, you find fields completed with the actual names.

Queue manager levels

If the action queue manager is not at IBM MQ 8.0.0 or later, some fields are not displayed, and some values cannot be entered. A few objects and actions are disallowed. In such cases, a secondary window opens asking for you to confirm that you want to proceed.

Using the function keys and command line with the ISPF control panels on z/OS

To use the panels, you must use the function keys or enter the equivalent commands in the ISPF control panel command area.

- [Function keys](#)
 - [Processing your actions](#)
 - [“Displaying IBM MQ user messages” on page 436](#)
 - [Canceling your actions](#)
 - [Getting help](#)
- [Using the command line](#)

Function keys

The function keys have special settings for IBM MQ. (This means that you cannot use the ISPF default values for the function keys; if you have previously used the `KEYLIST OFF` ISPF command anywhere, you must type `KEYLIST ON` in the command area of any operations and control panel and then press **Enter** to enable the IBM MQ settings.)

These function key settings can be displayed on the panels, as shown in [“Using the operations and control panels on z/OS” on page 428](#). If the settings are not shown, type `PFSHOW` in the command area of any operations and control panel and then press **Enter**. To remove the display of the settings, use the command `PFSHOW OFF`.

The function key settings in the operations and control panels conform to CUA standards. Although you can change the key setting through normal ISPF procedures (such as the `KEYLIST` utility), you are not recommended to do so.

Note: Using the `PFSHOW` and `KEYLIST` commands affects any other logical ISPF screens that you have, and their settings remain when you leave the operations and control panels.

Processing your actions

Press **Enter** to carry out the action requested on a panel. The information from the panel is sent to the queue manager for processing.

Each time you press **Enter** in the panels, IBM MQ generates one or more operator messages. If the operation was successful, you get confirmation message `CSQ9022I`, otherwise you get some error messages.

Displaying IBM MQ user messages

Press function key F10 in any panel to see the IBM MQ user messages.

Canceling your actions

On the initial panel, both F3 and F12 exit the operations and control panels and return you to ISPF. No information is sent to the queue manager.

On any other panel, press function keys F3 or F12 to leave the current panel **ignoring any data you have typed since last pressing Enter**. Again, no information is sent to the queue manager.

- F3 takes you straight back to the initial panel.
- F12 takes you back to the previous panel.

Getting help

Each panel has help panels associated with it. The help panels use the ISPF protocols:

- Press function key F1 on any panel to see general help (extended help) about the task.
- Press function key F1 with the cursor on any field to see specific help about that field.
- Press function key F5 from any field help panel to get the general help.
- Press function key F3 to return to the base panel, that is, the panel from which you pressed function key F1.
- Press function key F6 from any help panel to get help about the function keys.

If the help information carries on into a second or subsequent pages, a **More** indicator is displayed in the upper-right of the panel. Use these function keys to navigate through the help pages:

- F11 to get to the next help page (if there is one).
- F10 to get back to the previous help page (if there is one).

Using the command line

You never need to use the command line to issue the commands used by the operations and control panels because they are available from function keys. The command line is provided to allow you to enter normal ISPF commands (like **PFSHOW**).

The ISPF command `PANELID ON` displays the name of the current CSQOREXX panel.

The command line is initially displayed in the default position at the bottom of the panels, regardless of what ISPF settings you have. You can use the `SETTINGS ISPF` command from any of the operations and control panels to change the position of the command line. The settings are remembered for subsequent sessions with the operations and control panels.

▶ z/OS

Using the IBM MQ for z/OS utilities

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

IBM MQ for z/OS supplies a set of utility programs to help you perform various administrative tasks, including the following:

- Manage message security policies.
- Perform backup, restoration, and reorganization tasks.
- Issue commands and process object definitions.
- Generate data-conversion exits.
- Modify the bootstrap data set.
- List information about the logs.
- Print the logs.

- Set up Db2 tables and other Db2 utilities.
- Process messages on the dead-letter queue.

The message security policy utility

The message security policy utility (CSQ0UTIL) runs as a stand-alone utility to manage message security policies. See [The message security policy utility \(CSQ0UTIL\)](#) for more information.

The CSQUTIL utility

This is a utility program provided to help you with backup, restore and reorganize tasks. See [“Using the CSQUTIL utility for IBM MQ for z/OS” on page 438](#).

The data conversion exit utility

The IBM MQ for z/OS data conversion exit utility (**CSQUCVX**) runs as a stand-alone utility to create data conversion exit routines.

The change log inventory utility

The IBM MQ for z/OS change log inventory utility program (**CSQJU003**) runs as a stand-alone utility to change the bootstrap data set (BSDS). You can use the utility to perform the following functions:

- Add or delete active or archive log data sets.
- Supply passwords for archive logs.

The print log map utility

The IBM MQ for z/OS print log map utility program (**CSQJU004**) runs as a stand-alone utility to list the following information:

- Log data set name and log RBA association for both copies of all active and archive log data sets. If dual logging is not active, there is only one copy of the data sets.
- Active log data sets available for new log data.
- Contents of the queue of checkpoint records in the bootstrap data set (BSDS).
- Contents of the archive log command history record.
- System and utility time stamps.

The log print utility

The log print utility program (**CSQ1LOGP**) is run as a stand-alone utility. You can run the utility specifying:

- A bootstrap data set (BSDS)
- Active logs (with no BSDS)
- Archive logs (with no BSDS)

The queue sharing group utility

The queue sharing group utility program (**CSQ5PQSG**) runs as a stand-alone utility to set up Db2 tables and perform other Db2 tasks required for queue sharing groups.

The active log preformat utility

The active log preformat utility (**CSQJUFMT**) formats active log data sets before they are used by a queue manager. If the active log data sets are preformatted by the utility, log write performance is improved on the queue manager's first pass through the active logs.

The dead-letter queue handler utility

The dead-letter queue handler utility program (**CSQUDLQH**) runs as a stand-alone utility. It checks messages that are on the dead-letter queue and processes them according to a set of rules that you supply to the utility.

The queue load and unload utility

The queue load and unload utility copies or moves the contents of a queue, or its messages, to a file. The utility was originally shipped as the **QLOAD** utility in IBM MQ Supportpac MO03. From IBM MQ 8.0 it is integrated into the product as executable module **CSQUDMSG** in the SCSQLOAD library, with an alias of QLOAD for compatibility. Sample JCL is provided as member CSQ4QLOD in SCSQPROC.

The equivalent utility for Multiplatforms is called **dmpmqmsg**. For details of the available options, including the differences for z/OS, see [dmpmqmsg \(queue load and unload\)](#).

You can also reload messages as described in [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#) and [Restoring messages from a data set to a queue \(SLOAD\) on z/OS](#).

Using the CSQUTIL utility for IBM MQ for z/OS

The CSQUTIL utility program is provided with IBM MQ for z/OS to help you perform backup, restoration, and reorganization tasks, and to issue commands and process object definitions.

About this task

Use this utility program to invoke the following functions. For example, you can issue commands from a sequential data set using the COMMAND function of the CSQUTIL utility. This function transfers the commands, as messages, to the *system-command input queue* and waits for the response, which is printed together with the original commands in SYSPRINT.

For more information about the CSQUTIL utility program, see [IBM MQ utility program \(CSQUTIL\)](#).

Procedure

- [COMMAND](#)
Use this function to issue MQSC commands, to record object definitions, and to make client-channel definition files.
- [COPY](#)
Use this function to read the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, and put them into a sequential file and retain the original queue.
- [COPYPAGE](#)
Use this function to copy whole page sets to larger page sets.
- [EMPTY](#)
Use this function to delete the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, retaining the definitions of the queues.
- [FORMAT](#)
Use this function to format IBM MQ for z/OS page sets.
- [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#)

Use this function to restore the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set from a sequential file created by the COPY function.

- [PAGEINFO](#)

Use this function to extract page set information from one or more page sets.

- [RESETPAGE](#)

Use this function to copy whole page sets to other page set data sets and reset the log information in the copy.

- [SCOPY](#)

Use this function to copy the contents of a queue to a data set while the queue manager is offline.

- [SDEFS](#)

Use this function to produce a set of define commands for objects while the queue manager is offline.

- [SLOAD](#)

Use this function to restore messages from the destination data set of an earlier COPY or SCOPY operation. SLOAD processes a single queue.

- [SWITCH](#)

Use this function to switch or query the transmission queue associated with cluster-sender channels.

Using the Command Facility on z/OS

Use the editor to enter or amend MQSC commands to be passed to the queue manager.

From the primary panel, CSQOPRIA, select option **8 Command**, to start the Command Facility.

You are presented with an edit session of a sequential file, *prefix*.CSQUTIL.COMMANDS, used as input to the CSQUTIL COMMAND function; see [Issuing commands to IBM MQ](#).

You do not need to prefix commands with the command prefix string (CPF).

You can continue MQSC commands on subsequent lines by terminating the current line with the continuation characters + or -. Alternatively, use line edit mode to provide long MQSC commands or the values of long attribute values within the command.

line edit

To use line edit, move the cursor to the appropriate line in the edit panel and use **F4** to display a single line in a scrollable panel. A single line can be up to 32 760 bytes of data.

To leave line edit:

- **F3 exit** saves changes made to the line and exits
- **F12 cancel** returns to the edit panel discarding changes made to the line.

To discard changes made in the edit session, use **F12 cancel** to terminate the edit session leaving the contents of the file unchanged. Commands are not executed.

Executing commands

When you have finished entering MQSC commands, terminate the edit session with **F3 exit** to save the contents of the file and invoke CSQUTIL to pass the commands to the queue manager. The output from command processing is held in file *prefix*.CSQUTIL.OUTPUT. An edit session opens automatically on this file so that you can view the responses. Press **F3 exit** to exit this session and return to the main menu.

Working with IBM MQ objects on z/OS

Many of the tasks described in this documentation involve manipulating IBM MQ objects. The object types are queue managers, queues, process definitions, namelists, channels, client connection channels, listeners, services, and authentication information objects.

- [Defining simple queue objects](#)
- [Defining other types of objects](#)
- [Working with object definitions](#)
- [Working with namelists](#)

Defining simple queue objects

To define a new object, use an existing definition as the basis for it. You can do this in one of three ways:

- By selecting an object that is a member of a list displayed as a result of options selected on the initial panel. You then enter action type 2 (**Define like**) in the action field next to the selected object. Your new object has the attributes of the selected object, except the disposition. You can then change any attributes in your new object as you require.
- On the initial panel, select the **Define like** action type, enter the type of object that you are defining in the **Object type** field, and enter the name of a specific existing object in the **Name** field. Your new object has the same attributes as the object you named in the **Name** field, except the disposition. You can then change any attributes in your new object definition as you require.
- By selecting the **Define like** action type, specifying an object type and then leaving the **Name** field blank. You can then define your new object and it has the default attributes defined for your installation. You can then change any attributes in your new object definition as you require.

Note: You do not enter the name of the object you are defining on the initial panel, but on the **Define** panel you are presented with.

The following example demonstrates how to define a local queue using an existing queue as a template.

Defining a local queue

To define a local queue object from the operations and control panels, use an existing queue definition as the basis for your new definition. There are several panels to complete. When you have completed all the panels and you are satisfied that the attributes are correct, press Enter to send your definition to the queue manager, which then creates the actual queue.

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

Action	2 (Define like)
Object type	QLOCAL
Name	QUEUE.YOU.LIKE. This is the name of the queue that provides the attributes for your new queue.

Press Enter to display the **Define a Local Queue** panel. The queue name field is blank so that you can supply the name for the new queue. The description is that of the queue upon which you are basing this new definition. Over type this field with your own description for the new queue.

The values in the other fields are those of the queue upon which you are basing this new queue, except the disposition. You can over type these fields as you require. For example, type Y in the **Put enabled** field (if it is not already Y) if suitably authorized applications can put messages on this queue.

You get field help by moving the cursor into a field and pressing function key F1. Field help provides information about the values that can be used for each attribute.

When you have completed the first panel, press function key F8 to display the second panel.

Hints:

1. Do not press Enter at this stage, otherwise the queue will be created before you have a chance to complete the remaining fields. (If you do press Enter prematurely, do not worry; you can always alter your definition later on.)
2. Do not press function keys F3 or F12, or the data you typed will be lost.

Press function key F8 repeatedly to see and complete the remaining panels, including the trigger definition, event control, and backout reporting panels.

When your local queue definition is complete

When your definition is complete, press Enter to send the information to the queue manager for processing. The queue manager creates the queue according to the definition you have supplied. If you do not want the queue to be created, press function key F3 to exit and cancel the definition.

Defining other types of objects

To define other types of object, use an existing definition as the base for your new definition as explained in [Defining a local queue](#).

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

Action	2 (Define like)
Object type	QALIAS, NAMELIST, PROCESS, CHANNEL, and other resource objects.
Name	Leave blank or enter the name of an existing object of the same type.

Press Enter to display the corresponding DEFINE panels. Complete the fields as required and then press Enter again to send the information to the queue manager.

Like defining a local queue, defining another type of object generally requires several panels to be completed. Defining a namelist requires some additional work, as described in [“Working with namelists”](#) on page 442.

Working with object definitions

When an object has been defined, you can specify an action in the **Action** field, to alter, display, or manage it.

In each case, you can either:

- Select the object you want to work with from a list displayed as a result of options selected on the initial panel. For example, having entered 1 in the **Action** field to display objects, Queue in the **Object type** field, and * in the **Name** field, you are presented with a list of all queues defined in the system. You can then select from this list the queue with which you need to work.
- Start from the initial panel, where you specify the object you are working with by completing the **Object type** and **Name** fields.

Altering an object definition

To alter an object definition, specify action 3 and press Enter to see the ALTER panels. These panels are very similar to the DEFINE panels. You can alter the values you want. When your changes are complete, press Enter to send the information to the queue manager.

Displaying an object definition

If you want to see the details of an object without being able to change them, specify action 1 and press Enter to see the DISPLAY panels. Again, these panels are similar to the DEFINE panels except that you cannot change any of the fields. Change the object name to display details of another object.

Deleting an object

To delete an object, specify action 4 (Manage) and the **Delete** action is one of the actions presented on the resulting menu. Select the **Delete** action.

You are asked to confirm your request. If you press function key F3 or F12, the request is canceled. If you press Enter, the request is confirmed and passed to the queue manager. The object you specified is then deleted.

Note: You cannot delete most types of channel object unless the channel initiator is started.

Working with namelists

When working with namelists, proceed as you would for other objects.

For the actions DEFINE LIKE or ALTER, press function key F11 to add names to the list or to change the names in the list. This involves working with the ISPF editor and all the normal ISPF edit commands are available. Enter each name in the namelist on a separate line.

When you use the ISPF editor in this way, the function key settings are the normal ISPF settings, and **not** those used by the other operations and control panels.

If you need to specify lowercase names in the list, specify CAPS(OFF) on the editor panel command line. When you do this, all the namelists that you edit in the future are in lowercase until you specify CAPS(ON).

When you have finished editing the namelist, press function key F3 to end the ISPF edit session. Then press Enter to send the changes to the queue manager.

Attention: If you do not press Enter at this stage but press function key F3 instead, you lose any updates that you have typed in.

Implementing the system using multiple cluster transmission queues

It makes no difference if the channel is used in a single cluster, or an overlapping cluster. When the channel is selected and started, the channel selects the transmission queue depending on the definitions.

Procedure

- If you are using the DEFCLXQ option, see [“Using the automatic definition of queues and switching” on page 442.](#)
- If you are using a staged approach, see [“Changing your cluster-sender channels using a phased approach” on page 443.](#)

Using the automatic definition of queues and switching

Use this option if you are planning on using the DEFCLXQ option. There will be a queue created for every channel, and every new channel.

Procedure

1. Review the definition of the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE and change the attributes if required.

This queue is defined in member SCSQPROC (csq4insx).

2. Create the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE model queue.
3. Apply security policies for this model queue, and the SYSTEM.CLUSTER.TRANSMIT.** queues.

For z/OS the channel initiator started task user ID needs:

- Control access to CLASS(MQADMIN) for

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelname
```

- Update access to CLASS(MQQUEUE) for

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelname
```

Changing your cluster-sender channels using a phased approach

Use this option if you are planning on using a staged approach. This process allows you to move to the new cluster-sender channels at various times to suit the needs of your enterprise.

Before you begin

- Identify your business applications, and which channels are used.
- For the queues you use, display the clusters they are in.
- Display the channels to show the connection names, the names of the remote queue managers, and which clusters the channel supports.

About this task

- Create a transmission queue. On z/OS you might want to consider which page set you use for the queue.
- Set up security policy for the queue.
- Change any queue monitoring to include this queue name.
- Decide which channels are to use this transmission queue. The channels should have a similar name, so generic characters ' * ' in the CLCHNAME identify the channel.
- When you are ready to use the new function, alter the transmission queue to specify the name of the channels to use this transmission queue. For example CLUSTER1.TOPARIS, or CLUSTER1.* or *.TOPARIS
- Start the channels

Procedure

1. Use the DIS CLUSQMGR(xxxx) XMITQ command to display the cluster sender channels defined in the cluster, where xxxx is the name of the remote queue manager.
2. Set up the security profile for the transmission queue and give the queue access to the channel initiator.
3. Define the transmission queue to be used, and specify USAGE(XMITQ) INDXTYPE(CORRELID) SHARE and CLCHNAME(value)

The channel initiator started task user ID needs the following access:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel1
update class(MQQUEUE ssid.SYSTEM.CLUSTER.TRANSMIT.channel1
```

and the user ID using the SWITCH command needs the following access:

```
alter cl(MQADMIN) ssid.QUEUE.queueName
```

4. Stop and restart the channels.

The channel change occurs when the channel starts using an MQSC command, or you use CSQUTIL. You can identify which channels need to be restarted using the SWITCH CHANNEL(*) STATUS of CSQUTIL

If you have problems when the channel is started, stop the channel, resolve the problems, and restart the channel.

Note that you can change the CLCHNAME attribute as often as you need to.

The value of CLCHNAME used is the one when the channel is started, so you can change the CLCHNAME definition while the channel continues to use the definitions from the time that it started. The channel uses the new definition when it is restarted.

Undoing a change to a transmission queue on z/OS

You need to have a process to backout a change if the results are not as you expect.

What can go wrong?

If the new transmission queue is not what you expect:

1. Check the CLCHNAME is as you expect
2. Review the job log to check if the switch process has finished. If not, wait and check the new transmission queue of the channel later.

If you are using multiple cluster transmission queues, it is important that you design the transmission queues definitions explicitly and avoid complicated overlapping configuration. In this way, you can make sure that if there are problems, you can go back to the original queues and configuration.

If you encounter problems during the move to using a different transmission queue, you must resolve any problems before you can proceed with the change.

An existing change request must complete before a new change request can be made. For example, you:

1. Define a new transmission queue with a maximum depth of one and there are 10 messages waiting to be sent.
2. Change the transmission queue to specify the channel name in the CLCHNAME parameter.
3. Stop and restart the channel. The attempt to move the messages fails and reports the problems.
4. Change the CLCHNAME parameter on the transmission queue to be blank.
5. Stop and restart the channel. The channel continues to try and complete the original request, so the channel continues to use the new transmission queue.
6. Need to resolve the problems and restart the channel so the moving of messages completes successfully.

Next time the channel is restarted it picks up any changes, so if you had set CLCHNAME to blanks, the channel will not use the specified transmission queue.

In this example, changing the CLCHNAME on the transmission queue to blanks does not necessarily mean that the channel uses the SYSTEM.CLUSTER.TRANSMIT queue, as there might be other transmission queues whose CLCHNAME parameter match the channel name. For example, a generic name, or the queue manager attribute DEFCLXQ might be set to channel, so the channel uses a dynamic queue instead of the SYSTEM.CLUSTER.TRANSMIT queue.

Writing programs to administer IBM MQ for z/OS

You can write your own application programs to administer a queue manager. Use this topic to understand the requirements for writing your own administration programs.

Start of General-use programming interface information

This set of topics contains hints and guidance to enable you to issue IBM MQ commands from an IBM MQ application program.

Note: In this topic, the MQI calls are described using C-language notation. For typical invocations of the calls in the COBOL, PL/I, and assembler languages, see [Function calls](#).

Understanding how it all works

In outline, the procedure for issuing commands from an application program is as follows:

1. Build an IBM MQ command into a type of IBM MQ message called a *request message*. The command can be in MQSC or PCF format.
2. Send (use MQPUT) this message to a special queue called the system-command input queue. The IBM MQ command processor runs the command.
3. Retrieve (use MQGET) the results of the command as *reply messages* on the reply-to queue. These messages contain the user messages that you need to determine whether your command was successful and, if it was, what the results were.

Then it is up to your application program to process the results.

This set of topics contains:

Preparing queues for administration programs

Administration programs require a number of predefined queues for system command input and receiving responses.

This section applies to commands in the MQSC format. For the equivalent in PCF, see [“IBM MQ PCF\(Programmable Command Format\) 사용”](#) on page 25.

Before you can issue any MQPUT or MQGET calls, you must first define, and then open, the queues you are going to use.

Defining the system-command input queue

The system-command input queue is a local queue called SYSTEM.COMMAND.INPUT. The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a default definition for the system-command input queue. For compatibility with IBM MQ on other platforms, an alias of this queue, called SYSTEM.ADMIN.COMMAND.QUEUE is also supplied. See [Sample definitions supplied with IBM MQ](#) for more information.

Defining a reply-to queue

You must define a reply-to queue to receive reply messages from the IBM MQ command processor. It can be any queue with attributes that allow reply messages to be put on it. However, for normal operation, specify these attributes:

- USAGE(NORMAL)
- NOTRIGGER (unless your application uses triggering)

Avoid using persistent messages for commands, but if you choose to do so, the reply-to queue must not be a temporary dynamic queue.

The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a definition for a model queue called SYSTEM.COMMAND.REPLY.MODEL. You can use this model to create a dynamic reply-to queue.

Note: Replies generated by the command processor can be up to 15 000 bytes in length.

If you use a permanent dynamic queue as a reply-to queue, your application should allow time for all PUT and GET operations to complete before attempting to delete the queue, otherwise MQRC2055 (MQRC_Q_NOT_EMPTY) can be returned. If this occurs, try the queue deletion again after a few seconds.

Opening the system-command input queue

Before you can open the system-command input queue, your application program must be connected to your queue manager. Use the MQI call MQCONN or MQCONNX to do this.

Then use the MQI call MQOPEN to open the system-command input queue. To use this call:

1. Set the **Options** parameter to MQOO_OUTPUT
2. Set the MQOD object descriptor fields as follows:

ObjectType

MQOT_Q (the object is a queue)

ObjectName

SYSTEM.COMMAND.INPUT

ObjectQMgrName

If you want to send your request messages to your local queue manager, leave this field blank. This means that your commands are processed locally.

If you want your IBM MQ commands to be processed on a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

Opening a reply-to queue

To retrieve the replies from an IBM MQ command, you must open a reply-to queue. One way of doing this is to specify the model queue, SYSTEM.COMMAND.REPLY.MODEL in an MQOPEN call, to create a permanent dynamic queue as the reply-to queue. To use this call:

1. Set the **Options** parameter to MQOO_INPUT_SHARED
2. Set the MQOD object descriptor fields as follows:

ObjectType

MQOT_Q (the object is a queue)

ObjectName

The name of the reply-to queue. If the queue name you specify is the name of a model queue object, the queue manager creates a dynamic queue.

ObjectQMgrName

To receive replies on your local queue manager, leave this field blank.

DynamicQName

Specify the name of the dynamic queue to be created.

Using the command server

The command server is an IBM MQ component that works with the command processor component. You can send formatted messages to the command server which interprets the messages, runs the administration requests, and sends responses back to your administration application.

The command server reads request messages from the system-command input queue, verifies them, and passes the valid ones as commands to the command processor. The command processor processes the commands and puts any replies as reply messages on to the reply-to queue that you specify. The first reply message contains the user message CSQN205I. See “[Interpreting the reply messages from the command server](#)” on page 450 for more information. The command server also processes channel initiator and queue sharing group commands, wherever they are issued from.

Identifying the queue manager that processes your commands

The queue manager that processes the commands you issue from an administration program is the queue manager that owns the system-command input queue that the message is put onto.

Starting the command server

Normally, the command server is started automatically when the queue manager is started. It becomes available as soon as the message CSQ9022I 'START QMGR' NORMAL COMPLETION is

returned from the START QMGR command. The command server is stopped when all the connected tasks have been disconnected during the system termination phase.

You can control the command server yourself using the START CMDSERV and STOP CMDSERV commands. To prevent the command server starting automatically when IBM MQ is restarted, you can add a STOP CMDSERV command to your CSQINP1 or CSQINP2 initialization data sets. However, this is not recommended as it prevents any channel initiator or queue sharing group commands being processed.

The STOP CMDSERV command stops the command server as soon as it has finished processing the current message, or immediately if no messages are being processed.

If the command server has been stopped by a STOP CMDSERV command in the program, no other commands from the program can be processed. To restart the command server, you must issue a START CMDSERV command from the z/OS console.

If you stop and restart the command server while the queue manager is running, all the messages that are on the system-command input queue when the command server stops are processed when the command server is restarted. However, if you stop and restart the queue manager after the command server is stopped, only the persistent messages on the system-command input queue are processed when the command server is restarted. All nonpersistent messages on the system-command input queue are lost.

Sending commands to the command server

For each command, you build a message containing the command, then put it onto the system-command input queue.

Building a message that includes IBM MQ commands

You can incorporate IBM MQ commands in an application program by building request messages that include the required commands. For each such command you:

1. Create a buffer containing a character string representing the command.
2. Issue an MQPUT call specifying the buffer name in the **buffer** parameter of the call.

The simplest way to do this in C is to define a buffer using 'char'. For example:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

When you build a command, use a null-terminated character string. Do not specify a command prefix string (CPF) at the start of a command defined in this way. This means that you do not have to alter your command scripts if you want to run them on another queue manager. However, you must take into account that a CPF is included in any response messages that are put onto the reply-to queue.

The command server folds all lowercase characters to uppercase unless they are inside quotation marks.

Commands can be any length up to a maximum 32 762 characters.

Putting messages on the system-command input queue

Use the MQPUT call to put request messages containing commands on the system-command input queue. In this call you specify the name of the reply-to queue that you have already opened.

To use the MQPUT call:

1. Set these MQPUT parameters:

Hconn

The connection handle returned by the MQCONN or MQCONNX call.

Hobj

The object handle returned by the MQOPEN call for the system-command input queue.

BufferLength

The length of the formatted command.

Buffer

The name of the buffer containing the command.

2. Set these MQMD fields:

MsgType

MQMT_REQUEST

Format

MQFMT_STRING or MQFMT_NONE

If you are not using the same code page as the queue manager, set *CodedCharSetId* as appropriate and set MQFMT_STRING, so that the command server can convert the message. Do not set MQFMT_ADMIN, as that causes your command to be interpreted as PCF.

ReplyToQ

Name of your reply-to queue.

ReplyToQMgr

If you want replies sent to your local queue manager, leave this field blank. If you want your IBM MQ commands to be sent to a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

3. Set any other MQMD fields, as required. You should normally use nonpersistent messages for commands.
4. Set any *PutMsgOpts* options, as required.

If you specify MQPMO_SYNCPOINT (the default), you must follow the MQPUT call with a syncpoint call.

Using MQPUT1 and the system-command input queue

If you want to put just one message on the system-command input queue, you can use the **MQPUT1** call. This call combines the functions of an **MQOPEN**, followed by an **MQPUT** of one message, followed by an **MQCLOSE**, all in one call. If you use this call, modify the parameters accordingly. See [Putting one message on a queue using the MQPUT1 call](#) for details.

Retrieving replies to your commands

The command server sends a response to a reply queue for each request message it receives. Any administration application must receive, and handle the reply messages.

When the command processor processes your commands, any reply messages are put onto the reply-to queue specified in the MQPUT call. The command server sends the reply messages with the same persistence as the command message it received.

Waiting for a reply

Use the MQGET call to retrieve a reply from your request message. One request message can produce several reply messages. For details, see [“Interpreting the reply messages from the command server” on page 450](#).

You can specify a time interval that an MQGET call waits for a reply message to be generated. If you do not get a reply, use the checklist beginning in topic [“If you do not receive a reply” on page 451](#).

To use the MQGET call:

1. Set these parameters:

Hconn

The connection handle returned by the MQCONN or MQCONNX call.

Hobj

The object handle returned by the MQOPEN call for the reply-to queue.

Buffer

The name of the area to receive the reply.

BufferLength

The length of the buffer to receive the reply. This must be a minimum of 80 bytes.

- To ensure that you only get the responses from the command that you issued, you must specify the appropriate *MsgId* and *CorrelId* fields. These depend on the report options, MQMD_REPORT, you specified in the MQPUT call:

MQRO_NONE

Binary zero, '00...00' (24 nulls).

MQRO_NEW_MSG_ID

Binary zero, '00...00' (24 nulls).

This is the default if none of these options has been specified.

MQRO_PASS_MSG_ID

The *MsgId* from the MQPUT.

MQRO_NONE

The *MsgId* from the MQPUT call.

MQRO_COPY_MSG_ID_TO_CORREL_ID

The *MsgId* from the MQPUT call.

This is the default if none of these options has been specified.

MQRO_PASS_CORREL_ID

The *CorrelId* from the MQPUT call.

For more details on report options, see [Report options and message flags](#).

- Set the following *GetMsgOpts* fields:

Options

MQGMO_WAIT

If you are not using the same code page as the queue manager, set MQGMO_CONVERT, and set *CodedCharSetId* as appropriate in the MQMD.

WaitInterval

For replies from the local queue manager, try 5 seconds. Coded in milliseconds, this becomes 5 000. For replies from a remote queue manager, and channel control and status commands, try 30 seconds. Coded in milliseconds, this becomes 30 000.

Discarded messages

If the command server finds that a request message is not valid, it discards this message and writes the message [CSQN205I](#) to the named reply-to queue. If there is no reply-to queue, the CSQN205I message is put onto the dead-letter queue. The return code in this message shows why the original request message was not valid:

00D5020F It is not of type MQMT_REQUEST.

00D50210 It has zero length.

00D50212 It is longer than 32 762 bytes.

00D50211 It contains all blanks.

00D5483E It needed converting, but *Format* was not MQFMT_STRING.

Other See [Command server codes](#)

The command server reply message descriptor

For any reply message, the following MQMD message descriptor fields are set:

<i>MsgType</i>	MQMT_REPLY
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>Priority</i>	As for the MQMD in the message you issued.
<i>Persistence</i>	As for the MQMD in the message you issued.
<i>CorrelId</i>	Depends on the MQPUT report options.
<i>ReplyToQ</i>	None.

The command server sets the *Options* field of the MQPMO structure to MQPMO_NO_SYNCPOINT. This means that you can retrieve the replies as they are created, rather than as a group at the next syncpoint.

Interpreting the reply messages from the command server

Each request message correctly processed by IBM MQ produces at least two reply messages. Each reply message contains a single IBM MQ user message.

The length of a reply depends on the command that was issued. The longest reply you can get is from a **DISPLAY NAMELIST** command, and that can be up to 15 000 bytes in length.

The first user message, CSQN205I, always contains:

- A count of the replies (in decimal), which you can use as a counter in a loop to get the rest of the replies. The count includes this first message.
- The return code from the command preprocessor.
- A reason code, which is the reason code from the command processor.

This message does not contain a CPF.

For example:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
```

The COUNT field is 8 bytes long and is right-justified. It always starts at position 18, that is, immediately after COUNT=. The RETURN field is 8 bytes long in character hexadecimal and is immediately after RETURN= at position 35. The REASON field is 8 bytes long in character hexadecimal and is immediately after REASON= at position 52.

If the RETURN= value is 00000000 and the REASON= value is 00000004, the set of reply messages is incomplete. After retrieving the replies indicated by the CSQN205I message, issue a further MQGET call to wait for a further set of replies. The first message in the next set of replies is again CSQN205I, indicating how many replies there are, and whether there are still more to come.

See the [IBM MQ for z/OS 메시지, 완료 및 이유 코드](#) documentation for more details about the individual messages.

If you are using a non-English language feature, the text and layout of the replies are different from those shown here. However, the size and position of the count and return codes in message CSQN205I are the same.

If you do not receive a reply

There are a series of steps you can take if you do not receive a response to request to the command server.

If you do not receive a reply to your request message, work through this checklist:

- Is the command server running?
- Is the *WaitInterval* long enough?
- Are the system-command input and reply-to queues correctly defined?
- Were the MQOPEN calls to these queues successful?
- Are both the system-command input and reply-to queues enabled for MQPUT and MQGET calls?
- Have you considered increasing the MAXDEPTH and MAXMSGL attributes of your queues?
- Are you are using the *CorrelId* and *MsgId* fields correctly?
- Is the queue manager still running?
- Was the command built correctly?
- Are all your remote links defined and operating correctly?
- Were the MQPUT calls correctly defined?
- Has the reply-to queue been defined as a temporary dynamic queue instead of a permanent dynamic queue? (If the request message is persistent, you must use a permanent dynamic queue for the reply.)

When the command server generates replies but cannot write them to the reply-to queue that you specify, it writes them to the dead-letter queue.

Passing commands using MGCRE

With appropriate authorization, an application program can make requests to multiple queue managers using a z/OS service routine.

If you have the correct authorization, you can pass IBM MQ commands from your program to multiple queue managers by the MGCRE (SVC 34) z/OS service. See the [z/OS MVS Programming: Authorized Assembler Services Guide](#) for more information.

The value of the CPF identifies the particular queue manager to which the command is directed. For information about CPFs, see [User IDs for command security and command resource security](#) and [“Issuing queue manager commands on z/OS”](#) on page 414.

If you use MGCRE, you can use a Command and Response Token (CART) to get the direct responses to the command.

Examples of commands and their replies

Use this topic as a series of examples of commands to the command server and the responses from the command server.

Here are some examples of commands that could be built into IBM MQ messages, and the user messages that are the replies. Unless otherwise stated, each line of the reply is a separate message.

- [Messages from a DEFINE command](#)
- [Messages from a DELETE command](#)
- [Messages from DISPLAY commands](#)
- [Messages from commands with CMDSCOPE](#)
- [Messages from commands that generate commands with CMDSCOPE](#)

Messages from a DEFINE command

The following command:

```
DEFINE QLOCAL(Q1)
```

produces these messages:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000  
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

These reply messages are produced on normal completion.

Messages from a DELETE command

The following command:

```
DELETE QLOCAL(Q2)
```

produces these messages:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000008  
CSQM125I +CSQ1 CSQMUQLC QLOCAL(Q2) QSGDISP(QMGR) WAS NOT FOUND  
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D44002'  
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

These messages indicate that a local queue called Q2 does not exist.

Messages from DISPLAY commands

The following examples show the replies from some DISPLAY commands.

Finding out the name of the dead-letter queue

If you want to find out the name of the dead-letter queue for a queue manager, issue this command from an application program:

```
DISPLAY QMGR DEADQ
```

The following three user messages are returned, from which you can extract the required name:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000  
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE )  
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION
```

Messages from the DISPLAY QUEUE command

The following examples show how the results from a command depend on the attributes specified in that command.

Example 1

You define a local queue using the command:

```
DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE
```

If you issue the following command from an application program:

```
DISPLAY QUEUE(Q1) SHARE GET DESCR
```

these three user messages are returned:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1)                                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Note: The second message, CSQM401I, is shown here occupying four lines.

Example 2

Two queues have names beginning with the letter A:

- A1 is a local queue with its PUT attribute set to DISABLED.
- A2 is a remote queue with its PUT attribute set to ENABLED.

If you issue the following command from an application program:

```
DISPLAY QUEUE(A*) PUT
```

these four user messages are returned:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1)                                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2)                                ) TYPE(
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Note: The second and third messages, CSQM401I and CSQM406I, are shown here occupying three and two lines.

Messages from the DISPLAY NAMELIST command

You define a namelist using the command:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```


If you issue the following command from an application program:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

the following three user messages are returned:

```
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1
GDISP(QMGR  ) NAMCOUNT(    2) NAMES(Q1
,SAMPLE_QUEUE
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY NAMELIST' NORMAL COMPLETION
```

Note: The second message, CSQM407I, is shown here occupying three lines.

Messages from commands with CMDSCOPE

The following examples show the replies from commands that have been entered with the CMDSCOPE attribute.

Messages from the ALTER PROCESS command

The following command:

```
ALT PRO(V4) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'ALT PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I  COUNT=    5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ26
CSQM125I !MQ26 CSQMMSGP PROCESS(V4) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E !MQ26 CSQMMSGP FAILURE REASON CODE X'00D44002'
CSQ9023E !MQ26 CSQMMSGP ' ALT PRO' ABNORMAL COMPLETION
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' ALT PRO' NORMAL COMPLETION
CSQN205I  COUNT=    2, RETURN=0000000C, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). The command was successful on MQ25 but the process definition did not exist on MQ26, so the command failed on that queue manager.

Messages from the DISPLAY PROCESS command

The following command:

```
DIS PRO(V*) CMDSCOPE(*)
```

produces the following messages:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ26
CSQM408I !MQ26 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ26 PROCESS(V3) QSGDISP(QMGR)
CSQ9022I !MQ26 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 7, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ25
CSQM408I !MQ25 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ25 PROCESS(V2) QSGDISP(GROUP)
CSQM408I !MQ25 PROCESS(V3) QSGDISP(QMGR)
CSQM408I !MQ25 PROCESS(V4) QSGDISP(QMGR)
CSQ9022I !MQ25 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS PRO' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about all the processes on each queue manager with names starting with the letter V.

Messages from the DISPLAY CHSTATUS command

The following command:

```
DIS CHS(VT) CMDSCOPE(*)
```

produces the following messages:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS CHS' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ25
CSQM422I !MQ25 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ25 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ26
CSQM422I !MQ26 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ26 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS CHS' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about channel status on each queue manager.

Messages from the STOP CHANNEL command

The following command:

```
STOP CHL(VT) CMDSCOPE(*)
```

produces these messages:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'STOP CHL' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQM134I !MQ25 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Channel VT was stopped on each queue manager.

Messages from commands that generate commands with CMDSCOPE

The following command:

```
DEF PRO(V2) QSGDISP(GROUP)
```

produces these messages:

```

CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQM122I !MQ25 CSQMMSGP ' DEF PRO' COMPLETED FOR QSGDISP(GROUP)
CSQN138I !MQ25 'DEFINE PRO' command generated for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ26
CSQ9022I !MQ26 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DEFINE PRO' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25. When the object was created on the shared repository, another command was generated and sent to all the active queue managers in the queue sharing group (MQ25 and MQ26).

z/OS

Managing IBM MQ resources on z/OS

Use the links in this topic to find out how to manage the resources used by IBM MQ for z/OS, for example, managing log files, data sets, page sets, buffer pools, and coupling facility structures.

Use the following links for details of the different administrative tasks you might have to complete while using IBM MQ for z/OS:

- [“Managing the logs” on page 457](#)
- [“Managing the bootstrap data set \(BSDS\)” on page 466](#)
- [“Managing page sets” on page 473](#)
- [“How to back up and recover page sets” on page 480](#)
- [“How to back up and restore queues using CSQUTIL” on page 483](#)
- [“Managing buffer pools” on page 483](#)

- [“Managing queue sharing groups and shared queues on z/OS” on page 485](#)

Related concepts

[IBM MQ for z/OS concepts](#)

[“Administering IBM MQ for z/OS” on page 413](#)

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

[“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS” on page 414](#)

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

[“Recovery and restart on z/OS” on page 494](#)

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

Related tasks

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

[IBM MQ utilities on z/OS reference](#)

Related reference

[“Using the IBM MQ for z/OS utilities” on page 436](#)

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

[Programmable command formats reference](#)

Managing the logs

Use this topic to understand how to manage your IBM MQ log files, including the log archiving process, using log record compression, log record recovery, and printing log records.

This topic describes the tasks involved in managing the IBM MQ logs. It contains these sections:

Archiving logs with the ARCHIVE LOG command

An authorized operator can archive the current IBM MQ active log data sets whenever required using the **ARCHIVE LOG** command.

When you issue the **ARCHIVE LOG** command, IBM MQ truncates the current active log data sets, then runs an asynchronous offload process, and updates the BSDS with a record of the offload process.

The **ARCHIVE LOG** command has a **MODE(QUIESCE)** option. With this option, IBM MQ jobs and users are quiesced after a commit point, and the resulting point of consistency is captured in the current active log before it is offloaded.

Consider using the **MODE(QUIESCE)** option when planning a backup strategy for off site recovery. It creates a system-wide point of consistency, which minimizes the number of data inconsistencies when the archive log is used with the most current backup page set copy during recovery. For example:

```
ARCHIVE LOG MODE(QUIESCE)
```

If you issue the **ARCHIVE LOG** command without specifying a **TIME** parameter, the quiesce time period defaults to the value of the **QUIESCE** parameter of the CSQ6ARVP macro. If the time required for the **ARCHIVE LOG MODE(QUIESCE)** to complete is less than the time specified, the command completes successfully; otherwise, the command fails when the time period expires. You can specify the time period explicitly by using the **TIME** option, for example:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

This command specifies a quiesce period of up to 60 seconds before **ARCHIVE LOG** processing occurs.

Attention: Using the **TIME** option when time is critical can significantly disrupt IBM MQ availability for all jobs and users that use IBM MQ resources.

By default, the command is processed asynchronously from the time you submit the command. (To process the command synchronously with other IBM MQ commands use the **WAIT (YES)** option with **QUIESCE**, but be aware that the z/OS console is locked from IBM MQ command input for the entire **QUIESCE** period.)

During the quiesce period:

- Jobs and users on the queue manager are allowed to go through commit processing, but are suspended if they try to update any IBM MQ resource after the commit.
- Jobs and users that only read data can be affected, since they might be waiting for locks held by jobs or users that were suspended.
- New tasks can start, but they cannot update data.

The output from the **DISPLAY LOG** command uses the message CSQV400I to indicate that a quiesce is in effect.

For example:

```
CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter      Initial value      SET value
-----
INBUFF         60
OUTBUFF       400
MAXRTU         2
MAXARCH        2
TWOACTV       YES
TWOARCH       YES
TWOBSDS       YES
OFFLOAD       YES
MAXCNOFF      0
WRTHRSH       20
DEALLCT        0
COMPLG        NONE
ZHYWRITE      NO
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full zHyperWrite Encrypted DSName
  1    68 NO          NO      VICY.CSQ1.LOGCOPY1.DS01
  2    68 NO          NO      VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2019-08-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION
```

When all updates are quiesced, the quiesce history record in the BSDS is updated with the date and time that the active log data sets were truncated, and with the last-written RBA in the current active log data sets. IBM MQ truncates the current active log data sets, switches to the next available active log data sets, and issues message CSQJ311I stating that the offload process started.

If updates cannot be quiesced before the quiesce period expires, IBM MQ issues message CSQJ317I, and **ARCHIVE LOG** processing terminates. The current active log data sets are not truncated, nor switched to the next available log data sets, and the offload process is not started.

Whether the quiesce was successful or not, all suspended users and jobs are then resumed, and IBM MQ issues message CSQJ312I, stating that the quiesce is ended and update activity is resumed.

If **ARCHIVE LOG** is issued when the current active log is the last available active log data set, the command is not processed, and IBM MQ issues the following message:

```
CSQJ319I - csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING
WILL BE TERMINATED
```

If **ARCHIVE LOG** is issued when another **ARCHIVE LOG** command is already in progress, the new command is not processed, and IBM MQ issues the following message:

```
CSQJ318I - ARCHIVE LOG COMMAND ALREADY IN PROGRESS
```

For information about the messages issued during archiving, see [Messages for IBM MQ for z/OS](#).

Restarting the log archive process after a failure

If there is a problem during the log archive process (for example, a problem with allocation or tape mounts), the archiving of the active log might be suspended. You can cancel the archive process and restart it by using the following command:

```
ARCHIVE LOG CANCEL OFFLOAD
```

This command cancels any offload processing currently in progress, and restarts the archive process. It starts with the oldest log data set that has not been archived, and proceeds through all active log data sets that need offloading. Any log archive operations that have been suspended are restarted.

Use this command only if you are sure that the current log archive task is no longer functioning, or if you want to restart a previous attempt that failed. This is because the command might cause an abnormal termination of the offload task, which might result in a dump.

Controlling archiving and logging

You can control compression, printing, archiving, recovery and logging with using the CSQ6LOGP, CSQ6ARVP, and CSQ6SYSP macros. Note, that changes to private objects only are logged in IBM MQlogs. Changes to GROUP objects (like shared inbound channels) are also logged, because the definitions are propagated around the group and held locally.

Many aspects of archiving and logging are controlled by parameters set using the CSQ6LOGP, CSQ6ARVP and CSQ6SYSP macros of the system parameter module when the queue manager is customized. See [Tailor your system parameter module](#) for details of these macros.

Some of these parameters can be changed while a queue manager is running using the IBM MQ MQSC SET LOG, SET SYSTEM and SET ARCHIVE commands. They are shown in [Table 28 on page 459](#):

SET command	Parameters
LOG	WRTHRSH, MAXARCH, DEALLCT, MAXRTU, COMPLOG
ARCHIVE	All
SYSTEM	LOGLOAD

You can display the settings of all the parameters using the MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) and [DISPLAY SYSTEM](#) commands. These commands also show status information about archiving and logging.

Controlling log compression

You can enable and disable the compression of log records using either

- The SET and DISPLAY LOG commands in MQSC; see [The MQSC commands](#)
- Invoking PCF interface. See [“IBM MQ PCF\(Programmable Command Format\) 소개” on page 24](#)
- Using the CSQ6LOGP macro in the system parameter module; see [Using CSQ6LOGP](#)

Printing log records

You can extract and print log records using the CSQ1LOGP utility. For instructions, see [The log print utility](#).

Recovering logs

Normally, you do not need to back up and restore the IBM MQ logs, especially if you are using dual logging. However, in rare circumstances, such as an I/O error on a log, you might need to recover the logs. Use Access Method Services to delete and redefine the data set, and then copy the corresponding dual log into it.

Discarding archive log data sets

You can discard your archive log data sets and choose to discard the logs automatically or manually.

You must keep enough log data to be able to perform unit of work recovery, page set media recovery if a page set is lost, or CF structure media recovery if a CF structure is lost. Do not discard archive log data sets that might be required for recovery; if you discard these archive log data sets you might not be able to perform required recovery operations.

If you have confirmed that your archive log data sets can be discarded, you can do this in either of the following ways:

- [Automatic archive log data set deletion](#)
- [Manually deleting archive log data sets](#)

Automatic archive log data set deletion

You can use a DASD or tape management system to delete archive log data sets automatically. The retention period for IBM MQ archive log data sets is specified by the retention period field ARCRETN in the CSQ6ARVP installation macro (see the [Using CSQ6ARVP](#) for more information).

The default for the retention period specifies that archive logs are to be kept for 9999 days (the maximum).

Important: You can change the retention period but you must ensure that you can accommodate the number of backup cycles that you have planned for.

.

IBM MQ uses the retention period value as the value for the JCL parameter RETPD when archive log data sets are created.

The retention period set by the MVS™/DFP storage management subsystem (SMS) can be overridden by this IBM MQ parameter. Typically, the retention period is set to the smaller value specified by either IBM MQ or SMS. The storage administrator and IBM MQ administrator must agree on a retention period value that is appropriate for IBM MQ.

Note: IBM MQ does not have an automated method to delete information about archive log data sets from the BSDS, because some tape management systems provide external manual overrides of retention periods. Therefore, information about an archive log data set can still be in the BSDS long after the data set retention period has expired and the data set has been scratched by the tape management system. Conversely, the maximum number of archive log data sets might have been exceeded and the data from the BSDS might have been dropped before the data set has reached its expiration date.

If archive log data sets are deleted automatically, remember that the operation does not update the list of archive logs in the BSDS. You can update the BSDS with the change log inventory utility, as described in [“Changing the BSDS” on page 467](#). The update is not essential. Recording old archive logs wastes space in the BSDS, but does no other harm.

Manually deleting archive log data sets

You must keep all the log records as far back as the lowest RBA identified in messages CSQI024I and CSQI025I. This RBA is obtained using the DISPLAY USAGE command that you issued when creating a point of recovery using [Method 1: Full backup](#).

Read [Creating a point of recovery for non-shared resources before discarding any logs](#).

Locate and discard archive log data sets

Having established the minimum log RBA required for recovery, you can find archive log data sets that contain only earlier log records by performing the following procedure:

1. Use the print log map utility to print the contents of the BSDS. For an example of the output, see [The print log map utility](#).
2. Find the sections of the output titled ARCHIVE LOG COPY n DATA SETS. If you use dual logging, there are two sections. The columns labeled STARTRBA and ENDRBA show the range of RBAs contained in each volume. Find the volumes with ranges that include the minimum RBA you found with messages CSQI024I and CSQI025I. These are the earliest volumes you need to keep. If you are using dual-logging, there are two such volumes.

If no volumes have an appropriate range, one of the following cases applies:

- The minimum RBA has not yet been archived, and you can discard all archive log volumes.
- The list of archive log volumes in the BSDS wrapped around when the number of volumes exceeded the number allowed by the MAXARCH parameter of the CSQ6LOGP macro. If the BSDS does not register an archive log volume, that volume cannot be used for recovery. Therefore, consider adding information about existing volumes to the BSDS. For instructions, see [“Changes for archive logs” on page 470](#).

Also consider increasing the value of MAXARCH. For information, see the [Using CSQ6LOGP](#).

3. Delete any archive log data set or volume with an ENDRBA value that is less than the STARTRBA value of the earliest volume you want to keep. If you are using dual logging, delete both such copies.

Because BSDS entries wrap around, the first few entries in the BSDS archive log section might be more recent than the entries at the end. Look at the combination of date and time and compare their ages. Do not assume that you can discard all entries before the entry for the archive log containing the minimum LOGRBA.

Delete the data sets. If the archives are on tape, erase the tapes. If they are on DASD, run a z/OS utility to delete each data set. Then, if you want the BSDS to list only existing archive volumes, use the change log inventory utility (CSQJU003) to delete entries for the discarded volumes. See [“Changes for archive logs” on page 470](#) for an example.

The effect of log shunting

Long running transactions can cause unit of work log records which span log data sets. IBM MQ handles this scenario by using log shunting, a technique which moves the log records to optimize the quantity of log data retained, and queue manager restart time.

When a unit of work is considered to be long, a representation of each log record is written further down the log. This is known as *log shunting*. It is described more fully in [Log files](#).

The queue manager uses these shunted log records instead of the originals after a failure, to ensure unit of work integrity. There are two benefits to this:

- the quantity of log data which must be retained for unit of work coordination is reduced
- less log data must be traversed at queue manager restart time, so the queue manager is restarted more quickly

Shunted log records do not contain sufficient information for media recovery operations.

Data held in the log is used for two distinct purposes; media recovery and unit of work coordination. If a media failure occurs which affects either a CF structure or page set, the queue manager can recover the media to the point of failure by restoring a prior copy and updating this using data contained in the log.

Persistent activity performed in a unit of work is recorded on the log so that in the event of a failure, it can either be backed out or locks can be recovered on changed resources. The quantity of log data you need to retain to enable queue manager recovery is affected by these two elements.

For media recovery, you must retain sufficient log data to be able to perform media recovery from at least the most recent media copy and to be able to back out. (Your site may stipulate the ability to recover from older backups.) For unit of work integrity, you must retain the log data for your oldest in flight or indoubt units of work.

To assist you with managing the system, the queue manager detects old units of work at each log archive and reports them in messages CSQJ160 and CSQJ161. An internal task reads unit of work log information for these old units of work and rewrites it in a more succinct form to the current position in the log. Message CSQR026 indicates when this has happened. The MQSC command DISPLAY USAGE TYPE(DATASET) can also help you to manage the retention of log data. The command reports the following three pieces of recovery information:

1. How much of the log must be retained for unit of work recovery.
2. How much of the log must be retained for media recovery of page sets.
3. For a queue manager in a queue sharing group, how much of the log must be retained for media recovery of CF structures.

For each of these pieces of information, an attempt is made to map the oldest log data required into a data set. As new units of work start and stop, (1) would be expected to move to a more recent position in the log. If it is not moving, the long running UOW messages warn you that there is an issue. (2) relates to page set media recovery if the queue manager were to be shut down now and restarted. It does not know about when you last backed up your page sets, or which backup you might have to use if there was a page set failure. It normally moves to a more recent position in the log during checkpoint processing as changes held in the buffer pools are written to the page sets. In (3), the queue manager does know about CF structure backups taken either on this queue manager or on other queue managers in the queue sharing group. However, CF structure recovery requires a merge of log data from all queue managers in the queue sharing group which have interacted with the CF structure since the last backup. This means that the log data is identified by a log record sequence number, (or LRSN), which is timestamp based and so applicable across the entire queue sharing group rather than an RBA which would be different on different queue managers in the queue sharing group. It normally moves to a more recent position in the log as BACKUP CFSTRUCT commands are performed on either this or other queue managers in the queue sharing group.

Resetting the queue manager's log

Use this topic to understand how to reset the queue manager's log.

You must not allow the queue manager log RBA to wrap around from the end of the log RBA range to 0, as this leads to a queue manager outage and all persistent data will become unrecoverable. The end of the log RBA is either a value of FFFFFFFFFF (if 6-byte RBAs as in use), or FFFFFFFFFFFFFFFF (if 8-byte RBAs are in use).

The queue manager issues messages [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031D](#), and [CSQJ032E](#) to indicate that the used log range is significant and that you should plan to take action to avoid an unplanned outage.

The queue manager terminates with reason code [00D10257](#) when the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFFC0000000000 (if 8-byte log RBAs are in use).

If 6-byte log RBAs are in use, consider converting the queue manager to use 8-byte log RBAs rather than resetting the queue manager's log, following the process described in [“Implementing the larger log Relative Byte Address”](#) on page 465. Converting a queue manager to use 8-byte log RBAs requires a shorter outage than resetting the log, and increases the period of time before you have to reset the log.

Message [CSQJ034I](#), issued during queue manager initialization, indicates the end of the log RBA range for the queue manager as configured, and can be used to determine whether 6-byte or 8-byte log RBAs are in use.

The procedure to follow to reset the queue manager's log is as follows:

1. Resolve any unresolved units of work. The number of unresolved units of work is displayed at queue manager startup in message CSQR005I as the INDOUBT count. At each checkpoint, and at queue manager shutdown, the queue manager automatically issues the command **DISPLAY CONN(*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED)** to provide information about unresolved units of work.

See [How in-doubt units of recovery are resolved](#) for information on resolving units of recovery. The ultimate recourse is to use the **RESOLVE INDOUBT** MQSC command to manually resolve indoubt units of recovery.
2. Shut down the queue manager cleanly.

You can use either **STOP QMGR** or **STOP QMGR MODE(FORCE)** as both these commands flush any changed pages from bufferpools to the page sets.
3. If a queue manager is part of a queue sharing group, take CFSTRUCT backups on other queue managers for all structures in the queue sharing group. This ensures that the most recent backups are not in this queue manager's log, and that this queue manager's log is not required for CFSTRUCT recovery.
4. Define new logs and BSDS using CSQJU003 (see [The change log inventory utility](#) for more information on using the change log inventory utility).
5. Run **CSQUTIL RESETPAGE** against all the page sets for this queue manager (see [Copying a page and resetting the log](#) for more information on using this function). Note that page set RBAs can be reset independently, so multiple concurrent jobs (for example, one per page set) can be submitted to reduce the elapsed time for this step.
6. Restart the queue manager

Warning messages

When IBM MQ detects that the end of the log is approaching, it issues console messages in the following order, which indicate that a log reset should be planned. In this section the messages show 6-byte log RBA values. The same console messages are issued when IBM MQ is running in 8-byte log RBA mode but with different values; see [“Warning thresholds” on page 464](#) for the 8-byte log RBA thresholds.

1. When IBM MQ detects that the end of the log is approaching in the near future, (approximately 94% full) IBM MQ issues console message CSQI045I, as in the following example:

```
CSQI045I -CSQ7 CSQILCUR Log RBA has reached 0000F00000000000.
Plan a log reset
```

2. IBM MQ issues the following CSQI046E error console message when the end of the log is near (approximately 97% full). This informs the IBM MQ administrator to take action soon.

```
CSQI046E -CSQ7 CSQILCUR Log RBA has reached 0000F80000000000.
Perform a log reset
```

3. After the CSQI046E message is issued, at the next log switch, IBM MQ issues the following CSQJ032E console message with the word WARNING:

```
CSQJ032E -CSQ7 CSQJW307 WARNING - APPROACHING END OF
THE LOG RBA RANGE OF 0000FFFFFFFFFFFF. CURRENT LOG RBA IS 0000F80000022000.
```

4. After the CSQI046E and CSQJ032E console messages are issued, IBM MQ issues one more error message, which does not require immediate IBM MQ administrator intervention. IBM MQ issues console message CSQI047E (when the log is approximately 99% full):

```
CSQI047E -CSQ7 CSQILCUR Log RBA has reached 0000FF0000000000.
Stop queue manager and reset logs
```

5. When the log RBA reaches FF8000000000, IBM MQ increases the urgency of the situation and issues console message CSQJ032E with the word CRITICAL:

```
CSQJ032E -CSQ7 CSQJW009 CRITICAL - APPROACHING END OF THE LOG RBA RANGE OF 0000FFFFFFFFFFFF.
CURRENT LOG RBA IS 0000FFF7FFFFDFFF.
```

6. If the queue manager is started when the log RBA is almost at the maximum, the following CSQJ031D console message is issued. This stage requires the input of the IBM MQ administrator:.

```
CSQJ031D -CSQ7 CSQYSTRT THE LOG RBA RANGE MUST BE RESET.
REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN
```

7. IBM MQ startup remains suspended until a reply is given to message CSQJ031D.

The purpose of these messages is to give the IBM MQ administrator time to plan for a system outage to reset the logs. In an ideal configuration, there are at least two queue managers, possibly in a queue sharing group (QSG), sharing the workload. When one is down for maintenance the other can continue to receive work.

The severity of console messages that IBM MQ issues becomes greater as the RBA gets closer to the end. Ideally your IBM MQ administrator should plan to reset the log RBA when the first console message is seen.

If the warning and error console messages are ignored, IBM MQ terminates with reason code 5C6-00D10257 when the log RBA reaches FFF800000000, at which point IBM MQ determines that the available range is too small for the queue manager to continue. When this point is reached, the only option is to take an outage and either reset the log or extend the size of the log RBA.

Note: When the end of the log is reached it is not possible to resolve any in-flight units of work (UOW); these are lost during the log reset process. Enough of the RBA range should be left to start the queue manager and resolve any UOW. Because IBM MQ issues console messages several times to inform that the end of the log is approaching, a log reset should be planned.

The preferred option to avoid losing any in-flight UOW is to extend the log RBA to use 8 bytes. This means that a log RBA reset will not be necessary for a long period.

Warning thresholds

The following table lists the thresholds, based on the length of the log RBA.

Console message	6-byte log RBA	8-byte log RBA
CSQI045I	0000F00000000000	FFFF800000000000
CSQI046E	0000F80000000000	FFFFC00000000000
CSQI047E	0000FF8000000000	FFFFFC0000000000
CSQJ032E	0000FF8000000000 0000FF8000000000	FFFFFC0000000000 FFFFFC0000000000
CSQJ031D	0000FF8000000000	FFFFFC0000000000

Notes:

1. For message CSQJ032E, the first number applies to the WARNING text and the second number applies to the CRITICAL text in the console message.
2. Message CSQJ031D is issued at IBM MQ initialization only.

Related concepts

[“Implementing the larger log Relative Byte Address” on page 465](#)

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.

Implementing the larger log Relative Byte Address

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.



Attention: You only have to carry out the following procedure to enable this feature if your queue managers were created before IBM MQ 9.3.0, as queue managers created at IBM MQ 9.3.0 and later already have this feature enabled.

See [Planning to increase the maximum addressable log range](#) for considerations when planning to enable 8 byte log RBA.

Perform these instructions, in the order shown, to enable 8 byte log RBA on a single IBM MQ for z/OS queue manager. For queue managers in a queue sharing group, perform the steps on each queue manager in turn.

1. Allocate new BSDS data sets with similar attributes to the current BSDS. You can tailor sample CSQ4BSDS and delete any irrelevant statement, or you can use your existing JCL, but change the BSDS name to something like ++HLQ++.NEW.BSDS01.

Notes:

- a. Check the attributes of your new BSDS before submitting the job to allocate the new BSDS. The only attribute that might change is the size of the BSDS.
 - b. The new BSDS contains more data than the current BSDS, therefore, you must ensure that the new data sets are allocated with sufficient available space. The sample JCL in `thlqual.SCSQPROC(CSQ4BSDS)` contains the recommended values when defining a new BSDS.
2. Shut down the queue manager cleanly.
 3. Run the [BSDS conversion utility \(CSQJUCNV\)](#) to convert the existing BSDS to the new BSDS data sets. This usually takes a few seconds to run.

Your existing BSDS will not be changed during this process, and you can use that for the initialization of the queue manager in the case of an unsuccessful conversion.

4. Rename the current BSDS to become the old BSDS, and the new BSDS to become the current BSDS, so that the new data sets are used when you next restart the queue manager. You can use the DFSMS Access Method Services ALTER command, for example:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

Ensure that you also issue commands to rename both the data and index portions of the VSAM cluster.

5. Restart the queue manager. It should start in the same amount of time as it would have done when using 6 byte log RBA.

If the queue manager does not restart successfully due to a failure to access the converted BSDS, attempt to identify the cause of the failure, resolve the problem and retry the operation. If required, contact your IBM support center for assistance.

If necessary, the change can be backed out at this point by:

- a. Renaming the current BSDS to become the new BSDS.
 - b. Renaming the old BSDS to become the current BSDS.
 - c. Restarting the queue manager.
6. Once the queue manager has been successfully restarted with the converted BSDS, do not attempt to start the queue manager using the old BSDS.

7. Message [CSQJ034I](#) is issued during queue manager initialization to indicate the end of the log RBA for the queue manager as configured. Confirm that the end of the log RBA range displayed is FFFFFFFFFFFFFFFF. This indicates that 8 byte log RBA is in use.

Related tasks

[Planning to increase the maximum addressable log range](#)

Related reference

[Larger log Relative Byte Address](#)

[The BSDS conversion utility \(CSQJUCNV\)](#)

Managing the bootstrap data set (BSDS)

The bootstrap data set (BSDS) is used to reference log data sets, and log records. Use this topic to understand how you can examine, change, and recover the BSDS.

For more information, see [The bootstrap data set](#).

This topic describes the tasks involved in managing the bootstrap data set. It contains these sections:

- [“Finding out what the BSDS contains” on page 466](#)
- [“Changing the BSDS” on page 467](#)
- [“Recovering the BSDS” on page 471](#)

Finding out what the BSDS contains

You can use the print log map utility (CSQJU004) to examine the contents of the BSDS.

The print log map utility (CSQJU004) is a batch utility that lists the information stored in the BSDS. For instructions on running it, see [The print log map utility](#).

The BSDS contains:

- [Time stamps](#)
- [Active log data set status](#)

Time stamps in the BSDS

The output of the print log map utility shows the time stamps, which are used to record the date and time of various system events, that are stored in the BSDS.

The following time stamps are included in the header section of the report:

SYSTEM TIMESTAMP

Reflects the date and time the BSDS was last updated. The BSDS time stamp can be updated when:

- The queue manager starts.
- The write threshold is reached during log write activities. Depending on the number of output buffers you have specified and the system activity rate, the BSDS might be updated several times a second, or might not be updated for several seconds, minutes, or even hours. For details of the write threshold, see the WRTHRSH parameter of the CSQ6LOGP macro in [Using CSQ6LOGP](#).
- IBM MQ drops into a single BSDS mode from its normal dual BSDS mode due to an error. This can occur when a request to get, insert, point to, update, or delete a BSDS record is unsuccessful. When this error occurs, IBM MQ updates the time stamp in the remaining BSDS to force a time stamp mismatch with the disabled BSDS.

UTILITY TIMESTAMP

The date and time the contents of the BSDS were altered by the change log inventory utility (CSQJU003).

The following time stamps are included in the active and archive log data sets portion of the report:

Active log date

The date the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

Active log time

The time the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

Archive log date

The date the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

Archive log time

The time the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

Active log data set status

The BSDS records the status of an active log data set as one of the following:

NEW

The data set has been defined but never used by IBM MQ, or the log was truncated to a point before the data set was first used. In either case, the data set starting and ending RBA values are reset to zero.

REUSABLE

Either the data set has been defined but never used by IBM MQ, or the data set has been offloaded. In the print log map output, the start RBA value for the last REUSABLE data set is equal to the start RBA value of the last archive log data set.

NOT REUSABLE

The data set contains records that have not been offloaded.

STOPPED

The offload processor encountered an error while reading a record, and that record could not be obtained from the other copy of the active log.

TRUNCATED

Either:

- An I/O error occurred, and IBM MQ has stopped writing to this data set. The active log data set is offloaded, beginning with the starting RBA and continuing up to the last valid record segment in the truncated active log data set. The RBA of the last valid record segment is lower than the ending RBA of the active log data set. Logging is switched to the next available active log data set, and continues uninterrupted.

or

- An ARCHIVE LOG function has been called, which has truncated the active log.

The status appears in the output from the print log map utility.

Changing the BSDS

You do not have to take special steps to keep the BSDS updated with records of logging events because IBM MQ does that automatically.

However, you might want to change the BSDS if you do any of the following:

- Add more active log data sets.
- Copy active log data sets to newly allocated data sets, for example, when providing larger active log allocations.

- Move log data sets to other devices.
- Recover a damaged BSDS.
- Discard outdated archive log data sets.

You can change the BSDS by running the change log inventory utility (CSQJU003). Only run this utility when the queue manager is inactive, or you might get inconsistent results. The action of the utility is controlled by statements in the SYSIN data set. This section shows several examples. For complete instructions, see [The change log inventory utility](#).

You can copy an active log data set only when the queue manager is inactive because IBM MQ allocates the active log data sets as exclusive (DISP=OLD) at queue manager startup.

Changes for active logs

Use this topic to understand how you can change the active logs using the BSDS.

You can add to, delete from, and record entries in the BSDS for active logs using the change log utility. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

See these sections for more information:

- [Adding record entries to the BSDS](#)
- [Deleting information about the active log data set from the BSDS](#)
- [Recording information about the log data set in the BSDS](#)
- [Increasing the size of the active log](#)
- [The use of CSQJUFMT](#)

Adding record entries to the BSDS

If an active log has been flagged as "stopped", it is not reused for logging; however, it continues to be used for reading. Use the access method services to define new active log data sets, then use the change log inventory utility to register the new data sets in the BSDS. For example, use:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSNAMES=MQM111.LOGCOPY2.DS10,COPY2
```

If you are copying the contents of an old active log data set to the new one, you can also give the RBA range and the starting and ending time stamps on the NEWLOG function.

Deleting information about the active log data set from the BSDS

To delete information about an active log data set from the BSDS, you could use:

```
DELETE DSNAMES=MQM111.LOGCOPY1.DS99
DELETE DSNAMES=MQM111.LOGCOPY2.DS99
```

Recording information about the log data set in the BSDS

To record information about an existing active log data set in the BSDS, use:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

You might need to insert a record containing this type of information in the BSDS because:

- The entry for the data set has been deleted, but is needed again.

- You are copying the contents of one active log data set to another data set.
- You are recovering the BSDS from a backup copy.

Increasing the size of the active log

There are two methods of achieving this process.

1. When the queue manager is active:
 - a. Define new larger log data sets using JCL.
 - b. Add the new log data sets to the active queue manager using the MQSC DEFINE LOG command.
 - c. Use the MQSC ARCHIVE LOG command to move the current active log, to be a new larger log.
 - d. Wait for the archive of the smaller active log data set to complete.
 - e. Shut down the queue manager, using the CSQJU003 utility to remove the old small active logs.
 - f. Restart the queue manager.
2. When the queue manager is inactive:
 - a. Stop the queue manager. This step is required because IBM MQ allocates all active log data sets for its exclusive use when it is active.
 - b. Use Access Method Services ALTER with the NEWNAME option to rename your active log data sets.
 - c. Use Access Method Services DEFINE to define larger active log data sets.

By reusing the old data set names, you do not have to run the change log inventory utility to establish new names in the BSDSs. The old data set names and the correct RBA ranges are already in the BSDSs.
 - d. Use Access Method Services REPRO to copy the old (renamed) data sets into their appropriate new data sets.

Note: This step can take a long time, so your enterprise could be out of action for this period.
 - e. Start the queue manager.

If all your log data sets are the same size, your system will be operationally more consistent and efficient. If the log data sets are not the same size, it is more difficult to track your system's logs, and so space can be wasted.

The use of CSQJUFMT

Do not run a CSQJUFMT format when increasing the size of an active log.

If you run CSQJUFMT (in order to provide a performance advantage the first time the queue manager writes to the new active log) you receive messages:

```
IEC070I 203-204,XS95GTLX,REPRO02,OUTPUT,B857,SPMG02, 358
IEC070I MG.W.MG4E.LOGCOPY1.DS02,MG.W.MG4E.LOGCOPY1.DS02.DATA,
IDC3302I ACTION ERROR ON MG.W.MG4E.LOGCOPY1.DS02
IDC3351I ** VSAM I/O RETURN CODE IS 28 - RPLFDBWD = X'2908001C'
IDC31467I MAXIMUM ERROR LIMIT REACHED.

IDC0005I NUMBER OF RECORDS PROCESSED WAS 0
```

In addition, if you use the Access Method Services REPRO, ensure that you define a new empty log.

If you use REPRO to copy the old (renamed) data set into its respective new data set, the default is NOREPLACE.

This means that REPRO does not replace a record that is already on the designated data set. When formatting is done on the data set, the RBA value is reset. The net result is a data set that is not empty after formatting.

Changes for archive logs

Use this topic to understand how to change the archive logs.

You can add to, delete from, and change the password of, entries in the BSDS for archive logs. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

- [Adding an archive log](#)
- [Deleting an archive log](#)
- [Changing the password of an archive log](#)

Adding an archive log

When the recovery of an object depends on reading an existing archive log data set, the BSDS must contain information about that data set so that IBM MQ can find it. To register information about an existing archive log data set in the BSDS, use:

```
NEWLOG DSN=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,  
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

Deleting an archive log

To delete an entire archive log data set on one or more volumes, use:

```
DELETE DSN=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

Changing the password of an archive log

If you change the password of an existing archive log data set, you must also change the information in the BSDS.

1. List the BSDS, using the print log map utility.
2. Delete the entry for the archive log data set with the changed password, using the DELETE function of the CSQJU003 utility (see topic [The change log inventory utility](#)).
3. Name the data set as for a new archive log data set. Use the NEWLOG function of the CSQJU003 utility (see topic [The change log inventory utility](#)), and give the new password, the starting and ending RBAs, and the volume serial numbers (which can be found in the print log map utility output, see [The print log map utility](#)).

To change the password for new archive log data sets, use:

```
ARCHIVE PASSWORD= password
```

To stop placing passwords on new archive log data sets, use:

```
ARCHIVE NOPASSWD
```

Note: Only use the ARCHIVE utility function if you do not have an external security manager.

Changing the high-level qualifier (HLQ) for the logs and BSDS

Use this topic to understand the procedure required to change the high-level qualifier (HLQ).

Before you begin

You must end the queue manager normally before copying any of the logs or data sets to the new data sets. This is to ensure that the data is consistent and no recovery is needed during restart.

About this task

This task provides information about how to change the HLQ for the logs and BSDS. To do this, follow these steps:

Procedure

1. Run the log print utility CSQJU004 to record the log data set information. This information is needed later.
2. You can either:
 - a) run DSS backup and restore with rename on the log and BSDS data sets to be renamed, or
 - b) use AMS DEFINE and REPRO to create the HLQ data sets and copy the data from the old data sets.
3. Modify the MSTR and CHIN procedures to point to the new data sets.
4. Delete the old log information in the new copy of the BSDS using CSQJU003.
5. Define the new log data sets to the new BSDS using the NEWLOG function of CSQJU003.
Keep all information about each log the same, apart from the HLQ.
6. The new BSDS should reflect the same information that was recorded for the old logs in the old BSDS.
The HLQ should be the only thing that has changed.

What to do next

Compare the CSQJU004 output for the old and new BSDS to ensure that they look EXACTLY the same (except for the HLQs) before starting the queue manager.

Note: Care must be taken when performing these operations. Incorrect actions might lead to unrecoverable situations. Check the PRINT LOG MAP UTILITY output and make sure that all the information needed for recovery or restart has been included.

Recovering the BSDS

If IBM MQ is operating in dual BSDS mode and one BSDS becomes damaged, forcing IBM MQ into single BSDS mode, IBM MQ continues to operate without a problem (until the next restart).

To return the environment to dual BSDS mode:

1. Use Access Method Services to rename or delete the damaged BSDS and to define a new BSDS with the same name as the damaged BSDS. Example control statements can be found in job CSQ4BREC in thlqual.SCSQPROC.
2. Issue the IBM MQ command RECOVER BSDS to make a copy of the valid BSDS in the newly allocated data set and to reinstate dual BSDS mode.

If IBM MQ is operating in single BSDS mode and the BSDS is damaged, or if IBM MQ is operating in dual BSDS mode and both BSDSs are damaged, the queue manager stops and does not restart until the BSDS data sets are repaired. In this case:

1. Locate the BSDS associated with the most recent archive log data set. The data set name of the most recent archive log appears on the job log in the last occurrence of message CSQJ003I, which indicates that offload processing has been completed successfully. In preparation for the rest of this procedure, it is a good practice to keep a log of all successful archives noted by that message:

- If archive logs are on DASD, the BSDS is allocated on any available DASD. The BSDS name is like the corresponding archive log data set name; change only the first letter of the last qualifier, from A to B, as in this example:

Archive log name

CSQ.ARCHLOG1. **A** 0000001

BSDS copy name

CSQ.ARCHLOG1. **B** 0000001

- If archive logs are on tape, the BSDS is the first data set of the first archive log volume. The BSDS is not repeated on later volumes.
2. If the most recent archive log data set has no copy of the BSDS (for example, because an error occurred when offloading it), locate an earlier copy of the BSDS from earlier offload processing.
 3. Rename *damaged* BSDSs using the Access Method Services ALTER command with the NEWNAME option. If you want to delete a damaged BSDS, use the Access Method Services DELETE command. For each damaged BSDS, use Access Method Services to define a new BSDS as a replacement data set. Job CSQ4BREC in thlqual.SCSQPROC contains Access Method Services control statements to define a new BSDS.
 4. Use the Access Method Services REPRO command to copy the BSDS from the archive log to one of the replacement BSDSs you defined in step “3” on page 472. Do not copy any data to the second replacement BSDS, you do that in step “5” on page 473.

- a. Print the contents of the replacement BSDS.

Use the print log map utility (CSQJU004) to print the contents of the replacement BSDS. This enables you to review the contents of the replacement BSDS before continuing your recovery work.

- b. Update the archive log data set inventory in the replacement BSDS.

Examine the output from the print log map utility and check that the replacement BSDS does not contain a record of the archive log from which the BSDS was copied. If the replacement BSDS is an old copy, its inventory might not contain all archive log data sets that were created more recently. The BSDS inventory of the archive log data sets must be updated to reflect the current subsystem inventory.

Use the change log inventory utility (CSQJU003) NEWLOG statement to update the replacement BSDS, adding a record of the archive log from which the BSDS was copied. If the archive log data set is password-protected, use the PASSWORD option of the NEWLOG function. Also, if the archive log data set is cataloged, ensure that the CATALOG option of the NEWLOG function is properly set to CATALOG=YES. Use the NEWLOG statement to add any additional archive log data sets that were created later than the BSDS copy.

- c. Update passwords in the replacement BSDS.

The BSDS contains passwords for the archive log data sets and for the active log data sets. To ensure that the passwords in the replacement BSDS reflect the current passwords used by your installation, use the change log inventory ARCHIVE utility function with the PASSWORD option.

- d. Update the active log data set inventory in the replacement BSDS.

In unusual circumstances, your installation might have added, deleted, or renamed active log data sets since the BSDS was copied. In this case, the replacement BSDS does not reflect the actual number or names of the active log data sets your installation currently has in use.

If you need to delete an active log data set from the replacement BSDS log inventory, use the change log inventory utility DELETE function.

If you need to add an active log data set to the replacement BSDS log inventory, use the change log inventory utility NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

If you need to rename an active log data set in the replacement BSDS log inventory, use the change log inventory utility DELETE function, followed by the NEWLOG function. Ensure that the RBA range

is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

- e. Update the active log RBA ranges in the replacement BSDS.

Later, when the queue manager restarts, it compares the RBAs of the active log data sets listed in the BSDS with the RBAs found in the actual active log data sets. If the RBAs do not agree, the queue manager does not restart. The problem is magnified when an old copy of the BSDS is used. To solve this problem, use the change log inventory utility (CSQJU003) to adjust the RBAs found in the BSDS using the RBAs in the actual active log data sets. You do this by:

- Using the print log records utility (CSQ1LOGP) to print a summary report of the active log data set. This shows the starting and ending RBAs.
- Comparing the actual RBA ranges with the RBA ranges you have just printed, when the RBAs of all active log data sets are known.

If the RBA ranges are equal for all active log data sets, you can proceed to the next recovery step without any additional work.

If the RBA ranges are not equal, adjust the values in the BSDS to reflect the actual values. For each active log data set that needs to have the RBA range adjusted, use the change log inventory utility DELETE function to delete the active log data set from the inventory in the replacement BSDS. Then use the NEWLOG function to redefine the active log data set to the BSDS. If the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function.

- f. If only two active log data sets are specified for each copy of the active log, IBM MQ can have difficulty during queue manager restart. The problem can arise when one of the active log data sets is full and has not been offloaded, while the second active log data set is close to filling. In this case, add a new active log data set for each copy of the active log and define each new active log data set in the replacement BSDS log inventory.

Use the Access Method Services DEFINE command to define a new active log data set for each copy of the active log and use the change log inventory utility NEWLOG function to define the new active log data sets in the replacement BSDS. You do not need to specify the RBA ranges on the NEWLOG statement. However, if the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function. Example control statements to accomplish this task can be found in job CSQ4LREC in thlqual.SCSQPROC.

5. Copy the updated BSDS to the second new BSDS data set. The BSDSs are now identical.

Use the print log map utility (CSQJU004) to print the contents of the second replacement BSDS at this point.

6. See [Active log problems](#) for information about what to do if you have lost your current active log data set.
7. Restart the queue manager using the newly constructed BSDS. IBM MQ determines the current RBA and what active logs need to be archived.

Managing page sets

Use this topic to understand how to manage the page sets associated with a queue manager.

This topic describes how to add, copy, and generally manage the page sets associated with a queue manager. It contains these sections:

- [“How to change the high-level qualifier \(HLQ\) for the page sets” on page 474](#)
- [“How to add a page set to a queue manager” on page 474](#)
- [“What to do when one of your page sets becomes full” on page 474](#)
- [“How to balance loads on page sets” on page 475](#)
- [How to increase the size of a page set](#)

- [“How to reduce a page set” on page 478](#)
- [“How to reintroduce a page set” on page 479](#)
- [“How to back up and recover page sets” on page 480](#)
- [“How to delete page sets” on page 483](#)
- [“How to back up and restore queues using CSQUTIL” on page 483](#)

See [Page sets](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

How to change the high-level qualifier (HLQ) for the page sets

This task gives information on how to change the HLQ for the page sets. To perform this task, do the following:

1. Define the new HLQ page sets.
2. If the size allocation is the same as the old page sets, copy the existing page set using REPRO to the empty new HLQ page sets.
3. If you are increasing the size of the page sets, use the FORMAT function of CSQUTIL to format the destination pages, and then the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set.

For more information, see [Formatting page sets \(FORMAT\)](#), and [Expanding a page set \(COPYPAGE\)](#).

4. Change the CSQP00xx DD statement in the queue manager procedure to point to the new HLQ page sets.

Restart the queue manager and verify the changes to the page sets.

How to add a page set to a queue manager

This description assumes that you have a queue manager that is already running. You might need to add a page set if, for example, your queue manager has to cope with new applications using new queues.

To add a new page set, use the following procedure:

1. Define and format the new page set. You can use the sample JCL in thlqual.SCSQPROC(CSQ4PAGE) as a basis. For more information, see [Formatting page sets \(FORMAT\)](#).

Take care not to format any page sets that are in use, unless this is what you intend. If so, use the FORCE option of the FORMAT utility function.

2. Use the DEFINE PSID command with the DSN option to associate the page set with a buffer pool.
3. Add the appropriate storage class definitions for your page set by issuing DEFINE STGCLASS commands.
4. Optionally, to document how your queue manager is configured:
 - a. Add the new page set to the started task procedure for your queue manager.
 - b. Add a definition for the new page set to your CSQINP1 initialization data set.
 - c. Add a definition for the new storage class to your CSQ4INYP1 initialization data set member.

For details of the DEFINE PSID and DEFINE STGCLASS commands, see [DEFINE PSID](#) and [DEFINE STGCLASS](#).

What to do when one of your page sets becomes full

You can find out about the utilization of page sets by using the IBM MQ command DISPLAY USAGE. For example, the command:


```
DISPLAY USAGE PSID(03)
```

displays the current state of the page set 03. This tells you how many free pages this page set has.

If you have defined secondary extents for your page sets, they are dynamically expanded each time they fill up. Eventually, all secondary extents are used, or no further disk space is available. If this happens, an application receives the return code MQRC_STORAGE_MEDIUM_FULL.

If an application receives a return code of MQRC_STORAGE_MEDIUM_FULL from an MQI call, this is a clear indication that there is not enough space remaining on the page set. If the problem persists or is likely to recur, you must do something to solve it.

You can approach this problem in a number of ways:

- Balance the load between page sets by moving queues from one page set to another.
- Expand the page set. See [“How to increase the size of a page set” on page 477](#) for instructions.
- Redefine the page set so that it can expand beyond 4 GB to a maximum size of 64 GB. See [Defining a page set to be larger than 4 GB](#) for instructions.

How to balance loads on page sets

Load balancing on page sets means moving the messages associated with one or more queues from one page set to another, less used, page set. Use this technique if it is not practical to expand the page set.

To identify which queues are using a page set, use the appropriate IBM MQ commands. For example, to find out which queues are mapped to page set 02, first, find out which storage classes map to page set 02, by using the command:

```
DISPLAY STGCLASS(*) PSID(02)
```

Then use the following command to find out which queues use which storage class:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

Moving a non-shared queue

To move queues and their messages from one page set to another, use the MQSC MOVE QLOCAL command (described in [MOVE QLOCAL](#)). When you have identified the queue or queues that you want to move to a new page set, follow this procedure for each of these queues:

1. Ensure that the queue you want to move is not in use by any applications (that is, IPPROCS and OPPROCS values from the DISPLAY QSTATUS command are zero) and that it has no uncommitted messages (the UNCOM value from the DISPLAY QSTATUS command is NO).

Note: The only way to ensure that this state continues is to change the security authorization of the queue temporarily. See [Profiles for queue security](#) for more information.

If you cannot do this, later stages in this procedure might fail if applications start to use the queue despite precautionary steps such as setting PUT(DISABLED). However, messages can never be lost by this procedure.

2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved, using the command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

Note: If this temporary queue already exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Define a new storage class that maps to the required page set, for example:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Add the new storage class definition to the CSQINP2 data sets ready for the next queue manager restart.

7. Redefine the queue that you are moving, by changing the storage class attribute:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 475.

8. Move the messages back to the new queue, using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. The queue created in step “3” on page 475 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

10. If the queue being moved was defined in the CSQINP2 data sets, change the STGCLASS attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 29 on page 477 shows an extract from a load balancing job.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
//          DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE STGCLASS(NEW) PSID(2)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) STGCLASS(NEW)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*

```

Figure 29. Extract from a load balancing job for a page set

How to increase the size of a page set

You can initially allocate a page set larger than 4 GB, See [Defining a page set to be larger than 4 GB](#)

A page set can be defined to be automatically expanded as it becomes full by specifying EXPAND(SYSTEM) or EXPAND(USER). If your page set was defined with EXPAND(NONE), you can expand it in either of two ways:

- Alter its definition to allow automatic expansion. See [Altering a page set to allow automatic expansion](#)
- Create a new, larger page set and copy the messages from the old page set to the new one. See [Moving messages to a new, larger page set](#)

Defining a page set to be larger than 4 GB

IBM MQ can use a page set up to 64 GB in size, provided the data set is defined with 'extended addressability' to VSAM. Extended addressability is an attribute which is conferred by an SMS data class.

Note: Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and, from z/OS V1.12, an archive log dataset can also reside in the EAS.

In the example shown in the following sample JCL, the management class 'EXTENDED' is defined to SMS with 'Extended addressability'. If your existing page set is not currently defined as having extended addressability, use the following method to migrate to an extended addressability format data set.

1. Stop the queue manager.
2. Use Access Method Services to rename the existing page set.
3. Define a destination page set, the same size as the existing page set, but with DATACLAS(EXTENDED).

Note: Extended-format data sets must be SMS managed. These are the mechanisms for requesting extended format for VSAM data sets:

- Using a data class that has a DSNTYPE value of EXT and the subparameter R or P to indicate required or preferred.
- Coding DSNTYPE=EXTREQ (extended format is required) or DSNTYPE=EXTPREF (extended format is preferred) on the DD statement.

- Coding the LIKE= parameter on the DD statement to refer to an existing extended format data set.

For more information, see [Restrictions on Defining Extended-Format Data Sets](#).

4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager.
6. Alter the page set to use system expansion, to allow it to continue growing beyond its current allocation.

The following JCL shows example Access Method Services commands:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER 'VICY.CSQ1.PAGE01' -
NEWNAME('VICY.CSQ1.PAGE01.OLD')
ALTER 'VICY.CSQ1.PAGE01.DATA' -
NEWNAME('VICY.CSQ1.PAGE01.DATA.OLD')
DEFINE CLUSTER (NAME('VICY.CSQ1.PAGE01') -
MODEL('VICY.CSQ1.PAGE01.OLD') -
DATACLAS(EXTENDED))
/*
```

Altering a page set to allow automatic expansion

Use the ALTER PSID command with the EXPAND(USER) or EXPAND(SYSTEM) options. See [ALTER PSID](#) and [Expanding a page set \(COPYPAGE\)](#) for general information on expanding page sets.

Moving messages to a new, larger page set

This technique involves stopping and restarting the queue manager. This deletes any nonpersistent messages that are not on shared queues at restart time. If you have nonpersistent messages that you do not want to be deleted, use load balancing instead. For more details, see [“How to balance loads on page sets” on page 475](#). In this description, the page set that you want to expand is referred to as the *source* page set; the new, larger page set is referred to as the *destination* page set.

Follow these steps:

1. Stop the queue manager.
2. Define the destination page set, ensuring that it is larger than the source page set, with a larger secondary extent value.
3. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets \(FORMAT\)](#) for more details.
4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager using the destination page set by doing one of the following:
 - Change the queue manager started task procedure to reference the destination page set.
 - Use Access Method Services to delete the source page set and then rename the destination page set, giving it the same name as that of the source page set.

Attention:

Before you delete any IBM MQ page set, be sure that you have made the required backup copies.

How to reduce a page set

Prevent all users, other than the IBM MQ administrator, from using the queue manager. For example; by changing the access security settings.

If you have a large page set that is mostly empty (as shown by the DISPLAY USAGE command), you might want to reduce its size. The procedure to do this involves using the COPY, FORMAT, and LOAD functions of CSQUTIL (see [IBM MQ utility program](#)). This procedure does not work for page set zero (0), as it is not practical to reduce the size of this page set; the only way to do so is by reinitializing your queue manager (see [“Reinitializing a queue manager” on page 501](#)). The prerequisite of this procedure is to try and remove all users from the system so that all UOWs are complete and the page sets are consistent.

1. Use the STOP QMGR command with the QUIESCE or FORCE attribute to stop the queue manager.
2. Run the SCOPY function of CSQUTIL with the PSID option, to copy all message data from the large page set and save them in a sequential data set.
3. Define a new smaller page set data set to replace the large page set.
4. Run the FORMAT TYPE(NEW) function of CSQUTIL against the page set that you created in step [“3” on page 479](#).
5. Restart the queue manager using the page set created in step [“3” on page 479](#).
6. Run the LOAD function of CSQUTIL to load back all the messages saved during step [“2” on page 479](#).
7. Allow all users access to the queue manager.
8. Delete the old large page set.

How to reintroduce a page set

In certain scenarios it is useful to be able to bring an old page set online again to the queue manager. Unless specific action is taken, when the old page set is brought online the queue manager will recognize that the page set recovery RBA stored in the page set itself and in the checkpoint records is old, and will therefore automatically start media recovery of the page set to bring it up to date.

Such media recovery can only be performed at queue manager restart, and is likely to take a considerable length of time, especially if archive logs held on tape must be read. However, normally in this circumstance, the page set has been offline for the intervening period and so the log contains no information pertinent to the page set recovery.

The following three choices are available:

Allow full media recovery to be performed.

1. Stop the queue manager.
2. Ensure definitions are available for the page set in both the started task procedure for the queue manager and in the CSQINP1 initialization data set.
3. Restart the queue manager.

Allow any messages on the page set to be destroyed.

This choice is useful where a page set has been offline for a long time (some months, for example) and it has now been decided to reuse it for a different purpose.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(NEW) option.
2. Add definitions for the page set to both the started task procedure for the queue manager and the CSQINP1 initialization data set.
3. Restart the queue manager.

Using the TYPE(NEW) option for formatting clears the current contents of the page set and tells the queue manager to ignore any historical information in the checkpoint about the page set.

Bring the page set online avoiding the media recovery process.

Use this technique only if you are sure that the page set has been offline since a clean shutdown of the queue manager. This choice is most appropriate where the page set has been offline for a short period, typically due to operational issues such as a backup running while the queue manager is being started.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(REPLACE) option.

2. Either add the page set back into the queue manager dynamically using the DEFINE PSID command with the DSN option or allow it to be added at a queue manager restart.

Using the TYPE(REPLACE) option for formatting checks that the page set was cleanly closed by the queue manager, and marks it so that media recovery will not be performed. No other changes are made to the contents of the page set.

How to back up and recover page sets

There are different mechanisms available for back up and recovery. Use this topic to understand these mechanisms.

This section describes the following topics:

- [“Creating a point of recovery for non-shared resources” on page 480](#)
- [“Backing up page sets” on page 481](#)
- [“Recovering page sets” on page 482](#)
- [How to delete page sets](#)

For information about how to create a point of recovery for shared resources, see [“Recovering shared queues” on page 488](#).

Creating a point of recovery for non-shared resources

IBM MQ can recover objects and non-shared persistent messages to their current state if both:

1. Copies of page sets from an earlier point exist.
2. All the IBM MQ logs are available to perform recovery from that point.

These represent a point of recovery for non-shared resources.

Both objects and messages are held on page sets. Multiple objects and messages from different queues can exist on the same page set. For recovery purposes, objects and messages cannot be backed up in isolation, so a page set must be backed up as a whole to ensure the correct recovery of the data.

The IBM MQ recovery log contains a record of all persistent messages and changes made to objects. If IBM MQ fails (for example, due to an I/O error on a page set), you can recover the page set by restoring the backup copy and restarting the queue manager. IBM MQ applies the log changes to the page set from the point of the backup copy.

There are two ways of creating a point of recovery:

Full backup

Stop the queue manager, which forces all updates on to the page sets.

This allows you to restart from the point of recovery, using only the backed up page set data sets and the logs from that point on.

Fuzzy backup

Take *fuzzy* backup copies of the page sets without stopping the queue manager.

If you use this method, and your associated logs later become damaged or lost, you cannot use the fuzzy page set backup copies to recover. This is because the fuzzy page set backup copies contain an inconsistent view of the state of the queue manager and are dependent on the logs being available. If the logs are not available, you need to return to the last set of backup page set copies taken while the subsystem was inactive ([Method 1](#)) and accept the loss of data from that time.

Method 1: Full backup

This method involves shutting the queue manager down. This forces all updates on to the page sets so that the page sets are in a consistent state.

1. Stop all the IBM MQ applications that are using the queue manager (allowing them to complete first). This can be done by changing the access security or queue settings, for example.
2. When all activity has completed, display and resolve any in-doubt units of recovery. (Use the commands `DISPLAY CONN` and `RESOLVE INDOUBT`, as described in [DISPLAY CONN](#) and [RESOLVE INDOUBT](#).)
This brings the page sets to a consistent state; if you do not do this, your page sets might not be consistent, and you are effectively doing a fuzzy backup.
3. Issue the `ARCHIVE LOG` command to ensure that the latest log data is written out to the log data sets.
4. Issue the `STOP QMGR MODE(QUIESCE)` command. Record the lowest RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information). You should keep the log data sets starting from the one indicated by the RBA value up to the current log data set.
5. Take backup copies of all the queue manager page sets (see [“Backing up page sets”](#) on page 481).

Method 2: Fuzzy backup

This method does not involve shutting the queue manager down. Therefore, updates might be in virtual storage buffers during the backup process. This means that the page sets are not in a consistent state, and can only be used for recovery with the logs.

1. Issue the `DISPLAY USAGE TYPE(ALL)` command, and record the RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information).
2. Take backup copies of the page sets (see [“Backing up page sets”](#) on page 481).
3. Issue the `ARCHIVE LOG` command, to ensure that the latest log data is written out to the log data sets. To restart from the point of recovery, you must keep the log data sets starting from the log data set indicated by the RBA value up to the current log data set.

Backing up page sets

To recover a page set, IBM MQ needs to know how far back in the log to go. IBM MQ maintains a log RBA number in page zero of each page set, called the *recovery log sequence number* (LSN). This number is the starting RBA in the log from which IBM MQ can recover the page set. When you back up a page set, this number is also copied.

If the copy is later used to recover the page set, IBM MQ must have access to all the log records from this RBA value to the current RBA. That means you must keep enough of the log records to enable IBM MQ to recover from the oldest backup copy of a page set you intend to keep.

Use `ADRDSSU COPY` function to copy the page sets.

For more information, see the [COPY DATASET Command Syntax for Logical Data Set](#) documentation .

For example:

```
//STEP2 EXEC PGM=ADRDSSU,REGION=6M
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
COPY -
DATASET(INCLUDE(SCENDATA.MQPA.PAGESET.*)) -
RENAMEU(SCENDATA.MQPA.PAGESET.** ,SCENDATA.MQPA.BACKUP1.** ) -
SPHERE -
REPUNC -
FASTREPLICATION(PREF ) -
CANCELERROR -
TOL(ENQF)
/*
//
```

If you copy the page set while the queue manager is running you must use a copy utility that copies page zero of the page set first. If you do not do this you could corrupt the data in your page set.

If the process of dynamically expanding a page set is interrupted, for example by power to the system being lost, you can still use ADRDSSU to take a backup of a page set.

If you perform an Access Method Services IDCAMS LISTCAT ENT('page set data set name') ALLOC, you will see that the HI-ALLOC-RBA is higher than the HI-USED-RBA.

The next time this page set fills up it is extended again, if possible, and the pages between the high used RBA and the highest allocated RBA are used, along with another new extent.

Backing up your object definitions

You should also back up copies of your object definitions. To do this, use the MAKEDEF feature of the CSQUTIL COMMAND function (described in [Issuing commands to IBM MQ \(COMMAND\)](#)).

Back up your object definitions whenever you take a backup copy of your queue manager, and keep the most current version.

Recovering page sets

If the queue manager has terminated due to a failure, the queue manager can normally be restarted with all recovery being performed during restart. However, such recovery is not possible if any of your page sets or log data sets are not available. The extent to which you can now recover depends on the availability of backup copies of page sets and log data sets.

To restart from a point of recovery you must have:

- A backup copy of the page set that is to be recovered.
- If you used the "fuzzy" backup process described in ["Method 2: Fuzzy backup"](#) on page 481, the log data set that included the recorded RBA value, the log data set that was made by the ARCHIVE LOG command, and all the log data sets between these.
- If you used full backup, but you do not have the log data sets following that made by the ARCHIVE LOG command, you do **not** need to run the FORMAT TYPE(REPLACE) function of the CSQUTIL utility against all your page sets.

To recover a page set to its current state, you must also have all the log data sets and records since the ARCHIVE LOG command.

There are two methods for recovering a page set. To use either method, the queue manager must be stopped.

Simple recovery

This is the simpler method, and is appropriate for most recovery situations.

1. Delete the page set you want to restore from backup.
2. Use the ADRDSSU COPY function to recover your page set from the backup copy..

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Restart the queue manager.
4. When the queue manager has restarted successfully, you can restart your applications
5. Reinststate your normal backup procedures for the restored page.

Advanced recovery

This method provides performance advantages if you have a large page set to recover, or if there has been much activity on the page set since the last backup copy was taken. However, it requires more manual intervention than the simple method, which might increase the risk of error and the time taken to perform the recovery.

1. Delete and redefine the page set you want to restore from backup.
2. Use ADRDSSU to copy the backup copy of the page set into the new page set. Define your new page set with a secondary extent value so that it can be expanded dynamically.

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Change the CSQINP1 definitions for your queue manager to make the buffer pool associated with the page set being recovered as large as possible. By making the buffer pool large, you might be able to keep all the changed pages resident in the buffer pool and reduce the amount of I/O to the page set.
4. Restart the queue manager.
5. When the queue manager has restarted successfully, stop it (using quiesce) and then restart it using the normal buffer pool definition for that page set. After this second restart completes successfully, you can restart your applications
6. Reinststate your normal backup procedures for the restored page.

What happens when the queue manager is restarted

When the queue manager is restarted, it applies all changes made to the page set that are registered in the log, beginning at the restart point for the page set. IBM MQ can recover multiple page sets in this way. The page set is dynamically expanded, if required, during media recovery.

During restart, IBM MQ determines the log RBA to start from by taking the lowest value from the following:

- Recovery LSN from the checkpoint log record for each page set.
- Recovery LSN from page zero in each page set.
- The RBA of the oldest incomplete unit of recovery in the system at the time the backup was taken.

All object definitions are stored on page set zero. Messages can be stored on any available page set.

Note: The queue manager cannot restart if page set zero is not available.

How to delete page sets

You delete a page set by using the DELETE PSID command; see [DELETE PSID](#) for details of this command.

You cannot delete a page set that is still referenced by any storage class. Use DISPLAY STGCLASS to find out which storage classes reference a page set.

The data set is deallocated from IBM MQ but is not deleted. It remains available for future use, or can be deleted using z/OS facilities.

Remove the page set from the started task procedure for your queue manager.

Remove the definition of the page set from your CSQINP1 initialization data set.

How to back up and restore queues using CSQUTIL

Use this topic as a reference for further information about back up and restore using CSQUTIL.

You can use the CSQUTIL utility functions for backing up and restoring queues. To back up a queue, use the COPY or SCOPY function to copy the messages from a queue onto a data set. To restore the queue, use the complementary function LOAD or SLOAD. For more information, see [IBM MQ utility program](#).

Managing buffer pools

Use this topic if you want to change or delete your buffer pools.

This topic describes how to alter and delete buffer pools. It contains these sections:

- [“How to change the number of buffers in a buffer pool” on page 484](#)
- [“How to delete a buffer pool” on page 484](#)

Buffer pools are defined during queue manager initialization, using [DEFINE BUFFPOOL](#) commands issued from the initialization input data set CSQINP1. Their attributes can be altered in response to business requirements while the queue manager is running, using the processes detailed in this topic. The queue manager records the current buffer pool attributes in checkpoint log records. These are automatically restored on subsequent queue manager restart, unless the buffer pool definition in CSQINP1 includes the REPLACE attribute.

Use the [DISPLAY USAGE](#) command to display the current buffer attributes.

You can also define buffer pools dynamically using the [DEFINE PSID](#) command with the DSN option.

If you change buffer pools dynamically, you should also update their definitions in the initialization data set CSQINP1.

See [z/OS 에 대한 계획](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

Note: Buffer pools use significant storage. When you increase the size of a buffer pool or define a new buffer pool ensure that sufficient storage is available. For more information, see [Address space storage](#).

How to change the number of buffers in a buffer pool

If a buffer pool is too small, the condition can result in message [CSQP020E](#) on the console, you can allocate more buffers to it using the ALTER BUFFPOOL command as follows:

1. Determine how much space is available for new buffers by looking at the [CSQY220I](#) messages in the log. The available space is reported in MB. As a buffer has a size of 4 KB, each MB of available space allows you to allocate 256 buffers. Do not allocate all the free space to buffers, as some is required for other tasks.

If the buffer pool uses fixed 4 KB pages, that is, its PAGECLAS attribute is FIXED4KB, ensure that there is sufficient real storage available on the LPAR.

2. If the reported free space is inadequate, release some buffers from another buffer pool using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool from which you want to reclaim space and *integer* is the new number of buffers to be allocated to this buffer pool, which must be smaller than the original number of buffers allocated to it.

3. Add buffers to the buffer pool you want to expand using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool to be expanded and *integer* is the new number of buffers to be allocated to this buffer pool, which must be larger than the original number of buffers allocated to it.

How to delete a buffer pool

When a buffer pool is no longer used by any page sets, delete it to release the virtual storage allocated to it.

You delete a buffer pool using the [DELETE BUFFPOOL](#) command. The command fails if any page sets are using this buffer pool.

See [“How to delete page sets” on page 483](#) for information about how to delete page sets.

Managing queue sharing groups and shared queues on z/OS

IBM MQ can use different types of shared resources, for example queue sharing groups, shared queues, and the coupling facility. Use this topic to review the procedures needed to manage these shared resources.

This section contains information about the following topics:

- [“Managing queue sharing groups” on page 485](#)
- [“Managing shared queues” on page 488](#)
- [“Managing group objects” on page 492](#)
- [“Managing the coupling facility” on page 493](#)

Managing queue sharing groups

You can add or remove a queue manager to a queue sharing group (QSG), and manage the associated Db2 tables.

This topic has sections about the following tasks:

- [“Setting up a queue sharing group” on page 485](#)
- [“Adding a queue manager to a queue sharing group” on page 486](#)
- [“Removing a queue manager from a queue sharing group” on page 487](#)
- [“Removing a queue sharing group from the Db2 tables” on page 487](#)
- [“Validating the consistency of Db2 definitions” on page 488](#)

Setting up a queue sharing group

Each queue sharing group has a name of up to four characters. The name must be unique in your network, and must be different from any queue manager names.

Follow these steps to set up a queue sharing group:

1. If this is the first queue sharing group to use the Db2 data-sharing group, [set up the Db2 environment](#).
2. [Set up the coupling facility](#).
3. Add the queue sharing group to the Db2 tables. Use the ADD QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQS).
4. Add a queue manager to the queue sharing group by following the steps in [“Adding a queue manager to a queue sharing group” on page 486](#)
5. Define application structures to IBM MQ by following the steps in [“Adding a coupling facility structure” on page 493](#).
6. If required, [migrate non-shared queues to shared queues](#).
7. For availability, create shared channels into and out of the queue sharing group.
 - For connections into the queue sharing group:
 - Set up a VIPA socket or hardware router to distribute workload between the available queue managers in the QSG.
 - Define a receiver channel with QSGDISP(GROUP), to ensure the channel definition is available on all queue managers in the QSG.
 - Start a listener with INDISP(GROUP), on each queue manager, for MCA channel connections into the QSG. Client connections into the QSG should still connect to a listener started with INDISP(QMGR).
 - Change applications to connect using the QSG name, rather than a specific queue manager name.

- Ensure that the channel authentication rules on all queue managers in the QSG are the same, to allow applications to connect to any queue manager in the QSG.
- For connections out of the queue sharing group:
 - Define a shared transmission queue.
 - Define the outbound channel with QSGDISP(GROUP) and DEFCDISP(SHARED).

If you convert an existing channel to a shared channel, you might need to issue the [RESET CHANNEL](#) command before starting the channel as the synchronization queue used by the channel will have changed.

Adding a queue manager to a queue sharing group

A queue manager can be added to an existing queue sharing group.

Note that:

- The queue sharing group must exist before you can add queue managers to it.
- A queue manager can be a member of only one queue sharing group.

Follow these steps to add a queue manager to a queue sharing group:

1. Perform the tasks in [implement ESM security controls for the queue sharing group](#) to grant the appropriate access to the queue manager and channel initiator user IDs.
2. If the queue sharing group has CF structures configured to offload data to SMDS, perform the tasks in [set up the SMDS environment](#).
3. Stop the queue manager.
4. Use the ADD QMGR function of the queue sharing group utility (CSQ5PQSG). This program is described in the [queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQM).
5. [Change your system parameter module](#) to add queue sharing group data:
 - a. Modify CSQ6SYSP to specify the QSGDATA parameter. See [using CSQ6SYSP](#) for more information.
 - b. Assemble and link the system parameter module. You might want to use a different name for the load module.
 - c. Change your startup process to use the new module.
6. Copy and tailor sample member thlqual.SCSQPROC(CSQ4INSS), which defines required CF structures and SYSTEM queues. Add the customized member to the CSQINP2 DD in the queue manager startup JCL.
7. Restart your queue manager using the queue sharing group system parameter module.
8. Optionally, migrate to security profiles prefixed by the queue sharing group name, instead of the queue manager name.
9. If shared channels are used for connections into the QSG, create channel authentication rules that mirror those on the other queue managers in the QSG, to allow applications to connect to any queue manager in the QSG.
10. 10. Optionally, do either of the following to allow applications connected to the queue manager in the QSG to put messages to queues hosted by other queue managers in the QSG:
 - Turn on [intra-group queuing](#) by issuing the command ALTER QMGR IGQ(ENABLED).
 - Define transmission queues and channels to the other queue managers in the QSG. Defining transmission queues with the same name as the target queue managers avoids the need to define remote queues and queue manager aliases.

Note: To add a queue manager to an existing queue sharing group containing queue managers running earlier versions of IBM MQ, you must first apply the coexistence PTF for the highest version of IBM MQ in the group to every earlier version queue manager in the group.

Removing a queue manager from a queue sharing group

You can only remove a queue manager from a queue sharing group if the queue manager's logs are not needed by another process, and all SMDS owned by the queue manager are empty.

See [Deleting shared message data sets](#) and [DELETE CFSTRUCT](#) for more information.

The logs are needed if they contain:

- The latest backup of one of the coupling facility (CF) application structures used by the queue sharing group
- Data needed by a future restore process, that is, the queue manager has used a recoverable structure since the time described by the last backup exclusion interval value.

If either or both of these points apply, or an SMDS owned by the queue manager contains messages, the queue manager cannot be removed. To determine which queue managers' logs are needed for a future restore process, use the MQSC DISPLAY CFSTATUS command with the TYPE(BACKUP) option (for details of this command, see [DISPLAY CFSTATUS](#)).

Use the following steps to remove a queue manager from a queue sharing group:

1. Stop any applications connected to the queue manager that put messages to shared queues.
2. Resolve any indoubt units of work involving this queue manager.
3. Determine if there are any messages in any SMDS owned by the queue manager by issuing the command DISPLAY USAGE TYPE(SMDS).
4. If there are offloaded messages for any application structure, wait until those messages have been retrieved from the queue. The number of offloaded messages reported by DISPLAY USAGE TYPE(SMDS) should be zero before proceeding.
5. Shut the queue manager down cleanly using STOP QMGR MODE(QUIESCE).
6. Wait for an interval at least equivalent to the value of the EXCLINT parameter you will specify in the BACKUP CFSTRUCT command in the next step.
7. On another queue manager, run a CF structure backup for each recoverable CF structure by using the MQSC BACKUP CFSTRUCT command and specifying an EXCLINT value as required in the previous step.
8. Confirm that the queue manager's logs are not needed to restore any CF structures, by inspecting the output from the command DISPLAY CFSTATUS(*) TYPE(BACKUP).
9. Use the REMOVE QMGR function of the CSQ5PQSG utility to remove the queue manager from the queue sharing group. This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQM).
10. Before restarting the queue manager, reset the QSGDATA system parameter to its default value, and recreate the system parameter module. See [Using CSQ6SYSP](#) for information about how to tailor your system parameters.

Note, that when removing the last queue manager in a queue sharing group, you must use the FORCE option, rather than REMOVE. This removes the queue manager from the queue sharing group, while not performing the consistency checks of the queue manager logs being required for recovery. You should only perform this operation if you are deleting the queue sharing group.

Removing a queue sharing group from the Db2 tables

To remove a queue sharing group from the Db2 tables, use the REMOVE QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQS).

You can only remove a queue sharing group from the common Db2 data-sharing group tables after you have removed all the queue managers from the queue sharing group (as described in [“Removing a queue manager from a queue sharing group”](#) on page 487).

When the queue sharing group record is deleted from the queue sharing group administration table, all objects and administrative information relating to that queue sharing group are deleted from other IBM MQ Db2 tables. This includes shared queue and group object information.

Validating the consistency of Db2 definitions

Problems for shared queues within a queue sharing group can occur if the Db2 object definitions have, for any reason, become inconsistent.

To validate the consistency of the Db2 object definitions for queue managers, CF structures, and shared queues, use the VERIFY QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#).

Managing shared queues

Use this topic to understand how to recover, move, and migrate shared queues.

This section describes the following tasks:

- [“Recovering shared queues” on page 488](#)
- [“Moving shared queues” on page 489](#)
- [“Migrating non-shared queues to shared queues” on page 491](#)
- [Suspending a Db2 connection](#)

Recovering shared queues

IBM MQ can recover persistent messages on shared queues if all:

- Backups of the CF structures containing the messages have been performed.
- All the logs for all queue managers in the queue sharing group are available, to perform recovery from the point the backups are taken.
- Db2 is available and the structure backup table is more recent than the most recent CF structure backup.

The messages on a shared queue are stored in a coupling facility (CF) structure. Persistent messages can be put onto shared queues, and like persistent messages on non-shared queues, they are copied to the queue manager log. The MQSC [BACKUP CFSTRUCT](#) and [RECOVER CFSTRUCT](#) commands are provided to allow the recovery of a CF structure in the unlikely event of a coupling facility failure. In such circumstances, any nonpersistent messages stored in the affected structure are lost, but persistent messages can be recovered. Any further application activity using the structure is prevented until the structure has been recovered.

To enable recovery, you must back up your coupling facility list structures frequently using the MQSC [BACKUP CFSTRUCT](#) command. The messages in the CF structure are written onto the active log data set of the queue manager making the backup. It writes a record of the backup to Db2: the name of the CF structure being backed up, the name of the queue manager doing the backup, the RBA range for this backup on that queue manager log, and the backup time. Back up CF list structures even if you are not actively using shared queues, for example, if you have set up a queue sharing group intending to use it in the future.

You can recover a CF structure by issuing an MQSC [RECOVER CFSTRUCT](#) command to the queue manager that can perform the recovery; you can use any queue manager in the queue sharing group. You can specify a single CF structure to be recovered, or you can recover several CF structures simultaneously.

As noted previously, it is important that you back up your CF list structures frequently, otherwise recovering a CF structure can take a long time. Moreover, the recovery process cannot be canceled.

The definition of a shared queue is kept in a Db2 database and can therefore be recovered if necessary using standard Db2 database procedures. See [Shared queues and queue sharing groups](#) for more information.

Moving shared queues

This section describes how to perform load balancing by moving a shared queue from one coupling facility structure to another. It also describes how to move a non-shared queue to a shared queue, and how to move a shared queue to a non-shared queue.

When you move a queue, you need to define a temporary queue as part of the procedure. This is because every queue must have a unique name, so you cannot have two queues of the same name, even if the queues have different queue dispositions. IBM MQ tolerates having two queues with the same name (as in step “2” on page 489), but you cannot use the queues.

- Moving a queue from one coupling facility structure to another
- Moving a non-shared queue to a shared queue
- Moving a shared queue to a non-shared queue

Moving a queue from one coupling facility structure to another

To move queues and their messages from one CF structure to another, use the MQSC [MOVE QLOCAL](#) command. When you have identified the queue or queues that you want to move to a new CF structure, use the following procedure to move each queue:

1. Ensure that the queue you want to move is not in use by any applications, that is, the queue attributes IPPROCS and OPPROCS are zero on all queue managers in the queue sharing group.
2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved using the following command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

Note: If this temporary queue exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Redefine the queue that is being moved, changing the CFSTRUCT attribute, using the following command:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 489.

7. Move the messages back to the new queue using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

8. The queue created in step “3” on page 489 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

9. If the queue being moved was defined in the CSQINP2 data sets, change the CFSTRUCT attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 30 on page 490 shows a sample job for moving a queue from one CF structure to another.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//      DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 30. Sample job for moving a queue from one CF structure to another

Moving a non-shared queue to a shared queue

The procedure for moving a non-shared queue to a shared queue is like the procedure for moving a queue from one CF structure to another (see “[Moving a queue from one coupling facility structure to another](#)” on page 489). Figure 31 on page 490 gives a sample job to do this.

Note: Remember that messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//      DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 31. Sample job for moving a non-shared queue to a shared queue

Moving a shared queue to a non-shared queue

The procedure for moving a shared queue to a non-shared queue is like the procedure for moving a queue from one CF structure to another (see [“Moving a queue from one coupling facility structure to another”](#) on page 489).

Figure 32 on page 491 gives a sample job to do this.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) STGCLASS(NEW) QSGDISP(QMGR)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 32. Sample job for moving a shared queue to a non-shared queue

Migrating non-shared queues to shared queues

There are two stages to migrating non-shared queues to shared queues:

- Migrating the first (or only) queue manager in the queue sharing group
- Migrating any other queue managers in the queue sharing group

Migrating the first (or only) queue manager in the queue sharing group

Figure 31 on page 490 shows an example job for moving a non-shared queue to a shared queue. Do this for each queue that needs migrating.

Note:

1. Messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.
2. You must use the correct index type for shared queues. If you migrate a transmission queue to be a shared queue, the index type must be MSGID.

If the queue is empty, or you do not need to keep the messages that are on it, migrating the queue is simpler. Figure 33 on page 492 shows an example job to use in these circumstances.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
DELETE QL(TEMP_QUEUE)
/*

```

Figure 33. Sample job for moving a non-shared queue without messages to a shared queue

Migrating any other queue managers in the queue sharing group

1. For each queue that does not have the same name as an existing shared queue, move the queue as described in [Figure 31 on page 490](#) or [Figure 33 on page 492](#).
2. For queues that have the same name as an existing shared queue, move the messages to the shared queue using the commands shown in [Figure 34 on page 492](#).

```

MOVE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(Queue_TO_MOVE)
DELETE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR)

```

Figure 34. Moving messages from a non-shared queue to an existing shared queue

Suspending a connection to Db2

If you want to apply maintenance or service to the Db2 tables or package related to shared queues without stopping your queue manager, you must temporarily disconnect queue managers in the data sharing group (DSG) from Db2.

To do this:

1. Use the MQSC command [SUSPEND QMGR FACILITY\(Db2 \)](#).
2. Do the binds.
3. Reconnect to Db2 using the MQSC command [RESUME QMGR FACILITY\(Db2 \)](#)

Note that there are restrictions on the use of these commands.



Attention: While the Db2 connection is suspended, the following operations will not be available. Therefore, you need to do this work during a time when your enterprise is at its least busy.

- Access to Shared queue objects for administration (define, delete,alter)
- Starting shared channels
- Storing messages in Db2
- Backup or recover CFSTRUCT

Managing group objects

Use this topic to understand how to work with group objects.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily, and IBM MQ allows you to refresh the page set copies from the repository copy. IBM MQ always tries to refresh the page set copies from the repository

copy on start-up (for channel objects, this is done when the channel initiator restarts). This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive.

There are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that would change the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that would delete that queue fails.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers. Check for and correct any problems with copy objects after adding, changing, or deleting a group object, and at queue manager or channel initiator restart.

Managing the coupling facility

Use this topic to understand how to add or remove coupling facility (CF) structures.

This section describes the following tasks:

- [“Adding a coupling facility structure” on page 493](#)
- [“Removing a coupling facility structure” on page 493](#)

Adding a coupling facility structure

To add a coupling facility structure, use the following procedure:

1. Define the CF structure in the CFRM policy data set. The information about setting up the coupling facility in [Set up the coupling facility](#) describes the rules for naming coupling facility structures, and how to define structures in the CFRM policy data set.
2. If you want to configure the structure to offload message data to SMDS, allocate and preformat data sets. See [creating a shared message data set](#) for details.
3. Define the structure to IBM MQ using the [DEFINE CFSTRUCT](#) command.

Removing a coupling facility structure

To remove a coupling facility structure, use the following procedure:

1. Use the following command to get a list of all the queues using the coupling facility structure that you want to delete:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Delete all the queues that use the structure.
3. Delete the CF structure from IBM MQ using the [DELETE CFSTRUCT](#) command.
4. If the structure was configured to offload message data to SMDS, delete the SMDS.
5. Remove the structure definition from your CFRM policy data set and run the IXCMIAPU utility. (This is the reverse of the customization task set up the coupling facility, described in [Set up the coupling facility](#).)

Tuning coupling facility list monitoring

Use this topic to understand coupling facility list monitoring

Coupling facility (CF) list monitoring is used to monitor the state of list structures containing IBM MQ shared queues. When a message is added to a shared queue, and the queue's depth transitions from zero to non-zero, the CF notifies all queue managers in the queue sharing group. When notified the queue

managers might perform a number of actions, including notifying trigger monitors that are using TRIGGER(FIRST), or applications which are performing a get-wait.

By default, the CF notifies all queue managers in the queue sharing group at the same time. In certain configurations this can cause problems, such as:

- Skewed workload distribution, where a large percentage of messages go to a particular queue manager in the queue sharing group, often the queue manager running on the fastest LPAR, or which is closest to the CF, or
- A large number of failed gets, resulting in wasted CPU time.

z/OS V2R3 introduces a new coupling facility resource manager (CFRM) attribute called **KEYRNOTIFYDELAY**, which can be used with list structures containing shared queues (that is, application structures, and not the admin structure), and which can, for certain workloads, minimize the effects of workload skewing and empty MQGET calls, or empty MQGET calls.

KEYRNOTIFYDELAY can only be set on structures in a CF, running at CFLEVEL 22 or higher.

Its value must be one to seven decimal digits, in a range from 0 to 1,000,000 microseconds. If set to a non-zero value and the depth of a queue transitions from zero to non-zero, the CF selects a single queue manager from the queue sharing group, and notifies that queue manager before all the other queue managers in the group.

The queue manager is selected in a round-robin manner. If the selected queue manager does not process the message inside the time interval described by **KEYRNOTIFYDELAY** all the other queue managers in the queue sharing group will also be notified.

More information on **KEYRNOTIFYDELAY** is available here: [Understanding Keyrange Monitoring Notification Delay](#).

Note that there are two similar CFRM attributes called **LISTNOTIFYDELAY** and **SUBNOTIFYDELAY**. Neither of these has any measurable effect on IBM MQ workload.

Recovery and restart on z/OS

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

Restarting IBM MQ

After a queue manager terminates there are different restart procedures needed depending on how the queue manager terminated. Use this topic to understand the different restart procedures that you can use.

This topic contains information about how to restart your queue manager in the following circumstances:

- [“Restarting after a normal shutdown” on page 494](#)
- [“Restarting after an abnormal termination” on page 495](#)
- [“Restarting if you have lost your page sets” on page 495](#)
- [“Restarting if you have lost your log data sets” on page 495](#)
- [Restarting if you have lost your CF structures](#)

Restarting after a normal shutdown

If the queue manager was stopped with the STOP QMGR command, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 420](#).

Restarting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting the queue manager after it has terminated abnormally is different from starting it after the STOP QMGR command has been issued. If the queue manager terminates abnormally, it terminates without being able to finish its work or take a termination checkpoint.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 420](#). When you restart a queue manager after an abnormal termination, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks.

Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies. This is described in [“Recovering units of work manually” on page 507](#).

Restarting if you have lost your page sets

If you have lost your page sets, you need to restore them from your backup copies before you can restart the queue manager. This is described in [“How to back up and recover page sets” on page 480](#).

The queue manager might take a long time to restart under these circumstances because of the length of time needed for media recovery.

Restarting if you have lost your log data sets

If, after stopping a queue manager (using the STOP QMGR command), both copies of the log are lost or damaged, you can restart the queue manager providing you have a consistent set of page sets (produced using [Method 1: Full backup](#)).

Follow this procedure:

1. Define new page sets to correspond to each existing page set in your queue manager. See [Task 15: Define your page sets](#) for information about page set definition.
Ensure that each new page set is larger than the corresponding source page set.
2. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets](#) for more details.
3. Use the RESETPAGE function of CSQUTIL to copy the existing page sets or reset them in place, and reset the log RBA in each page. See [Copying a page set and resetting the log](#) for more information about this function.
4. Redefine your queue manager log data sets and BSDS using CSQJU003 (see [The change log inventory utility](#)).
5. Restart the queue manager using the new page sets. To do this, you do one of the following:
 - Change the queue manager started task procedure to reference the new page sets. See [Task 6: Create procedures for the IBM MQ queue manager](#) for more information.
 - Use Access Method Services to delete the old page sets and then rename the new page sets, giving them the same names as the old page sets.

Attention: Before you delete any IBM MQ page set, ensure that you have made the required backup copies.

If the queue manager is a member of a queue sharing group, GROUP and SHARED object definitions are not normally affected by lost or damaged logs. However, if any shared-queue messages are involved in a unit of work that was covered by the lost or damaged logs, the effect on such uncommitted messages is unpredictable.

Note: If logs are damaged and the queue manager is a member of a queue sharing group, the ability to recover shared persistent messages might be lost. Issue a BACKUP CFSTRUCT command immediately on another active queue manager in the queue sharing group for all CF structures with the RECOVER(YES) attribute.

Restarting if you have lost your CF structures

You do not need to restart if you lose your CF structures, because the queue manager does not terminate.

Alternative site recovery on z/OS

You can recover a single queue manager or a queue sharing group, or consider disk mirroring.

See the following sections for more details:

- [Recovering a single queue manager at an alternative site](#)
- [Recovering a queue sharing group.](#)
 - [CF structure media recovery](#)
 - [Backing up the queue sharing group at the prime site](#)
 - [Recovering a queue sharing group at the alternative site](#)
- [Using disk mirroring](#)

Recovering a single queue manager at an alternative site

If a total loss of an IBM MQ computing center occurs, you can recover on another queue manager or queue sharing group at a recovery site. (See [“Recovering a queue sharing group at the alternative site”](#) on page 499 for the alternative site recovery procedure for a queue sharing group.)

To recover on another queue manager at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time, as possible.

At the recovery site:

- The recovery queue managers **must** have the same names as the lost queue managers.
- The system parameter module (for example, CSQZPARM) used on each recovery queue manager must contain the same parameters as the corresponding lost queue manager.

When you have done this, reestablish all your queue managers as described in the following procedure. This can be used to perform disaster recovery at the recovery site for a single queue manager. It assumes that all that is available are:

- Copies of the archive logs and BSDSs created by normal running at the primary site (the active logs will have been lost along with the queue manager at the primary site).
- Copies of the page sets from the queue manager at the primary site that are the same age or older than the most recent archive log copies available.

You can use dual logging for the active and archive logs, in which case you need to apply the BSDS updates to both copies:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA and ENDRBA of this log.

5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the STARTRBA and ENDRBA recorded in Step “4” on page 496.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Use CSQJU003 to add a restart control record to the BSDS. Specify CRESTART CREATE, ENDRBA=highrba, where highrba is the high RBA of the most recent archive log available (found in Step “4” on page 496), plus 1.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

9. Restart the queue manager with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.
REPLY Y TO CONTINUE, N TO CANCEL
```

Type Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

See [IBM MQ utilities on z/OS reference](#) for information about using CSQJU003 and CSQJU004.

The following example shows sample input statements for CSQJU003 for steps 6, 7, and 8:

```
* Step 6
DELETE DSNAME=MQM2.LOGCOPY1.DS01
DELETE DSNAME=MQM2.LOGCOPY1.DS02
DELETE DSNAME=MQM2.LOGCOPY1.DS03
DELETE DSNAME=MQM2.LOGCOPY1.DS04
DELETE DSNAME=MQM2.LOGCOPY2.DS01
DELETE DSNAME=MQM2.LOGCOPY2.DS02
DELETE DSNAME=MQM2.LOGCOPY2.DS03
DELETE DSNAME=MQM2.LOGCOPY2.DS04

* Step 7
NEWLOG DSNAME=MQM2.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS03,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS04,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY2.DS01,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS02,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS03,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS04,COPY2

* Step 8
CRESTART CREATE, ENDRBA=063000
```

The things you need to consider for restarting the channel initiator at the recovery site are like those faced when using ARM to restart the channel initiator on a different z/OS image. See [“Using ARM in an IBM MQ network”](#) on page 505 for more information. Your recovery strategy should also cover recovery of the IBM MQ product libraries and the application programming environments that use IBM MQ (CICS , for example).

Other functions of the change log inventory utility (CSQJU003) can also be used in disaster recovery scenarios. The HIGHRBA function allows the update of the highest RBA written and highest RBA offloaded values within the bootstrap data set. The CHECKPT function allows the addition of new checkpoint queue records or the deletion of existing checkpoint queue records in the BSDS.

Attention: These functions might affect the integrity of your IBM MQ data. Only use them in disaster recovery scenarios under the guidance of IBM service personnel.

Fast copy techniques

If copies of all the page sets and logs are made while the queue manager is frozen, the copies will be a consistent set that can be used to restart the queue manager at an alternative site. They typically enable a much faster restart of the queue manager, as there is little media recovery to be performed.

Use the SUSPEND QMGR LOG command to freeze the queue manager. This command flushes buffer pools to the page sets, takes a checkpoint, and stops any further log write activity. Once log write activity has been suspended, the queue manager is effectively frozen until you issue a RESUME QMGR LOG command. While the queue manager is frozen, the page sets and logs can be copied.

By using copying tools such as FLASHCOPY or SNAPSHOT to rapidly copy the page sets and logs, the time during which the queue manager is frozen can be reduced to a minimum.

Within a queue sharing group, however, the SUSPEND QMGR LOG command might not be such a good solution. To be effective, the copies of the logs must all contain the same point in time for recovery, which means that the SUSPEND QMGR LOG command must be issued on all queue managers within the queue sharing group simultaneously, and therefore the entire queue sharing group will be frozen for some time.

Recovering a queue sharing group

In the event of a prime site disaster, you can restart a queue sharing group at a remote site using backup data sets from the prime site. To recover a queue sharing group you need to coordinate the recovery across all the queue managers in the queue sharing group, and coordinate with other resources, primarily Db2. This section describes these tasks in detail.

- [CF structure media recovery](#)
- [Backing up the queue sharing group at the prime site](#)
- [Recovering a queue sharing group at the alternative site](#)

CF structure media recovery

Media recovery of a CF structure used to hold persistent messages on a shared queue, relies on having a backup of the media that can be forward recovered by the application of logged updates. Take backups of your CF structures periodically using the MQSC BACKUP CFSTRUCT command. All updates to shared queues (MQGETs and MQPUTs) are written on the log of the queue manager where the update is performed. To perform media recovery of a CF structure you must apply logged updates to that backup from the logs of all the queue managers that have used that CF structure. When you use the MQSC RECOVER CFSTRUCT command, IBM MQ automatically merges the logs from relevant queue managers, and applies the updates to the most recent backup.

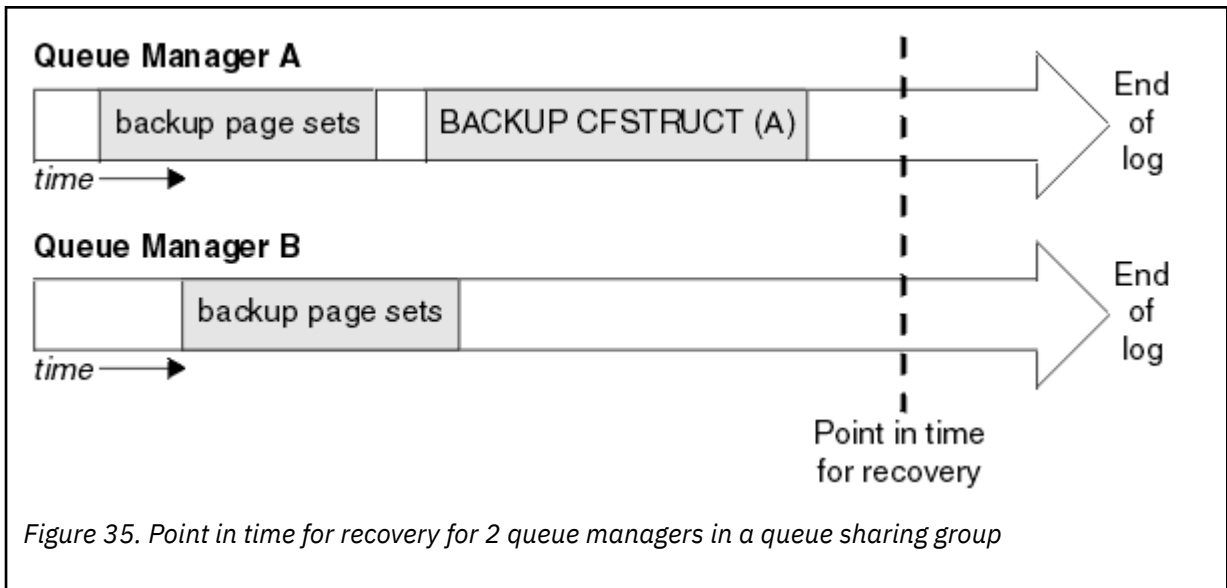
The CF structure backup is written to the log of the queue manager that processed the BACKUP CFSTRUCT command, so there are no additional data sets to be collected and transported to the alternative site.

Backing up the queue sharing group at the prime site

At the prime site you need to establish a consistent set of backups on a regular basis, which can be used in the event of a disaster to rebuild the queue sharing group at an alternative site. For a single queue manager, recovery can be to an arbitrary point in time, typically the end of the logs available at the remote site. However, where persistent messages have been stored on a shared queue, the logs of all the queue managers in the queue sharing group must be merged to recover shared queues, as any queue manager in the queue sharing group might have performed updates (MQPUTs or MQGETs) on the queue.

For recovery of a queue sharing group, you need to establish a point in time that is within the log range of the log data of all queue managers. However, as you can only **forward** recover media from the log, this point in time must be after the BACKUP CFSTRUCT command has been issued and after any page set backups have been performed. (Typically, the point in time for recovery might correspond to the end of a business day or week.)

The following diagram shows time lines for two queue managers in a queue sharing group. For each queue manager, fuzzy backups of page sets are taken (see [Method 2: Fuzzy backup](#)). On queue manager A, a BACKUP CFSTRUCT command is issued. Subsequently, an ARCHIVE LOG command is issued on each queue manager to truncate the active log, and copy it to media offline from the queue manager, which can be transported to the alternative site. End of log identifies the time at which the ARCHIVE LOG command was issued, and therefore marks the extent of log data typically available at the alternative site. The point in time for recovery must lie between the end of any page set or CF structure backups, and the earliest end of log available at the alternative site.



IBM MQ records information associated with the CF structure backups in a table in Db2. Depending on your requirements, you might want to coordinate the point in time for recovery of IBM MQ with that for Db2, or it might be sufficient to take a copy of the IBM MQ CSQ.ADMIN_B_STRBACKUP table after the BACKUP CFSTRUCT commands have finished.

To prepare for a recovery:

1. Create page set backups for each queue manager in the queue sharing group.
2. Issue a BACKUP CFSTRUCT command for each CF structure with the RECOVER(YES) attribute. You can issue these commands from a single queue manager, or from different queue managers within the queue sharing group to balance the workload.
3. Once all the backups have completed, issue an ARCHIVE LOG command to switch the active log and create copies of the logs and BSDS of each queue manager in the queue sharing group.
4. Transport the page set backups, the archived logs, the archived BSDS of all the queue managers in the queue sharing group, and your chosen Db2 backup information, off-site.

Recovering a queue sharing group at the alternative site

Before you can recover the queue sharing group, you need to prepare the environment:

1. If you have old information in your coupling facility from practice startups when you installed the queue sharing group, you need to clean this out first:

Note: If you do not have old information in the coupling facility, you can omit this step.

- a. Enter the following z/OS command to display the CF structures for this queue sharing group:

```
D XCF,STRUCTURE,STRNAME= qsgname
```

- b. For all structures that start with the queue sharing group name, use the z/OS command SETXCF FORCE CONNECTION to force the connection off those structures:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Delete all the CF structures using the following command for each structure:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Restore Db2 systems and data-sharing groups.
3. Recover the CSQ.ADMIN_B_STRBACKUP table so that it contains information about the most recent structure backups taken at the prime site.

Note: It is important that the STRBACKUP table contains the most recent structure backup information. Older structure backup information might require data sets that you have discarded as a result of the information given by a recent DISPLAY USAGE TYPE(DATASET) command, which would mean that your recovered CF structure would not contain accurate information.

4. Run the ADD QMGR command of the CSQ5PQSG utility for every queue manager in the queue sharing group. This will restore the XCF group entry for each queue manager.

When you run the utility in this scenario, the following messages are normal:

```
CSQU566I Unable to get attributes for admin structure, CF not found  
or not allocated  
CSQU546E Unable to add QMGR queue_manager_name entry,  
already exists in DB2 table CSQ.ADMIN_B_QMGR  
CSQU148I CSQ5PQSG Utility completed, return code=4
```

To recover the queue managers in the queue sharing group:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA, STARTLRSN, ENDRBA, and ENDLRSN values of this log.
5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the values recorded in Step “4” on page 500.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Calculate the *recoverylrsn* for the queue sharing group. The *recoverylrsn* is the lowest of the ENDLRSNs across all queue managers in the queue sharing group (as recorded in Step “4” on page 500), minus 1. For example, if there are two queue managers in the queue sharing group, and the ENDLRSN for one of them is B713 3C72 22C5, and for the other is B713 3D45 2123, the *recoverylrsn* is B713 3C72 22C4.
9. Use CSQJU003 to add a restart control record to the BSDS. Specify:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

where *recoverylrsn* is the value you recorded in Step “8” on page 500.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

You must add the CRESTART record to the BSDS for each queue manager within the queue sharing group.

10. Restart each queue manager in the queue sharing group with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Reply Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

The first IBM MQ queue manager started can rebuild the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group at this stage.

11. When the admin structure data for all queue managers has been rebuilt, issue a RECOVER CFSTRUCT command for each CF application structure.

If you issue the RECOVER CFSTRUCT command for all structures on a single queue manager, the log merge process is only performed once, so is quicker than issuing the command on a different queue manager for each CF structure, where each queue manager has to perform the log merge step.

When conditional restart processing is used in a queue sharing group, IBM MQ queue managers, performing peer admin rebuild, check that peers BSDS contain the same CRESTART LRSN as their own. This is to ensure the integrity of the rebuilt admin structure. It is therefore important to restart other peers in the QSG, so they can process their own CRESTART information, before the next unconditional restart of any member of the group.

Using disk mirroring

Many installations now use disk mirroring technologies such as IBM Metro Mirror (formerly PPRC) to make synchronous copies of data sets at an alternative site. In such situations, many of the steps detailed become unnecessary as the IBM MQ page sets and logs at the alternative site are effectively identical to those at the prime site. Where such technologies are used, the steps to restart a queue sharing group at an alternative site may be summarized as:

- Clear IBM MQ CF structures at the alternative site. (These often contain residual information from any previous disaster recovery exercise).
- Restore Db2 systems and all tables in the database used by the IBM MQ queue sharing group.
- Restart queue managers. Before IBM WebSphere MQ 7.0.1, it is necessary to restart each queue manager defined in the queue sharing group as each queue manager recovers its own partition of the admin structure during queue manager restart. After each queue manager has been restarted, those not on their home LPAR can be shut down again. The first IBM MQ queue manager started rebuilds the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group.
- After the admin structure has been rebuilt, recover the application structures.

IBM MQ for z/OS supports use of zHyperWrite when writing to active logs mirrored using Metro Mirror. zHyperWrite can help reduce the performance impact of using Metro Mirror; see [Using Metro Mirror with IBM MQ](#) for more information.

Reinitializing a queue manager

If the queue manager has terminated abnormally you might not be able to restart it. This could be because your page sets or logs have been lost, truncated, or corrupted. If this has happened, you might have to reinitialize the queue manager (perform a cold start).

Attention

Only perform a cold start if you cannot restart the queue manager any other way. Performing a cold start enables you to recover your queue manager and your object definitions; you will **not** be able to recover your message data. Check that none of the other restart scenarios described in this topic work for you before you do this.

When you have restarted, all your IBM MQ objects are defined and available for use, but there is no message data.

Note: Do not reinitialize a queue manager while it is part of a cluster. You must first remove the queue manager from the cluster (using RESET CLUSTER commands on the other queue managers in the cluster), then reinitialize it, and finally reintroduce it to the cluster as a new queue manager.

This is because during reinitialization, the queue manager identifier (QMID) is changed, so any cluster object with the old queue manager identifier must be removed from the cluster.

For further information see the following sections:

- [Reinitializing a queue manager that is not in a queue sharing group](#)
- [Reinitializing queue managers in a queue sharing group](#)

Reinitializing a queue manager that is not in a queue sharing group

To reinitialize a queue manager, follow this procedure:

1. Prepare the object definition statements that to be used when you restart the queue manager. To do this, either:
 - If page set zero is available, use the CSQUTIL SDEFS function (see [Producing a list of IBM MQ define commands](#)). You must get definitions for all object types (authentication information objects, CF structures, channels, namelists, processes, queues, and storage classes).
 - If page set zero is not available, use the definitions from the last time you backed up your object definitions.
2. Redefine your queue manager data sets (do not do this until you have completed step “1” on page 502).
See [creating the bootstrap and log data sets and defining your page sets](#) for more information.
3. Restart the queue manager using the newly defined and initialized log data sets, BSDS, and page sets. Use the object definition input statements that you created in step “1” on page 502 as input in the CSQINP2 initialization input data set.

Reinitializing queue managers in a queue sharing group

In a queue sharing group, reinitializing a queue manager is more complex. It might be necessary to reinitialize one or more queue managers because of page set or log problems, but there might also be problems with Db2 or the coupling facility to deal with. Because of this, there are a number of alternatives:

Cold start

Reinitializing the entire queue sharing group involves forcing all the coupling facilities structures, clearing all object definitions for the queue sharing group from Db2, deleting or redefining the logs and BSDS, and formatting page sets for all the queue managers in the queue sharing group.

Shared definitions retained

Delete or redefine the logs and BSDS, format page sets for all queue managers in the queue sharing group, and force all the coupling facilities structures. On restart, all messages will have been deleted. The queue managers re-create COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist and can be used.

Single queue manager reinitialized

Delete or redefine the logs and BSDS, and format page sets for the single queue manager (this deletes all its private objects and messages). On restart, the queue manager re-creates COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist, as do the messages on them, and can be used.

Point in time recovery of a queue sharing group

This is the alternative site disaster recovery scenario.

Shared objects are recovered to the point in time achieved by Db2 recovery (described in [A Db2 system fails](#)). Each queue manager can be recovered to a point in time achievable from the backup copies available at the alternative site.

Persistent messages can be used in queue sharing groups, and can be recovered using the MQSC RECOVER CFSTRUCT command. Note that this command recovers to the time of failure. However, there is no recovery of nonpersistent shared queue messages; they are lost unless you have made backup copies independently using the COPY function of the CSQUTIL utility program.

It is not necessary to try to restore each queue manager to the same point in time because there are no interdependencies between the local objects on different queue managers (which are what is actually being recovered), and the queue manager resynchronization with Db2 on restart creates or deletes COPY objects as necessary on a queue manager by queue manager basis.

Using the z/OS Automatic Restart Manager (ARM)

Use this topic to understand how you can use ARM to automatically restart your queue managers.

This section contains information about the following topics:

- [“What is the ARM?” on page 503](#)
- [“ARM policies” on page 504](#)
- [“Using ARM in an IBM MQ network” on page 505](#)

What is the ARM?

The z/OS Automatic Restart Manager (ARM) is a z/OS recovery function that can improve the availability of your queue managers. When a job or task fails, or the system on which it is running fails, ARM can restart the job or task without operator intervention.

If a queue manager or a channel initiator has failed, ARM restarts it on the same z/OS image. If z/OS, and hence a whole group of related subsystems and applications have failed, ARM can restart all the failed systems automatically, in a predefined order, on another z/OS image within the sysplex. This is called a *cross-system restart*.

Restart the channel initiator by ARM only in exceptional circumstances. If the queue manager is restarted by ARM, restart the channel initiator from the CSQINP2 initialization data set (see [“Using ARM in an IBM MQ network” on page 505](#)).

You can use ARM to restart a queue manager on a different z/OS image within the sysplex in the event of z/OS failure. The network implications of IBM MQ ARM restart on a different z/OS image are described in [“Using ARM in an IBM MQ network” on page 505](#).

To enable automatic restart:

- Set up an ARM couple data set.
- Define the automatic restart actions that you want z/OS to perform in an *ARM policy*.
- Start the ARM policy.

Also, IBM MQ must register with ARM at startup (this happens automatically).

Note: If you want to restart queue managers in different z/OS images automatically, you must define every queue manager as a subsystem in each z/OS image on which that queue manager might be restarted, with a sysplex wide unique four character subsystem name.

ARM couple data sets

Ensure that you define the couple data sets required for ARM, and that they are online and active before you start any queue manager for which you want ARM support. IBM MQ automatic ARM registration fails if the couple data sets are not available at queue manager startup. In this situation, IBM MQ assumes that the absence of the couple data set means that you do not want ARM support, and initialization continues.

See *z/OS MVS Setting up a Sysplex* for information about ARM couple data sets.

ARM policies

The Automatic Restart Manager policies are user-defined rules that control ARM functions that can control any restarts of a queue manager.

ARM functions are controlled by a user-defined *ARM policy*. Each z/OS image running a queue manager instance that is to be restarted by ARM must be connected to an ARM couple data set with an active ARM policy.

IBM provides a default ARM policy. You can define new policies, or override the policy defaults by using the *administrative data utility* (IXCMIAPU) provided with z/OS. *z/OS MVS Setting up a Sysplex* describes this utility, and includes full details of how to define an ARM policy.

Figure 36 on page 504 shows an example of an ARM policy. This sample policy restarts any queue manager within a sysplex, if either the queue manager failed, or a whole system failed.

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
RESTART_GROUP(DEFAULT)
ELEMENT(*)
RESTART_ATTEMPTS(0) /* Jobs not to be restarted by ARM */
RESTART_GROUP(GROUP1)
ELEMENT(SYSMQMGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Figure 36. Sample ARM policy

For more information see:

- [Defining an ARM policy](#)
- [Activating an ARM policy](#)
- [Registering with ARM](#)

Defining an ARM policy

Set up your ARM policy as follows:

- Define RESTART_GROUPS for each queue manager instance that also contain any CICS or IMS subsystems that connect to that queue manager instance. If you use a subsystem naming convention, you might be able to use the '?' and '*' wild-card characters in your element names to define RESTART_GROUPS with minimum definition effort.
- Specify TERMTYPE(ELEMTERM) for your channel initiators to indicate that they will be restarted only if the channel initiator has failed and the z/OS image has not failed.

- Specify `TERMTYPE(ALLTERM)` for your queue managers to indicate that they will be restarted if either the queue manager has failed or the z/OS image has failed.
- Specify `RESTART_METHOD(BOTH, PERSIST)` for both queue managers and channel initiators. This tells ARM to restart using the JCL it saved (after resolution of system symbols) during the last startup. It tells ARM to do this irrespective of whether the individual element failed, or the z/OS image failed.
- Accept the default values for all the other ARM policy options.

Activating an ARM policy

To start your automatic restart management policy, issue the following z/OS command:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

When the policy is started, all systems connected to the ARM couple data set use the same active policy. Use the `SETXCF STOP` command to disable automatic restarts.

Registering with ARM

IBM MQ registers automatically as an *ARM element* during queue manager startup (subject to ARM availability). It deregisters during its shutdown phase, unless requested not to.

At startup, the queue manager determines whether ARM is available. If it is, IBM MQ registers using the name `SYSMQMGR ssid`, where *ssid* is the four character queue manager name, and `SYSMQMGR` is the element type.

The `STOP QMGR MODE(QUIESCE)` and `STOP QMGR MODE(FORCE)` commands deregister the queue manager from ARM (if it was registered with ARM at startup). This prevents ARM restarting this queue manager. The `STOP QMGR MODE(RESTART)` command does not deregister the queue manager from ARM, so it is eligible for immediate automatic restart.

Each channel initiator address space determines whether ARM is available, and if so registers with the element name `SYSMQCH ssid`, where *ssid* is the queue manager name, and `SYSMQCH` is the element type.

The channel initiator is always deregistered from ARM when it stops normally, and remains registered only if it ends abnormally. The channel initiator is always deregistered if the queue manager fails.

Using ARM in an IBM MQ network

You can set up your queue manager so that the channel initiators and associated listeners are started automatically when the queue manager is restarted.

To ensure fully automatic queue manager restart on the same z/OS image for both LU 6.2 and TCP/IP communication protocols:

- Start your listeners automatically by adding the appropriate `START LISTENER` command to the `CSQINPX` data set.
- Start your channel initiator automatically by adding the appropriate `START CHINIT` command to the `CSQINP2` data set.

For restarting a queue manager with TCP/IP or LU6.2, see

- [“Restarting on a different z/OS image with TCP/IP” on page 506](#)
- [“Restarting on a different z/OS image with LU 6.2” on page 507](#)

See [Task 13: Customize the initialization input data sets](#) for information about the `CSQINP2` and `CSQINPX` data sets.

Restarting on a different z/OS image with TCP/IP

If you are using TCP/IP as your communication protocol, and you are using virtual IP addresses, you can configure these to recover on other z/OS images, allowing channels connecting to that queue manager to reconnect without any changes. Otherwise, you can reallocate a TCP/IP address after moving a queue manager to a different z/OS image only if you are using clusters or if you are connecting to a queue sharing group using a WLM dynamic Domain Name System (DNS) logical group name.

- [When using clustering](#)
- [When connecting to a queue sharing group](#)

When using clustering

z/OS ARM responds to a system failure by restarting the queue manager on a different z/OS image in the same sysplex; this system has a different TCP/IP address to the original z/OS image. The following explains how you can use IBM MQ clusters to reassign a queue manager's TCP/IP address after it has been moved by ARM restart to a different z/OS image.

When a client queue manager detects the queue manager failure (as a channel failure), it responds by reallocating suitable messages on its cluster transmission queue to a different server queue manager that hosts a different instance of the target cluster queue. However, it cannot reallocate messages that are bound to the original server by affinity constraints, or messages that are in doubt because the server queue manager failed during end-of-batch processing. To process these messages, do the following:

1. Allocate a different cluster-receiver channel name and a different TCP/IP port to each z/OS queue manager. Each queue manager needs a different port so that two systems can share a single TCP/IP stack on a z/OS image. One of these is the queue manager originally running on that z/OS image, and the other is the queue manager that ARM will restart on that z/OS image following a system failure. Configure each port on each z/OS image, so that ARM can restart any queue manager on any z/OS image.
2. Create a different channel initiator command input file (CSQINPX) for each queue manager and z/OS image combination, to be referenced during channel initiator startup.

Each CSQINPX file must include a START LISTENER PORT(port) command specific to that queue manager, and an ALTER CHANNEL command for a cluster-receiver channel specific to that queue manager and z/OS image combination. The ALTER CHANNEL command needs to set the connection name to the TCP/IP name of the z/OS image on which it is restarted. It must include the port number specific to the restarted queue manager as part of the connection name.

The start-up JCL of each queue manager can have a fixed data set name for this CSQINPX file, and each z/OS image must have a different version of each CSQINPX file on a non-shared DASD volume.

If an ARM restart occurs, IBM MQ advertises the changed channel definition to the cluster repository, which in turn publishes it to all the client queue managers that have expressed an interest in the server queue manager.

The client queue manager treats the server queue manager failure as a channel failure, and tries to restart the failed channel. When the client queue manager learns the new server connection-name, the channel restart reconnects the client queue manager to the restarted server queue manager. The client queue manager can then resynchronize its messages, resolve any in-doubt messages on the client queue manager's transmission queue, and normal processing can continue.

When connecting to a queue sharing group

When connecting to a queue sharing group through a TCP/IP dynamic Domain Name System (DNS) logical group name, the connection name in your channel definition specifies the logical group name of your queue sharing group, not the host name or IP address of a physical machine. When this channel starts, it connects to the dynamic DNS and is then connected to one of the queue managers in the queue sharing group. This process is explained in [Setting up communication for IBM MQ for z/OS using queue sharing groups](#).

In the unlikely event of an image failure, one of the following occurs:

- The queue managers on the failing image de-register from the dynamic DNS running on your sysplex. The channel responds to the connection failure by entering RETRYING state and then connects to the dynamic DNS running on the sysplex. The dynamic DNS allocates the inbound request to one of the remaining members of the queue sharing group that is still running on the remaining images.
- If no other queue manager in the queue sharing group is active and ARM restarts the queue manager and channel initiator on a different image, the group listener registers with dynamic DNS from this new image. This means that the logical group name (from the connection name field of the channel) connects to the dynamic DNS and is then connected to the same queue manager, now running on a different image. No change was required to the channel definition.

For this type of recovery to occur, the following points must be noted:

- On z/OS, the dynamic DNS runs on one of the z/OS images in the sysplex. If this image were to fail, the dynamic DNS needs to be configured so that there is a secondary name server active in the sysplex, acting as an alternative to the primary name server. Information about primary and secondary dynamic DNS servers can be found in the *OS/390® SecureWay CS IP Configuration* manual.
- The TCP/IP group listener might have been started on a particular IP address that might not be available on this z/OS image. If so, the listener might need to be started on a different IP address on the new image. If you are using virtual IP addresses, you can configure these to recover on other z/OS images so that no change to the START LISTENER command is required.

Restarting on a different z/OS image with LU 6.2

If you use only LU 6.2 communication protocols, carry out the following procedure to enable network reconnect after automatic restart of a queue manager on a different z/OS image within the sysplex:

- Define each queue manager within the sysplex with a unique subsystem name.
- Define each channel initiator within the sysplex with a unique LUNAME. This is specified in both the queue manager attributes and in the START LISTENER command.

Note: The LUNAME names an entry in the APPC side table, which in turn maps this to the actual LUNAME.

- Set up a shared APPC side table, which is referenced by each z/OS image within the sysplex. This should contain an entry for each channel initiator's LUNAME. See *z/OS MVS Planning: APPC/MVS Management* for information about this.
- Set up an APPCPM xx member of SYS1.PARMLIB for each channel initiator within the sysplex to contain an LUADD to activate the APPC side table entry for that channel initiator. These members should be shared by each z/OS image. The appropriate SYS1.PARMLIB member is activated by a z/OS command SET APPC= xx, which is issued automatically during ARM restart of the queue manager (and its channel initiator) on a different z/OS image, as described in the following text.
- Use the LU62ARM queue manager attribute to specify the xx suffix of this SYS1.PARMLIB member for each channel initiator. This causes the channel initiator to issue the required z/OS command SET APPC= xx to activate its LUNAME.

Define your ARM policy so that it restarts the channel initiator only if it fails while its z/OS image stays up; the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ command START CHINIT. Do not restart the channel initiator automatically if its z/OS image also fails, instead use commands in the CSQINP2 and CSQINPX data sets to start the channel initiator and listeners.

Recovering units of work manually

You can manually recover units of work CICS, IMS, RRS, or other queue managers in a queue sharing group. You can use queue manager commands to display the status of the units of work associated with each connection to the queue manager.

This topic contains information about the following subjects:

- [“Displaying connections and threads” on page 508](#)
- [“Recovering CICS units of recovery manually” on page 508](#)
- [“Recovering IMS units of recovery manually” on page 512](#)
- [“Recovering RRS units of recovery manually” on page 513](#)
- [“Recovering units of recovery on another queue manager in the queue sharing group” on page 514](#)

Displaying connections and threads

You can use the `DISPLAY CONN` command to get information about connections to queue managers and their associated units of work. You can display active units of work to see what is currently happening, or to see what needs to be terminated to allow the queue manager to shut down, and you can display unresolved units of work to help with recovery.

Active units of work

To display only active units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

Unresolved units of work

An unresolved unit of work, also known as an "in-doubt thread", is one that is in the second pass of the two-phase commit operation. Resources are held in IBM MQ on its behalf. To display unresolved units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

External intervention is needed to resolve the status of unresolved units of work. This might only involve starting the recovery coordinator (CICS, IMS, or RRS) or might involve more, as described in the following sections.

Recovering CICS units of recovery manually

Use this topic to understand what happens when the CICS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

What happens when the CICS adapter restarts

Whenever a connection is broken, the adapter has to go through a *restart phase* during the *reconnect process*. The restart phase resynchronizes resources. Resynchronization between CICS and IBM MQ enables in-doubt units of work to be identified and resolved.

Resynchronization can be caused by:

- An explicit request from the distributed queuing component
- An implicit request when a connection is made to IBM MQ

If the resynchronization is caused by connecting to IBM MQ, the sequence of events is:

1. The connection process retrieves a list of in-doubt units of work (UOW) IDs from IBM MQ.
2. The UOW IDs are displayed on the console in CSQC313I messages.
3. The UOW IDs are passed to CICS.
4. CICS initiates a resynchronization task (CRSY) for each in-doubt UOW ID.

5. The result of the task for each in-doubt UOW is displayed on the console.

You need to check the messages that are displayed during the connect process:

CSQC313I

Shows that a UOW is in doubt.

CSQC400I

Identifies the UOW and is followed by one of these messages:

- CSQC402I or CSQC403I shows that the UOW was resolved successfully (committed or backed out).
- CSQC404E, CSQC405E, CSQC406E, or CSQC407E shows that the UOW was not resolved.

CSQC409I

Shows that all UOWs were resolved successfully.

CSQC408I

Shows that not all UOWs were resolved successfully.

CSQC314I

Warns that UOW IDs highlighted with a * are not resolved automatically. These UOWs must be resolved explicitly by the distributed queuing component when it is restarted.

Figure 37 on page 509 shows an example set of restart messages displayed on the z/OS console.

```
CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCQCON CONNECT received from TERMID=PB62 TRANID=CKCN
+CSQC303I VICIC1 CSQCCON CSQCSERV loaded. Entry point is 850E8918
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F0E2178D25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F055B2AC25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6EFFF60D425 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F07AB56D22 is in doubt
+CSQC307I VICIC1 CSQCCON Successful connection to subsystem VC2
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAD18) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAA10) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BA708) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAE88) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAB80) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA878) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA570) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA268) connect
successful
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6EFFF60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully
```

Figure 37. Example restart messages

The total number of CSQC313I messages should equal the total number of CSQC402I plus CSQC403I messages. If the totals are not equal, there are UOWs that the connection process cannot resolve. Those UOWs that cannot be resolved are caused by problems with CICS (for example, a cold start) or with IBM MQ, or by distributing queuing. When these problems have been fixed, you can initiate another resynchronization by disconnecting and then reconnecting.

Alternatively, you can resolve each outstanding UOW yourself using the RESOLVE INDOUBT command and the UOW ID shown in message CSQC400I. You must then initiate a disconnect and a connect to clean up

the *unit of recovery descriptors* in CICS. You need to know the correct outcome of the UOW to resolve UOWs manually.

All messages that are associated with unresolved UOWs are locked by IBM MQ and no Batch, TSO, or CICS task can access them.

If CICS fails and an emergency restart is necessary, *do not* vary the GENERIC APPLID of the CICS system. If you do and then reconnect to IBM MQ, data integrity with IBM MQ cannot be guaranteed. This is because IBM MQ treats the new instance of CICS as a different CICS (because the APPLID is different). In-doubt resolution is then based on the wrong CICS log.

How to resolve CICS units of recovery manually

If the adapter ends abnormally, CICS and IBM MQ build in-doubt lists either dynamically or during restart, depending on which subsystem caused the abend.

Note: If you use the DFH\$INDB sample program to show units of work, you might find that it does not always show IBM MQ UOWs correctly.

When CICS connects to IBM MQ, there might be one or more units of recovery that have not been resolved.

One of the following messages is sent to the console:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

For details of what these messages mean, see the [CICS adapter and Bridge messages](#) messages.

CICS retains details of units of recovery that were not resolved during connection startup. An entry is purged when it no longer appears on the list presented by IBM MQ.

Any units of recovery that CICS cannot resolve must be resolved manually using IBM MQ commands. This manual procedure is rarely used within an installation, because it is required only where operational errors or software problems have prevented automatic resolution. *Any inconsistencies found during in-doubt resolution must be investigated.*

To resolve the units of recovery:

1. Obtain a list of the units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```

CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN (BC85772CBE3E0001)
EXTCONN (C3E2D8C3C7D9F0F940404040404040)
TYPE (CONN)
CONNOPTS (
MQCNO_STANDARD_BINDING
)
UOWLOGDA (2005-02-04)
UOWLOGTI (10.17.44)
UOWSTDA (2005-02-04)
UOWSTTI (10.17.44)
UOWSTATE (UNRESOLVED)
NID (IYRCSQ1 .BC8571519B60222D)
EXTURID (BC8571519B60222D)
QMURID (0000002BDA50)
URTYPE (CICS)
USERID (MQTEST)
APPLTAG (IYRCSQ1)
ASID (0000)
APPLTYPE (CICS)
TRANSID (GP02)
TASKNO (0000096)
END CONN DETAILS

```

For CICS connections, the NID consists of the CICS applid and a unique number provided by CICS at the time the syncpoint log entries are written. This unique number is stored in records written to both the CICS system log and the IBM MQ log at syncpoint processing time. This value is referred to in CICS as the *recovery token*.

2. Scan the CICS log for entries related to a particular unit of recovery.

Look for a PREPARE record for the task-related installation where the recovery token field (JCSRMTKN) equals the value obtained from the network ID. The network ID is supplied by IBM MQ in the DISPLAY CONN command output.

The PREPARE record in the CICS log for the units of recovery provides the CICS task number. All other entries on the log for this CICS task can be located using this number.

You can use the CICS journal print utility DFHJUP when scanning the log. For details of using this program, see the *CICS Operations and Utilities Guide*.

3. Scan the IBM MQ log for records with the NID related to a particular unit of recovery. Then use the URID from this record to obtain the rest of the log records for this unit of recovery.

When scanning the IBM MQ log, note that the IBM MQ startup message CSQJ001I provides the start RBA for this session.

The print log records program (CSQ1LOGP) can be used for that purpose.

4. If you need to, do in-doubt resolution in IBM MQ.

IBM MQ can be directed to take the recovery action for a unit of recovery using an IBM MQ [RESOLVE INDOUBT](#) command.

To recover all threads associated with a specific *connection-name*, use the NID(*) option.

The command produces one of the following messages showing whether the thread is committed or backed out:

```

CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED

```

When performing in-doubt resolution, CICS and the adapter are not aware of the commands to IBM MQ to commit or back out units of recovery, because only IBM MQ resources are affected. However, CICS keeps details about the in-doubt threads that could not be resolved by IBM MQ. This information is purged

either when the list presented is empty, or when the list does not include a unit of recovery of which CICS has details.

Recovering IMS units of recovery manually

Use this topic to understand what happens when the IMS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

What happens when the IMS adapter restarts

Whenever the connection to IBM MQ is restarted, either following a queue manager restart or an IMS / START SUBSYS command, IMS initiates the following resynchronization process:

1. IMS presents the list of unit of work (UOW) IDs that it believes are in doubt to the IBM MQ IMS adapter one at a time with a resolution parameter of Commit or Backout.
2. The IMS adapter passes the resolution request to IBM MQ and reports the result back to IMS.
3. Having processed all the IMS resolution requests, the IMS adapter gets from IBM MQ a list of all UOWs that IBM MQ still holds in doubt that were initiated by the IMS system. These are reported to the IMS master terminal in message CSQQ008I.

Note: While a UOW is in doubt, any associated IBM MQ message is locked by IBM MQ and is not available to any application.

How to resolve IMS units of recovery manually

When IMS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved.

If IBM MQ has in-doubt units of recovery that IMS did not resolve, the following message is issued at the IMS master terminal:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

If this message is issued, IMS was either cold-started or it was started with an incomplete log tape. This message can also be issued if IBM MQ or IMS terminates abnormally because of a software error or other subsystem failure.

After receiving the CSQQ008I message:

- The connection remains active.
- IMS applications can still access IBM MQ resources.
- Some IBM MQ resources remain locked out.

If the in-doubt thread is not resolved, IMS message queues can start to build up. If the IMS queues fill to capacity, IMS terminates. You must be aware of this potential difficulty, and you must monitor IMS until the in-doubt units of recovery are fully resolved.

Recovery procedure

Use the following procedure to recover the IMS units of work:

1. Force the IMS log closed, using /SWI OLDS, and then archive the IMS log. Use the utility, DFSERA10, to print the records from the previous IMS log tape. Type X '3730' log records indicate a phase-2 commit request and type X '38' log records indicate an abort request. Record the requested action for the last transaction in each dependent region.
2. Run the DL/I batch job to back out each PSB involved that has not reached a commit point. The process might take some time because transactions are still being processed. It might also lock up

a number of records, which could affect the rest of the processing and the rest of the message queues.

3. Produce a list of the in-doubt units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3E2C5C3F240404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-15)
UOWLOGTI(16.39.43)
UOWSTDA(2005-02-15)
UOWSTTI(16.39.43)
UOWSTATE(UNRESOLVED)
NID(IM8F .BC45A794D3810344)
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0000)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

For IMS, the NID consists of the IMS connection name and a unique number provided by IMS. The value is referred to in IMS as the *recovery token*. For more information, see the [IMS documentation](#).

4. Compare the NIDs (IMSID plus OASN in hexadecimal) displayed in the DISPLAY THREAD messages with the OASNs (4 bytes decimal) shown in the DFSERA10 output. Decide whether to commit or back out.
5. Perform in-doubt resolution in IBM MQ with the [RESOLVE INDOUBT](#) command, as follows:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

To recover all threads associated with *connection-name*, use the NID(*) option. The command results in one of the following messages to indicate whether the thread is committed or backed out:

```
CSQV414I THREAD network-id COMMIT SCHEDULED
CSQV415I THREAD network-id BACKOUT SCHEDULED
```

When performing in-doubt resolution, IMS and the adapter are not aware of the commands to IBM MQ to commit or back out in-doubt units of recovery because only IBM MQ resources are affected.

Recovering RRS units of recovery manually

Use this topic to understand the how to determine if there are in-doubt RRS units of recovery, and how to manually resolve those units of recovery.

When RRS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved. If IBM MQ has in-doubt units of recovery that RRS did not resolve, one of the following messages is issued at the z/OS console:

- CSQ3011I
- CSQ3013I

- CSQ3014I
- CSQ3016I

Both IBM MQ and RRS provide tools to display information about in-doubt units of recovery, and techniques for manually resolving them.

In IBM MQ, use the DISPLAY CONN command to display information about in-doubt IBM MQ threads. The output from the command includes RRS unit of recovery IDs for those IBM MQ threads that have RRS as a coordinator. This can be used to determine the outcome of the unit of recovery.

Use the RESOLVE INDOUBT command to resolve the IBM MQ in-doubt thread manually. This command can be used to either commit or back out the unit of recovery after you have determined what the correct decision is.

Recovering units of recovery on another queue manager in the queue sharing group

Use this topic to identify, and manually recover units of recovery on other queue managers in a queue sharing group.

If a queue manager that is a member of a queue sharing group fails and cannot be restarted, other queue managers in the group can perform peer recovery, and take over from it. However, the queue manager might have in-doubt units of recovery that cannot be resolved by peer recovery because the final disposition of that unit of recovery is known only to the failed queue manager. These units of recovery are resolved when the queue manager is eventually restarted, but until then, they remain in doubt.

This means that certain resources (for example, messages) might be locked, making them unavailable to other queue managers in the group. In this situation, you can use the DISPLAY THREAD command to display these units of work on the inactive queue manager. If you want to resolve these units of recovery manually to make the messages available to other queue managers in the group, you can use the RESOLVE INDOUBT command.

When you issue the DISPLAY THREAD command to display units of recovery that are in doubt, you can use the QMNAME keyword to specify the name of the inactive queue manager. For example, if you issue the following command:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

You receive the following messages:

```
CSQV436I +CSQ1 INDOUBT THREADS FOR QM01 -
NAME  THREAD-XREF  URID  NID
USER1  0000000000000000000000000000 CSQ:0001.0
USER2  0000000000000000000000000000 CSQ:0002.0
DISPLAY THREAD REPORT COMPLETE
```

If the queue manager specified is active, IBM MQ does not return information about in-doubt threads, but issues the following message:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Use the IBM MQ command RESOLVE INDOUBT to resolve the in-doubt threads manually. Use the QMNAME keyword to specify the name of the inactive queue manager in the command.

This command can be used to commit or back out the unit of recovery. The command resolves the shared portion of the unit of recovery only; any local messages are unaffected and remain locked until the queue manager restarts, or reconnects to CICS, IMS, or RRS batch.

z/OS IBM MQ and IMS

IBM MQ provides two components to interface with IMS, the IBM MQ - IMS adapter, and the IBM MQ - IMS bridge. These components are commonly called the IMS adapter, and the IMS bridge.

z/OS Operating the IMS adapter

Use this topic to understand how to operate the IMS adapter, which connects IBM MQ to IMS systems.

Note: The IMS adapter does not incorporate any operations and control panels.

This topic contains the following sections:

- [“Controlling IMS connections” on page 515](#)
- [“Connecting from the IMS control region” on page 515](#)
- [“Displaying in-doubt units of recovery” on page 517](#)
- [“Controlling IMS dependent region connections” on page 519](#)
- [“Disconnecting from IMS” on page 521](#)
- [“Controlling the IMS trigger monitor” on page 522](#)

z/OS Controlling IMS connections

Use this topic to understand the IMS operator commands which control and monitor the connection to IBM MQ.

IMS provides the following operator commands to control and monitor the connection to IBM MQ:

/CHANGE SUBSYS

Deletes an in-doubt unit of recovery from IMS.

/DISPLAY OASN SUBSYS

Displays outstanding recovery elements.

/DISPLAY SUBSYS

Displays connection status and thread activity.

/START SUBSYS

Connects the IMS control region to a queue manager.

/STOP SUBSYS

Disconnects IMS from a queue manager.

/TRACE

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA® Operator's Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

z/OS Connecting from the IMS control region

Use this topic to understand the mechanisms available to connect from IMS to IBM MQ.

IMS makes one connection from its control region to each queue manager that uses IMS. IMS must be enabled to make the connection in one of these ways:

- Automatically during either:
 - A cold start initialization.
 - A warm start of IMS, if the IBM MQ connection was active when IMS was shut down.
- In response to the IMS command:

```
/START SUBSYS sysid
```

where *sysid* is the queue manager name.

The command can be issued regardless of whether the queue manager is active.

The connection is not made until the first IBM MQ API call to the queue manager is made. Until that time, the IMS command /DIS SUBSYS shows the status as 'NOT CONN'.

The order in which you start IMS and the queue manager is not significant.

IMS cannot re-enable the connection to the queue manager automatically if the queue manager is stopped with a STOP QMGR command, the IMS command /STOP SUBSYS, or an abnormal end. Therefore, you must make the connection by using the IMS command /START SUBSYS.

If an IMS command is seen in the queue manager console log similar to this:

```
MODIFY IMS*,SS*
```

check the IMS master log and ensure that IBM MQ has RACF authority to issue IMS Adapter MODIFY commands.

Initializing the adapter and connecting to the queue manager

The adapter is a set of modules loaded into the IMS control and dependent regions, using the IMS external Subsystem Attach Facility.

This procedure initializes the adapter and connects to the queue manager:

1. Read the subsystem member (SSM) from IMS.PROCLIB. The SSM chosen is an IMS EXEC parameter. There is one entry in the member for each queue manager to which IMS can connect. Each entry contains control information about an IBM MQ adapter.

2. Load the IMS adapter.

Note: IMS loads one copy of the adapter modules for each IBM MQ instance that is defined in the SSM member.

3. Attach the external subsystem task for IBM MQ.

4. Run the adapter with the CTL EXEC parameter (IMSID) as the connection name.

The process is the same whether the connection is part of initialization or a result of the IMS command /START SUBSYS.

If the queue manager is active when IMS tries to make the connection, the following messages are sent:

- to the z/OS console:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- to the IMS master terminal:

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```


When IMS tries to make the connection and *the queue manager is not active*, the following messages are sent to the IMS master terminal each time an application makes an MQI call:

```
CSQQ001I IMS/TM imsid not connected to queue manager ssnm.  
Notify message accepted  
DFS3607I MQM1 SUBSYSTEM ID EXIT FAILURE, FC = 0286, RC = 08,  
JOBNAME = IMSEMPR1
```

If you get DFS3607I messages when you start the connection to IMS or on system startup, this indicates that the queue manager is not available. To prevent a large number of messages being generated, you must do one of the following:

1. Start the relevant queue manager.
2. Issue the IMS command:

```
/STOP SUBSYS
```

so that IMS does not expect to connect to the queue manager.

If you do neither, a DFS3607I message and the associated CSQQ001I message are issued each time a job is scheduled in the region and each time a connection request to the queue manager is made by an application.

Thread attachment

In an MPP or IFP region, IMS makes a thread connection when the first application program is scheduled into that region, even if that application program does not make an IBM MQ call. In a BMP region, the thread connection is made when the application makes its first IBM MQ call (MQCONN or MQCONNX). This thread is retained for the duration of the region or until the connection is stopped.

For both the message driven and non-message driven regions, the recovery thread cross-reference identifier, *Thread-xref*, associated with the thread is:

```
PSTid + PSBname
```

where:

PSTid

Partition specification table region identifier

PSBname

Program specification block name

You can use connection IDs as unique identifiers in IBM MQ commands, in which case IBM MQ automatically inserts these IDs into any operator message that it generates.

Displaying in-doubt units of recovery

You can display in-doubt units of recovery and attempt to recover them.

The operational steps used to list and recover in-doubt units of recovery in this topic are for relatively simple cases only. If the queue manager ends abnormally while connected to IMS, IMS might commit or back out work without IBM MQ being aware of it. When the queue manager restarts, that work is termed *in doubt*. A decision must be made about the status of the work.

To display a list of in-doubt units of recovery, issue the command:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ responds with a message like the following:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-11-02)
UOWLOGTI(12.27.58)
UOWSTDA(2004-11-02)
UOWSTTI(12.27.58)
UOWSTATE(UNRESOLVED)
NID(CSQ1CHIN.BC0F5F1C86FC0766)
EXTURID(000000000000001F000000007472616E5F6964547565204E6F762020...)
QMURID(0000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

For an explanation of the attributes in this message, see the description of the [DISPLAY CONN](#) command.

Recovering in-doubt units of recovery

To recover in-doubt units of recovery, issue this command:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

where:

connection-name

The IMS system ID.

ACTION

Indicates whether to commit (COMMIT) or back out (BACKOUT) this unit of recovery.

net-node.number

The associated net-node.number.

When you have issued the RESOLVE INDOUBT command, one of the following messages is displayed:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

Resolving residual recovery entries

At given times, IMS builds a list of residual recovery entries (RREs). RREs are units of recovery about which IBM MQ might be in doubt. They arise in several situations:

- If the queue manager is not active, IMS has RREs that cannot be resolved until the queue manager is active. These RREs are not a problem.
- If the queue manager is active and connected to IMS, and if IMS backs out the work that IBM MQ has committed, the IMS adapter issues message CSQQ010E. If the data in the two systems must be consistent, there is a problem. For information about resolving this problem, see [“Recovering IMS units of recovery manually” on page 512](#).
- If the queue manager is active and connected to IMS, there might still be RREs even though no messages have informed you of this problem. After the IBM MQ connection to IMS has been established, you can issue the following IMS command to find out if there is a problem:

```
/DISPLAY OASN SUBSYS sysid
```

To purge the RRE, issue one of the following IMS commands:

```
/CHANGE SUBSYS sysid RESET  
/CHANGE SUBSYS sysid RESET OASN nnnn
```

where *nnnn* is the originating application sequence number listed in response to your +CSQ1 DISPLAY command. This is the schedule number of the program instance, giving its place in the sequence of invocations of that program since the last IMS cold start. IMS cannot have two in-doubt units of recovery with the same schedule number.

These commands reset the status of IMS ; they do not result in any communication with IBM MQ.

Controlling IMS dependent region connections

You can control, monitor, and, when necessary, terminate connections between IMS and IBM MQ.

Controlling IMS dependent region connections involves the following activities:

- [Connecting from dependent regions](#)
- [Region error options](#)
- [Monitoring the activity on connections](#)
- [Disconnecting from dependent regions](#)

Connecting from dependent regions

The IMS adapter used in the control region is also loaded into dependent regions. A connection is made from each dependent region to IBM MQ. This connection is used to coordinate the commitment of IBM MQ and IMS work. To initialize and make the connection, IMS does the following:

1. Reads the subsystem member (SSM) from IMS.PROCLIB.

A subsystem member can be specified on the dependent region EXEC parameter. If it is not specified, the control region SSM is used. If the region is never likely to connect to IBM MQ, to avoid loading the adapter, specify a member with no entries.

2. Loads the IBM MQ adapter.

For a batch message program, the load is not done until the application issues its first messaging command. At that time, IMS tries to make the connection.

For a message-processing program region or IMS fast-path region, the attempt is made when the region is initialized.

Region error options

If the queue manager is not active, or if resources are not available when the first messaging command is sent from application programs, the action taken depends on the error option specified on the SSM entry. The options are:

R

The appropriate return code is sent to the application.

Q

The application ends abnormally with abend code U3051. The input message is re-queued.

A

The application ends abnormally with abend code U3047. The input message is discarded.

Monitoring the activity on connections

A thread is established from a dependent region when an application makes its first successful IBM MQ request. You can display information about connections and the applications currently using them by issuing the following command from IBM MQ:

```
+CSQ1 DISPLAY CONN(*) ALL
```

The command produces a message like the following:

```
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-12-15)
UOWLOGTI(16.39.43)
UOWSTDA(2004-12-15)
UOWSTTI(16.39.43)
UOWSTATE(ACTIVE)
NID( )
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0049)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

For the control region, *thread-xref* is the special value CONTROL. For dependent regions, it is the PSTid concatenated with the PSBname. *auth-id* is either the user field from the job card, or the ID from the z/OS started procedures table.

For an explanation of the displayed list, see the description of message CSQV402I in the [IBM MQ for z/OS 메시지, 완료 및 이유 코드](#) documentation.

IMS provides a display command to monitor the connection to IBM MQ. It shows which program is active on each dependent region connection, the LTERM user name, and the control region connection status. The command is:

```
/DISPLAY SUBSYS name
```

The status of the connection between IMS and IBM MQ is shown as one of:

```
CONNECTED  
NOT CONNECTED  
CONNECT IN PROGRESS  
STOPPED  
STOP IN PROGRESS  
INVALID SUBSYSTEM NAME= name  
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING
```

The thread status from each dependent region is one of the following:

```
CONN  
CONN, ACTIVE (includes LTERM of user)
```

Disconnecting from dependent regions

To change values in the SSM member of IMS.PROCLIB, you disconnect a dependent region. To do this, you must:

1. Issue the IMS command:

```
/STOP REGION
```

2. Update the SSM member.
3. Issue the IMS command:

```
/START REGION
```

Disconnecting from IMS

The connection is ended when either IMS or the queue manager terminates. Alternatively, the IMS master terminal operator can explicitly break the connection.

To terminate the connection between IMS and IBM MQ, use the following IMS command:

```
/STOP SUBSYS sysid
```

The command sends the following message to the terminal that issued it, typically the master terminal operator (MTO):

```
DFS058I STOP COMMAND IN PROGRESS
```

The IMS command:

```
/START SUBSYS sysid
```

is required to reestablish the connection.

Note: The IMS command /STOP SUBSYS is not completed if an IMS trigger monitor is running.

Controlling the IMS trigger monitor

You can use the CSQQTRMN transaction to stop, and start the IMS trigger monitor.

The IMS trigger monitor (the CSQQTRMN transaction) is described in the [Setting up the IMS trigger monitor](#).

To control the IMS trigger monitor see:

- [Starting CSQQTRMN](#)
- [Stopping CSQQTRMN](#)

Starting CSQQTRMN

1. Start a batch-oriented BMP that runs the program CSQQTRMN for each initiation queue you want to monitor.
2. Modify your batch JCL to add a DDname of CSQQUT1 that points to a data set containing the following information:

```
QMGRNAME=q_manager_name    Comment: queue manager name
INITQUEUEUENAME=init_q_name  Comment: initiation queue name
LTERM=lterm                  Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES         Comment: Send error messages to console
```

where:

q_manager_name	The name of the queue manager (if this is blank, the default nominated in CSQQDEFV is assumed)
init_q_name	The name of the initiation queue to be monitored
lterm	The IMS LTERM name for the destination of error messages (if this is blank, the default value is MASTER).
CONSOLEMESSAGES= YES	Requests that messages sent to the nominated IMS LTERM are also sent to the z/OS console. If this parameter is omitted or misspelled, the default is NOT to send messages to the console.

3. Add a DD name of CSQQUT2 if you want a printed report of the processing of CSQQUT1 input.

Note:

1. The data set CSQQUT1 is defined with LRECL=80. Other DCB information is taken from the data set. The DCB for data set CSQQUT2 is RECFM=VBA and LRECL=125.

2. You can put only one keyword on each record. The keyword value is delimited by the first blank following the keyword; this means that you can include comments. An asterisk in column 1 means that the whole input record is a comment.
3. If you misspell either of the QMGRNAME or LTERM keywords, CSQQTRMN uses the default for that keyword.
4. Ensure that the subsystem is started in IMS (by the /START SUBSYS command) before submitting the trigger monitor BMP job. If it is not started, your trigger monitor job terminates with abend code U3042.

Stopping CSQQTRMN

Once started, CSQQTRMN runs until either the connection between IBM MQ and IMS is broken due to one of the following events:

- the queue manager ending
- IMS ending

or a z/OS STOP **jobname** command is entered.

Controlling the IMS bridge

Use this topic to understand the IMS commands that you can use to control the IMS bridge.

There are no IBM MQ commands to control the IBM MQ-IMS bridge. However, you can stop messages being delivered to IMS in the following ways:

- For non-shared queues, by using the ALTER QLOCAL(xxx) GET(DISABLED) command for all bridge queues.
- For clustered queues, by using the SUSPEND QMGR CLUSTER(xxx) command. This is effective only when another queue manager is also hosting the clustered bridge queue.
- For clustered queues, by using the SUSPEND QMGR FACILITY(IMSBRIDGE) command. No further messages are sent to IMS, but the responses for any outstanding transactions are received from IMS.

To start sending messages to IMS again, issue the RESUME QMGR FACILITY(IMSBRIDGE) command.

You can also use the MQSC command DISPLAY SYSTEM to display whether the bridge is suspended.

See [MQSC commands](#) for details of these commands.

For further information see:

- [“Starting and stopping the IMS bridge” on page 523](#)
- [“Controlling IMS connections” on page 524](#)
- [Controlling bridge queues](#)
- [“Resynchronizing the IMS bridge” on page 525](#)
- [Working with tpipe names](#)
- [Deleting messages from IMS](#)
- [Deleting tpipes](#)
- [“IMS Transaction Expiration” on page 527](#)

Starting and stopping the IMS bridge

Start the IBM MQ bridge by starting OTMA. Either use the IMS command:

```
/START OTMA
```


or start it automatically by specifying OTMA=YES in the IMS system parameters. If OTMA is already started, the bridge starts automatically when queue manager startup has completed. An IBM MQ event message is produced when OTMA is started.

Use the IMS command:

```
/STOP OTMA
```

to stop OTMA communication. When this command is issued, an IBM MQ event message is produced.

Controlling IMS connections

IMS provides these operator commands to control and monitor the connection to IBM MQ:

/DEQUEUE TMEMBER *tmember* TPIPE *tpipe*

Removes messages from a Tpipe. Specify PURGE to remove all messages or PURGE1 to remove the first message only.

/DISPLAY OTMA

Displays summary information about the OTMA server and clients, and client status.

/DISPLAY TMEMBER *name*

Displays information about an OTMA client.

/DISPLAY TRACE TMEMBER *name*

Displays information about what is being traced.

/SECURE OTMA

Sets security options.

/START OTMA

Enables communications through OTMA.

/START TMEMBER *tmember* TPIPE *tpipe*

Starts the named Tpipe.

/STOP OTMA

Stops communications through OTMA.

/STOP TMEMBER *tmember* TPIPE *tpipe*

Stops the named Tpipe.

/TRACE

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA Operators Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

Controlling bridge queues

To stop communicating with the queue manager with XCF member name *tmember* through the bridge, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE ALL
```

To resume communication, issue the following IMS command:

```
/START TMEMBER tmember TPIPE ALL
```

The Tpipes for a queue can be displayed using the MQ DISPLAY QUEUE command.

To stop communication with the queue manager on a single Tpipe, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE tpipe
```

One or two Tpipes are created for each active bridge queue, so issuing this command stops communication with the IBM MQ queue. To resume communication, use the following IMS command :

```
/START TMEMBER tmember TPIPE tpipe
```

Alternatively, you can alter the attributes of the IBM MQ queue to make it get inhibited.

Resynchronizing the IMS bridge

The IMS bridge is automatically restarted whenever the queue manager, IMS, or OTMA are restarted.

The first task undertaken by the IMS bridge is to resynchronize with IMS. This involves IBM MQ and IMS checking sequence numbers on every synchronized Tpipe. A synchronized Tpipe is used when persistent messages are sent to IMS from an IBM MQ - IMS bridge queue using commit mode zero (commit-then-send).

If the bridge cannot resynchronize with IMS, the IMS sense code is returned in message CSQ2023E and the connection to OTMA is stopped. If the bridge cannot resynchronize with an individual IMS Tpipe, the IMS sense code is returned in message CSQ2025E and the Tpipe is stopped. If a Tpipe has been cold started, the recoverable sequence numbers are automatically reset to 1.

If the bridge discovers mismatched sequence numbers when resynchronizing with a Tpipe, message CSQ2020E is issued. Use the IBM MQ command RESET TPIPE to initiate resynchronization with the IMS Tpipe. You need to provide the XCF group and member name, and the name of the Tpipe; this information is provided by the message.

You can also specify:

- A new recoverable sequence number to be set in the Tpipe for messages sent by IBM MQ, and to be set as the partner's receive sequence number. If you do not specify this, the partner's receive sequence number is set to the current IBM MQ send sequence number.
- A new recoverable sequence number to be set in the Tpipe for messages received by IBM MQ, and to be set as the partner's send sequence number. If you do not specify this, the partner's send sequence number is set to the current IBM MQ receive sequence number.

If there is an unresolved unit of recovery associated with the Tpipe, this is also notified in the message. Use the IBM MQ command RESET TPIPE to specify whether to commit the unit of recovery, or back it out. If you commit the unit of recovery, the batch of messages has already been sent to IMS, and is deleted from the bridge queue. If you back the unit of recovery out, the messages are returned to the bridge queue, to be later sent to IMS.

Commit mode 1 (send-then-commit) Tpipes are not synchronized.

Considerations for Commit mode 1 transactions

In IMS, commit mode 1 (CM1) transactions send their output replies before sync point.

A CM1 transaction might not be able to send its reply, for example because:

- The Tpipe on which the reply is to be sent is stopped
- OTMA is stopped
- The OTMA client (that is, the queue manager) has gone away
- The reply-to queue and dead-letter queue are unavailable

For these reasons, the IMS application sending the message pseudo-abends with code U0119. The IMS transaction and program are not stopped in this case.

These reasons often prevent messages being sent into IMS, as well as replies being delivered from IMS. A U0119 abend can occur if:

- The Tpipe, OTMA, or the queue manager is stopped while the message is in IMS
- IMS replies on a different Tpipe to the incoming message, and that Tpipe is stopped
- IMS replies to a different OTMA client, and that client is unavailable.

Whenever a U0119 abend occurs, both the incoming message to IMS and the reply messages to IBM MQ are lost. If the output of a CMO transaction cannot be delivered for any of these reasons, it is queued on the Tpipe within IMS.

Working with tpipe names

Many of the commands used to control the IBM MQ - IMS bridge require the *tpipe* name. Use this topic to understand how you can find further details of the tpipe name.

You need *tpipe* names for many of the commands that control the IBM MQ - IMS bridge. You can get the tpipe names from DISPLAY QUEUE command and note the following points:

- tpipe names are assigned when a local queue is defined
- a local queue is given two tpipe names, one for sync and one for non-sync
- tpipe names will not be known to IMS until after some communication between IMS and IBM MQ specific to that particular local queue takes place
- For a tpipe to be available for use by the IBM MQ - IMS bridge its associated queue must be assigned to a Storage Class that has the correct XCF group and member name fields completed

Deleting messages from IMS

A message that is destined for IBM MQ through the IMS bridge can be deleted if the Tmember/Tpipe is stopped. To delete one message for the queue manager with XCF member name *tmember*, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

To delete all the messages on the Tpipe, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

Deleting tpipes

You cannot delete IMS tpipes yourself. They are deleted by IMS at the following times:

- Synchronized tpipes are deleted when IMS is cold started.

- Non-synchronized tpipes are deleted when IMS is restarted.

IMS Transaction Expiration

An expiration time is associated with a transaction; any IBM MQ message can have an expiration time associated with it. The expiration interval is passed from the application, to IBM MQ, using the MQMD.Expiry field. The time is the duration of a message before it expires, expressed as a value in tenths of a second. An attempt to perform the MQGET of a message, later than it has expired, results in the message being removed from the queue and expiry processing performed. The expiration time decreases as a message flows between queue managers on an IBM MQ network. When an IMS message is passed across the IMS bridge to OTMA, the remaining message expiry time is passed to OTMA as a transaction expiration time.

If a transaction has an expiration time specified, OTMA expires the input transactions in three different places in IMS:

- input message receiving from XCF
- input message enqueueing time
- application GU time

No expiration is performed after the GU time.

The transaction EXPRTIME can be provided by:

- IMS transaction definition
- IMS OTMA message header
- IMS DFSINSX0 user exit
- IMS CREATE or UPDATE TRAN commands

IMS indicates that it has expired a transaction by abending a transaction with 0243, and issuing a message. The message issued is either DFS555I in the non-shared-queues environment, or DFS2224I in the shared-queues environment.

z/OS

Operating Advanced Message Security on z/OS

The Advanced Message Security address space accepts commands using the z/OS MODIFY command.

Procedure

- Modify Advanced Message Security on z/OS.

To enter commands for the Advanced Message Security (AMS) address space, use the z/OS MODIFY command.

For example:

```
F qmgrAMS, cmd
```

where *qmgr* is the prefix of the started task name.

The following table describes the MODIFY commands that are accepted:

Table 29. Advanced Message Security address space MODIFY commands		
Command	Option	Description
DISPLAY		Display version information

Table 29. Advanced Message Security address space MODIFY commands (continued)		
Command	Option	Description
REFRESH	KEYRING POLICY ALL	Refresh the key ring certificates, security policies, or both.
SMFAUDIT	SUCCESS FAILURE ALL	Set whether SMF auditing is required when AMS successfully protects or unprotects messages, when AMS fails to protect or unprotect messages, or both.
SMFTYPE	0 - 255	Set the SMF record type to be generated when AMS protects or unprotects messages. To disable SMF auditing specify a record type of 0.

Note: To specify an option it must be separated by a comma. For example:

```
F qmgrAMSM,REFRESH KEYRING
F qmgrAMSM,SMFAUDIT ALL
F qmgrAMSM,SMFTYPE 180
```

- Refresh Advanced Message Security on z/OS.

Changes that are made effective by issuing the **REFRESH** command apply to applications that issue MQOPEN after the **REFRESH** command has completed. Existing applications that have a queue open, continue to use the options from when the application opened the queue. To use the new values, the application has to close and reopen the queue.

- Start and stop AMS on z/OS.

You do not need to enter a command to start or stop the Advanced Message Security address space. The AMS address space is started automatically when the queue manager is started if AMS has been enabled with the **SPLCAP** parameter of CSQ6SYSP, and is stopped when the queue manager is stopped.

IBM MQ Internet Pass-Thru 관리

이 절에서는 IBM MQ Internet Pass-Thru 관리 방법(MQIPT)을 설명합니다.



IBM MQ Internet Pass-Thru 구성에 설명된 대로 `mqipt.conf` 구성 파일을 변경하여 MQIPT를 구성하십시오. MQIPT를 재시작하지 않고 변경사항이 적용되도록 MQIPT 새로 고침기를 포함하여 MQIPT를 관리하려면, **mqiptAdmin** 명령을 사용하십시오. **mqiptAdmin** 명령을 사용한 MQIPT 관리에 대한 정보는 [530 페이지의 『명령행을 사용하여 MQIPT 관리』](#)의 내용을 참조하십시오.

MQIPT 시작 및 중지

명령행에서 MQIPT를 시작하거나 시스템 시작 시 자동으로 시작되도록 할 수 있습니다. **mqiptAdmin** 명령을 사용하여 MQIPT를 중지할 수 있습니다.

명령행에서 MQIPT 시작

MQIPT가 다음과 같은 설치 디렉토리에 설치됩니다.

-  Windows 시스템의 C:\MQIPT(C:\MQIPT\bin에 실행 가능 스크립트가 있음)
-  Linux and AIX 시스템의 /opt/mqipt(/opt/mqipt/bin에 실행 가능 스크립트가 있음)

또한 MQIPT는 구성 파일 `mcipt.conf` 및 실행 시 MQIPT에서 출력된 모든 파일이 포함된 홈 디렉토리를 사용합니다. 처음으로 MQIPT가 호출될 때 MQIPT 홈 디렉토리의 다음 서브디렉토리가 자동으로 작성됩니다.

- `errors` - FFST(First Failure Support Technology) 및 추적 파일이 기록됩니다.
- `logs` 디렉토리 - 연결 로그가 보관됩니다.

MQIPT를 실행하는 사용자 ID에 이러한 디렉토리를 작성할 수 있는 권한이 있어야 하거나 해당 디렉토리가 이미 존재하고 사용자 ID에 이러한 디렉토리에 파일을 작성하고 읽고 쓸 수 있는 권한이 있어야 합니다. 또한 Java security manager 정책을 사용 중인 경우 보안 정책이 이러한 디렉토리에 필요한 권한을 부여해야 합니다. 보안 관리자 정책 설정에 대한 자세한 정보는 [Java security manager](#)의 내용을 참조하십시오.

설치 디렉토리를 홈 디렉토리로 사용할 수 있습니다. 이 디렉토리를 사용하는 경우 MQIPT를 실행하는 사용자 ID에 적절한 권한이 있으며 보안 관리자 정책이 올바르게 구성되었는지 확인해야 합니다.

MQIPT를 시작하려면 MQIPT 설치 디렉토리의 `bin` 디렉토리에 있는 `mcipt` 명령을 사용하십시오. 예를 들어, 다음 명령은 C:\mciptHome 디렉토리를 홈 디렉토리로 사용하는 MQIPT의 인스턴스를 시작합니다.

```
mcipt C:\mciptHome
```

`mcipt` 명령에 대한 자세한 정보는 [mcipt\(MQIPT 시작\)](#)의 내용을 참조하십시오.




`mcipt` 명령을 사용하여 시작되는 MQIPT 인스턴스에 제공할 이름을 지정할 수 있습니다. MQIPT 인스턴스의 이름은 명령 포트를 사용하지 않고 `mciptAdmin` 명령을 사용하여 MQIPT의 로컬 인스턴스를 관리하는 데 사용됩니다. 이 매개변수를 지정하지 않으면 MQIPT 홈 디렉토리의 이름이 MQIPT 인스턴스 이름으로 사용됩니다.

콘솔 메시지에 MQIPT의 상태가 표시됩니다. 오류가 발생하는 경우 [IBM MQ Internet Pass-Thru 문제점 해결](#)을 참조하십시오. 다음 메시지는 MQIPT가 성공적으로 시작된 경우 출력의 예입니다.

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mcipt.conf
MQCPI152 MQIPT name is C:\mciptHome
MQCPI021 Password checking has been enabled on the command port
MQCPI144 MQ Advanced capabilities not enabled
MQCPI011 The path C:\mciptHome\logs will be used to store the log files
MQCPI006 Route 1414 is starting and will forward messages to :
MQCPI034 ....examplehost(1414)
MQCPI035 ....using MQ protocols
MQCPI057 ....trace level 5 enabled
MQCPI078 Route 1414 ready for connection requests
```

자동으로 MQIPT 시작

시스템이 시작될 때 자동으로 시작하는 시스템 서비스로 MQIPT를 설치할 수 있습니다. `mciptService` 명령을 사용하여 MQIPT 서비스를 설치 및 설치 제거하십시오.

-  Windows 시스템에서 `mciptService` 명령은 MQIPT를 Windows 서비스로 설치합니다.
-   AIX and Linux 시스템에서 `mciptService` 명령은 시스템이 부팅될 때 시작되는 System V init 서비스로 MQIPT를 설치합니다. System V init를 지원하지 않는 Linux 시스템에서는 `systemd`와 같은 다른 방법을 사용하여 MQIPT를 서비스로 관리하십시오.

MQIPT 서비스가 시작될 때 모든 활성 MQIPT 라우트가 시작됩니다. 서비스가 중지되면 모든 라우트가 즉시 종료됩니다.

시스템에 MQIPT 설치가 둘 이상인 경우에도 시스템에서 MQIPT 서비스 하나만 설치할 수 있습니다.

`mciptService` 명령에 대한 자세한 정보는 [mciptService\(MQIPT 서비스 관리\)](#)를 참조하십시오.

MQIPT 중지

-`stop` 매개변수와 함께 `mciptAdmin` 명령을 사용하여 MQIPT를 중지할 수 있습니다.

예를 들어, 다음 명령은 **mqiptAdmin** 명령과 동일한 사용자 ID로 로컬로 실행 중인 이름이 **mqipt1** 인 MQIPT의 인스턴스를 중지합니다.

```
mqiptAdmin -stop -n ipt1
```

mqiptAdmin 명령은 다음 방법 중 하나를 사용하여 관리할 MQIPT의 활성 인스턴스에 연결합니다.

- 명령 포트를 사용하지 않고 MQIPT의 로컬 인스턴스에 연결하여.
- 명령 포트에 대한 네트워크 연결을 작성하여.

mqiptAdmin 명령을 사용하여 명령 포트에 명령을 전송함으로써 MQIPT를 중지하려면 **RemoteShutDown** 특성을 **true**로 설정하여 원격 종료를 사용으로 설정해야 합니다.

mqiptAdmin 명령을 사용한 MQIPT 관리에 대한 자세한 정보는 530 페이지의 『명령행을 사용하여 MQIPT 관리』의 내용을 참조하십시오.

비밀번호 암호화 키 지정

MQIPT 구성에 기본 키가 아닌 암호화 키를 사용하여 암호화된 비밀번호가 포함된 경우, 시작 시 MQIPT가 읽을 수 있는 파일에 비밀번호 암호화 키를 제공해야 합니다.

비밀번호 암호화 키 파일

MQIPT가 저장 및 사용할 수 있도록 암호화되는 비밀번호는 사용자가 제공하는 암호화 키를 사용하여 암호화할 수 있습니다. 암호화 키를 제공하지 않으면 기본 암호화 키가 사용됩니다. 비밀번호 암호화 키를 지정하지 않아도 되지만 이를 수행하는 것이 보다 안전합니다. 고유 암호화 키를 지정하지 않으면 기본 암호화 키가 사용됩니다.

비밀번호 암호화 키를 제공하는 경우 비밀번호 및 MQIPT를 암호화하는 데 사용되는 **mqiptPW** 명령으로 액세스할 수 있는 파일에 저장되어야 합니다. 이 파일의 콘텐츠에 대한 유일한 제한은 이 파일이 최소 하나 이상의 문자와 한 개의 텍스트 행만 포함해야 한다는 것입니다.

참고: 권한이 없는 사용자가 암호화 키를 읽는 것을 방지하기 위해 적절한 파일 권한이 비밀번호 암호화 키 파일에 설정되어 있는지 확인해야 합니다. **mqiptPW** 명령을 실행하는 사용자와 MQIPT가 실행되는 사용자만 비밀번호 암호화 키를 읽기 위한 권한이 필요합니다.

동일한 비밀번호 암호화 키가 MQIPT의 인스턴스에 대해 저장된 모든 비밀번호를 암호화하고 복호화하는 데 사용됩니다. 따라서 MQIPT 설치 시마다 단일 비밀번호 암호화 키 파일만 필요합니다.

MQIPT 설치를 위한 비밀번호 암호화 키가 변경되는 경우 암호화된 비밀번호 모두 새 암호화 키를 사용하여 다시 암호화되어야 합니다.

MQIPT 시작

비밀번호 암호화 키 파일의 기본 이름은 **MQIPT_HOME_DIR/mqipt_cred.key**입니다. 여기서 **MQIPT_HOME_DIR**은 **mqipt.conf** 구성 파일이 저장된 디렉토리입니다. 자동으로 시작되는 서비스로 MQIPT를 실행할 계획인 경우 기본 이름을 사용하여 비밀번호 암호화 키 파일을 작성해야 합니다.

기본 이름 외의 이름으로 비밀번호 암호화 키 파일이 작성되는 경우 MQIPT 시작 시 이 파일의 이름을 제공해야 합니다. 비밀번호 암호화 키 파일의 이름은 우선순위에 따라 다음 방법 중 하나를 사용하여 지정할 수 있습니다.

1. MQIPT를 시작하는 데 사용되는 **mqipt** 명령의 **-sf** 매개변수.
2. **MQS_MQIPTCRED_KEYFILE** 환경 변수.
3. **com.ibm.mq.ipt_cred.keyfile** Java 특성.

비밀번호 암호화 키 파일 이름을 제공하지 않으면 파일이 존재하는 경우 기본 파일 이름이 사용됩니다. 기본 비밀번호 암호화 키 파일이 없는 경우 기본 비밀번호 암호화 키가 사용됩니다.

명령행을 사용하여 MQIPT 관리

명령행에서 **mqiptAdmin** 명령을 사용하여 MQIPT를 관리할 수 있습니다.

mqiptAdmin 명령을 사용하여 다음 관리 기능을 수행할 수 있습니다.

- MQIPT의 활성 로컬 인스턴스를 나열합니다.
- 구성 파일을 변경한 후, MQIPT의 인스턴스를 새로 고치십시오.
- MQIPT의 인스턴스를 중지하십시오.

mqiptAdmin 명령은 MQIPT 설치 디렉토리의 bin 서브디렉토리에 있습니다.

mqiptAdmin 명령은 다음 방법 중 하나를 사용하여 관리할 MQIPT의 활성 인스턴스에 연결합니다.

- 명령 포트에 대한 네트워크 연결을 작성하여.
- 명령 포트를 사용하지 않고 MQIPT의 로컬 인스턴스에 연결하여.

mqiptAdmin 명령은 이전 버전의 MQIPT와 호환되지만, 명령을 사용하여 MQIPT mqiptAdmin 명령의 버전보다 더 높은 버전을 관리할 수 없습니다. 다른 버전의 MQIPT를 포함하는 환경에서는 mqiptAdmin 명령의 최신 버전을 사용해야 합니다.

mqiptAdmin 명령의 구문에 대한 자세한 정보는 [mqiptAdmin\(관리 MQIPT\)](#)을 참조하십시오.

명령 포트가 없는 로컬 관리

MQIPT의 로컬 인스턴스는 명령 포트를 사용하지 않고 관리할 수 있습니다. 로컬 관리를 통해 관리하고자 하는 MQIPT와 동일한 시스템에서 실행되는 경우에만 mqiptAdmin 명령을 사용하여 MQIPT를 관리할 수 있습니다.

명령 포트를 사용하지 않고 mqiptAdmin이 MQIPT의 로컬 인스턴스를 관리할 수 있는 권한을 부여받으려면, MQIPT 인스턴스가 동일한 시스템 및 mqiptAdmin과 동일한 사용자 ID로 실행 중이어야 합니다. 또는 AIX and Linux, mqiptAdmin은 root로 실행될 수 있습니다.

기본적으로 로컬 관리가 사용 가능합니다. 로컬 관리를 사용 안함으로 설정하려면, LocalAdmin 구성 특성을 사용하십시오. LocalAdmin 특성에 대한 자세한 정보는 [LocalAdmin](#)을 참조하십시오.

MQIPT의 로컬 인스턴스를 관리하려면 각 인스턴스에 이름을 지정해야 합니다. mqipt 명령으로 MQIPT를 시작할 때 -n 매개변수를 사용하여 MQIPT의 인스턴스에 이름을 지정할 수 있습니다. MQIPT 시작 시 이름을 지정하지 않으면, 홈 디렉토리의 이름은 MQIPT 인스턴스의 이름으로 사용됩니다. 예를 들어, 다음 명령어는 MQIPT로 시작하고 이름 ipt1을 인스턴스에 지정합니다.

```
mqipt /opt/mqipt1 -n ipt1
```

인스턴스에 이름이 있으면 -n 매개변수를 사용하여 mqiptAdmin 명령에 이름을 지정하여 해당 인스턴스를 관리할 수 있습니다. 예를 들어, 다음 명령어는 이름이 ipt1인 MQIPT의 로컬 인스턴스를 중지합니다.

```
mqiptAdmin -stop -n ipt1
```

-list 매개변수와 함께 mqiptAdmin 명령을 사용하여 명령 포트를 사용하지 않고 mqiptAdmin 명령에 관리 권한이 부여된 MQIPT의 모든 로컬 활성 인스턴스를 나열할 수 있습니다. 예를 들어, 다음 명령어는 mqiptAdmin 명령을 시작한 사용자에게 관리 권한이 부여된 MQIPT의 모든 로컬 활성 인스턴스를 나열합니다.

```
mqiptAdmin -list
```

명령 포트를 사용한 관리

비보안인 하나의 명령 포트와 TLS로 보안되는 하나의 명령 포트를 사용하여 MQIPT를 구성할 수 있습니다. 이 명령 포트를 사용하여 원격 시스템에서 또는 관리하고자 하는 MQIPT 인스턴스와 동일한 시스템에 있는 사용자로 MQIPT를 관리할 수 있습니다.

MQIPT의 이전 버전은 안전하지 않은 명령 포트에 실행된 관리 명령만 허용합니다.

참고: 비보안 명령 포트에 대한 연결은 암호화되지 않으므로 MQIPT 액세스 비밀번호를 포함한 비보안 명령 포트에 네트워크를 통해 전송된 데이터를 네트워크의 다른 사용자에게 볼 수 있습니다.

MQIPT가 명령 포트에서 mqiptAdmin 명령으로 실행된 명령을 청구하려면 mqipt.conf 구성 파일의 글로벌 섹션에 있는 CommandPort 또는 SSLCommandPort 특성에 대한 값을 지정해야 합니다.

MQIPT 명령 포트 중 하나를 사용하기 전에 [기타 보안 고려사항](#) 에서 보안 고려사항을 검토하십시오. 명령 포트에서 수신한 명령에 대한 인증 사용을 고려하십시오. 명령 포트 인증에 대한 자세한 정보는 [535 페이지의 『명령 포트 인증』](#) 의 내용을 참조하십시오.

명령 포트를 사용하여 MQIPT의 인스턴스를 관리하려면, MQIPT가 실행 중인 호스트의 네트워크 주소 및 명령 포트 번호를 **mqiptAdmin** 명령에 대한 매개변수로 지정하십시오. 예를 들어, `mqipt.example.com`에서 실행 중이고 포트 1890에서 청취하도록 구성된 비보안 명령 포트가 있는 MQIPT 인스턴스를 새로 고치려면 다음 명령을 실행하십시오.

```
mqiptAdmin -refresh -r mqipt.example.com:1890
```

호스트 이름과 포트 번호를 지정하지 않은 경우, **mqiptAdmin**은 localhost, 포트 1881에 연결하려고 시도합니다.

TLS 명령 포트를 사용한 MQIPT 관리에 대한 자세한 정보는 [532 페이지의 『TLS 명령어 포트를 사용하여 MQIPT 관리』](#) 의 내용을 참조하십시오.



TLS 명령어 포트를 사용하여 MQIPT 관리

TLS 명령 포트를 사용하여 **mqiptAdmin** 명령으로 실행된 관리 명령을 청취하도록 MQIPT 를 구성할 수 있습니다. TLS 명령 포트를 사용하면 **mqiptAdmin** 및 MQIPT사이의 네트워크에서 MQIPT 액세스 비밀번호와 같은 민감한 데이터를 보호합니다. 이 프로시저를 통해 TLS 명령 포트를 사용하여 MQIPT를 관리하고 TLS 명령 포트를 구성하십시오.

이 태스크 정보

TLS 명령 포트는 PKCS #12 키 저장소 또는 PKCS #11 암호화 토큰 인터페이스를 지원하는 암호화 하드웨어에 저장된 서버 인증서로 구성되어야 합니다. 명령 포트 서버 인증은 TLS 데이터 교환 중에 **mqiptAdmin** 명령으로 전송됩니다. 이 태스크는 사용자가 신뢰할 수 있는 인증 기관 (CA) 에서 새 서버 인증서를 요청하고 해당 인증서가 파일로 사용자에게 리턴된다고 가정합니다. **mqiptAdmin** 명령은 서버 인증서에 서명한 CA의 CA 인증서를 사용하여 명령 포트 인증서를 유효성 검증합니다. CA 인증서는 **mqiptAdmin** 명령으로 액세스할 수 있는 PKCS #12 키 저장소에 저장되어야 합니다.



클라이언트 인증서 인증은 TLS 명령 포트에서 지원되지 않습니다. 명령 포트에 실행된 관리 명령에 대해 인증을 사용하려면 [535 페이지의 『명령 포트 인증』](#) 의 내용을 참조하십시오.

이 프로시저는   **mqiptKeytool** 명령을 사용하여 TLS 명령 포트를 사용하는 데 필요한 키 저장소 및 디지털 인증서를 관리하는 방법을 설명합니다. MQIPT 에서 사용하는 키 저장소 관리에 대한 자세한 정보는 [MQIPT 키 저장소 관리](#)를 참조하십시오.

프로시저

1. 다음 단계를 수행하여 MQIPT의 인스턴스에 대한 TLS 명령 포트를 구성하십시오.

a) PKCS #12 키 저장소에서 공개 및 개인 키 쌍과 연관된 TLS 명령 포트 서버 인증서를 작성하십시오.

  TLS 명령 포트 서버 인증서를 포함하는 키 저장소를 작성하려면 다음 명령을 입력하십시오.

```
mqiptKeytool -genkeypair -keystore filename -storetype pkcs12 -storepass password
             -dname distinguished_name -alias label
             -keyalg key_algorithm -keysize key_size -sigalg sig_algorithm
```

설명:

-keystore 파일 이름
키 저장소 이름을 지정합니다.

-storepass 암호
키 저장소 비밀번호를 지정합니다.

-alias 레이블
인증서 레이블을 지정합니다.

-keyalg 키 알고리즘

키 쌍을 작성하는 데 사용되는 알고리즘을 지정합니다.

-keysize 키 크기

키 크기를 지정합니다.



-sigalg 알고리즘

인증서에 서명하는 데 사용되는 알고리즘을 지정합니다.

-dname 구별 이름

큰따옴표로 묶인 X.500 식별 이름을 지정합니다.

- b) CA 서명된 TLS 명령 포트 서버 인증서에 대한 인증서 요청을 작성하십시오.

  인증서 요청을 작성하려면 다음 명령을 입력하십시오.

```
mqiptKeytool -certreq -keystore filename -storetype pkcs12 -storepass password
             -alias label -file certreq_filename
```

설명:

-keystore 파일 이름

키 저장소 이름을 지정합니다.

-storepass 암호

키 저장소 비밀번호를 지정합니다.

-alias 레이블



인증서 레이블을 지정합니다.

-file certreq_filename

인증서 요청의 파일 이름을 지정합니다.

- c) 533 페이지의 『1.b』 단계에서 작성된 인증서 요청 파일을 서명될 CA에 전송하십시오.

- d) CA가 사용자에게 서명된 인증서를 전송한 후 키 저장소로 서명된 인증서를 수신하십시오.

  서명된 인증서를 키 저장소에 수신하려면 다음 명령을 입력하십시오.

```
mqiptKeytool -importcert -keystore cert_filename -storetype pkcs12 -storepass password
             -file cert_filename
```

여기서 *cert_filename* 은 인증서를 포함하는 파일의 이름이고, *filename* 은 키 저장소의 이름이며, *password* 는 키 저장소 비밀번호입니다.

- e) **mqiptPW** 명령을 사용하여 키 저장소 비밀번호를 암호화하십시오.

다음 명령을 입력하십시오.

```
mqiptPW -sf encryption_key_file
```

여기서, *encryption_key_file*은 MQIPT 설치를 위한 비밀번호 암호화 키가 포함된 파일의 이름입니다. MQIPT 설치에서 기본 비밀번호 암호화 키를 사용하는 경우 **-sf** 매개변수를 지정하지 않아도 됩니다. 프롬프트가 표시되면 암호화할 키 저장소 비밀번호를 입력하십시오.

mqiptPW 명령에 대한 자세한 정보는 [키 링 비밀번호 암호화](#)를 참조하십시오.

- f) *mqipt.conf* 구성 파일을 편집하고 다음 특성을 지정하여 TLS 명령 포트를 구성하십시오.

- i) **SSLCommandPort** 특성 값을 TLS 명령 포트 번호로 설정하십시오.
- ii) **SSLCommandPortKeyRing** 특성의 값을 532 페이지의 『1.a』 단계에서 작성된 키 저장소의 파일 이름으로 설정하십시오.
- iii) **SSLCommandPortKeyRingPW**의 값을 533 페이지의 『1.e』 단계의 **mqiptPW** 명령에 의한 문자열 출력으로 설정하십시오.
- iv) 533 페이지의 『1.b』 단계에서 인증서 요청을 작성할 때 지정된 TLS 명령 포트 인증서의 레이블 이름으로 **SSLCommandPortSiteLabel** 특성 값을 설정하십시오.
- v) TLS 명령 포트에 대한 인바운드 연결을 특정 네트워크 인터페이스의 연결로 제한하려면 **SSLCommandPortListenerAddress** 특성 값을 MQIPT이(가) 실행 중인 시스템의 네트워크 인터

페이스 중 하나에 속하는 네트워크 주소로 설정하십시오. 예를 들어, TLS 명령 포트에 대한 인바운드 연결을 로컬 시스템의 연결로만 제한하려면 **SSLCommandPortListenerAddress** 특성의 값을 localhost(으)로 설정하십시오.



g) MQIPT를 시작하거나 새로 고쳐서 TLS 명령 포트를 사용으로 설정하십시오.

MQIPT는 다음과 같은 콘솔 메시지를 실행하여 적용되는 TLS 명령 포트 구성을 표시합니다.

```
MQCPI155 Listening for control commands on port 1882 on local address * using TLS
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file c:\\iptHome\\ssl\\commandport.p12
MQCPI072 .....and certificate label mqiptadmin
```

2. **mqiptAdmin** 명령이 MQIPT를 관리하는 데 사용되는 시스템에서, 다음 단계를 수행하여 **mqiptAdmin**이 TLS명령 포트에 연결할 수 있습니다.

a) **mqiptAdmin** 명령으로 신뢰 저장소로 사용할 PKCS #12 키 저장소에 TLS 명령 포트 인증서를 서명한 CA의 CA 인증서를 가져오십시오.

  CA 인증서를 가져오려면 다음 명령을 입력하십시오.

```
mqiptKeytool -importcert -keystore filename -storetype pkcs12 -storepass password
              -file cert_filename -alias certlabel
```

설명:

filename

작성할 키 저장소의 이름을 지정합니다.

비밀번호

키 저장소 비밀번호를 지정합니다.

certlabel

CA 인증서에 부여할 레이블을 지정합니다.

cert_filename

CA 인증서를 포함하는 파일의 이름을 지정합니다.

b) **mqiptPW** 명령을 사용하여 키 저장소 비밀번호를 암호화하십시오.

다음 명령을 입력하십시오.

```
mqiptPW -sf encryption_key_file
```

여기서, *encryption_key_file*은 비밀번호 암호화 키를 포함하는 파일의 이름입니다. 비밀번호 암호화 키 파일은 MQIPT 구성에서 비밀번호를 암호화하는 데 사용된 파일과 다를 수 있습니다. **-sf** 매개변수를 사용하여 암호화 키 파일을 지정하지 않으면 기본 비밀번호 암호화 키가 사용됩니다. 프롬프트가 표시되면 암호화할 키 저장소 비밀번호를 입력하십시오.

mqiptPW 명령에 대한 자세한 정보는 [키 링 비밀번호 암호화](#)를 참조하십시오.

c) **mqiptAdmin** 명령에서 사용할 특성 파일을 작성하고 다음 특성을 지정하십시오.

```
SSLClientCAKeyRing=key_ring_file_name
SSLClientCAKeyRingPW=key_ring_password
PasswordProtectionKeyFile=encryption_key_file
```

설명:

key_ring_file_name

534 페이지의 『2.a』 단계에서 작성된 키 저장소의 이름입니다.

key_ring_password

534 페이지의 『2.b』 단계에서 **mqiptPW** 명령에 의한 암호화된 비밀번호 출력입니다.

encryption_key_file

비밀번호 암호화 키가 포함된 파일의 이름입니다. 암호화 키 파일이 534 페이지의 『2.b』 단계에서 키 저장소 비밀번호를 암호화하는 데 사용된 경우에만 **PasswordProtectionKeyFile** 특성을 지정해야 합니다.

- d) **mqiptAdmin** 명령을 실행하여 MQIPT를 관리하고, **-s** 매개변수를 지정하여 TLS 연결이 필요함을 표시하고, **-p** 매개변수를 지정하여 534 페이지의 『2.c』 단계에서 작성된 특성 파일의 이름을 지정하십시오. 예를 들어, 다음 명령을 입력하여 새로 고치기 명령을 TLS 명령 포트에 전송하여 MQIPT의 인스턴스를 새로 고치십시오.

```
mqiptAdmin -refresh -r hostname:port -s -p properties_file
```

mqiptAdmin 명령은 다음과 같은 메시지를 발행하여 MQIPT에 대한 연결이 TLS로 보호됨을 확인합니다.

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

다음에 수행할 작업

TLS 명령 포트에서 수신한 명령에 대한 인증을 사용하려면 535 페이지의 『명령 포트 인증』의 단계를 수행하십시오.

명령 포트 인증

MQIPT는 비밀번호를 사용하여 비보안 명령 포트 및 TLS 명령 포트에 의해 수신된 명령을 인증하도록 구성될 수 있습니다. 명령 포트 인증을 사용하려면 이 프로시저를 사용하십시오.

이 태스크 정보

mqiptAdmin 명령은 명령이 명령 포트 인증을 사용하는 MQIPT 인스턴스의 명령 포트에 연결할 때 사용자에게 비밀번호를 입력하도록 프롬프트를 표시합니다. MQIPT 는 MQIPT 구성에 지정된 액세스 비밀번호에 대해 **mqiptAdmin** 명령에 입력된 비밀번호의 유효성을 검증합니다.

명령 포트 인증에 설정된 특성은 TLS 명령 포트와 비보안 명령 포트 모두에 적용됩니다.

프로시저

1. **mqiptPW** 명령을 사용하여 MQIPT 액세스 비밀번호를 암호화하십시오.

다음 명령을 입력하십시오.

```
mqiptPW -sf encryption_key_file
```

여기서, *encryption_key_file*은 MQIPT 설치에 대한 비밀번호 암호화 키를 포함한 파일의 이름입니다. MQIPT 설치에서 기본 비밀번호 암호화 키를 사용하는 경우 **-sf** 매개변수를 지정하지 않아도 됩니다. 프롬프트가 표시되면 암호화할 액세스 비밀번호를 입력하십시오.

MQIPT 구성에서 비밀번호 암호화에 대한 자세한 정보는 [저장된 비밀번호 암호화](#)를 참조하십시오.

2. **mqipt.conf** 구성 파일을 편집하여 다음 특성을 지정하십시오.

```
AccessPW=encrypted_password  
RemoteCommandAuthentication=auth_setting
```

설명:

암호화된 비밀번호

535 페이지의 『1』 단계에서 **mqiptPW** 명령에 의한 암호화된 비밀번호 출력입니다.

auth_setting

인증 요구사항입니다. 명령 포트 인증은 이 특성이 다음 값 중 하나로 설정된 경우 사용 가능합니다.

선택사항

비밀번호는 필요하지 않지만 비밀번호가 제공되는 경우에는 반드시 유효해야 합니다. 이 옵션은 마이그레이션 등의 경우에 유용할 수 있습니다.

필수

올바른 비밀번호는 명령 포트에서 수신한 각 명령과 함께 제공되어야 합니다.

이러한 특성에 대한 자세한 정보는 [MQIPT 글로벌 특성](#)을 참조하십시오.

3. 변경사항을 적용하려면 MQIPT를 시작하거나 새로 고치십시오.

MQIPT는 명령 포트 인증이 사용 가능한지 여부를 나타내는 메시지를 발행합니다. 예를 들어, MQIPT가 **mqiptAdmin** 명령이 실행될 때마다 올바른 비밀번호를 입력해야 하도록 구성된 경우에 다음 메시지가 발행됩니다.

```
MQCPI021 Password checking has been enabled on the command port
```

백업 작성

정기 백업 프로시저의 일부로 백업해야 하는 여러 MQIPT 파일이 있습니다.

다음 파일을 정기적으로 백업하십시오.

- 구성 파일 **mqipt.conf**
- **mqipt.conf**의 다음 특성에서 지정된 SSL/TLS 키 링 파일:
 - **SSLClientKeyRing**
 - **SSLClientCAKeyRing**
 - **SSLServerKeyRing**
 - **SSLServerCAKeyRing**
 - **SSLCommandPortKeyRing**
- **mqipt.conf**의 다음 특성에서 지정된 SSL/TLS 키 링 비밀번호 파일:
 - **SSLClientKeyRingPW**
 - **SSLClientCAKeyRingPW**
 - **SSLServerKeyRingPW**
 - **SSLServerCAKeyRingPW**
- MQIPT 구성에 기본 키 외의 암호화 키로 암호화되는 비밀번호가 포함되는 경우 비밀번호 암호화 키 파일입니다.
- **SecurityManagerPolicy**에서 지정된 정책 파일(해당 특성이 설정된 경우)
- **mqipt.conf**의 다음 특성에서 지정된 보안 엑시트 파일 및 인증서 엑시트 파일:
 - **SecurityExitName**
 - **SSLExitName**
- MQIPT 홈 디렉토리의 log 하위 디렉토리에 있는 연결 로그 파일(감사 목적으로 필요한 경우)

성능 조정

스레드 풀과 유희 제한시간 스펙의 조합을 사용하여 각 MQIPT 라우트의 상대 성능을 조정할 수 있습니다.

연결 스레드

각 MQIPT 라우트는 수신되는 통신 요청을 핸들링하는 동시에 실행 중인 스레드의 작업 풀에 지정됩니다. 초기화 시 해당 라우트의 **MinConnectionThreads** 속성에 지정된 크기로 스레드 풀이 작성되고 처음으로 수신되는 요청을 핸들링하기 위한 스레드가 지정됩니다. 이 요청이 도착하면 다른 스레드가 지정되며 다음으로 수신되는 요청을 위한 준비가 됩니다. 작업에 대한 모든 스레드가 지정되면 새 스레드가 작성되어 작업 풀에 추가되고 작업에 대해 지정됩니다.

이 방식으로 **MaxConnectionThreads**에 지정된 최대 스레드 수에 이를 때까지 풀이 증가합니다. 대화가 종료되거나 지정된 유희 제한시간 기간이 경과하면 스레드가 다시 풀에 해제됩니다. 최대 작업 스레드 수에 이르면 다음으로 수신되는 요청은 스레드가 작업 풀에 다시 해제될 때까지 대기합니다.

사용 가능한 스레드 수를 늘리면 요청이 대기해야 하는 시간을 줄일 수 있습니다. 그러나 이러한 증가를 사용할 수 있는 시스템 자원과 밸런싱해야 합니다.

유휴 제한시간

기본적으로 작업 스레드는 비활성으로 인해 종료되지 않습니다. 스레드가 대화에 지정되면 정상적으로 처리완료되거나, 라우트가 비활성화되거나, MQIPT가 종료될 때까지 해당 대화에 지정된 상태로 남아 있습니다. 선택적으로 **IdleTimeout** 특성에 유휴 제한시간 간격(분)을 지정하여 지정된 기간 동안 비활성 상태인 스레드가 재사용되도록 할 수 있습니다. 스레드를 작업 풀에 다시 배치하여 스레드가 재사용됩니다.

IBM MQ 활동이 간헐적이면 스레드가 계속 재사용되지 않도록 하트비트 간격을 MQIPT 제한시간 값보다 작은 값으로 설정하십시오.

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. 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.
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 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 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.

이 책에는 고객이 IBM MQ의 서비스를 얻기 위해 프로그램을 작성할 수 있도록 하는 의도된 프로그래밍 인터페이스에 대한 정보가 들어 있습니다.

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.com[®]는 전세계 여러 국가에 등록된 IBM Corporation의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/copytrade.shtml)에 있습니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

이 제품에는 Eclipse 프로젝트 (<https://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: