

9.4

IBM MQ 에 대한 모니터링 및 성능

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [363 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

모니터링 및 성능.....	5
IBM MQ 네트워크 모니터링.....	5
OpenTelemetry 통합.....	5
큐 관리자 상태 검사 동작.....	6
이벤트 모니터링.....	8
메시지 모니터링.....	56
회계 및 통계 메시지.....	126
애플리케이션 활동 추적.....	187
모니터링 및 활동 추적에 대한 시스템 토픽.....	269
실시간 모니터링.....	279
클러스터 모니터링.....	290
애플리케이션 밸런싱 모니터링.....	293
Monitoring performance and resource usage on z/OS.....	295
IBM MQ 네트워크 성능 조정.....	350
클라이언트 및 서버 연결 채널 조정.....	351
분산 발행/구독 네트워크 성능 조정.....	352
토픽 트리에서 불필요한 토픽 수 감소.....	360
Aspera gateway는 대기 시간이 긴 네트워크에서 성능을 향상시킬 수 있음.....	361
주의사항.....	363
프로그래밍 인터페이스 정보.....	364
상표.....	364

IBM MQ 모니터링 및 성능

이 절의 모니터링 정보와 지침 및 특정 성능 조정 팁을 사용하면 큐 관리자 네트워크의 성능을 개선할 수 있습니다.

이 태스크 정보

큐 관리자 네트워크의 크기와 복잡도에 따라서 네트워크 모니터링으로부터 다양한 정보를 얻을 수 있습니다. 해당 정보를 특정 성능 조정 팁에서 제공되는 정보와 함께 사용하여 네트워크 성능을 조정할 수 있습니다.

IBM MQ 네트워크 모니터링

큐 관리자 네트워크가 실행 중인 방식에 대한 특정 기타 정보와 통계를 확보하도록 다양한 모니터링 기술이 IBM MQ에서 사용 가능합니다. 이 절의 모니터링 정보 및 지침을 사용하면 큐 관리자 네트워크의 성능을 향상시키는 데 도움이 될 수 있습니다.

다음 목록에서는 큐 관리자 네트워크를 모니터링하는 이유에 대한 예를 제공합니다.

- 큐 관리자 네트워크에서 문제점을 감지합니다.
- 큐 관리자 네트워크에서 문제점의 원인을 판별하는 데 도움을 줍니다.
- 큐 관리자 네트워크의 효율성을 향상시킵니다.
- 큐 관리자 네트워크의 실행에 익숙해집니다.
- 큐 관리자 네트워크가 올바르게 실행 중인지 확인합니다.
- 특정 이벤트가 발생하면 메시지를 생성합니다.
- 메시지 활동을 기록합니다.
- 마지막으로 알려진 메시지 위치를 판별합니다.
- 실시간으로 큐 관리자 네트워크의 다양한 통계를 검사합니다.
- 감사 추적을 생성합니다.
- 애플리케이션 자원 사용법을 설명합니다.
- 용량 계획.

ALW

V 9.4.0

OpenTelemetry 통합

IBM MQ 를 OpenTelemetry 추적 시스템과 통합할 수 있습니다.

OpenTelemetry 추적

OpenTelemetry 추적을 사용하여 애플리케이션이 데이터 플로우 내에서 작동하는 방식을 관찰할 수 있습니다. 데이터 플로우는 여러 다른 애플리케이션을 통합할 수 있으며 종종 통합합니다. 추적은 전체 과정을 표시하고 각 개별 애플리케이션의 동작에 대한 인사이트를 제공할 수 있습니다. IBM MQ에서는 OpenTelemetry 추적 시스템과 통합할 수 있도록 하는 추적 서비스를 제공합니다.

IBM MQ OpenTelemetry 추적 서비스는 IBM MQ API 엑시트로 구현됩니다. 이는 IBM 지원 프로그램으로 제공되며, 이는 추적 서비스를 사용할 수 있는 인타이틀먼트가 있고 IBM MQ 인타이틀먼트의 일부로 지원을 받음을 의미합니다. IBM MQ 인타이틀먼트는 IBM Instana 지원 프로그램의 IBM MQ 추적 종료 구성요소 사용만 허용합니다.

IBM Instana 모니터링 시스템 및 IBM Instana 인타이틀먼트가 있는 추적 엑시트를 사용하지 않는 경우 엑시트 사용으로 인해 발생하는 문제를 IBM MQ 지원 센터에 보고해야 합니다.

종료는 <https://ibm.biz/mqinstanaexit>에서 다운로드할 수 있습니다.

IBM MQ Open Telemetry 추적 서비스 설치 및 구성

IBM MQ 추적을 설치하고 구성하는 방법에 대한 전체 세부사항은 [IBM MQ 추적](#)에서 찾을 수 있습니다.

IBM Instana IBM MQ 엑시트에 대해 지원되는 플랫폼의 세부사항은 [사내 구축형 IBM MQ에 지원되는 플랫폼](#)에서 찾을 수 있습니다.

Multi 큐 관리자 상태 검사 동작

큐 관리자는 안정적이고 신뢰할 수 있는 성능을 보장하기 위해 정기적인 상태 검사를 수행합니다. 이 주제에서는 큐 관리자가 수행하는 일부 상태 검사에 대해 설명하고 환경 요구사항을 기반으로 구성할 수 있는 방법에 대해 설명합니다.

대부분의 환경에서 기본 구성이 적합하며 이러한 검사가 수행되는 빈도를 변경할 필요가 없습니다. 기본 설정을 사용하는 경우에도 문제점이 발견될 때 큐 관리자가 작동하는 방식과 검사 실패의 원인이 될 수 있는 환경 문제점을 이해하는 것이 유용할 수 있습니다. 이 주제는 이러한 동작 중 일부를 설명하기 위한 것입니다.



경고: IBM 지원 센터에서 권장하지 않는 한, 이러한 검사의 빈도를 변경하지 마십시오.

큐 관리자의 서로 다른 컴포넌트는 다양한 방법을 사용하여 불일치를 발견하고 해결하며 이 토픽은 이러한 모든 메커니즘을 설명하기 위한 것이 아닙니다. 예를 들어, IBM MQ 프로세스는 다양한 메커니즘을 사용하여 종속된 다른 프로세스가 여전히 실행 중인지 확인합니다. 설명되는 동작은 환경 또는 기타 예기치 않은 상황을 발견하기 위해 실행 컨트롤러가 주기적으로 수행하는 동작입니다. (실행 컨트롤러는 대부분의 다른 큐 관리자 프로세스를 시작하고 관리하는 기본 IBM MQ 프로세스입니다.) 정기적인 검사이므로 적절한 튜닝 매개변수를 설정하여 어느 정도까지 수정할 수 있는 특정 간격으로 수행됩니다.

설명된 일부 검사는 전용 상태 검사 스레드에 의해 수행됩니다. 상태 검사 스레드 자체에서 문제점이 감지되면 경고 메시지 AMQ5066 이 큐 관리자 오류 로그에 기록됩니다.

이 주제에서 설명하는 동작은 향후 릴리스에서 변경될 수 있습니다. 예를 들어, 특정 플랫폼 또는 구성에서 다른 기본값이 보다 안정적인 것으로 관찰되는 경우입니다.

일반 상태 검사

큐 관리자는 일정한 간격으로 다양한 검사를 수행합니다. 기본적으로 이러한 검사는 10초마다 수행됩니다. (일부 경우, 검사는 오류를 보고하기 전에 두 번의 주기를 허용하므로 이러한 검사에 대해 20초간격이 됩니다.) 검사 중에 큐 관리자는 큐 관리자의 일부로 실행되는 다양한 프로세스가 여전히 실행 중인지 확인합니다. 고유 HA 큐 관리자의 경우 큐 관리자가 대기 인스턴스에 데이터를 복제하고 있는지 확인합니다.

이 때 중요한 검사가 실패하면 (예를 들어, amqzmc0 프로세스가 더 이상 실행되지 않는 경우) 큐 관리자는 계속 실행할 수 없습니다. 그러나 대부분의 검사는 더 이상 필요하지 않은 시스템 자원을 정리하기 위해 수행되며 단순히 메시지가 큐 관리자 오류 로그에 기록될 수 있습니다.

대부분의 경우 이러한 일반 상태 검사의 빈도를 변경할 필요가 없습니다. 큐 관리자 또는 운영 환경에 있는 대부분의 이벤트는 일반 상태 검사 프로세스가 이벤트를 감지하지 않아도 즉시 감지됩니다. 이 프로세스는 큐 관리자의 다른 위치에서 감지되지 않은 사항에 대한 주기적 검사 역할을 합니다. 필요한 경우 **ECHearTBeatLen** 튜닝 매개변수를 사용하여 빈도를 구성할 수 있습니다. 최소값은 10000밀리초 (10초) 입니다. 최대값은 60000밀리초 (60초) 입니다. 최대값 60000으로 설정하면 특정 검사에 대해 2분 지연이 발생할 수 있습니다.

로그 진행 상태를 확인합니다.

큐 관리자는 로그에 대한 쓰기가 합리적인 비율로 수행되는지 확인합니다. 이는 로거의 성능이 최적인지 확인하는 것이 아니라 추가 주의가 필요할 수 있는 조건을 발견하도록 디자인되었습니다. 예를 들어, 로그 파일이 저장되는 디스크가 특히 느린 경우 또는 큐 관리자가 컨테이너화된 환경에서 모든 작업을 수행하기에 충분한 CPU 시간을 수신하지 않는 경우입니다.

이 검사에 실패하면 큐 관리자가 수행하는 조치는 사용 중인 큐 관리자의 유형에 따라 다릅니다.

• 비HA 큐 관리자의 경우:

- xecL_W_PERFORMANCE_BOTTLENECK FDC가 기록됩니다. 이는 시스템의 일부 파트에 추가 주의가 필요할 수 있음을 나타내는 표시로 사용할 수 있습니다. 큐 관리자가 계속 실행 중입니다. 오류 디렉토리에 xecL_W_PERFORMANCE_BOTTLENECK FDC가 표시되는 경우 스토리지 또는 플랫폼 팀과 함께 작업하여

기본 시스템 자원이 IBM MQ 를 실행하기에 충분한지 여부를 파악해야 할 수 있습니다. IBM MQ 가 과다 할당된 노드의 컨테이너에서 실행되는 경우 IBM MQ 는 모든 메시징 워크로드를 수행하기에 충분한 스케줄된 CPU 시간을 수신하지 못할 수 있습니다.

- IBM MQ 9.3.0부터 경고 메시지 **AMQ5068W** 가 큐 관리자 오류 로그에 기록되고 **xecL_W_PERFORMANCE_BOTTLENECK FDC**가 기록되지 않습니다. **AMQ5068W** 메시지가 로그에 표시되는 경우 스토리지 또는 플랫폼 팀과 함께 작업하여 기본 시스템 자원이 IBM MQ 를 실행하기에 충분한지 여부를 파악해야 할 수 있습니다. IBM MQ 가 과다 할당된 노드의 컨테이너에서 실행되는 경우 IBM MQ 는 모든 메시징 워크로드를 수행하기에 충분한 스케줄된 CPU 시간을 수신하지 못할 수 있습니다. 5개의 **AMQ5068W** 경고 메시지가 연속적으로 기록되는 경우 **xecL_W_PERFORMANCE_BOTTLENECK FDC**가 기록됩니다.
- 다중 인스턴스 큐 관리자의 경우:
 - 로그 진행 상태 검사가 실패하면 기본 인스턴스가 종료됩니다. 대기 인스턴스가 사용 가능한 경우 시작되어 기본 인스턴스가 됩니다.
 - IBM MQ 9.3.0부터 기본 인스턴스는 종료하기 전에 대기 인스턴스가 사용 가능한지 확인합니다. 대기 큐 관리자가 기본 인스턴스로 장애 복구할 수 있는 경우 종료됩니다. 또한 경고 메시지 **AMQ5068W** 가 큐 관리자 오류 로그에 기록됩니다.
- 고유 HA큐 관리자에서 이 검사는 비HA 큐 관리자와 동일한 방식으로 작동합니다.
- RDQM (복제된 데이터 큐 관리자) 에서 이 검사는 비HA 큐 관리자와 동일한 방식으로 작동합니다.

IBM MQ 로그의 진행에 대한 문제점이 큐 관리자 자체의 성능 문제로 인해 발생할 수 있습니다.

큐 관리자가 조치를 수행하기 전에 두 번의 검사 주기 동안 대기하더라도 기본적으로 이 검사는 60초마다 수행됩니다. 즉, 기본 설정을 사용하면 큐 관리자가 오류 메시지를 기록하기 전에 2분이 경과해야 합니다 (또는 HA큐 관리자의 경우 장애 복구).

대부분의 경우 파일 시스템이 느리거나 큐 관리자에 적은 양의 CPU 시간이 할당된 경우에도 기본 작동이 적합합니다. 파일 잠금 (7 페이지의 『파일 잠금이 여전히 유지되는지 확인합니다.』 참조) 및 기본 파일 시스템 조작과 같은 다른 검사로 인해 이 검사가 수행되기 전에 기본 인스턴스가 장애 복구되기 때문입니다. 필요한 경우 **LivenessHeartBeatLen** 튜닝 매개변수를 사용하여 이 검사의 빈도를 구성할 수 있습니다. 구성할 수 있는 최대값은 600초 (10분) 입니다. 최소값 0은 검사를 모두 사용 안함으로 설정하는 효과가 있습니다. 비HA 큐 관리자의 경우 검사의 유일한 효과는 큐 관리자 오류 로그의 추가 경고 메시지입니다. 다중 인스턴스 큐 관리자의 경우 큐 관리자의 기본 인스턴스가 더 빠르게 (값을 줄여서) 또는 느리게 (값을 늘려서) 장애 복구되도록 **LivenessHeartBeatLen** 를 구성할 수 있습니다. 로그 진행 검사의 빈도를 줄이기 위해 값을 늘리는 것은 사용자 환경에서 때때로 매우 느린 파일 시스템 IO가 발생하지만 큐 관리자의 기본 인스턴스가 실행 중인 상태로 유지되기를 원하는 경우에 유용할 수 있습니다. 이는 대기 인스턴스에 자동으로 다시 연결하도록 설계되지 않은 애플리케이션이 있고 이를 다시 시작하기 위해 수동 개입이 필요한 경우에 유용할 수 있습니다.

참고: **ECHearBeatLen** 가 증가된 경우 이는 **LivenessHeartBeatLen** 검사의 타이밍에 영향을 줍니다. 로그 진행상태 검사는 일반 상태 검사가 수행될 때 수행되므로 일반 상태 검사 (**ECHearBeatLen**) 의 빈도를 줄이면 구성된 **LivenessHeartBeatLen** 이후 최대 30초까지 로그 진행상태 검사가 구성될 수 있습니다.

일반 로깅 파일 시스템 성능

▶ V 9.4.0

IBM MQ 9.4.0부터 이 스토리지에 대한 일반 읽기/쓰기 조작성이 예상보다 오래 걸리는 것으로 나타나는 경우 큐 관리자 오류 로그에서 경고 메시지 **AMQ6729W** 가 발행됩니다. **AMQ_IODELAY** 환경 변수를 사용하여 스토리지 성능 문제를 진단하는 데 도움을 주거나 이러한 지연에 대한 허용 한계를 늘리기 위해 진단 및 타이밍을 미세 조정할 수 있습니다. 자세한 정보는 **AMQ_IODELAY**, **AMQ_IODELAY_INMS** 및 **AMQ_IODELAY_FFST**를 참조하십시오.

파일 잠금이 여전히 유지되는지 확인합니다.

다중 인스턴스 큐 관리자의 경우, 실행 제어기는 주기적으로 기본 다중 인스턴스 파일에 대한 독점 잠금을 계속 보유하고 있는지 확인하기 위해 검사합니다. 대부분의 경우 NFS 서버의 문제점으로 인해 잠금이 유실되면 기본 인스턴스가 거의 즉시 (이 검사가 수행되기 전에) 장애 복구됩니다. 일반적이지 않은 파일 시스템 문제점이 발생하는 경우 기본 큐 관리자가 장애 복구되는지 확인하기 위해 추가 정기적인 파일 잠금 검사가 수행됩니다.

기본적으로 이러한 파일 잠금 검사는 20초마다 수행됩니다. 필요한 경우 **FileLockHeartBeatLen** 튜닝 매개 변수를 설정하여 이 값을 변경할 수 있습니다. 튜닝 매개변수의 기본값은 10초입니다. (큐 관리자는 20초마다 검사하는 기본 동작을 발생시키는 조치를 수행하기 전에 두 번의 검사 주기를 허용합니다.) 튜닝 매개변수의 최소값은 10초이고 최대값은 600초 (10분) 입니다.

참고: **ECHearBeatLen** 가 증가된 경우 이는 **FileLockHeartBeatLen** 검사의 타이밍에 영향을 줍니다. 파일 잠금 검사는 일반 상태 검사가 수행될 때 수행되므로 일반 상태 검사 (**ECHearBeatLen**) 의 빈도를 줄이면 구성된 **FileLockHeartBeatLen** 이후에 최대 30초까지 파일 잠금 검사가 구성될 수 있습니다.

사용자 애플리케이션 상태 확인

큐 관리자는 더 이상 실행되지 않는 로컬로 바인드된 애플리케이션이 종료하기 전에 MQDISC MQI 호출을 수행했는지 주기적으로 검사합니다. 이러한 검사는 6 페이지의 『일반 상태 검사』에 설명된 일반 상태 검사와 동시에 수행됩니다. 따라서 이러한 검사의 기본 간격은 10000밀리초 (10초) 이며 **ECHearBeatLen** 튜닝 매개변수의 값을 변경하면 수행되는 빈도가 변경됩니다. 이 검사는 주로 연결된 애플리케이션과 연관된 자원이 사용 가능한지 확인하기 위한 것이며 HA 또는 비HA 큐 관리자가 종료되거나 대체 인스턴스로 장애 복구되지 않습니다.

MQDISC MQI 호출을 발행하지 않고 종료된 IBM MQ 클라이언트 애플리케이션은 에이전트 프로세스에 의해 별도로 감지되며 연결과 연관된 모든 자원이 해제됩니다.

관련 개념

[고가용성 구성](#)

이벤트 모니터링

이벤트 모니터링은 큐 관리자 네트워크에서 도구 이벤트 발생을 감지하는 프로세스입니다. 도구 이벤트는 큐 관리자나 채널 인스턴스에서 감지되는 이벤트의 논리적 조합입니다. 이러한 이벤트 때문에 큐 관리자나 채널 인스턴스가 이벤트 메시지라는 특정 메시지를 이벤트 큐에 넣게 됩니다.

IBM MQ 도구 이벤트에서는 큐 관리자에서 오류, 경고 및 기타 중요한 발생에 대한 정보를 제공합니다. 이 이벤트를 사용하여 큐 관리자 네트워크에서 큐 관리자의 조작을 모니터링하여 다음 목표를 이룹니다.

- 큐 관리자 네트워크에서 문제점을 감지합니다.
- 큐 관리자 네트워크에서 문제점의 원인을 판별하는 데 도움을 줍니다.
- 감사 추적을 생성합니다.
- 큐 관리자 상태 변경에 반응합니다.

관련 참조

[11 페이지의 『이벤트 유형』](#)

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

[이벤트 메시지 참조](#)

[이벤트 메시지 형식](#)

IBM MQ 이벤트 메시지 발행

IBM MQ의 이벤트 메시지 발행 준비 방법을 설명합니다.

이 태스크 정보

이벤트 메시지는 SYSTEM.ADMIN.<feature name>.EVENT라는 특별히 이름 지정된 큐에 기록됩니다.

이러한 이벤트 큐와 관련하여 주의할 사항은 그 이름이 중요하다는 점입니다. 기본적으로 큐 관리자에서는 모든 이벤트 큐를 로컬 큐로 정의합니다. 하지만, 이러한 큐를 삭제하고 원격 큐로 다시 정의하여 모든 이벤트를 전용 이벤트 처리 큐 관리자로 보낼 수 있습니다. 또는, 토픽 오브젝트를 지정하는 알리어스 큐를 사용할 수 있습니다.

두 경우 모두 경로 재지정 기술을 사용하려면 이벤트 큐를 읽는 애플리케이션에서 읽으려는 큐의 이름을 하드 코딩하지 않아야 합니다. 따라서, 애플리케이션에서 읽을 큐를 구성할 수 있어야만 합니다.

다음 명령은 다음 가정을 사용하여 이벤트 메시지를 발행할 수 있도록 이벤트 큐를 다시 정의하는 방법을 보여줍니다. 다음과 같이 가정합니다.

- 이벤트 사용을 아직 시작하지 않았거나
- 기존 이벤트 큐에서 모든 메시지를 제거했고 이러한 단계 전에 로컬 큐를 삭제한 상태입니다.

이러한 단계는 재정의 중인 QMGR 및 CHANNEL 이벤트 큐만 표시하지만 모든 이벤트로 확장될 수도 있습니다.

참고: 이 토픽 문자열은 와일드카드를 사용하여 모든 이벤트로 또는 필요에 따라 특정 이벤트로 애플리케이션을 구독할 수 있도록 설계되었습니다.

프로시저

다음 명령을 실행하십시오.

```
DEFINE TOPIC(ADMIN.QMGR.EVENT) TOPICSTR('Events/QMgr')
DEFINE TOPIC(ADMIN.CHANNEL.EVENT) TOPICSTR('Events/Channel')

DEFINE QALIAS(SYSTEM.ADMIN.QMGR.EVENT) TARGETYPE(TOPIC) TARGET(ADMIN.QMGR.EVENT)
DEFINE QALIAS(SYSTEM.ADMIN.CHANNEL.EVENT) TARGETYPE(TOPIC) TARGET(ADMIN.CHANNEL.EVENT)

DEFINE QLOCAL(ADMIN.EVENT)
DEFINE QLOCAL(ADMIN.QMGR.EVENT)

DEFINE SUB(EVENTS.ALL) TOPICSTR('Events+') PSPROP(NONE)
DESTCLAS(PROVIDED) DEST(ADMIN.EVENT)
DEFINE SUB(EVENTS.QMGR) TOPICSTR('Events/QMgr') PSPROP(NONE)
DESTCLAS(PROVIDED) DEST(ADMIN.QMGR.EVENT)
```

이벤트 읽기 애플리케이션에서 큐의 이벤트 메시지를 읽을 수 있고 필요에 따라 위에 정의된 큐 중 하나에서 읽을 수 있도록 애플리케이션을 다시 구성할 수 있다고 가정합니다.

DEFINE SUB 명령에서 PSPROP(NONE) 구성을 사용하면 발행/구독 엔진에서 추가한 메시지 속성(예: MQTopicString)을 이벤트 메시지에 추가하지 않으므로 기존 애플리케이션의 작업이 변경되지 않게 할 수 있습니다.

또한 관리 DEFINE SUB 명령을 사용하는 대신 MQSUB 호출을 사용하여 애플리케이션에서 직접 구독하여 정보를 수신할 수도 있습니다.

이제 여러 애플리케이션에서 큐 관리자가 이벤트에 생성한 정보를 사용할 수도 있습니다.

도구 이벤트

도구 이벤트는 큐 관리자나 채널 인스턴스가 이벤트 큐에서 이벤트 메시지라는 특수 메시지를 감지하여 넣는 조건의 논리 결합입니다.

IBM MQ 도구 이벤트에서는 큐 관리자에서 오류, 경고 및 기타 중요한 발생에 대한 정보를 제공합니다. 이러한 이벤트를 사용하여 큐 관리자의 조작을 모니터링할 수 있습니다 (z/OS® 용 Tivoli® NetView® 와 같은 다른 메소드 사용).

10 페이지의 그림 1에서는 도구 이벤트의 개념을 설명합니다.

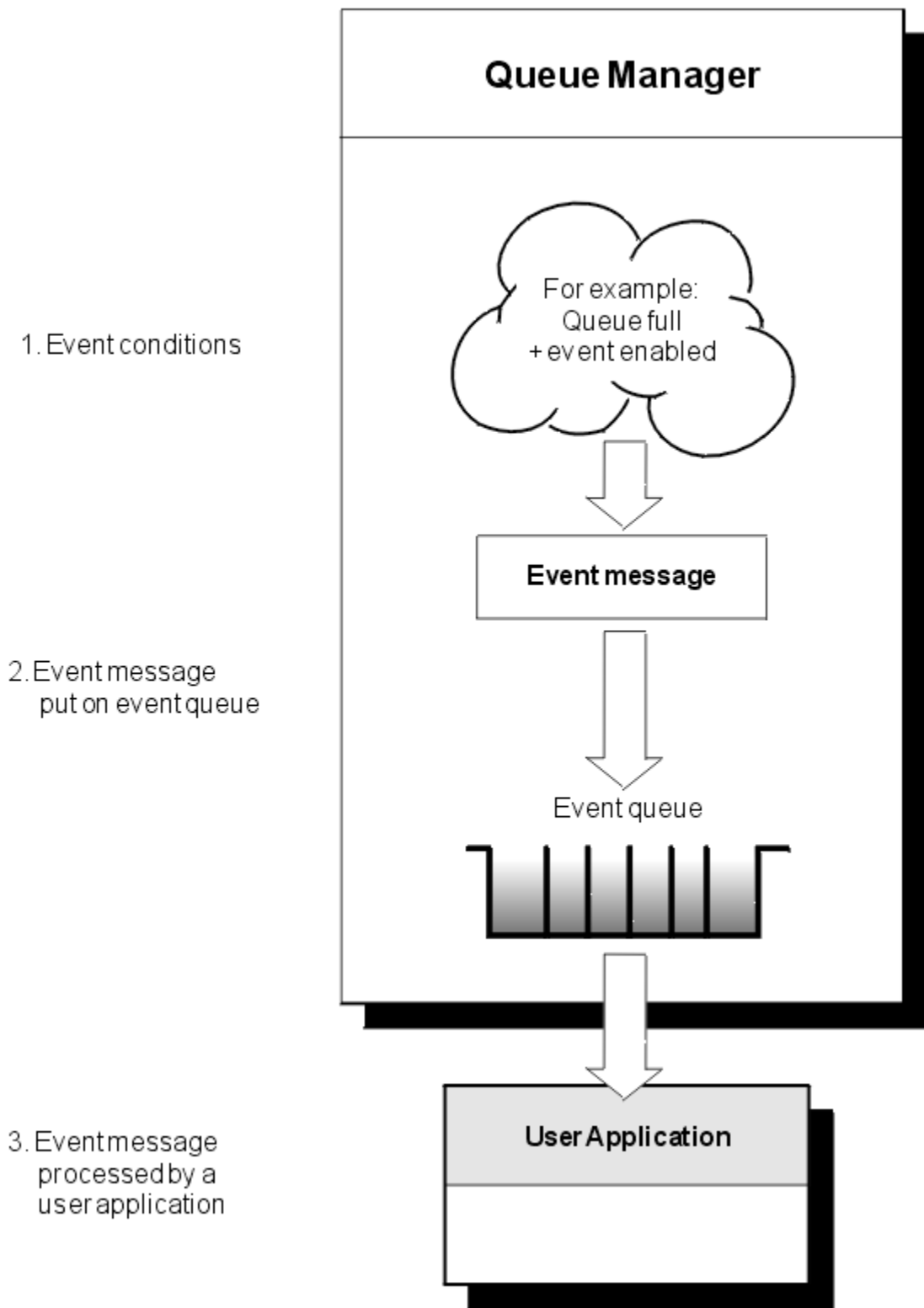


그림 1. 도구 이벤트 이해

이벤트 모니터링 애플리케이션

이벤트를 사용하여 큐 관리자를 모니터링하는 애플리케이션은 다음 요소를 포함해야 합니다.

1. 네트워크의 큐 관리자 간에 채널을 설정하십시오.

2. 필요한 데이터 변환을 구현하십시오. 데이터 변환의 일반 규칙이 적용됩니다. 예를 들어, z/OS 큐 관리자에서 UNIX 시스템 큐 관리자의 이벤트를 모니터링하는 경우 EBCDIC를 ASCII로 변환해야 합니다.

이벤트 큐를 통해 이벤트 알림

적절한 이벤트 큐가 정의된 경우, 이벤트가 발생하면 큐 관리자는 이벤트 큐에 이벤트 메시지를 넣습니다(put). 이벤트 메시지는 다음 단계를 수행하는 적당한 MQI 애플리케이션 프로그램을 작성하여 검색할 수 있는 이벤트에 관한 정보를 포함합니다.

- 큐에서 메시지를 가져오십시오.
- 이벤트 데이터를 추출하도록 메시지를 처리하십시오.

관련 정보는 이벤트 메시지의 형식을 설명합니다.

이벤트를 발생시키는 조건

다음 목록은 도구 이벤트를 발생시킬 수 있는 조건 예제를 제공합니다.

- 큐에 있는 메시지 수에 대한 임계값 한계에 도달했습니다.
- 채널 인스턴스가 시작되거나 중지됩니다.
- 큐 관리자가 활성이 되거나 중지하도록 요청됩니다.
- 애플리케이션이 IBM MQ for IBM i, AIX®, Linux®, and Windows 시스템에서 권한이 부여되지 않은 사용자 ID를 지정하는 큐를 열려고 시도합니다.
- 오브젝트가 작성, 삭제, 변경 또는 새로 고쳐집니다.
- MQSC 또는 PCF 명령이 실행됩니다.
- 큐 관리자는 새 로그 범위를 작성하기 시작합니다.
- 이벤트 조건이 충족되면 데드-레터 큐에 메시지를 넣습니다.

관련 개념

22 페이지의 『성능 이벤트』

성능 이벤트는 지정된 큐를 사용하는 애플리케이션의 성능에 영향을 줄 수 있는 조건과 관련됩니다. 성능 이벤트의 범위는 큐입니다. 한 큐에 대한 **MQPUT** 호출과 **MQGET** 호출은 다른 큐의 성능 이벤트 생성에 영향을 주지 않습니다.

53 페이지의 『멀티플랫폼에서 도구 이벤트를 모니터링하는 샘플 프로그램』

amqsevt는 큐 관리자가 작성할 수 있는 도구 이벤트를 형식화하며, IBM MQ for Multiplatforms와 함께 제공됩니다. 프로그램은 이벤트 큐에서 메시지를 읽고 읽을 수 있는 문자열로 형식화합니다.

이벤트 유형

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

IBM MQ 도구 이벤트는 다음 유형을 갖습니다.

- 큐 관리자 이벤트
- 채널 및 브릿지 이벤트
- 성능 이벤트
- 구성 이벤트
- 명령 이벤트
- 로거 이벤트
- 로컬 이벤트

각 큐 관리자에 대해 이벤트의 각 범주는 그 자신의 이벤트 큐를 갖습니다. 해당 범주의 모든 이벤트는 이벤트 메시지를 동일한 큐에 넣습니다(put).

이벤트 큐:

SYSTEM.ADMIN.QMGR.EVENT

포함하는 메시지 출처:

큐 관리자 이벤트

이벤트 큐:

SYSTEM.ADMIN.CHANNEL.EVENT
 SYSTEM.ADMIN.PERFM.EVENT
 SYSTEM.ADMIN.CONFIG.EVENT
 SYSTEM.ADMIN.COMMAND.EVENT
 SYSTEM.ADMIN.LOGGER.EVENT
 SYSTEM.ADMIN.PUBSUB.EVENT

포함하는 메시지 출처:

채널 이벤트
 성능 이벤트
 구성 이벤트
 명령 이벤트
 로거 이벤트

이벤트가 발행/구독에 관련되게 합니다. 멀티캐스트에서만 사용됩니다. 자세한 정보는 [멀티캐스트 애플리케이션 모니터링을 참조하십시오.](#)

도구 이벤트를 사용자 고유의 시스템 관리 애플리케이션에 통합함으로써, 많은 큐 관리자 사이, 많은 상이한 노드 사이 및 다중 IBM MQ 애플리케이션에 대한 활동을 모니터링할 수 있습니다. 특히 [12 페이지의 그림 2](#)에 표시된 대로 단일 노드 (IBM MQ 이벤트를 지원하는 노드의 경우) 에서 시스템의 모든 노드를 모니터링할 수 있습니다.

도구 이벤트는 사용자 작성 보고 메커니즘을 통해 이벤트를 운영자에게 제시할 수 있는 관리 애플리케이션으로 보고될 수 있습니다.

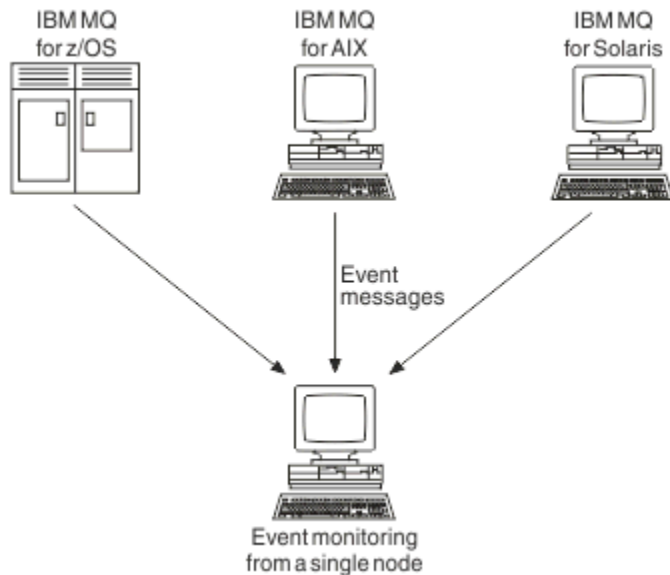


그림 2. 단일 노드에서 상이한 플랫폼 사이의 큐 관리자 모니터링

인스트루먼트이션 이벤트는 또한 다른 관리 네트워크 (예: z/OS용 Tivoli NetView) 의 에이전트 역할을 하는 애플리케이션이 보고서를 모니터링하고 적절한 경보를 작성할 수 있도록 합니다.

큐 관리자 이벤트

큐 관리자 이벤트는 큐 관리자 내 자원 사용과 관련이 있습니다. 예를 들어, 애플리케이션이 존재하지 않는 큐에 메시지를 넣으려고 시도하는 경우에 큐 관리자 이벤트가 생성됩니다.

다음 예는 큐 관리자 이벤트를 유발할 수 있는 조건을 보여줍니다.

- 애플리케이션이 실패하는 MQI 호출을 발행합니다. 호출의 이유 코드가 이벤트 메시지의 이유 코드와 동일합니다.

큐 관리자의 내부 조작 중에, 예를 들어 보고 메시지를 생성할 때 비슷한 조건이 발생할 수 있습니다. 이벤트 메시지의 이유 코드는 어떤 애플리케이션과도 연관되지 않은 경우에도 MQI 이유 코드와 일치할 수 있습니다. 이벤트 메시지 이유 코드가 MQI 이유 코드처럼 보이기 때문에 이벤트가 반드시 애플리케이션의 실패한 MQI 호출에 의해 유발되었다고 가정하지 마십시오.

- 명령이 큐 관리자에 발행되고 이 명령 처리가 이벤트를 유발합니다. 예를 들면, 다음과 같습니다.
 - 큐 관리자가 중지 또는 시작됩니다.

- 명령에 대한 연관된 사용자 ID가 권한 부여되지 않는 명령이 발행됩니다.

IBM MQ 는 SYSTEM.ADMIN.QMGR.EVENT 큐에 있으며, 다음 큐 관리자 이벤트 유형을 지원합니다.

ALW 권한(AIX, Linux, and Windows에만 해당됨)

권한 이벤트가 필요한 권한을 갖지 않는 큐를 열려고 시도하는 애플리케이션 또는 필수 권한을 갖지 않는 사용자 ID로부터 발행되는 명령 같은 권한 부여를 보고합니다. 권한 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [권한 부여되지 않음\(유형 1\)](#)
- [권한 부여되지 않음\(유형 2\)](#)
- [권한 부여되지 않음\(유형 3\)](#)
- [권한 부여되지 않음\(유형 4\)](#)
- [권한 부여되지 않음\(유형 5\)](#)
- [권한 부여되지 않음\(유형 6\)](#)

모든 권한 이벤트는 AIX, Linux, and Windows에서만 유효합니다.

금지

금지 이벤트는 큐가 Put 또는 Get에 대해 금지되는 큐 또는 토픽이 발행에 대해 금지되는 토픽에 대해 MQPUT 또는 MQGET 조작이 시도되었음을 표시합니다. 금지 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [Get 금지](#)
- [Put 금지](#)

로컬

애플리케이션 또는 큐 관리자가 로컬 큐 또는 기타 로컬 오브젝트에 액세스할 수 없는 경우(예: 오브젝트가 정의되지 않았기 때문), 큐 관리자는 로컬 이벤트 메시지를 생성할 수 있습니다. 로컬 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [알리어스 기본 큐 유형 오류](#)
- [알 수 없는 알리어스 기본 큐](#)
- [알 수 없는 오브젝트 이름](#)

원격

애플리케이션 또는 큐 관리자가 다른 큐 관리자에 있는 리모트 큐에 액세스할 수 없는 경우(예: 전송 큐가 올바르게 정의되지 않았기 때문), 큐 관리자는 원격 이벤트 메시지를 생성할 수 있습니다. 원격 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [기본 전송 큐 유형 오류](#)
- [기본 전송 큐 사용 오류](#)
- [큐 유형 오류](#)
- [리모트 큐 이름 오류](#)
- [전송 큐 유형 오류](#)
- [전송 큐 사용 오류](#)
- [알 수 없는 기본 전송 큐](#)
- [알 수 없는 리모트 큐 관리자](#)
- [알 수 없는 전송 큐](#)

시작 및 중지

시작 및 중지 이벤트는 큐 관리자가 시작되었거나 시작 또는 일시정지하도록 요청되었음을 표시합니다.

z/OS z/OS는 시작 이벤트만 지원합니다.

중지 이벤트는 SYSTEM.ADMIN.QMGR.EVENT 큐의 기본 메시지 지속성이 지속적으로 정의되지 않는 한 보고되지 않습니다. 시작 및 중지 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [큐 관리자 활성화](#)
- [큐 관리자 비활성](#)

이 목록에 있는 각 이벤트 유형에 대해 이벤트 유형을 사용 가능 또는 사용 불가능하도록 큐 관리자 속성을 설정할 수 있습니다.

채널 및 브릿지 이벤트

채널은 이들 이벤트를 해당 조작 동안 감지되는 조건의 결과로 보고합니다. 예를 들어, 채널 인스턴스가 중지될 때입니다.

채널 이벤트는 다음 상황에 생성됩니다.

- 명령이 채널을 시작 또는 중지할 때.
- 채널 인스턴스가 시작 또는 중지할 때.
- 채널이 메시지를 가져올 때 변환 오류 경고를 수신할 때.
- 채널을 자동으로 작성하려고 시도할 때. 시도가 성공하든 아니면 실패하든 이벤트가 생성됩니다.

참고: 클라이언트 연결은 채널 시작 또는 채널 중지 이벤트를 유발하지 않습니다.

명령이 채널을 시작하는 데 사용될 때 이벤트가 생성됩니다. 채널 인스턴스가 시작할 때 또 다른 이벤트가 생성됩니다. 그러나 리스너, **runmqchl** 명령 또는 큐 관리자 트리거 메시지에 의한 채널 시작은 이벤트를 생성하지 않습니다. 이 경우 채널 인스턴스가 시작할 때만 이벤트가 생성됩니다.

성공적인 채널 시작 또는 중지 명령은 최소한 두 이벤트를 생성합니다. 이들 이벤트는 채널에 의해 연결되는 두 큐 관리자 모두에 대해(이벤트를 지원하는 경우) 생성됩니다.

채널 이벤트가 이벤트 큐에 놓여지는 경우, 오류 조건이 큐 관리자가 이벤트를 작성하게 합니다.

채널 및 브릿지 이벤트에 대한 이벤트 메시지는 SYSTEM.ADMIN.CHANNEL.EVENT 큐에 놓여집니다.

채널 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [채널이 활성화됨](#)
- [채널 자동 정의 오류](#)
- [채널 자동 정의 정상](#)
- [채널 변환 오류](#)
- [채널이 활성화되지 않음](#)
- [채널이 시작됨](#)
- [채널이 중지됨](#)
- [채널이 사용자에게 의해 중지됨](#)
- [채널이 차단됨](#)

IMS 브릿지 이벤트(z/OS 전용)



이들 이벤트는 IMS 브릿지가 시작 또는 중지할 때 보고됩니다.

IMS 브릿지 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [브릿지가 시작됨](#)
- [브릿지가 중지됨](#)

SSL 이벤트

유일한 TLS 이벤트는 채널 SSL 오류 이벤트입니다. 이 이벤트는 TLS를 사용하는 채널이 TLS 연결을 설정하는 데 실패하는 경우에 보고됩니다.

SSL 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [채널 SSL 오류](#)

• 채널 SSL 경고

성능 이벤트

성능 이벤트는 자원이 임계값 조건에 도달했다는 알림입니다. 예를 들어, 큐 용량 한계에 도달했습니다.

성능 이벤트는 지정된 큐를 사용하는 애플리케이션의 성능에 영향을 줄 수 있는 조건과 관련됩니다. 이벤트 큐 자체에 대해서는 생성되지 않습니다.

이벤트 유형은 메시지 데이터의 명령 ID 필드에서 리턴됩니다.

큐 관리자가 큐 관리자 이벤트나 성능 이벤트 메시지를 이벤트 큐에 넣으려고 시도하고 일반적으로 이벤트를 작성하는 오류가 감지되는 경우, 다른 이벤트는 작성되지 않고 조치가 취해지지 않습니다.

작업 단위 내의 MQGET 및 MQPUT 호출은 작업 단위가 커밋 또는 백아웃되는지 여부와 상관없이 성능 이벤트를 생성할 수 있습니다.

성능 이벤트에 대한 이벤트 메시지는 SYSTEM.ADMIN.PERFM.EVENT 큐에 넣어집니다.

다음 두 유형의 성능 이벤트가 있습니다.


큐 용량 이벤트

큐 용량 이벤트는 큐의 메시지 수, 즉 큐가 얼마나 가득 차거나 비었는지와 관련됩니다. 이들 이벤트는 공유 큐에 대해 지원됩니다. 큐 용량 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [큐 용량 높음](#)
- [큐 용량 낮음](#)
- [큐 가득 찼](#)

큐 서비스 간격 이벤트

큐 서비스 간격 이벤트는 사용자가 지정한 시간 간격 내에서 메시지가 처리되는지 여부와 관련됩니다. 이들 이벤트는 공유 큐에 대해 지원되지 않습니다.

 IBM MQ for z/OS는 QSGDISP(SHARED) 큐에 대한 큐 용량 이벤트를 지원하지만, 서비스 간격 이벤트는 지원하지 않습니다. 큐 관리자 및 채널 이벤트는 공유 큐에 의해 영향을 받지 않은 상태로 있습니다. 큐 서비스 이벤트 메시지는 다음 이벤트 데이터를 포함할 수 있습니다.

- [큐 서비스 간격 높음](#)
- [큐 서비스 간격 정상](#)

구성 이벤트

구성 이벤트는 구성 이벤트가 명시적으로 요청될 때 또는 오브젝트가 작성, 수정 또는 삭제될 때 자동으로 생성됩니다.

구성 이벤트 메시지는 오브젝트의 속성에 관한 정보를 포함합니다. 예를 들어, 이름 목록 오브젝트가 작성되고 이름 목록 오브젝트의 속성에 관한 정보를 포함하는 경우 구성 이벤트 메시지가 생성됩니다.

구성 이벤트에 대한 이벤트 메시지는 SYSTEM.ADMIN.CONFIG.EVENT 큐에 넣어집니다.

구성 이벤트는 다음의 방법으로 동작합니다.

- 큐 관리자 구성에 대한 변경사항의 감사 추적은 SYSTEM.ADMIN.CONFIG.EVENT 큐에 기록되는 이벤트 메시지를 통해 보관됩니다. [ALTER QMGR](#) 명령에서 **CONFIGEV** 매개변수를 사용하여 이러한 이벤트가 기록되도록 할 수 있습니다.
- 이러한 이벤트는 DEFINE, ALTER 또는 DELETE 명령이 오브젝트에 작용하거나 MQSET 호출이 사용될 때 생성됩니다.
- 큐 관리자의 모든 오브젝트에 대한 이벤트 메시지를 작성하는 [REFRESH QMGR TYPE\(CONFIGEV\)](#) 명령을 사용하여 현재 큐 관리자 구성의 기준선 그림을 작성할 수 있습니다. 이 조작은 많은 시간이 소요될 수 있으므로 많은 오브젝트가 있는 경우, 명령에서 NAME 및 OBJECT 규정자를 사용하여 오브젝트를 더 작은 오브젝트 세트에 분류할 수 있습니다.
- 이벤트 메시지는 다음 4개의 가능한 이유 중 하나를 기록합니다.
 - MQRC_CONFIG_CHANGE_OBJECT
 - MQRC_CONFIG_CREATE_OBJECT

- MQRC_CONFIG_DELETE_OBJECT
- MQRC_CONFIG_REFRESH_OBJECT

MQRC_CONFIG_CHANGE_OBJECT, MQRC_CONFIG_CREATE_OBJECT 또는 MQRC_CONFIG_DELETE_OBJECT는 오브젝트에서 실행할 수 있는 각 MQSC 또는 PCF 명령에 대해 발생합니다.

MQRC_CONFIG_REFRESH_OBJECT는 기준선 그림을 작성할 때 그 이벤트 메시지를 기록하기 위해 발생합니다.

4가지 유형의 구성 이벤트가 있습니다.

오브젝트 작성 이벤트

오브젝트 작성 이벤트는 오브젝트가 작성될 때 생성됩니다. 이벤트 메시지는 오브젝트 작성 이벤트 데이터를 포함합니다.

오브젝트 변경 이벤트

오브젝트 변경 이벤트는 오브젝트가 변경될 때 생성됩니다. 이벤트 메시지는 오브젝트 변경 이벤트 데이터를 포함합니다.

오브젝트 삭제 이벤트

오브젝트 삭제 이벤트는 오브젝트가 삭제될 때 생성됩니다. 이벤트 메시지는 오브젝트 삭제 이벤트 데이터를 포함합니다.

오브젝트 새로 고치기 이벤트

오브젝트 새로 고치기 이벤트는 명시적인 새로 고치기 요청에 의해 생성됩니다. 이벤트 메시지는 오브젝트 새로 고치기 이벤트 데이터를 포함합니다.

명령 이벤트

MQSC 또는 PCF 명령이 성공적으로 실행될 때 명령 이벤트가 보고됩니다.

명령 이벤트 메시지는 명령의 원본, 컨텍스트 및 컨텐츠에 관한 정보를 포함합니다. 예를 들어, 명령 이벤트 메시지는 MQSC 명령 ALTER QLOCAL이 성공적으로 실행하는 경우에 그런 정보와 함께 생성됩니다.

명령 이벤트에 대한 이벤트 메시지는 SYSTEM.ADMIN.COMMAND.EVENT 큐에 넣어집니다.

명령 이벤트는 명령 이벤트 데이터를 포함합니다.

Multi 로거 이벤트

로거 이벤트는 선형 로깅을 사용하는 큐 관리자가 새 로그 익스텐트 **IBM i** 또는 IBM i에서는 새 저널 수 신자에 로그 레코드 쓰기를 시작할 때 보고됩니다. **z/OS** 로거 이벤트는 IBM MQ for z/OS에서 사용할 수 없습니다.

로거 이벤트 메시지는 매체 복원을 위해 또는 큐 관리자를 재시작하기 위해 큐 관리자에서 필요한 로그 익스텐트를 지정하는 정보를 포함합니다.

로거 이벤트에 대한 이벤트 메시지는 SYSTEM.ADMIN.LOGGER.EVENT 큐에 넣습니다.

로거 이벤트 메시지는 로거 이벤트 데이터를 포함합니다.

이벤트 메시지 데이터 요약

이벤트 메시지의 각 유형이 포함할 수 있는 이벤트 데이터에 관한 정보를 얻으려면 이 요약을 사용하십시오.

이벤트 유형	다음 토픽 참조
권한 이벤트	권한 부여되지 않음(유형 1)
	권한 부여되지 않음(유형 2)
	권한 부여되지 않음(유형 3)
	권한 부여되지 않음(유형 4)
	권한 부여되지 않음(유형 5)
	권한 부여되지 않음(유형 6)
채널 이벤트	채널이 활성화됨
	채널 자동 정의 오류
	채널 자동 정의 정상
	채널이 차단됨
	채널 변환 오류
	채널이 활성화되지 않음
	채널이 시작됨
	채널이 중지됨
	채널이 사용자에게 의해 중지됨
명령 이벤트	명령
구성 이벤트	오브젝트 작성
	변경 오브젝트
	오브젝트 삭제
	오브젝트 새로 고치기
IMS 브릿지 이벤트	브릿지가 시작됨
	브릿지가 중지됨
금지 이벤트	Get 금지
	Put 금지
로컬 이벤트	알리어스 기본 큐 유형 오류
	알 수 없는 알리어스 기본 큐
	알 수 없는 오브젝트 이름
로거 이벤트	로거
성능 이벤트	큐 용량 높음
	큐 용량 낮음
	큐 가득 참
	큐 서비스 간격 높음
	큐 서비스 간격 정상

이벤트 유형	다음 토픽 참조
원격 이벤트	기본 전송 큐 유형 오류
	기본 전송 큐 사용 오류
	큐 유형 오류
	리모트 큐 이름 오류
	전송 큐 유형 오류
	전송 큐 사용 오류
	알 수 없는 기본 전송 큐
	알 수 없는 리모트 큐 관리자
	알 수 없는 전송 큐
SSL 이벤트	채널 SSL 오류
시작 및 중지 이벤트	큐 관리자 활성화
	큐 관리자 비활성


이벤트 제어

이벤트의 유형에 따라서 큐 관리자, 큐 속성 또는 둘 다에 대한 적절한 값을 지정하여 이벤트를 사용 가능 또는 사용 불가능하게 설정할 수 있습니다.

생성되기 원하는 각 도구 이벤트를 사용 가능하게 설정해야 합니다. 예를 들어, 큐 가득 참 이벤트를 유발하는 조건은 다음과 같습니다.

- 큐 가득 참 이벤트가 지정된 큐에 대해 사용 가능하고,
- 애플리케이션이 해당 큐에 메시지를 넣기 위한 MQPUT 요청을 발행하지만, 큐가 가득 찼기 때문에 요청이 실패합니다.

다음 기술 중 하나를 사용하여 이벤트를 사용 가능 또는 사용 불가능하게 설정하십시오.

- IBM MQ 스크립트 명령(MQSC).
- 대응하는 IBM MQ PCF 명령.
-  z/OS의 큐 관리자에 대한 조작 및 제어판.
- IBM MQ Explorer.

참고: 명령에 의해서만 큐 및 큐 관리자 둘 다에 대한 이벤트와 관련된 속성을 설정할 수 있습니다. MQI 호출 MQSET은 이벤트와 관련된 속성을 지원하지 않습니다.

관련 개념

[9 페이지의 『도구 이벤트』](#)

도구 이벤트는 큐 관리자나 채널 인스턴스가 이벤트 큐에서 이벤트 메시지라는 특수 메시지를 감지하여 넣는 조건의 논리 결합입니다.

[z/OS에서 조작 및 제어판 사용](#)

관련 태스크

[관리 태스크 자동화](#)

[PCF\(Programmable Command Format\) 사용](#)

관련 참조

[11 페이지의 『이벤트 유형』](#)

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

[MQSC 명령](#)

큐 관리자 이벤트 제어

큐 관리자 속성을 사용하여 큐 관리자 이벤트를 제어합니다. 큐 관리자 이벤트를 사용 가능하게 설정하려면 적절한 큐 관리자 속성을 ENABLED로 설정하십시오. 큐 관리자 이벤트를 사용 불가능하게 설정하려면 적절한 큐 관리자 속성을 DISABLED로 설정하십시오.

큐 관리자 이벤트를 사용 또는 사용 불가능하게 설정하려면 MQSC 명령 **ALTER QMGR**을 사용하고 적절한 큐 관리자 속성을 지정하십시오. 19 페이지의 표 1은 큐 관리자 이벤트를 사용으로 설정하는 방법을 요약합니다. 큐 관리자 이벤트를 사용 불가능하게 설정하려면 적절한 매개변수를 DISABLED로 설정하십시오.

표 1. MQSC 명령을 사용하여 큐 관리자 이벤트 사용 가능	
이벤트	ALTER QMGR 매개변수
권한 금지 로컬 원격 시작 및 중지	AUTHOREV (ENABLED) INHIBTEV (ENABLED) LOCALEV (ENABLED) REMOTEEV (ENABLED) STRSTPEV (ENABLED)

채널 및 브릿지 이벤트 제어

큐 관리자 속성을 사용하여 채널 이벤트를 제어합니다. 채널 이벤트를 사용 가능하게 설정하려면 적절한 큐 관리자 속성을 ENABLED로 설정하십시오. 채널 이벤트를 사용 불가능하게 설정하려면 적절한 큐 관리자 속성을 DISABLED로 설정하십시오.

채널 이벤트를 사용 또는 사용 불가능으로 설정하려면 MQSC 명령 **ALTER QMGR**을 사용하고 적절한 큐 관리자 속성을 지정하십시오. 19 페이지의 표 2은 채널 및 브릿지 이벤트를 사용 가능하게 설정하는 방법을 요약합니다. 큐 관리자 이벤트를 사용 불가능하게 설정하려면 적절한 매개변수를 DISABLED로 설정하십시오.


제한사항:  채널 자동 정의 이벤트는 IBM MQ for z/OS에서 사용 불가능합니다.

표 2. MQSC 명령을 사용한 채널 및 브릿지 이벤트 사용 가능 설정	
이벤트	ALTER QMGR 매개변수
채널 채널 오류에만 관련됨 IMS 브릿지 SSL 채널 자동 정의	CHLEV (ENABLED) CHLEV (EXCEPTION) BRIDGEV (ENABLED) SSLEV (ENABLED) CHADEV(ENABLED)

CHLEV가 예외로 설정될 때, 다음 리턴 코드 및 대응하는 이유 규정자가 생성됩니다.

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_ERROR
- MQRC_CHANNEL_NOT_ACTIVATED
- MQRC_CHANNEL_STOPPED
 - 다음 ReasonQualifiers를 가짐:
 - MQRQ_CHANNEL_STOPPED_ERROR
 - MQRQ_CHANNEL_STOPPED_RETRY
 - MQRQ_CHANNEL_STOPPED_DISABLED
- MQRC_CHANNEL_STOPPED_BY_USER
- MQRC_CHANNEL_BLOCKED
 - 다음 ReasonQualifiers를 가짐:
 - MQRQ_CHANNEL_BLOCKED_NOACCESS

- MQRQ_CHANNEL_BLOCKED_USERID
- MQRQ_CHANNEL_BLOCKED_ADDRESS

성능 이벤트 제어

PERFMEV 큐 관리자 속성을 사용하여 성능 이벤트를 제어합니다. 성능 이벤트를 사용 가능하게 설정하려면 PERFMEV를 ENABLED로 설정하십시오. 성능 이벤트를 사용 불가능하게 설정하려면 PERFMEV 큐 관리자 속성을 DISABLED로 설정하십시오.

PERFMEV 큐 관리자 속성을 ENABLED로 설정하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR PERFMEV (ENABLED)
```

특정 성능 이벤트를 사용 가능으로 설정하려면 적절한 큐 속성을 설정하십시오. 또한, 이벤트를 유발하는 조건을 지정하십시오.

큐 용량 이벤트

기본적으로, 모든 큐 용량 이벤트는 사용 불가능합니다. 큐 용량 이벤트 중 하나에 대해 큐를 구성하려면,

1. 큐 관리자에서 성능 이벤트를 사용 가능하게 설정하십시오.
2. 필수 큐에서 이벤트를 사용 가능하게 설정하십시오.
3. 필요한 경우 최대 큐 용량의 백분율로 표현되는 한계를 적절한 수준으로 설정하십시오.

큐 서비스 간격 이벤트

큐 서비스 간격 이벤트에 대해 큐를 구성하려면 다음을 수행해야 합니다.

1. 큐 관리자에서 성능 이벤트를 사용 가능하게 설정하십시오.
2. 필요한 대로 큐에서 큐 서비스 간격 높음 또는 정상 이벤트에 대한 제어 속성을 설정하십시오.
3. 큐에 대한 QSVCINT 속성을 적절한 시간으로 설정하여 서비스 간격 시간을 지정하십시오.

참고: 사용 가능할 때 큐 서비스 간격 이벤트가 큐에 대한 MQI 호출이 발행될 때까지 반드시 기다리지 않고 적절한 시간에 생성될 수 있습니다. 그러나, MQI 호출이 메시지를 넣거나 제거하기 위해 큐에서 사용되는 경우 모든 적용 가능한 성능 이벤트가 해당 시간에 생성됩니다. 경과 시간이 서비스 간격 시간과 같아질 때는 이벤트가 생성되지 않습니다.

구성, 명령 및 로거 이벤트 제어

큐 관리자 속성 CONFIGEV, CMDEV 및 LOGGEREV를 사용하여 구성, 명령 및 로거 이벤트를 제어합니다. 이들 이벤트를 사용 가능하게 설정하려면 적절한 큐 관리자 속성을 ENABLED로 설정하십시오. 이들 이벤트를 사용 불가능하게 설정하려면 적절한 큐 관리자 속성을 DISABLED로 설정하십시오.

구성 이벤트

구성 이벤트를 사용 가능하게 설정하려면 CONFIGEV를 ENABLED로 설정하십시오. 구성 이벤트를 사용 불가능하게 설정하려면 CONFIGEV를 DISABLED로 설정하십시오. 예를 들어, 다음 MQSC 명령을 사용하여 구성 이벤트를 사용 가능으로 설정할 수 있습니다.

```
ALTER QMGR CONFIGEV (ENABLED)
```

명령 이벤트

명령 이벤트를 사용 가능하게 설정하려면 CMDEV를 ENABLED로 설정하십시오. DISPLAY MQSC 명령 및 조회 PCF 명령을 제외한 명령에 대한 명령 이벤트를 사용 가능하게 설정하려면 CMDEV를 NODISPLAY로 설정하십시오. 명령 이벤트를 사용 불가능하게 설정하려면 CMDEV를 DISABLED로 설정하십시오. 예를 들어, 다음 MQSC 명령을 사용하여 명령 이벤트를 사용 가능으로 설정할 수 있습니다.

```
ALTER QMGR CMDEV (ENABLED)
```

로거 이벤트

로거 이벤트를 사용 가능하게 설정하려면 LOGGEREV를 ENABLED로 설정하십시오. 로거 이벤트를 사용 불가능하게 설정하려면 LOGGEREV를 DISABLED로 설정하십시오. 예를 들어, 다음 MQSC 명령을 사용하여 로거 이벤트를 사용 가능으로 설정할 수 있습니다.

```
ALTER QMGR LOGGEREV(ENABLED)
```

이벤트 큐

이벤트가 발생할 때 큐 관리자는 이벤트 메시지를 정의된 이벤트 큐에 넣습니다. 이벤트 메시지가 이벤트에 관한 정보를 포함합니다.

이벤트 큐를 다음 중 하나로 정의할 수 있습니다.

- 로컬 큐
- 알리어스 큐
- 리모트 큐의 로컬 정의 또는
- 리모트 클러스터 큐

모든 이벤트 큐를 한 큐 관리자의 동일한 리모트 큐의 로컬 정의로서 정의하는 경우, 모니터링 활동을 중앙 집중할 수 있습니다.

이벤트 메시지가 전송 큐에 필요한 메시지 형식과 호환되지 않는 형식을 갖기 때문에 이벤트 큐를 전송 큐로 정의해서는 안 됩니다.

공유 이벤트 큐는 QSGDISP(SHARED) 값을 갖고 정의되는 로컬 큐입니다.

z/OS에서 공유 큐 정의에 대한 자세한 정보는 [공유 큐를 사용한 애플리케이션 프로그래밍](#)을 참조하십시오.

이벤트 큐가 사용 불가능할 때

이벤트 큐가 사용 불가능할 때 이벤트가 발생하는 경우 이벤트 메시지가 유실됩니다. 예를 들어, 이벤트의 범주에 대한 이벤트 큐를 정의하지 않은 경우 해당 범주에 대한 모든 이벤트 메시지가 유실됩니다. 예를 들어 이벤트 메시지가 데드 레터(미전달 메시지) 큐에서 저장되지 않습니다.

그러나, 이벤트 큐를 리모트 큐로 정의할 수 있습니다. 그런 다음, 해석된 큐에 메시지를 넣는 원격 시스템에 문제가 있는 경우 이벤트 메시지는 원격 시스템의 데드-레터 큐에 도달합니다.

이벤트 큐는 다음을 포함한 많은 여러 가지 이유 때문에 사용 불가능할 수 있습니다.

- 큐가 정의되지 않았습니다.
- 큐가 삭제되었습니다.
- 큐가 가득 찼습니다.
- 큐가 put 금지되었습니다.

이벤트 큐의 부재가 이벤트가 발생하는 것을 막지 않습니다. 예를 들어, 성능 이벤트 후에 큐 관리자는 큐 속성을 변경하고 큐 통계를 재설정합니다. 이 변경은 이벤트 메시지가 성능 이벤트 큐에 넣어지는지 여부와 상관없이 발생합니다. 구성 및 명령 이벤트의 경우에도 동일하게 적용됩니다.

트리거된 이벤트 큐 사용

이벤트가 생성될 때 이벤트 큐에 넣어지는 이벤트 메시지가 사용자가 작성한 모니터링 애플리케이션을 시작하도록 트리거를 갖는 이벤트 큐를 설정할 수 있습니다. 이 애플리케이션은 이벤트 메시지를 처리하고 적절한 조치를 취할 수 있습니다. 예를 들어, 특정 이벤트는 운영자에게 알려져야 할 수 있으며, 다른 이벤트는 일부 관리 태스크를 자동으로 수행하는 애플리케이션을 시작할 수 있습니다.

이벤트 큐는 그들과 연관된 트리거 조치를 가질 수 있으며 트리거 메시지를 작성할 수 있습니다. 그러나, 이들 트리거 메시지가 다시 정상적으로 이벤트를 생성하는 조건을 유발하는 경우 이벤트가 생성되지 않습니다. 이 인스턴스에서 이벤트를 생성하지 않는 것은 루핑이 발생하지 않도록 보장합니다.

관련 개념

[18 페이지의 『이벤트 제어』](#)

이벤트의 유형에 따라서 큐 관리자, 큐 속성 또는 둘 다에 대한 적절한 값을 지정하여 이벤트를 사용 가능 또는 사용 불가능하게 설정할 수 있습니다.

[22 페이지의 『이벤트 메시지의 형식』](#)

이벤트 메시지는 이벤트 및 그의 원인에 관한 정보를 포함하고 있습니다. 다른 IBM MQ 메시지처럼, 이벤트 메시지는 메시지 디스크립터와 메시지 데이터의 두 파트를 갖고 있습니다.

[공유 큐를 사용한 애플리케이션 프로그래밍](#)

[트리거 이벤트의 조건](#)

관련 참조

[QSGDisp \(MQLONG\)](#)

이벤트 메시지의 형식

이벤트 메시지는 이벤트 및 그의 원인에 관한 정보를 포함하고 있습니다. 다른 IBM MQ 메시지처럼, 이벤트 메시지는 메시지 디스크립터와 메시지 데이터의 두 파트를 갖고 있습니다.

- 메시지 디스크립터는 MQMD 구조를 기반으로 합니다.
- 메시지 데이터는 이벤트 헤더와 이벤트 데이터로 구성됩니다. 이벤트 헤더에는 이벤트 유형을 식별하는 이유 코드가 들어 있습니다. 이벤트 메시지 넣기 및 모든 후속 조치는 해당 이벤트를 유발한 MQI 호출에 의해 리턴되는 이유 코드에 영향을 주지 않습니다. 이벤트 데이터는 이벤트에 관한 추가 정보를 제공합니다.

일반적으로 시스템 관리 애플리케이션이 실행하는 엔터프라이즈의 요구사항에 맞게 조정된 해당 애플리케이션으로 이벤트 메시지를 처리합니다.

큐 공유 그룹의 큐 관리자가 이벤트 메시지 생성을 위한 조건을 감지할 때, 여러 큐 관리자가 공유 큐에 대한 이벤트 메시지를 생성할 수 있으므로 여러 개의 이벤트 메시지가 생성됩니다. 시스템이 여러 큐 관리자의 다중 이벤트 메시지를 상관시킬 수 있도록 하기 위해, 이들 이벤트 메시지는 메시지 디스크립터(MQMD)에서 설정된 고유한 상관 ID(*CorrelId*)를 갖고 있습니다.

관련 참조

[93 페이지의 『활동 보고서 MQMD\(메시지 디스크립터\)』](#)

이 페이지를 사용하여 활동 보고서의 MQMD 구조에 따라 포함된 값을 봅니다.

[97 페이지의 『활동 보고서 MQEPH\(임베드된 PCF 헤더\)』](#)

이 페이지를 사용하여 활동 보고서의 MQEPH 구조에 따라 포함된 값을 봅니다.

[98 페이지의 『활동 보고서 MQCFH\(PCF 헤더\)』](#)

이 페이지를 사용하여 활동 보고서의 MQCFH 구조에 따라 포함된 PCF 값을 봅니다.

[이벤트 메시지 참조](#)

[이벤트 메시지 형식](#)

[이벤트 메시지 MQMD\(메시지 디스크립터\)](#)

[이벤트 메시지 MQCFH\(PCF 헤더\)](#)

[이벤트 메시지 설명](#)

성능 이벤트

성능 이벤트는 지정된 큐를 사용하는 애플리케이션의 성능에 영향을 줄 수 있는 조건과 관련됩니다. 성능 이벤트의 범위는 큐입니다. 한 큐에 대한 MQPUT 호출과 MQGET 호출은 다른 큐의 성능 이벤트 생성에 영향을 주지 않습니다.

성능 이벤트 메시지는 반드시 큐에 대한 MQI 호출이 발행될 때까지 기다릴 필요없이 임의의 적절한 시간에 생성될 수 있습니다. 그러나 메시지를 넣거나 제거하기 위해 큐에서 MQI 호출을 사용하는 경우 해당 시간에 적절한 성능 이벤트가 생성됩니다.

생성되는 모든 성능 이벤트 메시지는 SYSTEM.ADMIN.PERFM.EVENT 큐에 위치합니다.

이벤트 데이터는 이벤트의 원인을 식별하는 이유 코드, 성능 이벤트 통계 세트 및 기타 데이터를 포함합니다. 성능 이벤트 메시지에서 리턴될 수 있는 이벤트 데이터의 유형이 다음 목록에서 설명됩니다.

- [큐 용량 높음](#)

- 큐 용량 낮음
- 큐 가득 참
- 큐 서비스 간격 높음
- 큐 서비스 간격 정상

성능 이벤트의 사용을 설명하는 예는 사용자가 적절한 IBM MQ 명령(MQSC)을 사용하여 큐 속성을 설정한다고 가정합니다. z/OS에서는 큐 관리자에 대한 조작 및 제어판을 사용하여 큐 속성을 설정할 수도 있습니다.

관련 참조

11 페이지의 『이벤트 유형』

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.


성능 이벤트 통계

이벤트 메시지의 성능 이벤트 데이터에는 이벤트에 관한 통계가 들어 있습니다. 지정된 큐의 작동을 분석하려면 이 통계를 사용하십시오.

이벤트 메시지의 이벤트 데이터에는 시스템 관리 프로그램에 대한 이벤트의 정보가 들어 있습니다. 모든 성능 이벤트의 경우 이벤트 데이터는 큐 관리자의 이름과 이벤트와 연관된 큐를 포함합니다. 이벤트 데이터는 또한 이벤트와 관련된 통계를 포함합니다. 23 페이지의 표 3에서는 큐의 작동을 분석하기 위해 사용할 수 있는 이벤트 통계를 요약합니다. 모든 통계는 통계가 마지막으로 재설정된 이후에 발행한 이벤트를 참조합니다.

표 3. 성능 이벤트 통계	
매개변수	설명
TimeSinceReset	통계가 마지막으로 재설정된 이후 경과 시간.
HighQDepth	통계가 마지막으로 재설정된 이후 큐에 있는 메시지의 최대 수.
MsgEnqCount	통계가 마지막으로 재설정된 이후, 큐에 들어간 메시지 수(큐에 대한 MQPUT 호출의 수).
MsgDeqCount	통계가 마지막으로 재설정된 이후, 큐에서 제거된 메시지 수(큐에 대한 MQGET 호출의 수).

성능 이벤트 통계는 다음 변경 중 하나가 발생할 때 재설정됩니다.

- 성능 이벤트가 발생합니다(통계가 모든 활성 큐 관리자에서 재설정됩니다).
- 큐 관리자가 중지하고 재시작합니다.
- PCF 명령인 Reset Queue Statistics가 애플리케이션 프로그램에서 실행됩니다.
-  z/OS에서만, RESET QSTATS 명령이 콘솔에서 실행됩니다.

관련 개념

22 페이지의 『성능 이벤트』

성능 이벤트는 지정된 큐를 사용하는 애플리케이션의 성능에 영향을 줄 수 있는 조건과 관련됩니다. 성능 이벤트의 범위는 큐입니다. 한 큐에 대한 MQPUT 호출과 MQGET 호출은 다른 큐의 성능 이벤트 생성에 영향을 주지 않습니다.

25 페이지의 『서비스 타이머』

큐 서비스 간격 이벤트는 서비스 타이머라고 부르는 내부 타이머를 사용하는데, 이것은 큐 관리자에 의해 제어됩니다. 서비스 타이머는 큐 서비스 간격 이벤트가 사용 가능한 경우에만 사용됩니다.

25 페이지의 『큐 서비스 간격 이벤트에 대한 규칙』

공식 규칙이 서비스 타이머가 설정되고 큐 서비스 간격 이벤트가 생성되는 시기를 제어합니다.

관련 태스크

26 페이지의 『큐 서비스 간격 이벤트 사용 가능』

큐 서비스 간격 이벤트에 대해 큐를 구성하려면 적절한 큐 관리자 및 큐 속성을 설정합니다.

관련 참조

[큐 용량 상한](#)

[큐 통계 재설정](#)

[RESET QSTATS](#)

큐 서비스 간격 이벤트

큐 서비스 간격 이벤트는 서비스 간격이라고 부르는 사용자 정의 시간 간격 안에서 큐에 대해 조작이 수행되었는지 여부를 표시합니다. 설치에 따라서, 큐 서비스 간격 이벤트를 사용하여 메시지가 충분히 빨리 큐에서 제거되는지 여부를 모니터링할 수 있습니다.

큐 서비스 간격 이벤트는 공유 큐에서 지원되지 않습니다.

다음 유형의 큐 서비스 간격 이벤트가 발생할 수 있는데, 여기서 *Get* 조작이라는 용어는 **MQGET** 호출 또는 큐에서 메시지를 제거하는 활동(예: **CLEAR QLOCAL** 명령 사용)을 의미합니다.

큐 서비스 간격 확인

다음 조작 중 하나 후에

- MQPUT 호출
- 비어 있지 않은 큐를 떠나는 Get 조작

서비스 간격으로 알려진 사용자 정의 기간 안에 Get 조작이 수행되었음을 표시합니다.

Get 조작만이 큐 서비스 간격 정상 이벤트 메시지를 유발할 수 있습니다. 큐 서비스 간격 정상 이벤트를 때로는 정상 이벤트라고 설명합니다.

큐 서비스 간격 높음

다음 조작 중 하나 후에

- MQPUT 호출
- 비어 있지 않은 큐를 떠나는 Get 조작

사용자 정의 서비스 간격 내에 Get 조작이 수행되지 않았음을 표시합니다.

Get 조작이나 MQPUT 호출 중 하나가 큐 서비스 간격 높음 이벤트 메시지를 유발할 수 있습니다. 큐 서비스 간격 높음 이벤트는 때로는 높음 이벤트로 설명됩니다.

큐 서비스 간격 정상 및 큐 서비스 간격 높음 이벤트를 둘 다 사용으로 설정하려면 `QServiceIntervalEvent` 제어 속성을 `High`로 설정하십시오. 큐 서비스 간격 정상 이벤트는 큐 서비스 간격 높음 이벤트가 생성될 때 자동으로 사용 가능하게 됩니다. 큐 서비스 간격 정상 이벤트를 독립적으로 사용 가능하게 설정할 필요가 없습니다.

정상 및 높음 이벤트는 상호 배타적이므로, 하나가 사용 가능하면 다른 하나는 사용 불가능합니다. 그러나 두 이벤트가 모두 동시에 사용 불가능할 수 있습니다.

24 페이지의 [그림 3](#)에서는 큐 용량 대 시간의 그래프를 보여줍니다. 시간 P1에서 애플리케이션이 큐에 메시지를 넣기 위해 MQPUT을 실행합니다. 시간 G1에 또 다른 애플리케이션이 큐에서 메시지를 제거하기 위해 MQGET을 실행합니다.

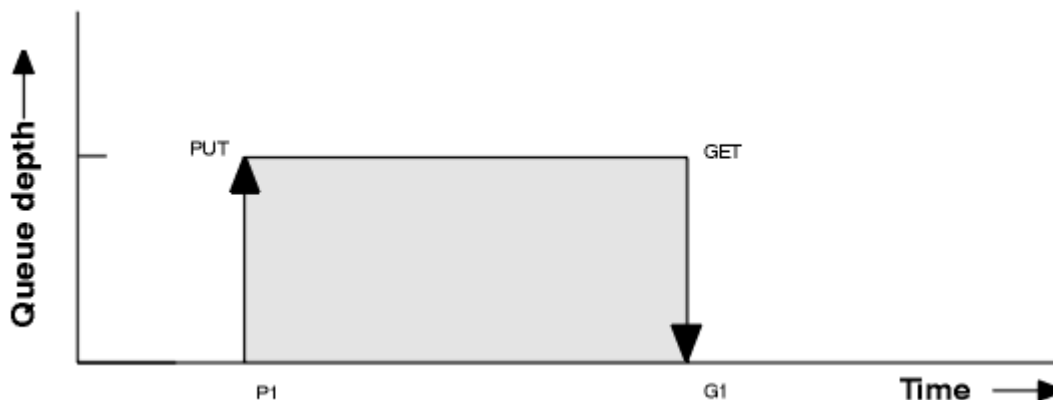


그림 3. 큐 서비스 간격 이벤트 이해

큐 서비스 간격 이벤트의 가능한 결과는 다음과 같습니다.

- PUT과 GET 사이의 경과 시간이 서비스 간격보다 작거나 같습니다.
 - 큐 서비스 간격 이벤트가 사용 가능한 경우 시간 G1에 큐 서비스 간격 정상 이벤트가 생성됩니다.
- PUT과 GET 사이의 경과 시간이 서비스 간격보다 큼니다.
 - 큐 서비스 간격 이벤트가 사용 가능한 경우 시간 G1에 큐 서비스 간격 높음 이벤트가 생성됩니다.

서비스 타이머 시작 및 이벤트 생성에 대한 알고리즘은 [25 페이지의 『큐 서비스 간격 이벤트에 대한 규칙』](#)에서 설명됩니다.

관련 참조

[큐 서비스 간격 확인](#)

[큐 서비스 간격 높음](#)

[QServiceIntervalEvent\(MQLONG\)](#)

[QServiceIntervalEvent\(10자리 부호화 정수\)](#)

서비스 타이머

큐 서비스 간격 이벤트는 서비스 타이머라고 부르는 내부 타이머를 사용하는데, 이것은 큐 관리자에 의해 제어됩니다. 서비스 타이머는 큐 서비스 간격 이벤트가 사용 가능한 경우에만 사용됩니다.

서비스 타이머는 정확하게 무엇을 측정합니까?

서비스 타이머는 비어있는 큐에 대한 MQPUT 호출이나 Get 조작과 다음 put 또는 get 사이의 경과 시간을 측정합니다(이들 두 조작 사이에 큐 용량이 0이 아니라고 가정함).

언제 서비스 타이머가 활성화됩니까?

서비스 타이머는 큐가 메시지를 갖고 있고(용량이 0이 아님) 큐 서비스 간격 이벤트가 사용 가능한 경우 항상 활성화(실행 중)입니다. 큐가 비게 되는 경우(큐 용량이 0), 타이머는 OFF 상태에 들어가고 다음 Put에서 재시작됩니다.

언제 서비스 타이머가 재설정됩니까?

서비스 타이머는 항상 Get 조작 후에 재설정됩니다. 또한 빈 큐에 대한 MQPUT 호출에 의해 재설정됩니다. 그러나 큐 서비스 간격 이벤트에서 반드시 재설정되지는 않습니다.

서비스 타이머는 어떻게 사용됩니까?

Get 조작이나 MQPUT 호출 뒤에, 큐 관리자가 서비스 타이머에 의해 측정된 경과 시간을 사용자 정의 서비스 간격에 대해 비교합니다. 이 비교의 결과는 다음과 같습니다.

- Get 조작이 있고 경과 시간이 서비스 간격보다 작거나 같고 이 이벤트가 사용 가능한 경우 정상 이벤트가 생성됩니다.
- 경과 시간이 서비스 간격보다 크고 이 이벤트가 사용 가능한 경우 높음 이벤트가 생성됩니다.

애플리케이션이 서비스 타이머를 읽을 수 있습니까?

아니오. 서비스 타이머는 애플리케이션에는 사용 불가능한 내부 타이머입니다.

TimeSinceReset 매개변수는 무엇입니까?

TimeSinceReset 매개변수는 이벤트 데이터에서 이벤트 통계의 일부로 리턴됩니다. 이벤트 통계가 재설정되지 않은 경우 연속적인 큐 서비스 간격 이벤트 사이의 시간을 지정합니다.

큐 서비스 간격 이벤트에 대한 규칙

공식 규칙이 서비스 타이머가 설정되고 큐 서비스 간격 이벤트가 생성되는 시기를 제어합니다.

서비스 타이머에 대한 규칙

서비스 타이머는 다음과 같이 0으로 재설정되고 재시작됩니다.

- 비어 있는 큐에 대한 MQPUT 호출 후.
- MQGET 호출 후에 큐가 비어있지 않은 경우 MQGET 호출 후.

타이머 재설정은 이벤트가 생성되었는지 여부에 의존하지 않습니다.

큐 관리자 시동 시에 큐 용량이 0보다 큰 경우 서비스 타이머가 시동 시간으로 설정됩니다.

Get 조작 후에 큐가 비어 있는 경우, 타이머는 OFF 상태에 들어갑니다.

큐 서비스 간격 높음 이벤트

큐 서비스 간격 이벤트가 사용 가능하게 설정(HIGH로 설정)되어야 합니다.

큐 서비스 간격 높음 이벤트는 큐 서비스 간격 정상 이벤트가 생성될 때 자동으로 사용 가능하게 됩니다.

서비스 시간이 서비스 간격보다 더 큰 경우, 다음 MQPUT 또는 Get 조작 시 또는 그 이전에 이벤트가 생성됩니다.

큐 서비스 간격 정상 이벤트

큐 서비스 간격 정상 이벤트는 큐 서비스 간격 높음 이벤트가 생성될 때 자동으로 사용 가능하게 됩니다.

서비스 시간(경과 시간)이 서비스 간격보다 작거나 같은 경우, 다음 Get 조작 시 또는 그 이전에 이벤트가 생성됩니다.

관련 태스크

26 페이지의 『큐 서비스 간격 이벤트 사용 가능』

큐 서비스 간격 이벤트에 대해 큐를 구성하려면 적절한 큐 관리자 및 큐 속성을 설정합니다.

큐 서비스 간격 이벤트 사용 가능

큐 서비스 간격 이벤트에 대해 큐를 구성하려면 적절한 큐 관리자 및 큐 속성을 설정합니다.

이 태스크 정보

높음 및 정상 이벤트는 상호 배타적입니다. 즉, 하나가 사용 가능하면 다른 하나는 자동으로 사용 불가능합니다.

- 높음 이벤트가 큐에서 생성될 때, 큐 관리자가 자동으로 높음 이벤트를 사용 불가능하게 하고 해당 큐에 대한 정상 이벤트를 사용 가능하게 합니다.
- 정상 이벤트가 큐에서 생성될 때, 큐 관리자가 자동으로 정상 이벤트를 사용 불가능하게 하고 해당 큐에 대한 높음 이벤트를 사용 가능하게 합니다.

큐 서비스 간격 이벤트	큐 속성
큐 서비스 간격 높음 큐 서비스 간격 확인 큐 서비스 간격 이벤트 없음	QSVCI EV (HIGH) QSVCI EV (OK) QSVCI EV (NONE)
서비스 간격	QSVCI NT (<i>tt</i>) 여기서 <i>tt</i> 는 밀리초 단위의 서비스 간격 시간입니다.

큐 서비스 간격 이벤트를 사용 가능하게 설정하려면 다음 단계를 수행하십시오.

프로시저

1. 큐 관리자 속성 **PERFMEV**를 ENABLED로 설정하십시오.
성능 이벤트가 큐 관리자에서 사용 가능합니다.
2. 큐의 큐 서비스 간격 높음 또는 정상 이벤트에 대해 제어 속성 **QSVCI EV**를 필요에 따라 설정하십시오.
3. 적절한 서비스 간격 시간을 지정하기 위해 큐에 대한 **QSVCI NT** 속성을 설정하십시오.

예

10초(10,000밀리초)의 서비스 간격 시간을 갖고 큐 서비스 간격 높음 이벤트를 사용 가능하게 설정하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR PERFMEV(ENABLED)
```

```
ALTER QLOCAL('MYQUEUE') QSVCINT(10000) QSVCIEV(HIGH)
```

큐 서비스 간격 이벤트 예

큐 서비스 간격 이벤트에서 얻을 수 있는 정보를 이해하려면 이 절의 예를 사용하십시오.

세 개의 하위 토픽 예가 큐 서비스 간격 이벤트 사용의 점진적으로 더 복잡한 설명을 제공합니다.

각 하위 토픽의 예에 동반되는 그림은 동일한 구조를 갖습니다.

- 그림 1은 개별 MQGET 호출 및 MQPUT 호출을 표시하는, 큐 용량 대 시간의 그래프입니다.
- 주석 섹션은 시간 제한조건의 비교를 표시합니다. 사용자가 고려해야 하는 세 개의 기간이 있습니다.
 - 사용자 정의 서비스 간격.
 - 서비스 타이머가 측정하는 시간.
 - 이벤트 통계가 마지막으로 재설정된 이후의 시간(이벤트 데이터의 TimeSinceReset).
- 이벤트 통계 요약 섹션은 임의의 순간에 사용 가능한 이벤트 및 생성되는 이벤트를 표시합니다.

예는 큐 서비스 간격 이벤트의 다음 측면을 설명합니다.

- 큐 용량이 시간에 따라 변하는 정도.
- 서비스 타이머가 측정하는 경과 시간이 서비스 간격과 비교하는 정도.
- 사용 가능한 이벤트.
- 생성되는 이벤트.

알아두기: 예 1은 메시지가 간헐적이고 다음 메시지가 도달하기 전에 각 메시지가 큐에서 제거되는 단순한 사례를 보여줍니다. 이벤트 데이터로부터 큐의 최대 메시지 수가 1이었음을 알고 있습니다. 그러므로, 각 메시지가 큐에 있었던 기간을 파악할 수 있습니다.

그러나 큐에 둘 이상의 메시지가 있고 MQGET 호출과 MQPUT 호출의 순서가 예측 불가능한 일반적인 경우, 큐 서비스 간격 이벤트를 사용하여 개별 메시지가 큐에 남아 있는 기간을 계산할 수 없습니다. 이벤트 데이터에서 리턴되는 TimeSinceReset 매개변수는 큐에 메시지가 없는 시간의 비율을 포함할 수 있습니다. 그러므로 이들 통계로부터 파생시키는 모든 결과는 암시적으로 이들 시간을 포함하도록 평균화됩니다.

관련 개념

[24 페이지의 『큐 서비스 간격 이벤트』](#)

큐 서비스 간격 이벤트는 서비스 간격이라고 부르는 사용자 정의 시간 간격 안에서 큐에 대해 조작이 수행되었는지 여부를 표시합니다. 설치에 따라서, 큐 서비스 간격 이벤트를 사용하여 메시지가 충분히 빨리 큐에서 제거되는지 여부를 모니터링할 수 있습니다.

[25 페이지의 『서비스 타이머』](#)

큐 서비스 간격 이벤트는 서비스 타이머라고 부르는 내부 타이머를 사용하는데, 이것은 큐 관리자에 의해 제어됩니다. 서비스 타이머는 큐 서비스 간격 이벤트가 사용 가능한 경우에만 사용됩니다.

큐 서비스 간격 이벤트: 예 1

MQGET 호출과 MQPUT 호출의 기본 순서로, 큐 용량은 항상 1 또는 0입니다.

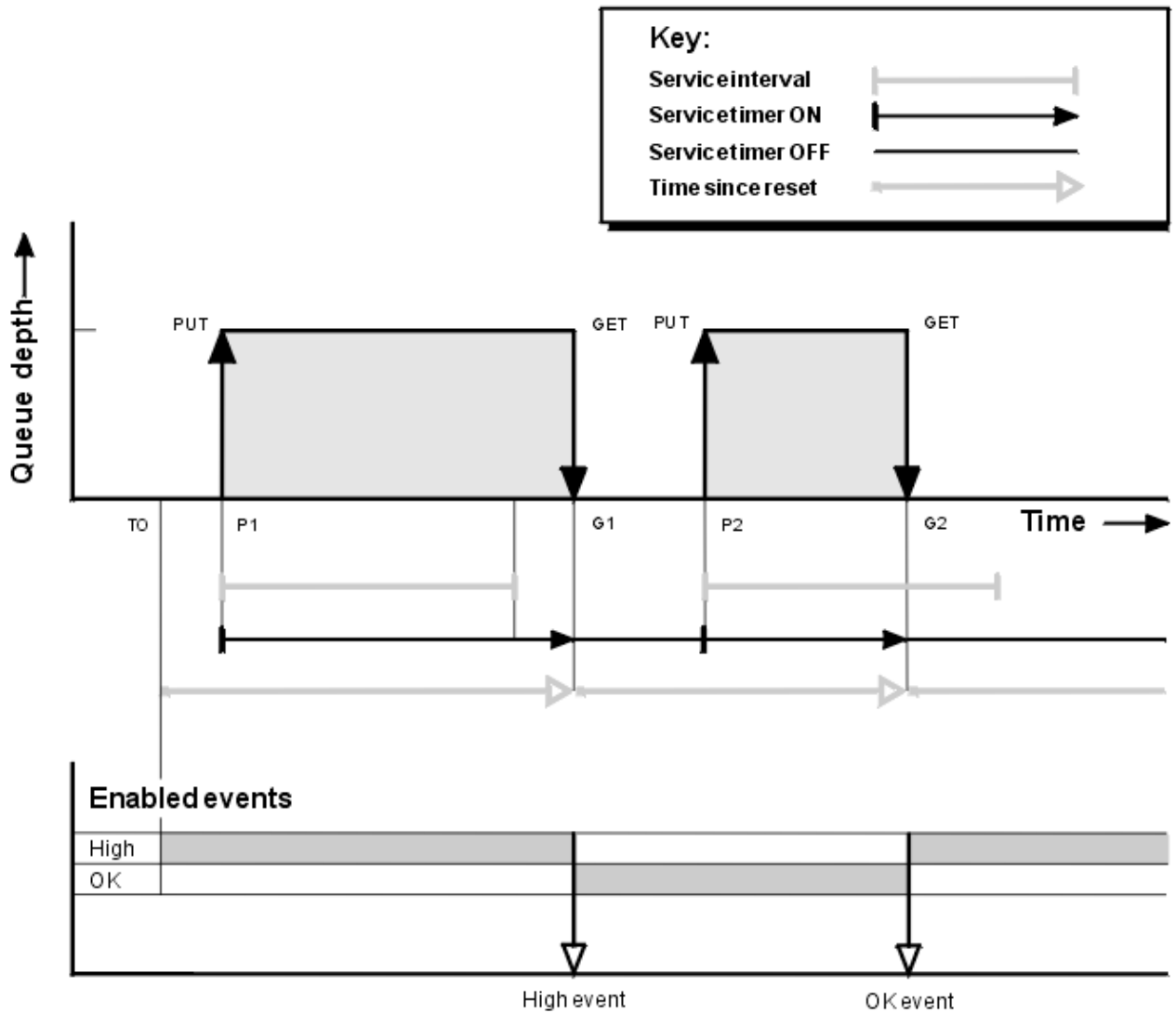


그림 4. 큐 서비스 간격 이벤트 - 예 1

주석

1. P1에서 애플리케이션이 빈 큐에 메시지를 넣습니다. 이것은 서비스 타이머를 시작합니다.
T0은 큐 관리자 시동 시간일 수 있음을 주의하십시오.
2. G1에서 다른 애플리케이션이 큐에서 메시지를 가져옵니다. P1과 G1 사이의 경과 시간이 서비스 간격보다 크기 때문에 큐 서비스 간격 높음 이벤트가 G1에서의 MQGET 호출에서 생성됩니다. 높음 이벤트가 생성될 때 큐 관리자는 다음과 같도록 이벤트 제어 속성을 재설정합니다.
 - a. 정상 이벤트가 자동으로 사용 가능하게 됩니다.
 - b. 높음 이벤트는 사용 불가능합니다.
 큐가 이제 비어 있기 때문에 서비스 타이머는 OFF 상태로 전환됩니다.
3. P2에서 두 번째 메시지가 큐에 들어갑니다. 이것은 서비스 타이머를 재시작합니다.
4. G2에 메시지가 큐에서 제거됩니다. 그러나 P2와 G2 사이의 경과 시간이 서비스 간격보다 작기 때문에 큐 서비스 간격 정상 이벤트가 G2에서 MQGET 호출에서 생성됩니다. 정상 이벤트가 생성될 때 큐 관리자는 다음과 같도록 제어 속성을 재설정합니다.
 - a. 높음 이벤트가 자동으로 사용 가능하게 됩니다.
 - b. 정상 이벤트는 사용 불가능합니다.

큐가 비어 있기 때문에 서비스 타이머는 다시 OFF 상태로 전환됩니다.

이벤트 통계 요약

29 페이지의 표 5에서는 이 예에 대한 이벤트 통계를 요약합니다.

표 5. 예 1에 대한 이벤트 통계 요약		
속성	이벤트 1	이벤트 2
이벤트 시간	T(G1)	T(G2)
이벤트의 유형	높음	확인
TimeSinceReset	T(G1) - T(0)	T(G2) - T(G1)
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

28 페이지의 그림 4의 중간 부분은 해당 큐에 대한 서비스 간격과 비교되는 서비스 타이머가 측정한 경과 시간을 보여줍니다. 큐 서비스 간격 이벤트가 발행할지 여부를 확인하려면 서비스 타이머를 나타내는 수평선(화살표를 갖는)의 길이를 서비스 간격을 나타내는 선의 길이와 비교하십시오. 서비스 타이머 선이 더 길고 큐 서비스 간격 높음 이벤트가 사용 가능한 경우, 큐 서비스 간격 높음 이벤트가 다음 GET에서 발생합니다. 타이머 선이 더 짧고 큐 서비스 간격 정상 이벤트가 사용 가능한 경우, 큐 서비스 간격 정상 이벤트가 다음 GET에서 발생합니다.

큐 서비스 간격 이벤트: 예 2

MQPUT 호출과 MQGET 호출의 순서로, 큐 용량은 항상 1 또는 0이 아닙니다.

이 예는 또한 예를 들어 시간 P2에서 이벤트가 생성되지 않고 재설정되는 타이머의 인스턴스를 보여줍니다.

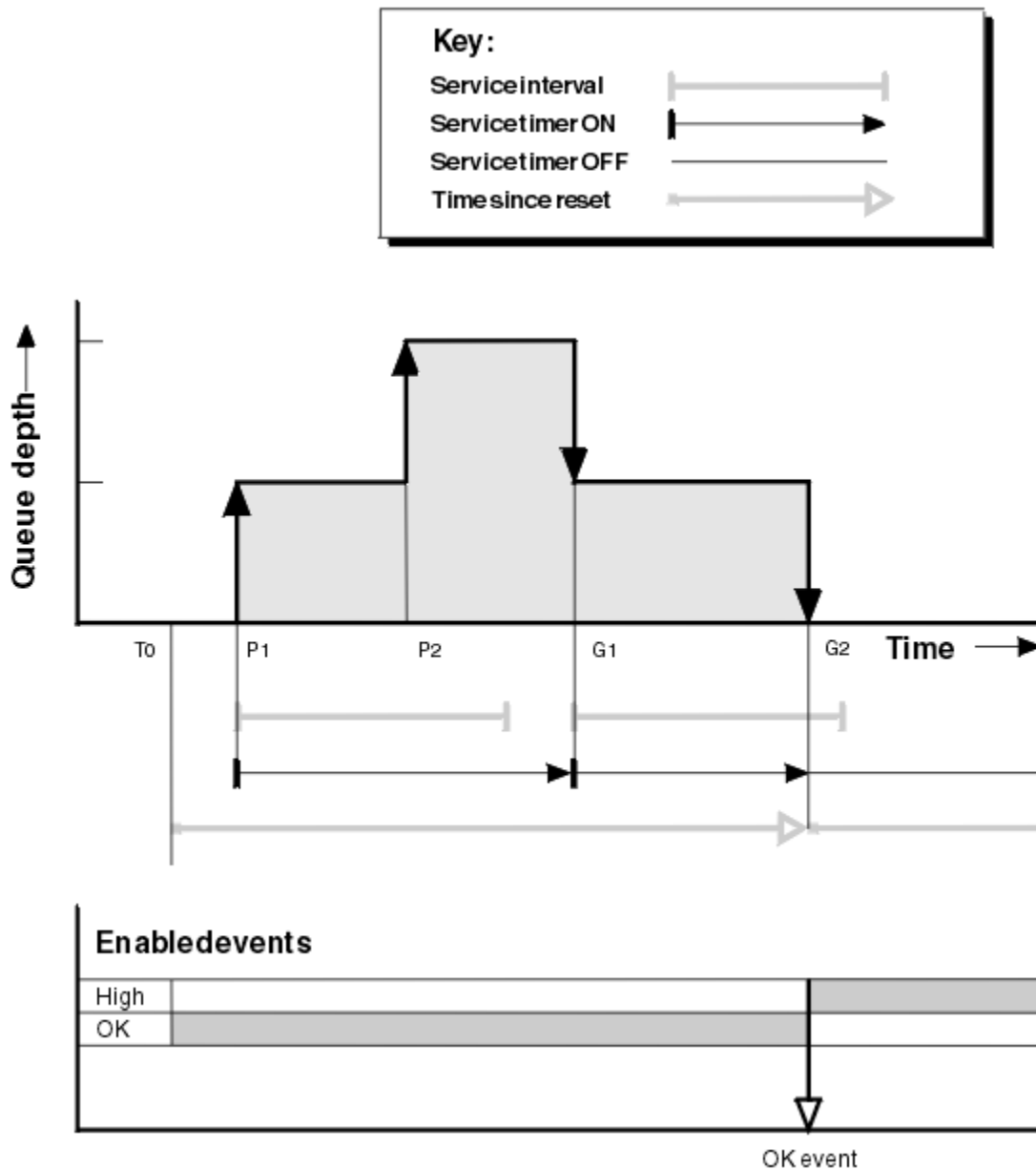


그림 5. 큐 서비스 간격 이벤트 - 예 2

주석

이 예에서 정상 이벤트가 초기에 사용 가능하고 큐 통계가 시간 T0에 재설정되었습니다.

1. P1에 첫 번째 PUT이 서비스 타이머를 시작합니다.
2. P2에서, PUT은 정상 이벤트를 유발할 수 없기 때문에 두 번째 PUT이 이벤트를 생성하지 않습니다.
3. G1에서 서비스 간격이 이제 초과되었으므로 정상 이벤트는 생성되지 않습니다. 그러나 MQGET 호출이 서비스 타이머가 재설정되게 만듭니다.
4. G2에서 서비스 간격 내에 두 번째 GET이 발생하고 이번에는 정상 이벤트가 생성됩니다. 큐 관리자는 다음과 같도록 이벤트 제어 속성을 재설정합니다.
 - a. 높음 이벤트가 자동으로 사용 가능하게 됩니다.
 - b. 정상 이벤트는 사용 불가능합니다.

큐가 이제 비어 있기 때문에 서비스 타이머는 OFF 상태로 전환됩니다.

이벤트 통계 요약

31 페이지의 표 6에서는 이 예에 대한 이벤트 통계를 요약합니다.

표 6. 예 2에 대한 이벤트 통계 요약	
속성	이벤트 2
이벤트 시간	T(G2)
이벤트의 유형	확인
TimeSinceReset	T(G2) - T(0)
HighQDepth	2
MsgEnqCount	2
MsgDeqCount	2

큐 서비스 간격 이벤트: 예 3

이전 예보다 더 산발적인 MQGET 호출과 MQPUT 호출의 순서입니다.

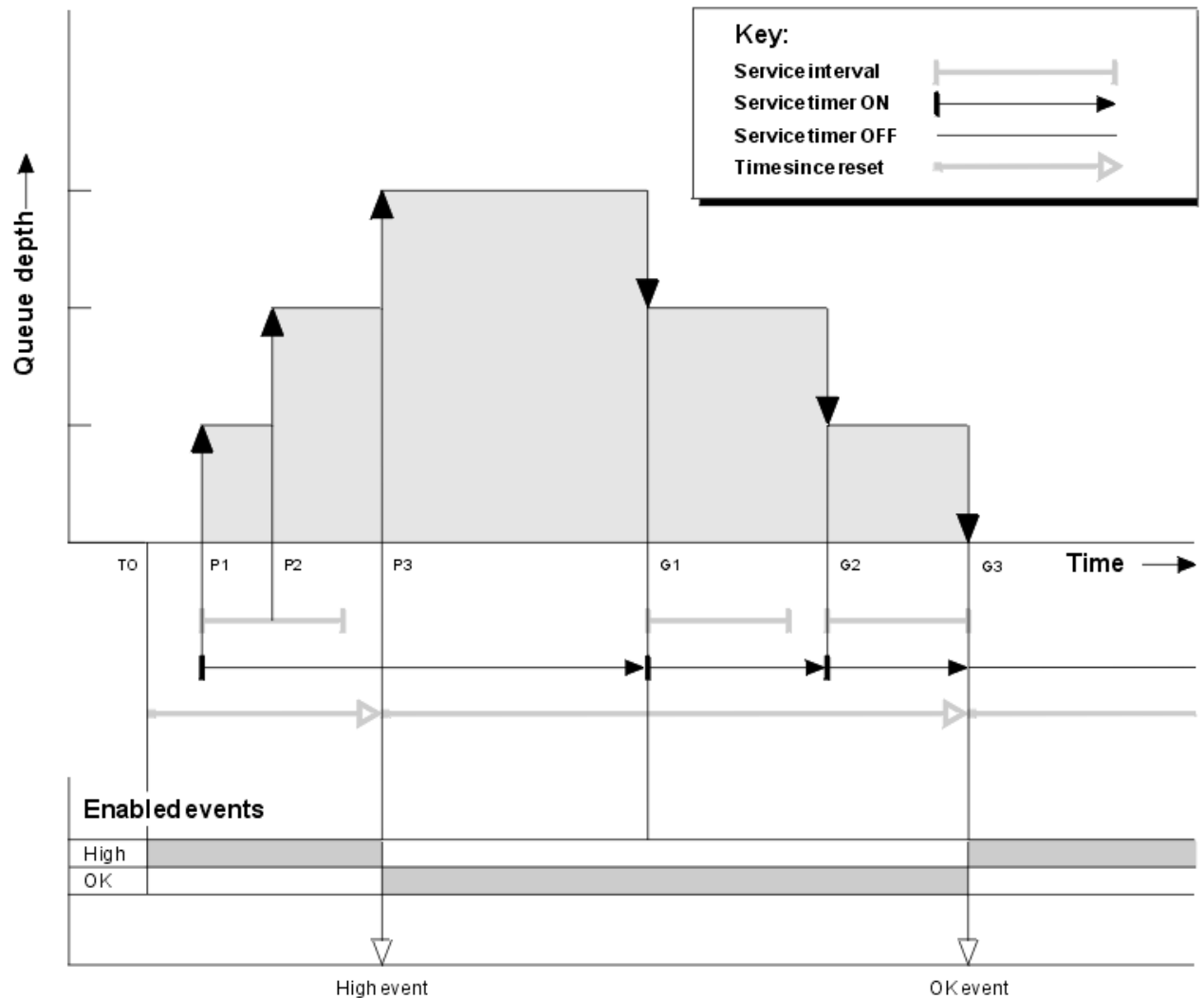


그림 6. 큐 서비스 간격 이벤트 - 예 3

주석

1. 시간 T(0)에서, 큐 통계가 재설정되고 큐 서비스 간격 높음 이벤트가 사용 가능합니다.
2. P1에 첫 번째 PUT이 서비스 타이머를 시작합니다.
3. P2에, 두 번째 PUT이 큐 용량을 2로 증가시킵니다. 서비스 간격 시간이 초과되지 않았기 때문에 높음 이벤트가 생성되지 않습니다.
4. P3에서 세 번째 PUT이 높음 이벤트가 생성되게 합니다. (타이머가 서비스 간격을 초과했습니다.) 큐 용량이 PUT 전에 0이 아니었기 때문에 타이머는 재설정되지 않습니다. 그러나 정상 이벤트가 사용 가능합니다.
5. G1에서 서비스 간격이 초과되었고 정상 이벤트가 사용 가능하기 때문에 MQGET 호출이 이벤트를 생성하지 않습니다. 그러나 MQGET 호출은 서비스 타이머를 재설정합니다.
6. G2에서 서비스 간격이 초과되었고 정상 이벤트가 사용 가능하기 때문에 MQGET 호출이 이벤트를 생성하지 않습니다. 다시, MQGET 호출이 서비스 타이머를 재설정합니다.
7. G3에서 세 번째 GET이 큐를 비우고 서비스 타이머는 서비스 간격과 동일합니다. 그러므로 정상 이벤트가 생성됩니다. 서비스 타이머가 재설정되고 높음 이벤트가 사용 가능합니다. MQGET 호출이 큐를 비우며, 이것은 타이머를 OFF 상태로 만듭니다.

이벤트 통계 요약

32 페이지의 표 7에서는 이 예에 대한 이벤트 통계를 요약합니다.

표 7. 예 3에 대한 이벤트 통계 요약		
속성	이벤트 1	이벤트 2
이벤트 시간	T(P3)	T(G3)
이벤트의 유형	높음	확인
TimeSinceReset	T(P3) - T(0)	T(G3) - T(P3)
HighQDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

큐 용량 이벤트

큐 용량 이벤트를 큐 용량, 즉 큐에 있는 메시지 수와 관련됩니다.

IBM MQ 애플리케이션에서 큐는 가득 차지 않아야 합니다. 가득 차는 경우 애플리케이션은 지정한 큐에 더 이상 메시지를 넣을 수 없습니다. 이것이 발생하는 경우 메시지가 유실되지 않더라도 가득 찬 큐는 상당한 불편을 유발할 수 있습니다. 메시지를 처리하는 애플리케이션이 덜어낼 수 있는 것보다 더 빨리 큐에 메시지가 놓여지는 경우 큐의 메시지 수가 쌓일 수 있습니다.

이 문제점의 솔루션은 특정 상황에 의존하지만 다음을 포함할 수 있습니다.

- 일부 메시지를 다른 큐로 전환.
- 큐에서 더 많은 메시지를 덜어내기 위한 새 애플리케이션 시작.
- 본질적이 아닌 메시지 트래픽 중지.
- 임시 최대를 극복하기 위해 큐 용량 증가.

문제점이 진행 중일 수 있다는 사전 경고가 예방 조치를 취하기가 더 쉽게 만듭니다. 이 목적을 위해 IBM MQ는 다음 큐 용량 이벤트를 제공합니다.

큐 용량 높음 이벤트

큐 용량이 큐 용량 상한이라고 부르는 사전 정의된 임계값으로 증가했음을 표시합니다.

큐 용량 낮음 이벤트

큐 용량이 큐 용량 하한이라고 부르는 사전 정의된 임계값으로 감소했음을 표시합니다.

큐 가득 참 이벤트

큐가 최대 용량에 도달했음, 즉 큐가 가득 찼음을 표시합니다.

큐 가득 참 이벤트는 애플리케이션이 최대 용량에 도달한 큐에 메시지를 넣으려고 시도할 때 생성됩니다. 큐 용량 높음 이벤트가 큐가 채워져 가고 있다는 사전 경고를 제공합니다. 이것은 이 이벤트를 수신했을 때 시스템 관리자가 어떤 예방 조치를 취해야 함을 의미합니다. 예방 조치가 성공하고 큐 용량이 안전한 수준으로 떨어지는 경우 큐 관리자가 큐 용량 낮음 이벤트를 생성하도록 큐 관리자를 구성할 수 있습니다.

첫 번째 큐 용량 이벤트 예는 큐가 가득 차게 되기를 막는 추정 조치의 효과를 보여줍니다.

관련 개념

36 페이지의 『[큐 용량 이벤트 예](#)』

큐 용량 이벤트로부터 얻을 수 있는 정보를 이해하려면 이들 예를 사용하십시오.

관련 참조

[큐 가득 참](#)

[큐 용량 상한](#)

[큐 용량 하한](#)

큐 용량 이벤트 사용 가능

임의의 큐 용량 이벤트에 대해 큐를 구성하려면 적절한 큐 관리자 및 큐 속성을 설정합니다.

이 태스크 정보

기본적으로, 모든 큐 용량 이벤트는 사용 불가능합니다. 사용 가능할 때 큐 용량 이벤트가 다음과 같이 생성됩니다.

- 큐 용량 높음 이벤트는 메시지를 큐에 넣어서 큐 용량이 **QDepthHighLimit**에 대해 설정된 값 이상이 될 때 생성됩니다.
 - 큐 용량 높음 이벤트는 동일한 큐에서의 큐 용량 낮음 이벤트에 의해 자동으로 사용 가능하게 됩니다.
 - 큐 용량 높음 이벤트는 동일한 큐에서 큐 용량 낮음 및 큐 가득 참 이벤트를 둘 다 자동으로 사용 가능하게 합니다.
- **Multi** 큐 용량 낮음 이벤트는 메시지가 GET 조작에 의해 큐에서 제거되어 큐 용량이 **QDepthLowLimit**에 대해 설정된 값 이하가 될 때 생성됩니다.
 - ▶ **z/OS** 큐 용량 낮음 이벤트는 메시지가 GET 조작에 의해 큐에서 제거되거나, 메시지가 만료되지 않았다면 조작에 의해 제거되었겠지만 만료로 인해 제거되어 큐 용량이 **QDepthLowLimit**에 대해 설정된 값 이하가 될 때 생성됩니다.
 - 큐 용량 낮음 이벤트는 동일한 큐에서의 큐 용량 높음 이벤트 또는 큐 가득 참 이벤트에 의해 자동으로 사용 가능하게 됩니다.
 - 큐 용량 낮음 이벤트는 동일한 큐에서 큐 용량 높음 및 큐 가득 참 이벤트를 둘 다 자동으로 사용 가능하게 합니다.
- 큐 가득 참 이벤트는 큐가 가득 찼기 때문에 애플리케이션이 큐에 메시지를 넣을 수 없을 때 생성됩니다.
 - 큐 가득 참 이벤트는 동일한 큐에서의 큐 용량 높음 또는 큐 용량 낮음 이벤트에 의해 자동으로 사용 가능하게 됩니다.
 - 큐 가득 참 이벤트는 동일한 큐에서 큐 용량 낮음 이벤트를 자동으로 사용 가능하게 합니다.

큐 용량 이벤트에 대해 큐를 구성하려면 다음 단계를 수행하십시오.

프로시저

1. 큐 관리자 속성 **PERFMEV**를 사용하여 큐 관리자에서 성능 이벤트를 사용 가능하게 하십시오. 이벤트가 **SYSTEM.ADMIN.PERFM.EVENT** 큐로 이동합니다.
2. 필수 큐에서 이벤트를 사용 가능하게 설정하려면 다음 속성 중 하나를 설정하십시오.
 - **QDepthHighEvent**(MQSC에서는 **QDPHIEV**)
 - **QDepthLowEvent**((MQSC에서는 **QDPLOEV**)
 - **QDepthMaxEvent**((MQSC에서는 **QDPMAXEV**)

3. 옵션: 한계를 설정하려면 최대 큐 용량의 백분율로서 다음 속성을 지정하십시오.

- `QDepthHighLimit`((MQSC에서는 **QDEPTHHI**)
- `QDepthLowLimit`((MQSC에서는 **QDEPTHLO**)

제한사항: **QDEPTHHI**는 **QDEPTHLO**보다 작지 않아야 합니다.

QDEPTHHI가 **QDEPTHLO**와 동일하면 큐 용량이 둘 중 어느 방향으로든 값을 초과할 때마다 이벤트 메시지가 생성됩니다. 큐 용량이 이러한 값보다 작으면 상한 임계값이 사용되고, 이러한 값보다 크면 하한 임계값이 사용되기 때문입니다.

결과

참고:

Multi 만료된 메시지가 GET 조작에 의해 큐에서 제거되어 큐 용량이 **QDepthLowLimit**에 대해 설정된 값 이하가 될 때는 큐 용량 낮음 이벤트가 생성되지 않습니다. IBM MQ는 성공적인 GET 조작 중에만 큐 용량 낮음 이벤트를 생성합니다. 그러므로, 만기된 메시지가 큐에서 제거될 때 큐 용량 낮음 메시지가 생성되지 않습니다. 또한, 이러한 만료된 메시지가 큐에서 제거된 후에는 **QDepthHighEvent** 및 **QDepthLowEvent**가 재설정되지 않습니다.

z/OS IBM MQ는 성공적인 파괴적 GET 조작, 또는 일치하는 메시지가 만료되지 않았다면 성공했을 파괴적 GET 조작 중에 큐 용량 낮음 이벤트 메시지를 생성합니다. 그렇지 않고 만료된 메시지가 일반 백그라운드 처리 중에 큐에서 제거된 경우에는 큐 용량 낮음 이벤트 메시지가 생성되지 않습니다. 또한 일반 백그라운드 처리 중에 만료된 메시지가 큐에서 제거된 후에는 **QDepthHighEvent** 및 **QDepthLowEvent**가 재설정되지 않습니다. 만료된 메시지 처리에 대한 자세한 정보는 [IBM MQ for z/OS에서 큐 관리자 조정을 참조하십시오](#).

예

80%에서 설정된 한계를 갖고 MYQUEUE 큐에서 큐 용량 높음 이벤트를 사용 가능하게 설정하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

20%에서 설정된 한계를 갖고 MYQUEUE 큐에서 큐 용량 낮음 이벤트를 사용 가능하게 설정하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHLO(20) QDPLOEV(ENABLED)
```

MYQUEUE 큐에서 큐 가득 참 이벤트를 사용 가능하게 설정하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDPMAXEV(ENABLED)
```

z/OS *Shared queues and queue depth events on z/OS*

On IBM MQ for z/OS, event monitoring is more straightforward for an application that uses shared queues if all the queue managers in the queue sharing group have the same setting for the **PERFMEV** attribute.

When a queue depth event occurs on a shared queue, and the queue manager attribute **PERFMEV** is set to **ENABLED**, the queue managers in the queue sharing group produce an event message. If **PERFMEV** is set to **DISABLED** on some of the queue managers, event messages are not produced by those queue managers, making event monitoring from an application more difficult. For more straightforward monitoring, give each queue manager the same setting for the **PERFMEV** attribute.

This event message that each queue manager generates represents its individual usage of the shared queue. If a queue manager performs no activity on the shared queue, various values in the event message are null or zero. You can use null event messages as follows:

- Ensure that each active queue manager in a queue sharing group generates one event message
- Highlight cases of no activity on a shared queue for the queue manager that produced the event message

Coordinating queue manager

When a queue manager issues a queue depth event, it updates the shared queue object definition to toggle the active performance event attributes. For example, depending on the definition of the queue attributes, a Queue Depth High event enables a Queue Depth Low and a Queue Full event. After updating the shared queue object successfully, the queue manager that detected the performance event initially becomes the *coordinating queue manager*.

If enabled for performance events, the coordinating queue manager performs the following actions:

1. Issues an event message that captures all shared queue performance data it has gathered since the last time an event message was created, or since the queue statistics were last reset. The message descriptor (MQMD) of this message contains a unique correlation identifier (*CorrelId*) created by the coordinating queue manager.
2. Broadcasts to all other *active* queue managers in the same queue sharing group to request the production of an event message for the shared queue. The broadcast contains the correlation identifier created by the coordinating queue manager for the set of event messages.

Having received a request from the coordinating queue manager, if there is an active queue manager in the queue sharing group that is enabled for performance events, that active queue manager issues an event message for the shared queue. The event message that is issued contains information about all the operations performed by the receiving (active) queue manager since the last time an event message was created, or since the statistics were last reset. The message descriptor (MQMD) of this event message contains the unique correlation identifier (*CorrelId*) specified by the coordinating queue manager.

When performance events occur on a shared queue, n event messages are produced, where n is a number from 1 to the number of active queue managers in the queue sharing group. Each event message contains data that relates to the shared queue activity for the queue manager that generated the event message.

Differences between shared and nonshared queues

Enabling queue depth events on shared queues differs from enabling them on nonshared queues. A key difference is that events are enabled for shared queues even if **PERFMEV** is DISABLED on the queue manager. This is not the case for nonshared queues.

Consider the following example, which illustrates this difference:

- QM1 is a queue manager with *PerformanceEvent* (**PERFMEV** in MQSC) set to DISABLED.
- SQ1 is a shared queue with **QSGDISP** set to (SHARED) QLOCAL in MQSC.
- LQ1 is a nonshared queue with **QSGDISP** set to (QMGR) QLOCAL in MQSC.

Both queues have the following attributes set on their definitions:

- **QDPHIEV (ENABLED)**
- **QDPLOEV (DISABLED)**
- **QDPMAXEV (DISABLED)**

If messages are placed on both queues so that the depth meets or exceeds the **QDEPTHHI** threshold, the **QDPHIEV** value on SQ1 switches to DISABLED. Also, **QDPLOEV** and **QDPMAXEV** are switched to ENABLED.

SQ1's attributes are automatically switched for each performance event at the time the event criteria are met.

In contrast the attributes for LQ1 remain unchanged until **PERFMEV** on the queue manager is ENABLED. This means that if, for example, the queue manager's **PERFMEV** attribute is ENABLED, DISABLED and then set to ENABLED again, the performance event settings on shared queues might not be consistent with those of nonshared queues, even though they might have initially been the same.

큐 용량 이벤트 예

큐 용량 이벤트로부터 얻을 수 있는 정보를 이해하려면 이들 예를 사용하십시오.

첫 번째 예는 큐 용량 이벤트의 기본 설명을 제공합니다. 두 번째 예는 더 방대하지만 원리는 첫 번째 예와 동일합니다. 두 예는 모두 다음과 같이 동일한 큐 정의를 사용합니다.

MYQUEUE1 큐는 1000 메시지의 최대 용량을 갖습니다. 큐 용량 상한은 80%이고 큐 용량 하한은 20%입니다. 초기에는 큐 용량 높음 이벤트가 사용 가능한 반면, 다른 큐 용량 이벤트는 사용 불가능합니다.

이 큐를 구성하는 IBM MQ 명령(MQSC)은 다음과 같습니다.

```
ALTER QMGR PERFMEV(ENABLED)

DEFINE QLOCAL('MYQUEUE1') MAXDEPTH(1000) QDPMAXEV(DISABLED) QDEPTHHI(80)
QDPHIEV(ENABLED) QDEPTHLO(20) QDPLOEV(DISABLED)
```

관련 개념

[32 페이지의 『큐 용량 이벤트』](#)

큐 용량 이벤트를 큐 용량, 즉 큐에 있는 메시지 수와 관련됩니다.

관련 태스크

[33 페이지의 『큐 용량 이벤트 사용 가능』](#)

임의의 큐 용량 이벤트에 대해 큐를 구성하려면 적절한 큐 관리자 및 큐 속성을 설정합니다.

관련 참조

[MQSC 명령](#)

큐 용량 이벤트: 예 1

큐 용량 이벤트의 기본 순서입니다.

[37 페이지의 그림 7](#)에서는 시간에 따른 큐 용량의 변화를 보여줍니다.

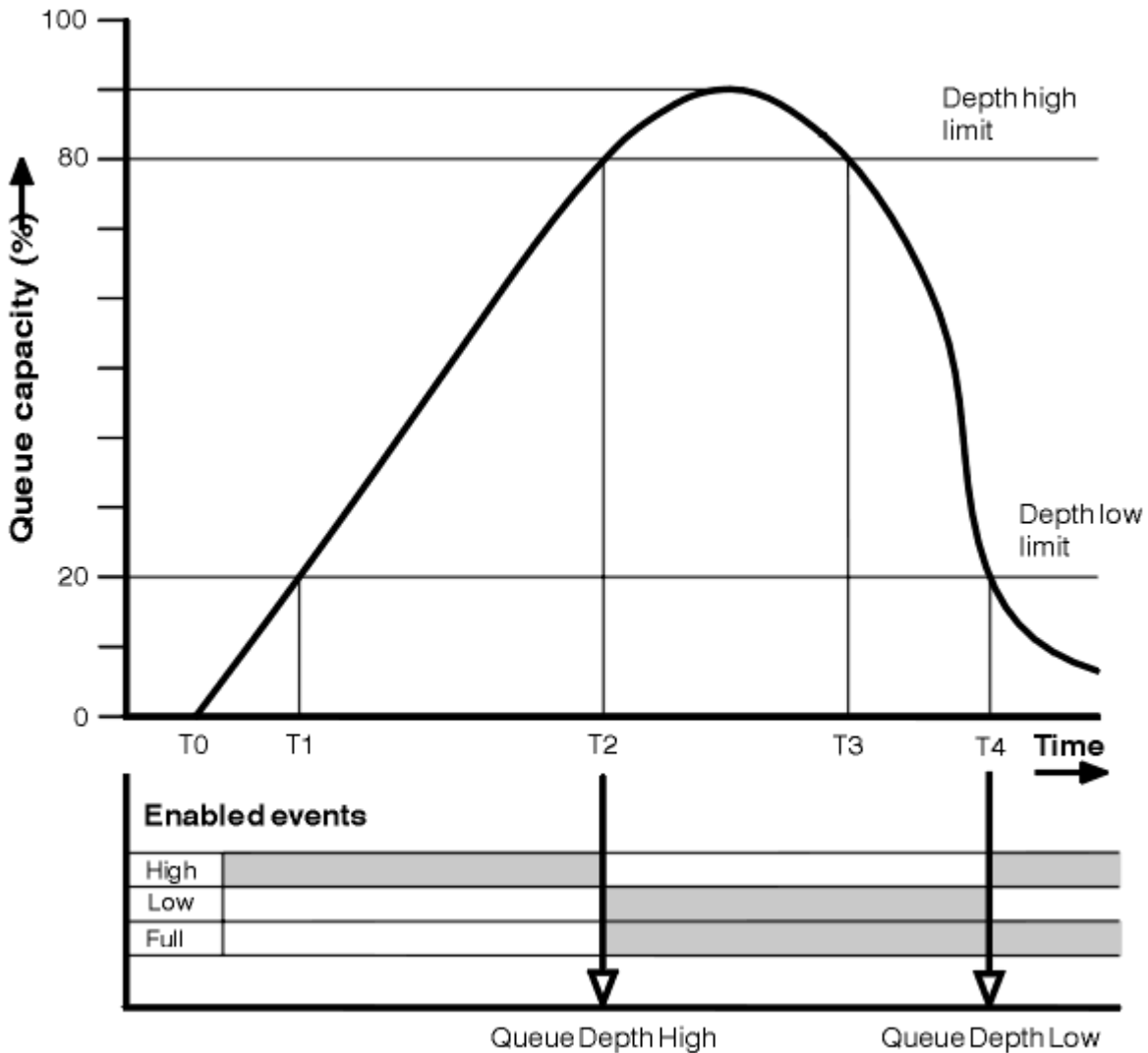


그림 7. 큐 용량 이벤트 (1)

주석

1. T(1)에서, 큐 용량이 증가하고(MQGET 호출보다 더 많은 MQPUT 호출) 큐 용량 하한을 지납니다. 이 때는 이벤트가 생성되지 않습니다.
2. 큐 용량이 T(2)까지 계속 증가하고, 이 때 용량 상한(80%)에 도달하고 큐 용량 높음 이벤트가 생성됩니다.
이것은 큐 가득 참 및 큐 용량 낮음 이벤트를 둘 다 사용 가능하게 합니다.
3. 이벤트에 의해 착수되는 (추정된) 예방 조치가 큐가 가득 차지 못하게 막습니다. 시간 T(3)까지, 큐 용량 상한에 다시 도달되었고, 이 시간부터 위입니다. 이 때는 이벤트가 생성되지 않습니다.
4. 용량 하한(20%)에 도달하고 큐 용량 낮음 이벤트가 생성되는 T(4)까지 큐 용량이 계속 감소합니다.
이것은 큐 가득 참 및 큐 용량 높음 이벤트를 둘 다 사용 가능하게 합니다.

이벤트 통계 요약

38 페이지의 표 8에서는 큐 이벤트 통계를 요약하고 38 페이지의 표 9에서는 사용 가능한 이벤트를 요약합니다.

표 8. 큐 용량 이벤트(예 1)에 대한 이벤트 통계 요약		
이벤트 통계	이벤트 2	이벤트 4
이벤트 시간	T(2)	T(4)
이벤트의 유형	큐 용량 상한	큐 용량 하한
TimeSinceReset	T(2) - T(0)	T(4) - T(2)
HighQDepth(재설정 이후 최대 큐 용량)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

표 9. 사용 가능한 이벤트를 표시하는 요약			
시간 기간	큐 용량 높음 이벤트	큐 용량 낮음 이벤트	큐 가득 참 이벤트
T(1) 이전	ENABLED	-	-
T(1) - T(2)	ENABLED	-	-
T(2) - T(3)	-	ENABLED	ENABLED
T(3) - T(4)	-	ENABLED	ENABLED
T(4) 이후	ENABLED	-	ENABLED

큐 용량 이벤트: 예 2

큐 용량 이벤트의 더 방대한 순서입니다.

39 페이지의 그림 8에서는 시간에 따른 큐 용량의 변화를 보여줍니다.

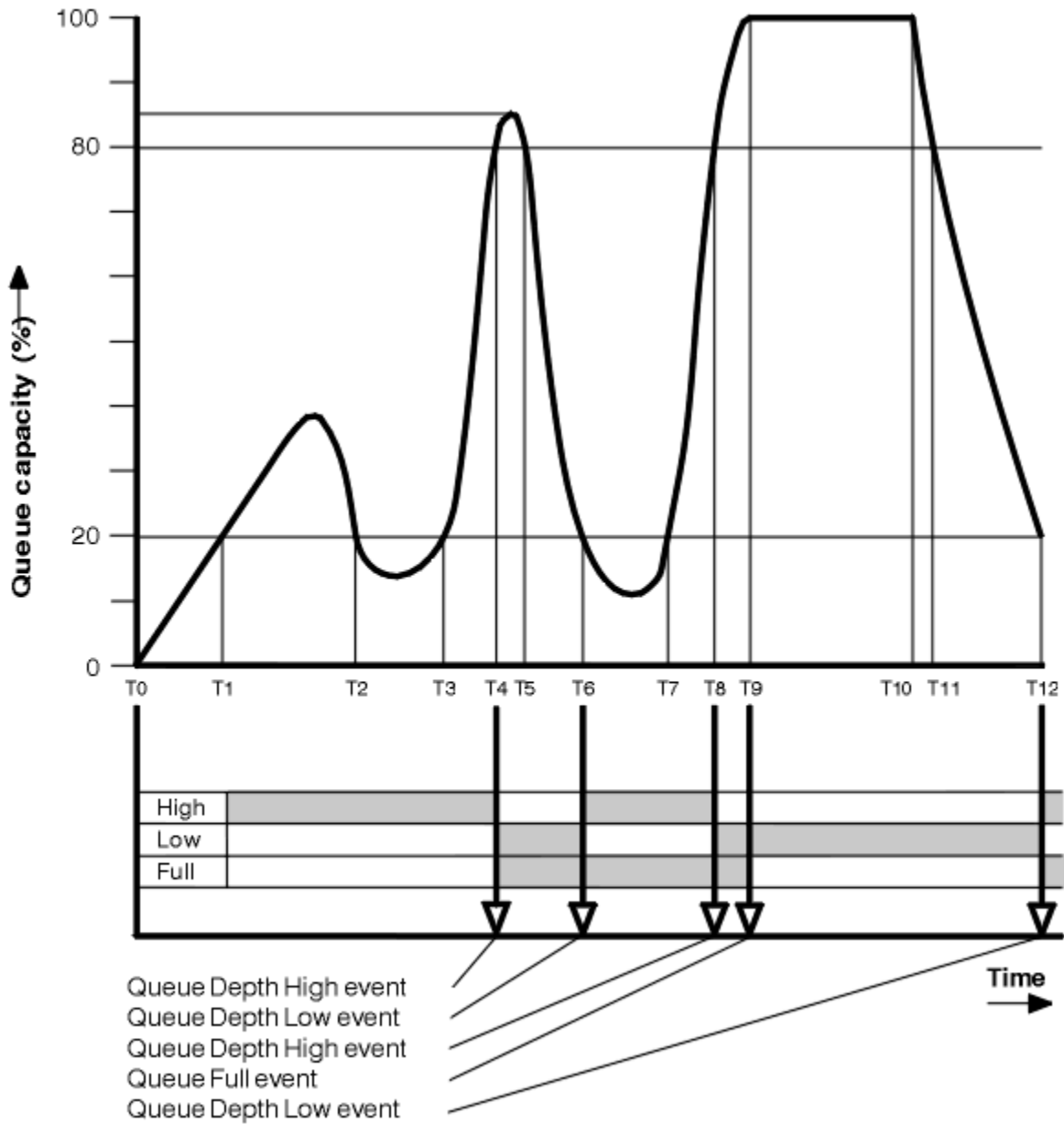


그림 8. 큐 용량 이벤트 (2)

주석

- 다음 시간에는 큐 용량 낮음 이벤트가 생성되지 않습니다.
 - T(1) (큐 용량이 증가하고, 사용 가능하지 않음)
 - T(2) (사용 가능하지 않음)
 - T(3) (큐 용량이 증가하고, 사용 가능하지 않음)
- T(4)에서 큐 용량 높음 이벤트가 발생합니다. 이것은 큐 가득 참 및 큐 용량 낮음 이벤트를 둘 다 사용 가능하게 합니다.
- T(9)에서 큐가 가득 찼기 때문에 큐에 넣을 수 없는 첫 번째 메시지 후에 큐 가득 참 이벤트가 발생합니다.
- T(12)에서 큐 용량 낮음 이벤트가 발생합니다.

이벤트 통계 요약

40 페이지의 표 10에서는 큐 이벤트 통계를 요약하고 40 페이지의 표 11에서는 이 예에 대해 다른 시간에 사용 가능한 이벤트를 요약합니다.

이벤트 통계	이벤트 4	이벤트 6	이벤트 8	이벤트 9	이벤트 12
이벤트 시간	T(4)	T(6)	T(8)	T(9)	T(12)
이벤트의 유형	큐 용량 상한	큐 용량 하한	큐 용량 상한	큐 가득 참	큐 용량 하한
TimeSinceReset	T(4) - T(0)	T(6) - T(4)	T(8) - T(6)	T(9) - T(8)	T(12) - T(9)
HighQDepth	800	855	800	1000	1000
MsgEnqCount	1645	311	1377	324	221
MsgDeqCount	845	911	777	124	1021

시간 기간	큐 용량 높음 이벤트	큐 용량 낮음 이벤트	큐 가득 참 이벤트
T(0) - T(4)	ENABLED	-	-
T(4) - T(6)	-	ENABLED	ENABLED
T(6) - T(8)	ENABLED	-	ENABLED
T(8) - T(9)	-	ENABLED	ENABLED
T(9) - T(12)	-	ENABLED	-
T(12) 이후	ENABLED	-	ENABLED

참고: 이벤트를 동기점을 벗어납니다. 그러므로 빈 큐를 가진 후 채워서 이벤트를 유발한 후, 동기점 관리자의 제어 하에 모든 메시지를 롤백할 수 있습니다. 그러나, 이벤트 사용 가능은 자동으로 설정되었으므로, 다음에 큐가 채워질 때 이벤트가 생성되지 않습니다.

구성 이벤트

구성 이벤트는 오브젝트가 작성, 변경 또는 삭제될 때 생성되며 명시적 요청에 의해서도 생성될 수 있는 알림입니다.

이벤트를 켜는 방법에 대한 정보는 20 페이지의 『구성, 명령 및 로거 이벤트 제어』의 내용을 참조하십시오.

구성 이벤트는 사용자에게 오브젝트 속성에 대한 변경을 알려줍니다. 4가지 유형의 구성 이벤트가 있습니다.

- 오브젝트 작성 이벤트
- 오브젝트 변경 이벤트
- 오브젝트 삭제 이벤트
- 오브젝트 새로 고치기 이벤트

이벤트 데이터는 다음 정보를 포함합니다.

원본 정보

변경이 수행된 큐 관리자, 변경을 수행한 사용자의 ID, 변경을 수행한 방식(예를 들어, 콘솔 명령으로) 등을 포함합니다.

컨텍스트 정보

명령 메시지의 메시지 데이터의 컨텍스트 정보의 복제본입니다.

컨텍스트 정보는 명령이 SYSTEM.COMMAND.INPUT 큐에 메시지로써 입력되었을 때만 이벤트 데이터에 포함됩니다.

오브젝트 ID

오브젝트의 이름, 유형 및 속성 지정으로 구성됩니다.

오브젝트 속성

오브젝트에 있는 모든 속성의 값으로 구성됩니다.

이벤트는 PCF 형식의 메시지입니다. 자세한 정보는 다음을 참조하십시오.

- [변경 오브젝트](#)
- [오브젝트 작성](#)
- [오브젝트 삭제](#)
- [오브젝트 새로 고치기](#)

오브젝트 변경 이벤트가 발생할 경우 변경 전 메시지와 변경 후 메시지가 생성됩니다. 일치하는 메시지를 식별할 수 있도록 이 메시지에는 다음 필드가 공통으로 있습니다.

- 동일한 상관 ID
- PCF 헤더:
 - 변경 전 메시지의 헤더: MsgSeqNumber': 1, 'Control': 'NOT_LAST'
 - 변경 후 메시지의 헤더: MsgSeqNumber': 2, 'Control': 'LAST'

생성되는 모든 구성 이벤트 메시지는 SYSTEM.ADMIN.CONFIG.EVENT 큐에 위치합니다.

관련 개념

15 페이지의 『구성 이벤트』

구성 이벤트는 구성 이벤트가 명시적으로 요청될 때 또는 오브젝트가 작성, 수정 또는 삭제될 때 자동으로 생성됩니다.

관련 참조

[오브젝트 작성](#)

[오브젝트 변경](#)

[오브젝트 삭제](#)

[오브젝트 새로 고치기](#)

11 페이지의 『이벤트 유형』

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

구성 이벤트 생성

구성 이벤트가 생성되게 하는 명령을 보고 구성 이벤트가 생성되지 않는 상황을 이해하려면 이 페이지를 사용하십시오.

구성 이벤트 메시지는 CONFIGEV 큐 관리자 속성이 ENABLED일 때 구성 이벤트 큐에 넣어집니다.

- 다음 명령 중 하나 또는 동등한 PCF가 실행됩니다.
 - DELETE AUTHINFO
 - DELETE CFSTRUCT
 - DELETE CHANNEL
 - DELETE NAMELIST
 - DELETE PROCESS
 - DELETE QMODEL/QALIAS/QREMOTE
 - DELETE STGCLASS
 - DELETE TOPIC
 - REFRESH QMGR
- 다음 명령 중 하나 또는 동등한 PCF가 오브젝트에 대한 변경이 없는 경우에도 실행됩니다.
 - DEFINE/ALTER AUTHINFO
 - DEFINE/ALTER CFSTRUCT

- DEFINE/ALTER CHANNEL
- DEFINE/ALTER NAMELIST
- DEFINE/ALTER PROCESS
- DEFINE/ALTER QMODEL/QALIAS/QREMOTE
- DEFINE/ALTER STGCLASS
- DEFINE/ALTER TOPIC
- DEFINE MAXSMSGS
- SET CHLAUTH
- CONFIGEV 속성이 DISABLED이고 ENABLED로 변경되지 않은 경우 ALTER QMGR
- 다음 명령 중 하나 또는 동등한 PCF가 큐에 대한 변경이 없는 경우에도 임시 동적이 아닌 로컬 큐에 대해 실행됩니다.
 - DELETE QLOCAL
 - DEFINE/ALTER QLOCAL
- 오브젝트에 대한 변경이 없는 경우에도 임시 동적 큐의 경우와는 다른 MQSET 호출이 실행됩니다.

구성 이벤트가 생성되지 않을 때

구성 이벤트 메시지는 다음 상황에서 생성되지 않습니다.

- 명령 또는 MQSET 호출이 실패할 때
- 큐 관리자가 이벤트 큐에 구성 이벤트를 넣으려고 시도하는 오류가 발생할 때. 그 경우 명령 또는 MQSET 호출이 완료하지만 이벤트 메시지는 생성되지 않습니다.
- 임시 동적 큐의 경우
- 내부 변경이 TRIGGER 큐 속성에 작성될 때
- REFRESH QMGR 명령에 의해 경우를 제외하고, SYSTEM.ADMIN.CONFIG.EVENT 구성 이벤트 큐의 경우
- 클러스터링 변경을 유발하는 REFRESH/RESET CLUSTER 및 RESUME/SUSPEND QMGR 명령의 경우
- 큐 관리자를 작성 또는 삭제할 때

관련 개념

[프로그래밍 가능 명령 형식 소개](#)

40 페이지의 『구성 이벤트』

구성 이벤트는 오브젝트가 작성, 변경 또는 삭제될 때 생성되며 명시적 요청에 의해서도 생성될 수 있는 알림입니다.

관련 참조

[MQSC 명령](#)

[MQSET - 오브젝트 속성 설정](#)

[MQSET - 오브젝트 속성 설정](#)

구성 이벤트 사용법

시스템에 관한 정보를 얻기 위해 구성 이벤트를 사용할 수 있는 방법을 보고 CMDSCOPE 같이 구성 이벤트의 사용에 영향을 줄 수 있는 요인을 이해하려면 이 페이지를 사용하십시오.

다음 목적을 위해 구성 이벤트를 사용할 수 있습니다.

1. 보고서가 생성될 수 있으며 시스템의 구조에 관한 정보가 생성될 수 있는 중앙 구성 저장소를 생성하고 유지 보수하기 위해.
2. 감사 추적을 생성하기 위해. 예를 들어, 오브젝트가 예기치 않게 변경되는 경우 변경을 작성한 사용자와 작성된 시간에 관한 정보가 저장될 수 있습니다.

이것은 특히 명령 이벤트도 사용 가능할 때 유용할 수 있습니다. MQSC 또는 PCF 명령이 구성 이벤트 및 명령 이벤트가 생성되도록 하는 경우, 이벤트 메시지가 둘 다 메시지 디스크립터에서 동일한 상관 ID를 공유합니다.

MQSET 호출 또는 다음 명령 중 하나의 경우,

- DEFINE 오브젝트
- ALTER 오브젝트
- DELETE 오브젝트

큐 관리자 속성 CONFIGEV가 사용 가능하지만 구성 이벤트 메시지가 구성 이벤트 큐에 넣어질 수 없는 경우, 예를 들어 이벤트 큐가 정의되지 않은 경우 명령이나 MQSET 호출이 상관없이 실행됩니다.

CMDSCOPE의 효과

CMDSCOPE가 사용되는 명령의 경우 구성 이벤트 메시지 또는 메시지들이 명령이 입력되는 곳이 아니라 명령이 실행되는 큐 관리자(들)에서 생성됩니다. 그러나 CMDSCOPE를 사용하는 명령이 소스 큐 관리자에 의해 생성된 명령인 경우에도 이벤트 데이터의 모든 원본 및 컨텍스트 정보가 입력된 원래 명령과 관련됩니다.

큐 공유 그룹이 현재 버전에 있지 않은 경우 큐 관리자를 포함하는 경우, 이전 버전에 있는 것이 아니라 현재 버전에 있는 큐 관리자에서 CMDSCOPE를 통해 실행되는 모든 명령에 대해 이벤트가 생성됩니다. 이것은 명령이 입력되는 큐 관리자가 이전 버전에 있는 경우에도 그런 경우에 컨텍스트 정보가 이벤트 데이터에 포함됨에도 불구하고 발생합니다.

관련 개념

[프로그래밍 가능 명령 형식 소개](#)

[40 페이지의 『구성 이벤트』](#)

구성 이벤트는 오브젝트가 작성, 변경 또는 삭제될 때 생성되며 명시적 요청에 의해서도 생성될 수 있는 알림입니다.

관련 참조

[MQSET - 오브젝트 속성 설정](#)

[MQSET - 오브젝트 속성 설정](#)

오브젝트 새로 고치기 구성 이벤트

오브젝트 새로 고치기 구성 이벤트는 명시적으로 요청될 때만 발생하기 때문에 다른 구성 이벤트와는 다릅니다.

작성, 변경 및 삭제 이벤트는 MQSET 호출이나 오브젝트를 변경하기 위한 명령에 의해 생성되지만 오브젝트 새로 고치기 이벤트는 MQSC 명령 REFRESH QMGR 또는 그의 동등한 PCF에 의해 명시적으로 요청될 때만 발생합니다.

REFRESH QMGR 명령은 구성 이벤트를 생성하는 다른 모든 명령과 다릅니다. 다른 모든 명령은 특정 오브젝트에 적용되며 해당 오브젝트에 대한 단일 구성 이벤트를 생성합니다. REFRESH QMGR 명령은 큐 관리자에 의해 저장되는 모든 오브젝트 정의를 잠재적으로 나타내는 많은 구성 이벤트 메시지를 생성할 수 있습니다. 선택되는 각 오브젝트에 대해 하나의 이벤트 메시지가 생성됩니다.

REFRESH QMGR 명령은 세 선택 기준의 조합을 사용하여 포함된 오브젝트의 수를 필터링합니다.

- 오브젝트 이름
- 오브젝트 유형
- 새로 고치기 간격

REFRESH QMGR 명령에 어떤 선택 기준도 지정하지 않는 경우, 각 선택 기준에 대해 기본값이 사용되며 큐 관리자에 의해 저장되는 모든 오브젝트 정의에 대해 구성 새로 고치기 이벤트 메시지가 생성됩니다. 이것은 허용할 수 없는 처리 시간 및 이벤트 메시지 생성을 유발할 수 있습니다. 몇 가지 선택 기준 지정을 고려하십시오.

새로 고치기 이벤트를 생성하는 REFRESH QMGR 명령은 다음 상황에서 사용될 수 있습니다.

- 오브젝트가 최근에 조작되었는지 여부와 상관없이 시스템에 있는 오브젝트의 일부 또는 전부에 관한 구성 데이터를 원하는 경우, 예를 들어 구성 이벤트가 처음 사용 가능할 때.

각각이 서로 다른 오브젝트의 선택을 갖지만 모두가 포함되는 여러 명령의 사용을 고려하십시오.

- SYSTEM.ADMIN.CONFIG.EVENT 큐에 오류가 있었던 경우. 이 상황에서는 작성, 변경 또는 삭제 이벤트에 대해 구성 이벤트 메시지가 생성되지 않습니다. 큐의 오류가 정정되었을 때, 큐 관리자 새로 고치기 명령이 이벤트 메시지의 생성을 요청하는 데 사용될 수 있는데 큐에 오류가 있는 동안 유실되었습니다. 이 상황에서는 새로 고치기 간격을 큐가 사용 불가능한 시간으로 설정할 것을 고려하십시오.

관련 개념

[40 페이지의 『구성 이벤트』](#)

구성 이벤트는 오브젝트가 작성, 변경 또는 삭제될 때 생성되며 명시적 요청에 의해서도 생성될 수 있는 알림입니다.

관련 참조

[REFRESH QMGR](#)

[큐 관리자 새로 고치기](#)

명령 이벤트

명령 이벤트는 MQSC 또는 PCF 명령이 성공적으로 실행했다는 알림입니다.

이벤트 데이터는 다음 정보를 포함합니다.

원본 정보

명령이 실행된 큐 관리자, 명령을 실행한 사용자의 ID 및 명령이 실행된 방법(예를 들어, 콘솔 명령에 의해)으로 구성됩니다.

컨텍스트 정보

명령 메시지의 메시지 데이터의 컨텍스트 정보의 복제본입니다. 명령이 메시지를 사용하여 입력되지 않은 경우 컨텍스트 정보는 생략됩니다.

컨텍스트 정보는 명령이 SYSTEM.COMMAND.INPUT 큐에 메시지로서 입력되었을 때만 이벤트 데이터에 포함됩니다.

명령 정보

실행된 명령의 유형입니다.

명령 데이터

- PCF 명령의 경우 명령 데이터의 복제본
- MQSC 명령의 경우 명령 텍스트

명령 데이터 형식이 원래 명령의 형식과 반드시 일치하지는 않습니다. 예를 들어, [멀티플랫폼에서 원래 요청](#)이 MQSC 명령인 경우에도 명령 데이터 형식은 항상 PCF 형식입니다.

생성되는 모든 명령 이벤트 메시지는 명령 이벤트 큐 SYSTEM.ADMIN.COMMAND.EVENT에 위치합니다.

관련 참조

[명령](#)

[11 페이지의 『이벤트 유형』](#)

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

명령 이벤트 생성

명령 이벤트가 생성되게 하는 상황을 보고 명령 이벤트가 생성되지 않는 상황을 이해하려면 이 페이지를 사용하십시오.

명령 이벤트 메시지는 다음 상황에서 생성됩니다.

- CMDEV 큐 관리자 속성이 ENABLED로 지정되고 MQSC 또는 PCF 명령이 성공적으로 실행할 때.
- CMDEV 큐 관리자 속성이 NODISPLAY로 지정되고 DISPLAY 명령(MQSC) 및 Inquire 명령(PCF)을 제외한 모든 명령이 성공적으로 실행할 때.
- MQSC 명령 ALTER QMGR 또는 PCF 명령인 Change Queue Manager를 실행하고 CMDEV 큐 관리자 속성이 다음 조건 중 하나를 만족할 때:
 - 변경 후 CMDEV가 DISABLED로 지정됨
 - 변경 전 CMDEV가 DISABLED로 지정되지 않았음

명령이 명령 이벤트 큐 SYSTEM.ADMIN.COMMAND.EVENT에 대해 실행하는 경우, 큐가 아직 존재하고 put 금지되지 않은 경우 명령 이벤트가 생성됩니다.

명령 이벤트가 생성되지 않을 때

명령 이벤트 메시지는 다음 상황에서 생성되지 않습니다.

- 명령이 실패할 때
- 큐 관리자가 이벤트 큐에 명령 이벤트를 넣으려는 오류를 만날 때. 이 경우 상관없이 명령이 실행하지만 이벤트 메시지가 생성되지 않습니다.
- MQSC 명령 REFRESH QMGR TYPE (EARLY)의 경우
- MQSC 명령 START QMGR MQSC의 경우
- 매개변수 LOG가 지정되는 경우 MQSC 명령 SUSPEND QMGR의 경우
- 매개변수 LOG가 지정되는 경우 MQSC 명령 RESUME QMGR의 경우

관련 개념

44 페이지의 『명령 이벤트』

명령 이벤트는 MQSC 또는 PCF 명령이 성공적으로 실행했다는 알림입니다.

관련 참조

[REFRESH QMGR](#)

[START QMGR](#)

[SUSPEND QMGR](#)

[RESUME QMGR](#)

[SUSPEND QMGR, RESUME QMGR 및 클러스터](#)

명령 이벤트 사용법

명령 이벤트를 사용하여 실행한 명령의 감사 추적을 생성할 수 있는 방법을 보려면 이 페이지를 사용하십시오.

예를 들어, 오브젝트가 예기치 않게 변경되는 경우 변경을 작성한 사용자와 작성된 시간에 관한 정보가 저장될 수 있습니다. 이것은 특히 구성 이벤트도 사용 가능할 때 유용할 수 있습니다. MQSC 또는 PCF 명령이 명령 이벤트 및 구성 이벤트가 생성되도록 하는 경우, 이벤트 메시지가 둘 다 메시지 디스크립터에서 동일한 상관 ID를 공유합니다.

명령 이벤트 메시지가 생성되지만 명령 이벤트 큐에 넣을 수 없는 경우, 예를 들어 명령 이벤트 큐가 정의되지 않은 경우 명령 이벤트가 생성된 명령이 상관없이 아직 실행합니다.

CMDSCOPE의 효과

CMDSCOPE가 사용되는 명령의 경우 명령 이벤트 메시지 또는 메시지들이 명령이 입력되는 곳이 아니라 명령이 실행하는 큐 관리자(들)에서 생성됩니다. 그러나 CMDSCOPE를 사용하는 명령이 소스 큐 관리자에 의해 생성된 명령인 경우에도 이벤트 데이터의 모든 원본 및 컨텍스트 정보가 입력된 원래 명령과 관련됩니다.

관련 개념

44 페이지의 『명령 이벤트』

명령 이벤트는 MQSC 또는 PCF 명령이 성공적으로 실행했다는 알림입니다.

44 페이지의 『명령 이벤트 생성』



명령 이벤트가 생성되게 하는 상황을 보고 명령 이벤트가 생성되지 않는 상황을 이해하려면 이 페이지를 사용하십시오.

관련 참조

[MQSC 명령](#)

[그룹 내의 PCF 명령 및 응답](#)

Multi 로거 이벤트

로거 이벤트는 큐 관리자가 새 로그 익스텐트  또는 IBM i에서 저널 수신자에 쓰기 시작했다는 알림입니다.  로거 이벤트 메시지는 IBM MQ for z/OS에서 사용할 수 없습니다.

이벤트 데이터는 다음 정보를 포함합니다.

- 현재 로그 익스텐트의 이름.
- 재시작 복구를 위해 필요한 가장 빠른 로그 익스텐트의 이름.
- 매체 복원을 위해 필요한 가장 이른 로그 범위의 이름.
- 로그 익스텐트가 위치하는 디렉토리.
- 아카이브 알림이 필요한 가장 이른 로그 익스텐트의 이름.

로그 프로그램 이벤트는 아카이브 로그 관리를 위해 생성됩니다. 즉, **ARCHLOG** 가 변경되고 **ARCHLOG** 값이 로그 프로그램 이벤트 메시지에 포함될 때 생성됩니다.

이 모든 매개변수에 대한 자세한 정보는 [DISPLAY QMSTATUS](#)를 참조하십시오.

CURRLOG 및 **ARCHLOG** 값을 사용하여 아카이브해야 하는 항목을 판별할 수 있습니다. **CURRLOG**가 변경될 때마다 **CURRLOG** 미만의 숫자를 가진 파일이 아카이브를 위해 전송될 수 있으며 익스텐트에 대해 아카이브가 완료되고 나면 [SET LOG](#)를 호출하여 큐 관리자에게 알려야 합니다.

ARCHLOG는 아카이브가 필요한 가장 오래된 범위입니다. 해당 익스텐트를 아카이브하고 [SET LOG](#)를 호출하여 큐 관리자가 아카이브되었음을 통지하면 큐 관리자는 **ARCHLOG**를 순차적으로 다음 익스텐트로 이동합니다. 큐 관리자는 **ARCHLOG**보다 오래된 익스텐트를 삭제하거나 재사용하도록 스케줄링합니다.

이 범위에 대해 **SET LOG** 가 호출되면 **ARCHLOG** 가 변경되므로 새 이벤트가 발행되고 새 이벤트 이후에 아카이브해야 하는 사항을 알아야 합니다.

어떤 이유로 아카이브 프로세스가 실패하고 알림이 크게 빌드되는 경우 관리자가 수동으로 [RESET QMGR TYPE \(ARCHLOG\)](#) 명령을 실행할 수 있습니다. 이는 사용자가 지정한 범위를 포함하여 이전의 모든 범위를 재사용하거나 삭제할 수 있음을 큐 관리자에게 알립니다.

생성되는 모든 로거 이벤트 메시지는 로거 이벤트 큐 SYSTEM.ADMIN.LOGGER.EVENT에 위치합니다.

관련 참조

로거

11 페이지의 『이벤트 유형』

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

Multi

로거 이벤트 생성

로거 이벤트가 생성되지 않는 상황을 보고 로거 이벤트가 생성되지 않는 상황을 이해하려면 이 페이지를 사용하십시오.

다음 상황에서 로거 이벤트 메시지가 생성됩니다.

- LOGGEREV 큐 관리자 속성이 ENABLED로 지정되고 큐 관리자가 새 로그 익스텐트 또는 IBM i에서는 저널 수신자에 쓰기 시작할 때.
- LOGGEREV 큐 관리자 속성이 ENABLED로 지정되고 큐 관리자가 시작할 때.
- LOGGEREV 큐 관리자 속성이 DISABLED에서 ENABLED로 변경될 때.
- LOGGEREV 큐 관리자 속성에 해당 큐 관리자가 아카이브 알림(ARCHLOG)을 대기 중인 가장 오래된 로그 익스텐트의 이름이 아카이브되었다는 알림이 전송될 때

팁: [RESET QMGR MQSC](#) 명령을 사용하여 큐 관리자가 새 로그 익스텐트에 쓰기 시작하도록 요청할 수 있습니다.

로거 이벤트가 생성되지 않을 때

로거 이벤트 메시지는 다음 상황에서 생성되지 않습니다.



- 큐 관리자가 순환 로깅을 사용하도록 구성될 때.

이 경우 LOGGEREV 큐 관리자 속성이 DISABLED로 설정되며 변경할 수 없습니다.

- 큐 관리자가 이벤트 큐에 로거 이벤트를 넣으려는 오류를 만날 때. 이 경우 이벤트를 유발한 조치가 완료되지만 이벤트 메시지가 생성되지 않습니다.

관련 개념

45 페이지의 『로거 이벤트』

로거 이벤트는 큐 관리자가 새 로그 익스텐트  또는 IBM i에서 저널 수신자에 쓰기 시작했다는 알림입니다.  로거 이벤트 메시지는 IBM MQ for z/OS에서 사용할 수 없습니다.

관련 참조

[LoggerEvent\(MQLONG\)](#)

[LoggerEvent\(10자리 부호화 정수\)](#)

[RESET QMGR](#)

로거 이벤트 사용법



로거 이벤트를 사용하여 큐 관리자 재시작 또는 매체 복원에 더 이상 필요하지 않은 로그 익스텐트를 판별할 수 있는 방법을 보려면 이 페이지를 사용하십시오.

활성 로그 디렉토리로부터 제거하기 전에 재해 복구를 위해 테이프 같은 매체로 여분의 로그 익스텐트를 아카이브할 수 있습니다. 여분의 로그 익스텐트의 정기적인 제거는 디스크 공간 사용량을 최소로 유지합니다.

예를 들어 이벤트 큐가 정의되지 않았기 때문에 LOGGEREV 큐 관리자 속성이 사용 가능하지만 로거 이벤트 메시지를 로거 이벤트 큐에 넣을 수 없는 경우, 해당 이벤트를 유발한 조치는 상관없이 계속됩니다.

관련 개념

45 페이지의 『로거 이벤트』

로거 이벤트는 큐 관리자가 새 로그 익스텐트  또는 IBM i에서 저널 수신자에 쓰기 시작했다는 알림입니다.  로거 이벤트 메시지는 IBM MQ for z/OS에서 사용할 수 없습니다.

관련 참조

[LoggerEvent\(MQLONG\)](#)

[LoggerEvent\(10자리 부호화 정수\)](#)

46 페이지의 『로거 이벤트 생성』

로거 이벤트가 생성되게 하는 상황을 보고 로거 이벤트가 생성되지 않는 상황을 이해하려면 이 페이지를 사용하십시오.

로거 이벤트 큐를 모니터링하는 샘플 C 프로그램

새 이벤트 메시지에 대한 로거 이벤트 큐를 모니터링하고 해당 메시지를 읽고 메시지의 콘텐츠를 stdout에 넣는 샘플 C 프로그램을 보려면 이 페이지를 사용하십시오.

```

/*****
/*
/* Program name: AMQSLOG0.C
/*
/* Description: Sample C program to monitor the logger event and
/* display formatted message content to stdout when a logger
/* event occurs
/* <copyright
/* notice="lm-source-program"
/* pids="5724-H72,"
/* years="2005, 2024"
/* crc="186943832" >
/* Licensed Materials - Property of IBM
/*
/* 5724-H72,
/*
/* (C) Copyright IBM Corp. 2005, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* </copyright>
/*****
/*
/* Function: AMQSLOG is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and displays the
/* formatted contents of the message to stdout.
/*
/*****

```

```

/*                                                                    */
/* AMQSLOG has 1 parameter - the queue manager name (optional, if not  */
/* specified then the default queue manager is implied)                */
/*                                                                    */
/*****
/*****
/***** Includes *****/
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include <cmqc.h>          /* MQI constants*/
#include <cmqcf.h>        /* PCF constants*/

/*****
/***** Constants *****/
/*****

#define MAX_MESSAGE_LENGTH 8000

typedef struct _ParmTableEntry
{
    MQLONG  ConstVal;
    PMQCHAR Desc;
} ParmTableEntry;

ParmTableEntry ParmTable[] =
{
    {0, "", "Queue Manager Name"},
    {MQCA_Q_MGR_NAME, "Queue Manager Name"},
    {MQCMD_LOGGER_EVENT, "Logger Event Command"},
    {MQRC_LOGGER_STATUS, "Logger Status"},
    {MQCACF_ARCHIVE_LOG_EXTENT_NAME, "Archive Log Extent"},
    {MQCACF_CURRENT_LOG_EXTENT_NAME, "Current Log Extent"},
    {MQCACF_RESTART_LOG_EXTENT_NAME, "Restart Log Extent"},
    {MQCACF_MEDIA_LOG_EXTENT_NAME, "Media Log Extent"},
    {MQCACF_LOG_PATH, "Log Path"}
};

#if defined(MQ_64_BIT)
    #define Int32
#else
    #define Int32 "l"
#endif

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

static void ProcessPCF(MQHCONN hConn,
                      MQHOBJ hEventQueue,
                      PMQCHAR pBuffer);

static PMQCHAR ParmToString(MQLONG Parameter);

/*****
/***** Function: main *****/
/*****

int main(int argc, char * argv[])
{
    MQLONG  CompCode;
    MQLONG  Reason;
    MQHCONN hConn = MQHC_UNUSABLE_HCONN;
    MQOD    ObjDesc = { MQOD_DEFAULT };
    MQCHAR  QMName[MQ_Q_MGR_NAME_LENGTH+1] = "";
    MQCHAR  LogEvQ[MQ_Q_NAME_LENGTH] = "SYSTEM.ADMIN.LOGGER.EVENT";
    MQHOBJ  hEventQueue = MQHO_UNUSABLE_HOBJ;
    PMQCHAR pBuffer = NULL;

    printf("\n/*****\n");
    printf("/* Sample Logger Event Monitor start */\n");
    printf("/*****\n");

    /***** Parse any command line options *****/
    if (argc > 1)
    {
        strncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);

```

```

}

pBuffer = (PMQCHAR)malloc(MAX_MESSAGE_LENGTH);
if (pBuffer == NULL)
{
    printf("Can't allocate %d bytes\n", MAX_MESSAGE_LENGTH);
    goto MOD_EXIT;
}

/*****
/* Connect to the specified (or default) queue manager          */
*****/
MQCONN( QMName,
        &hConn,
        &CompCode,
        &Reason);

if (Reason != MQRC_NONE)
{
    printf("MQCONN ended with reason code %" Int32 "d\n", Reason);
    goto MOD_EXIT;
}

/*****
/* Open the logger event queue for input                        */
*****/
strncpy(ObjDesc.ObjectQMgrName, QMName, MQ_Q_MGR_NAME_LENGTH);
strncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

MQOPEN( hConn,
        &ObjDesc,
        MQOO_INPUT_EXCLUSIVE,
        &hEventQueue,
        &CompCode,
        &Reason );

if (Reason != MQRC_NONE)
{
    printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %" Int32 "d\n",
          ObjDesc.ObjectQMgrName,
          ObjDesc.ObjectName,
          Reason);
    goto MOD_EXIT;
}
else
{
    /*****
    /* Start processing event messages                          */
    *****/
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:
if (pBuffer != NULL)
{
    free(pBuffer);
}

/*****
/* Close the logger event queue                                */
*****/
if (hEventQueue != MQHO_UNUSABLE_HOBJ)
{
    MQCLOSE(hConn, &hEventQueue, MQCO_NONE, &CompCode, &Reason);
}

/*****
/* Disconnect                                                  */
*****/
if (hConn != MQHC_UNUSABLE_HCONN)
{
    MQDISC(&hConn, &CompCode, &Reason);
}

return 0;
}

/*****
/* Function: ProcessPCF                                        */
*****/
/*

```

```

/* Input Parameters: Handle to queue manager connection */
/* Handle to the opened logger event queue object */
/* Pointer to a memory buffer to store the incoming PCF */
/* message */
/* */
/* Output Parameters: None */
/* */
/* Logic: Wait for messages to appear on the logger event queue and display */
/* their formatted contents. */
/* */
/* */
/*****/

static void ProcessPCF(MQHCONN hConn,
                     MQHOBJ hEventQueue,
                     PMQCHAR pBuffer)
{
    MQCFH * pCfh;
    MQCFST * pCfst;
    MQGMO Gmo = { MQGMO_DEFAULT };
    MQMD Mqmd = { MQMD_DEFAULT };
    PMQCHAR pPCFCmd;
    MQLONG CompCode = MQCC_OK;
    MQLONG Reason = MQRC_NONE;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    Gmo.Options = MQGMO_WAIT +
                 MQGMO_CONVERT +
                 MQGMO_FAIL_IF QUIESCING;
    Gmo.WaitInterval = MQWI_UNLIMITED; /* Set timeout value */

    /*****/
    /* Process response Queue */
    /*****/
    while (Reason == MQRC_NONE)
    {
        memcpy(&Mqmd.MsgId, MQMI_NONE, sizeof(Mqmd.MsgId));
        memset(&Mqmd.CorrelId, 0, sizeof(Mqmd.CorrelId));

        MQGET( hConn,
              hEventQueue,
              &Mqmd,
              &Gmo,
              MAX_MESSAGE_LENGTH,
              pBuffer,
              &MsgLen,
              &CompCode,
              &Reason );

        if (Reason != MQRC_NONE)
        {
            switch(Reason)
            {
                case MQRC_NO_MSG_AVAILABLE:
                    printf("Timed out");
                    break;

                default:
                    printf("MQGET ended with reason code %" Int32 "d\n", Reason);
                    break;
            }
            goto MOD_EXIT;
        }

        /*****/
        /* Only expect PCF event messages on this queue */
        /*****/
        if (memcmp(Mqmd.Format, MQFMT_EVENT, MQ_FORMAT_LENGTH))
        {
            printf("Unexpected message format '%8.8s' received\n", Mqmd.Format);
            continue;
        }

        /*****/
        /* Build the output by parsing the received PCF message, first the */
        /* header, then each of the parameters */
        /*****/
        pCfh = (MQCFH *)pBuffer;

        if (pCfh->Reason != MQRC_NONE)
        {
            printf("-----\n");

```

```

printf("Event Message Received\n");

Parm = ParmToString(pCfh->Command);
if (Parm != NULL)
{
    printf("Command   :%s \n",Parm);
}
else
{
    printf("Command   :%" Int32 "d \n",pCfh->Command);
}

printf("CompCode  :%" Int32 "d\n"    ,pCfh->CompCode);

Parm = ParmToString(pCfh->Reason);
if (Parm != NULL)
{
    printf("Reason    :%s \n",Parm);
}
else
{
    printf("Reason    :%" Int32 "d \n",pCfh->Reason);
}
}

pPCFCmd = (PMQCHAR) (pCfh+1);
printf("-----\n");
while(pCfh->ParameterCount-->0)
{
    pCfst = (MQCFST *) pPCFCmd;
    switch(pCfst->Type)
    {
        case MQCFT_STRING:
            Parm = ParmToString(pCfst->Parameter);
            if (Parm != NULL)
            {
                printf("%-32s",Parm);
            }
            else
            {
                printf("%-32" Int32 "d",pCfst->Parameter);
            }

            fwrite(pCfst->String, pCfst->StringLength, 1, stdout);
            pPCFCmd += pCfst->StrucLength;
            break;

        default:
            printf("Unrecognised datatype %" Int32 "d returned\n", pCfst->Type);
            goto MOD_EXIT;
    }
    putchar('\n');
}
}

```

샘플 출력

이 애플리케이션은 다음 양식의 출력을 생성합니다.

```

/*****
/* Sample Logger Event Monitor start */
*****/
-----
Event Message Received
Command :Logger Event Command
CompCode :0
Reason  :Logger Status
-----
Queue Manager Name      CSIM
Current Log Extent      AMQA000001
Restart Log Extent      AMQA000001
Media Log Extent        AMQA000001
Log Path                 QMCSIM
-----

```

관련 개념

47 페이지의 『로거 이벤트 사용법』

로거 이벤트를 사용하여 큐 관리자 재시작 또는 매체 복원에 더 이상 필요하지 않은 로그 익스텐트를 판별할 수 있는 방법을 보려면 이 페이지를 사용하십시오.

[45 페이지의 『명령 이벤트 사용법』](#)

명령 이벤트를 사용하여 실행한 명령의 감사 추적을 생성할 수 있는 방법을 보려면 이 페이지를 사용하십시오.

관련 참조

[46 페이지의 『로거 이벤트 생성』](#)

로거 이벤트가 생성되지 않는 상황을 보고 로거 이벤트가 생성되지 않는 상황을 이해하려면 이 페이지를 사용하십시오.

권한 구성 이벤트

권한 구성 이벤트는 명령행, MQSC, PCF 또는 해당 IBM i 명령을 통해 보안 제어 조작이 변경될 때 출력됩니다.

이벤트 데이터는 다음 정보를 포함합니다.

원본 정보

변경이 수행된 큐 관리자, 변경을 수행한 사용자의 ID, 변경을 수행한 방식(예를 들어, 콘솔 명령으로) 등을 포함합니다.

컨텍스트 정보

명령 메시지의 메시지 데이터의 컨텍스트 정보의 복제본입니다.

컨텍스트 정보는 SYSTEM.ADMIN.COMMAND.QUEUE 큐에 명령이 메시지로 입력될 때 이벤트 데이터에 포함됩니다.

권한 레코드 ID

프로파일 이름과 권한 레코드의 오브젝트 유형을 포함합니다.

오브젝트 속성

권한 레코드에 있는 모든 속성의 값을 포함합니다.

권한 레코드 변경 이벤트의 경우 변경 이전의 정보를 포함한 메시지 하나와 변경 후의 정보를 포함한 다른 메시지 즉, 두 개의 메시지가 생성됩니다.

생성되는 모든 이벤트 메시지는 SYSTEM.ADMIN.CONFIG.EVENT 큐에 위치합니다.

관련 참조

[11 페이지의 『이벤트 유형』](#)

큐 관리자 또는 채널 인스턴스가 보고할 수 있는 도구 이벤트의 유형을 보려면 이 페이지를 사용하십시오.

권한 구성 이벤트 생성

이 페이지를 사용하여 권한 구성 이벤트를 생성시키는 상황을 보고 권한 구성 이벤트가 생성되지 않는 상황을 이해하십시오.

권한 구성 이벤트는 권한 레코드 속성의 변경에 대해 알립니다. 세 가지 유형의 권한 구성 이벤트가 있습니다.

- [권한 레코드 변경](#)
- [권한 레코드 삭제](#)
- [권한 레코드 새로 고치기](#)

CONFIGEV 큐 관리자 속성이 *ENABLED*로 설정되고 다음 명령이나 해당 MQSC 명령이 발행되면 권한 레코드가 실제로 변경되지 않아도 권한 이벤트 메시지를 구성 이벤트 큐에 넣습니다.

- [권한 레코드 삭제](#) PCF 명령
- [권한 레코드 설정](#) PCF 명령
- [setmqaut](#) 제어 명령
- [RVKMQMAUT](#) CL 명령
- [GRMQMAUT](#) CL 명령

권한 구성 이벤트가 생성되지 않을 때

다음 상황에서는 권한 구성 이벤트 메시지가 생성되지 않습니다.

- 명령이 실패할 때
- 큐 관리자가 이벤트 큐에 메시지를 넣으려 시도하는 중에 오류가 발생할 때, 이 경우 명령이 완료되지만 이벤트 메시지는 생성되지 않음
- 큐 관리자를 작성 또는 삭제할 때
- 삭제 명령의 **AUTHREC** 옵션과 무관하게 오브젝트가 삭제될 때. 해당 명령 이벤트는 개별 사용자에게 대한 권한 레코드에 적용되지 않는 이 조작을 보여줍니다.

관련 개념

44 페이지의 『명령 이벤트』

명령 이벤트는 MQSC 또는 PCF 명령이 성공적으로 실행했다는 알림입니다.

관련 참조

[REFRESH QMGR](#)

Multi 멀티플랫폼에서 도구 이벤트를 모니터링하는 샘플 프로그램

amqsevt는 큐 관리자가 작성할 수 있는 도구 이벤트를 형식화하며, IBM MQ for Multiplatforms와 함께 제공됩니다. 프로그램은 이벤트 큐에서 메시지를 읽고 읽을 수 있는 문자열로 형식화합니다.

샘플 프로그램으로서 소스와 2진이 둘 다 제공됩니다. 샘플은 IBM i를 포함한 모든 멀티플랫폼에서 제공됩니다.

단일 2진 파일 **amqsevt**(또는 **amqsevt.exe**)가 샘플 파일 세트에서 제공되고 샘플 **bin(tools\c\samples\bin 또는 bin64)** 디렉토리에 설치됩니다.

소스 파일 **amqsevta.c**도 샘플 파일 세트에 제공되며 샘플 디렉토리(즉, Windows의 경우 **tools\c\samples**)에 설치됩니다.

프로그램은 **MQCB**를 사용하여 메시지를 검색하여 다중 이벤트 큐에서 읽고 다중 토픽에 구독할 수 있음을 참고하십시오.

클라이언트로서 실행할 때, 샘플은 z/OS를 포함한 모든 큐 관리자에 연결할 수 있습니다.

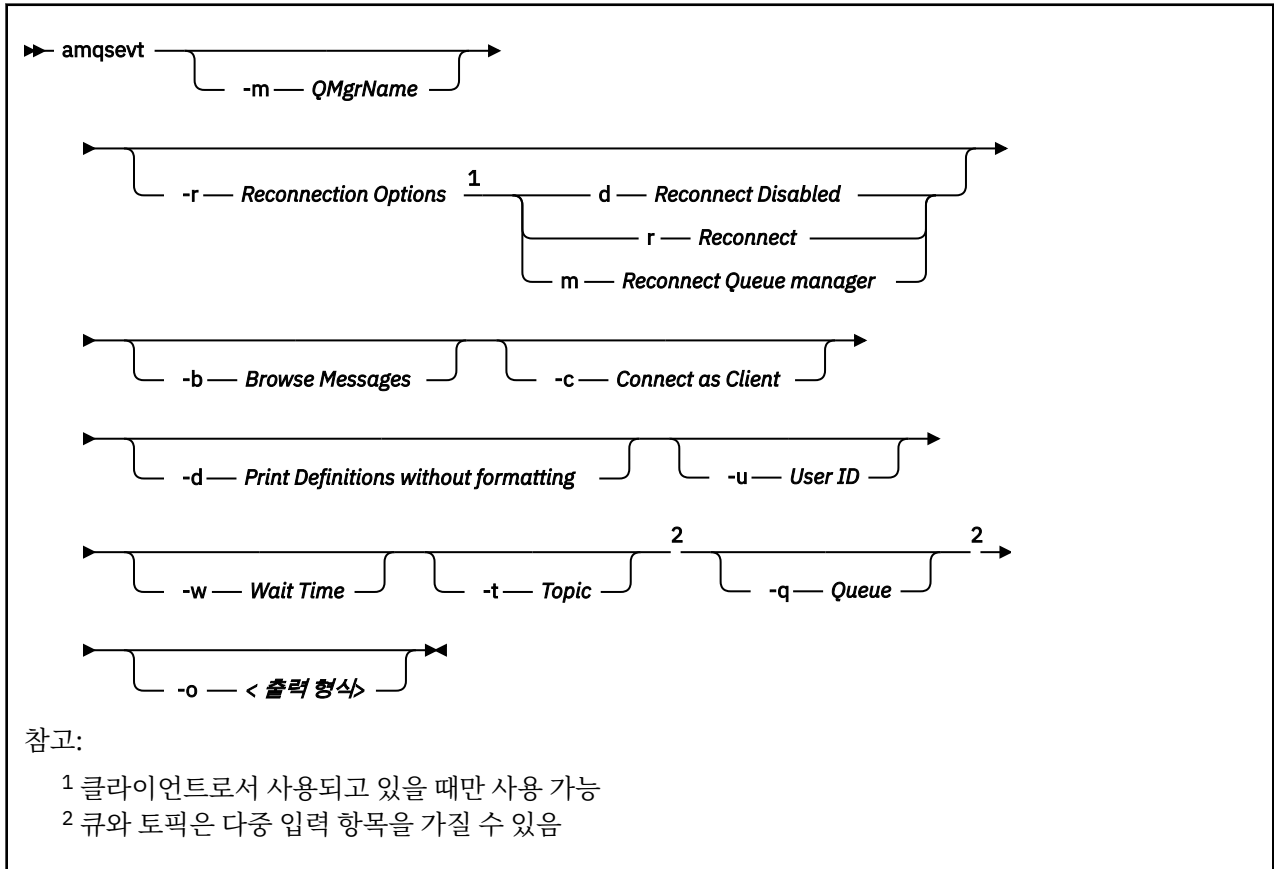


주의: 어떤 매개변수도 지정하지 않고 프로그램을 사용할 수 있지만, 그 경우 프로그램은 기본 큐 관리자에 연결하고 이벤트 큐의 표준 세트(**SYSTEM.ADMIN.*.EVENT**)에서 메시지를 읽으려고 시도합니다.

이 상황에서 프로그램은 사용자가 Enter 키를 눌러서 프로그램을 종료할 때까지 영원히 메시지를 대기합니다.

그러나, 설명되는 다양한 옵션과 함께 프로그램을 사용할 가능성이 더 큽니다.

구문



선택적 매개변수

-m *QueueManagerName*

이벤트 읽기를 위한 특정 큐 관리자를 지정하십시오.

-r *Reconnection Options*

클라이언트로서 사용될 때 자동 재연결 옵션입니다. 가능한 값은 다음과 같습니다.

d 클라이언트 다시 연결 사용 불가능

r 클라이언트 다시 연결

m 큐 관리자 다시 연결

-b 메시지를 파괴적으로 읽지 않고 레코드만 찾아보기

-c 연결을 클라이언트로서 선택합니다.

-d 두 번째 예에서 사용되는 인쇄 모드를 선택합니다. MQI 상수는 헤더 파일에 나타나는 그대로 인쇄됩니다.

-u *User ID*

특정 사용자를 지정하고 비밀번호를 요청하는 프롬프트가 나타나게 합니다.

-w *Wait*

지정된 시간(초) 안에 이벤트 메시지가 도달하지 않은 경우 프로그램이 종료하게 합니다.

시간을 지정하지 않는 경우 프로그램은 사용자가 Enter 키를 누를 때만 정상적으로 종료함을 참고하십시오.

-t Topic 및

-q Queue

-q 및 -t 옵션은 둘 다 명령행에서 여러 번 제공할 수 있습니다.

그러므로, 일부 표준 큐로부터 및 토픽에서 (이벤트가 토픽으로 송신되고 있는 경우) 프로그램의 단일 실행으로부터 읽을 수 있습니다.

큐나 토픽이 명령행에서 이름 지정되지 않는 경우 기본 이벤트 큐가 열립니다.

참고: 프로그램은 z/OS 큐 관리자에 클라이언트로 연결되어 있는지 감지하고 z/OS 에 SYSTEM.ADMIN.LOGGER.EVENT 큐.

토픽이 사용될 때 프로그램은 종료할 때 모든 것이 정리되도록 관리되는 큐와 함께 지속 가능하지 않은 구독을 사용합니다.

-o <output format>

출력의 형식. 가능한 값은 다음과 같습니다.

text

표준 텍스트 형식; 이는 기본값입니다.

json

표준 JSON 형식; JSON을 이해하는 모든 애플리케이션은 이 출력을 가져 와서 직접 처리 할 수 있습니다.

샘플 출력

다음 세 가지 예는 프로그램에서 출력을 표시합니다.

첫 번째 예는 프로그램이 필드의 MQI 정의를 사용하고 출력을 더 읽기 쉽게 만들기 위해 출력을 형식화하는 기본 형식화 옵션을 사용합니다.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : Queue Mgr Event
Reason               : Unknown Alias Base Queue
Event created        : 2015/06/17 13:47:07.02 GMT
Queue Mgr Name       : V8003_A
Queue Name           : EVT.NO.BASE.QUEUE
Base Object Name     : EVT.NOT.DEFINED
Appl Type            : Unix
Appl Name            : amqsput
Base Type            : Queue
```

두 번째 예는 MQI 상수를 변환하려고 시도하지 않는 -d 옵션을 사용하여 대체 형식화를 보여줍니다. 이것은 특정 MQI 값을 찾는 일부 스크립트 도구에 바람직할 수 있습니다.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : MQCMD_Q_MGR_EVENT
Reason               : MQRC_UNKNOWN_ALIAS_BASE_Q
Event created        : 2015/06/17 13:52:48.18 GMT
MQCA_Q_MGR_NAME      : V8003_A
MQCA_Q_NAME          : EVT.NO.BASE.QUEUE
MQCA_BASE_OBJECT_NAME : EVT.NOT.DEFINED
MQIA_APPL_TYPE       : MQAT_UNIX
MQCACF_APPL_NAME     : amqsput
MQIA_BASE_TYPE       : MQOT_Q
```

세 번째 예는 명령에서 JSON 출력을 표시합니다.

```
amqsevt -m QM1 -q SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE -o json
```

```
{
  "eventSource" : { "objectName": "SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE",
                   "objectType" : "Queue" },
  "eventType" : {
    "name" : "Activity Trace",
    "value" : 209
  },
  "eventReason" : {
    "name" : "None",
    "value" : 0
  },
  "eventCreation" : {
    "timeStamp" : "2018-07-10T12:44:26Z",
    "epoch" : 1531226666
  },
  "eventData" : {
    "queueMgrName" : "QM1",
    "hostName" : "<yourhostname>",
    "startDate" : "2018-07-10",
    "startTime" : "13:44:25",
    "endDate" : "2018-07-10",
    "endTime" : "13:44:26",
    "commandLevel" : 910,
    ...
  }
}
```

사용법 예

다음 예는 둘 이상의 큐를 사용하는 방법을 보여줍니다.

```
amqsevt -m QM1 -q SYSTEM.ADMIN.QMGR.EVENT -q SYSTEM.ADMIN.PERM.EVENT -w 1
```

관련 개념

[8 페이지의 『이벤트 모니터링』](#)

이벤트 모니터링은 큐 관리자 네트워크에서 도구 이벤트 발생을 감지하는 프로세스입니다. 도구 이벤트는 큐 관리자나 채널 인스턴스에서 감지되는 이벤트의 논리적 조합입니다. 이러한 이벤트 때문에 큐 관리자나 채널 인스턴스가 이벤트 메시지라는 특정 메시지를 이벤트 큐에 넣게 됩니다.

[9 페이지의 『도구 이벤트』](#)

도구 이벤트는 큐 관리자나 채널 인스턴스가 이벤트 큐에서 이벤트 메시지라는 특수 메시지를 감지하여 넣는 조건의 논리 결합입니다.

관련 참조

[C 프로그래밍](#)

[47 페이지의 『로거 이벤트 큐를 모니터링하는 샘플 C 프로그램』](#)

새 이벤트 메시지에 대한 로거 이벤트 큐를 모니터링하고 해당 메시지를 읽고 메시지의 콘텐츠를 stdout에 넣는 샘플 C 프로그램을 보려면 이 페이지를 사용하십시오.

메시지 모니터링

메시지 모니터링은 큐 관리자 네트워크를 통해 메시지를 가져오는 라우트를 식별하는 프로세스입니다. 활동 유형 및 메시지를 위해 수행된 활동의 순서를 식별하여, 메시지 라우트를 판별할 수 있습니다.

메시지는 큐 관리자 네트워크를 통해 전달되면 다양한 프로세스가 메시지를 위해 활동을 수행합니다. 다음 기술 중 하나를 사용하여 메시지 라우트를 판별하십시오.

- IBM MQ 표시 라우트 애플리케이션(dspmqrte)
- 활동 기록
- 라우트 추적 메시징

이 기술 모두는 큐 관리자 네트워크를 통해 전달될 때 메시지에 수행된 활동에 관한 정보를 포함한 특수 메시지를 생성합니다. 이 특수 메시지에서 리턴된 정보를 사용하여 다음 목표를 이룹니다.

- 메시지 활동을 기록합니다.
- 마지막으로 알려진 메시지 위치를 판별합니다.
- 큐 관리자 네트워크에서 라우팅 문제를 감지합니다.

- 큐 관리자 네트워크에서 라우팅 문제의 원인 판별을 돕습니다.
- 큐 관리자 네트워크가 올바르게 실행 중인지 확인합니다.
- 큐 관리자 네트워크의 실행에 익숙해집니다.
- 발행된 메시지를 추적합니다.

관련 개념

메시지의 유형

활동 및 조작

활동은 애플리케이션이 메시지를 위해 수행하는 별개의 조치입니다. 활동은 조작으로 구성되는데, 조작은 애플리케이션이 수행하는 작업의 단일 부분입니다.

다음 조치가 활동의 예입니다.

- 메시지 채널 에이전트(MCA)는 전송 큐에서 채널을 따라 메시지를 송신합니다.
- MCA가 채널에서 메시지를 받아서 대상 큐에 넣습니다.
- 애플리케이션이 큐에서 메시지를 가져와서 응답의 응답 메시지에 놓습니다.
- IBM MQ 발행/구독 엔진이 메시지를 처리합니다.

활동은 하나 이상의 조작으로 구성됩니다. 조작은 애플리케이션이 수행하는 작업의 단일 조각입니다. 예를 들어, 전송 큐에서 채널을 따라 메시지를 송신하는 MCA의 활동은 다음 조작으로 구성됩니다.

1. 전송 큐에서 메시지 가져오기(*Get* 조작).
2. 채널을 따라 메시지 송신(송신 조작).

발행/구독 네트워크에서, 메시지를 처리하는 IBM MQ 발행/구독 엔진의 활동은 다음의 다중 조작으로 구성될 수 있습니다.

1. 메시지를 토픽 문자열에 넣기(*Put* 조작).
2. 메시지의 수령으로 간주되는 각 구독자에 대한 하나 이상의 조작(발행 조작, 제거된 발행 조작 또는 제외된 발행 조작).

활동의 정보

메시지가 큐 관리자 네트워크를 통해 라우트되므로 정보를 기록하여 메시지에 수행되는 활동의 순서를 식별할 수 있습니다. 메시지에 대해 수행되는 활동의 순서로부터 큐 관리자 네트워크를 통해 메시지의 라우트를 판별할 수 있으며, 다음 정보를 얻을 수 있습니다.

메시지의 마지막으로 알려진 위치

메시지가 그의 의도된 목적지에 도달하지 않는 경우, 전체 또는 부분적 메시지 라우트로부터 메시지의 마지막 알려진 위치를 판별할 수 있습니다.

큐 관리자 네트워크의 구성 문제

큐 관리자 네트워크를 통한 메시지의 라우트를 연구할 때, 메시지가 예상한 곳으로 이동하지 않았음을 확인할 수 있습니다. 이것이 발생할 수 있는 많은 이유가 있는데, 예를 들어 채널이 비활성인 경우 메시지는 대체 라우트를 택할 수 있습니다.

발행/구독 애플리케이션의 경우, 구독자에게 발행되는 결과로 큐 관리자 네트워크에서 이동하는 토픽 및 모든 메시지에 발행되는 메시지의 라우트를 판별할 수도 있습니다.

그런 상황에서 시스템 관리자는 큐 관리자 네트워크에 문제가 있는지 여부를 판별할 수 있으며, 문제가 있는 경우 이를 정정할 수 있습니다.

메시지 라우트

메시지 라우트를 판별하는 이유에 따라서, 다음의 일반 접근법을 사용할 수 있습니다.

라우트 추적 메시지를 위해 기록된 활동 정보 사용

라우트 추적 메시지는 특정 목적을 위한 활동 정보를 기록합니다. 이들을 사용하여 큐 관리자 네트워크의 구성 문제를 판별하거나 메시지의 마지막으로 알려진 위치를 판별할 수 있습니다. 의도된 목적지에 도달하지

않은 메시지의 마지막으로 알려진 위치를 판별하기 위해 라우트 추적 메시지가 생성되는 경우 원래 메시지를 흉내낼 수 있습니다. 이것은 라우트 추적 메시지에 원래 메시지가 취한 라우트를 따르는 가장 큰 기회를 제공합니다.

IBM MQ 표시 라우트 애플리케이션은 라우트 추적 메시지를 생성할 수 있습니다.

원래 메시지를 위해 기록된 활동 정보 사용

활동 기록을 위해 모든 메시지를 사용 가능하게 하고 그 대신 기록되는 활동 정보를 가질 수 있습니다. 메시지가 의도된 목적지에 도달하지 않는 경우, 기록된 활동 정보를 사용하여 메시지의 마지막으로 알려진 위치를 판별할 수 있습니다. 원래 메시지의 활동 정보를 사용하면 마지막으로 알려진 위치로 이끄는 가장 정확한 가능한 메시지 라우트를 판별할 수 있습니다. 이 접근법을 사용하려면 원래 메시지가 활동 기록에 대해 사용 가능해야 합니다.

경고: 활동 기록을 위해 큐 관리자 네트워크의 모든 메시지를 사용 가능하게 하는 것은 피하십시오. 활동 기록을 위해 사용 가능한 메시지는 그 대신 생성되는 많은 활동 보고서를 가질 수 있습니다. 큐 관리자 네트워크의 모든 메시지가 활동 기록에 대해 사용 가능한 경우, 큐 관리자 네트워크 트래픽이 허용할 수 없는 레벨로 증가할 수 있습니다.

관련 개념

56 페이지의 『메시지 모니터링』

메시지 모니터링은 큐 관리자 네트워크를 통해 메시지를 가져오는 라우트를 식별하는 프로세스입니다. 활동 유형 및 메시지를 위해 수행된 활동의 순서를 식별하여, 메시지 라우트를 판별할 수 있습니다.

58 페이지의 『메시지 라우트 기술』

활동 기록 및 라우트 추적 메시징은 메시지가 큐 관리자 네트워크를 따라 라우트될 때 메시지에 대한 활동 정보를 기록할 수 있게 하는 기술입니다.

64 페이지의 『라우트 추적 메시징』

라우트 추적 메시징은 라우트 추적 메시지를 사용하여 메시지에 대한 활동 정보를 기록하는 기술입니다. 라우트 추적 메시징은 큐 관리자 네트워크로 라우트 추적 메시지 송신을 포함합니다.

관련 태스크

[사용자 고유의 메시지 채널 에이전트 작성](#)

메시지 라우트 기술

활동 기록 및 라우트 추적 메시징은 메시지가 큐 관리자 네트워크를 따라 라우트될 때 메시지에 대한 활동 정보를 기록할 수 있게 하는 기술입니다.

활동 기록

메시지가 적절한 보고서 옵션이 지정된 경우, 메시지가 큐 관리자 네트워크를 통해 라우트될 때 애플리케이션이 활동 보고서를 생성할 것을 요청합니다. 애플리케이션이 메시지를 위해 활동을 수행할 때, 활동 보고서가 생성되고 적절한 위치로 전달될 수 있습니다. 활동 보고서에는 메시지에 수행된 활동에 관한 정보가 들어 있습니다.

활동 보고서를 사용하여 수집된 활동 정보는 메시지 라우트가 판별될 수 있기 전에 순서대로 정렬되어야 합니다.

라우트 추적 메시징

라우트 추적 메시징은 큐 관리자 네트워크로 라우트 추적 송신을 포함하는 기술입니다. 애플리케이션이 라우트 추적 메시지를 위해 활동을 수행할 때, 활동 정보가 라우트 추적 메시지의 메시지 데이터에 누적될 수 있거나 활동 보고서가 생성될 수 있습니다. 활동 정보가 라우트 추적 메시지의 메시지 데이터에 누적되는 경우, 대상 큐에 도달할 때 라우트 추적 메시지의 모든 정보를 포함하는 라우트 추적 응답 메시지가 생성되어 적절한 위치로 전달될 수 있습니다.

라우트 추적 메시지가 그 대신에 수행되는 활동의 순서를 기록하는 데 전용되기 때문에 활동 보고서를 요청하는 정상 메시지와 비교하여 사용 가능한 추가 처리 옵션이 있습니다.

활동 기록과 라우트 추적 메시징의 비교

활동 기록과 라우트 추적 메시징은 둘 다 큐 관리자 네트워크를 통해 메시지가 사용한 라우트를 판별하기 위한 활동 정보를 제공할 수 있습니다. 두 메소드는 고유한 장점을 갖고 있습니다.

이점	활동 기록	라우트 추적 메시징
메시지의 마지막 알려진 위치를 판별할 수 있습니까?	예	예
큐 관리자 네트워크의 구성 문제를 판별할 수 있습니까?	예	예
임의의 메시지가 요청할 수 있습니까? (라우트 추적 메시지와 사용으로 제한되지 않음)	예	아니오
메시지 데이터가 수정되지 않은 채로 있음	예	아니오
메시지가 정상적으로 처리됨	예	아니오
활동 정보가 메시지 데이터에 누적될 수 있음	아니오	예
대상 큐로 선택적인 메시지 전달	아니오	예
메시지가 무한 루프에 갇힌 경우 감지하고 다룰 수 있음	아니오	예
활동 정보를 신뢰할 수 있는 순서로 넣을 수 있음	아니오	예
활동 정보를 표시하기 위해 애플리케이션이 제공됨	아니오	예

메시지 라우트 완전성

일부 경우에는 메시지를 위해 수행되는 활동의 전체 순서를 식별하는 것이 불가능하므로, 부분 메시지 라우트만 판별할 수 있습니다. 메시지 라우트의 완전성은 메시지가 라우트되는 큐 관리자 네트워크에 직접적으로 영향을 받습니다.

큐 관리자에 연결된 MCA 및 사용자 작성 애플리케이션은 메시지 대신 수행되는 활동과 관련된 정보를 기록할 수 있습니다. 활동 정보의 기록은 큐 관리자 속성 ACTIVREC 및 ROUTEREC에 의해 제어됩니다. 큐 관리자 네트워크는 완전한 메시지 라우트를 판별할 수 있습니다.

활동 정보가 저장되는 방법

IBM MQ는 활동 보고서, 라우트 추적 메시지 또는 라우트 추적 응답 메시지에 활동 정보를 저장합니다. 각 경우에 정보는 Activity PCF 그룹이라는 구조에 저장됩니다. 라우트 추적 메시지 또는 라우트 추적 응답 메시지는 메시지 에서 수행되는 활동 수에 따라서 많은 Activity PCF 그룹을 포함할 수 있습니다. 활동 보고서는 모든 기록된 활동에 대해 별도의 활동 보고서가 생성되기 때문에 하나의 Activity PCF 그룹을 포함합니다.

라우트 추적 메시징을 사용하면 추가 정보를 기록할 수 있습니다. 이 추가 정보는 TraceRoute PCF 그룹이라는 구조에 저장됩니다. TraceRoute PCF 그룹은 추가 활동 정보를 저장하고 큐 관리자 네트워크를 통해 라우트될 때 라우트 추적 메시지가 처리되는 방법을 판별하는 옵션을 지정하는 데 사용되는 많은 PCF 구조를 포함합니다.

관련 개념

[60 페이지의 『활동 기록』](#)

활동 기록은 큐 관리자 네트워크를 통해 메시지가 전달되는 라우트를 판별하는 기술입니다. 메시지가 취하는 라우트를 판별하려면 메시지를 위해 수행되는 활동이 기록됩니다.

[64 페이지의 『라우트 추적 메시징』](#)

라우트 추적 메시징은 라우트 추적 메시지를 사용하여 메시지에 대한 활동 정보를 기록하는 기술입니다. 라우트 추적 메시징은 큐 관리자 네트워크로 라우트 추적 메시지 송신을 포함합니다.

관련 참조

[68 페이지의 『TraceRoute PCF 그룹』](#)

TraceRoute PCF 제어 그룹의 속성이 라우트 추적 메시지의 작동을 제어합니다. TraceRoute PCF 그룹은 모든 라우트 추적 메시지의 메시지 데이터에 있습니다.

[99 페이지의 『활동 보고서 메시지 데이터』](#)

이 페이지를 사용하여 활동 보고서 메시지의 Activity PCF 그룹에 포함된 매개변수를 봅니다. 일부 매개변수는 특정 조작이 수행되었을 때에만 리턴됩니다.

활동 기록

활동 기록은 큐 관리자 네트워크를 통해 메시지가 전달되는 라우트를 판별하는 기술입니다. 메시지가 취하는 라우트를 판별하려면 메시지를 위해 수행되는 활동이 기록됩니다.

활동 레코드를 사용할 때, 메시지를 위해 수행된 각 활동을 활동 보고서에 기록할 수 있습니다. 활동 보고서는 보고 메시지의 유형입니다. 각 활동 보고서에는 메시지를 위해 활동을 수행한 애플리케이션에 대한 정보, 활동 발생 시기 및 활동의 일부로 수행된 조작에 관한 정보가 포함됩니다. 일반적으로 활동 보고서는 함께 수집된 응답 대상 큐에 전달됩니다. 메시지와 관련된 활동 보고서를 연구해서 큐 관리자 네트워크를 통해 메시지가 취한 라우트를 판별할 수 있습니다.

활동 보고서 사용법

메시지가 큐 관리자 네트워크를 통해 라우트되면 활동 보고서가 생성될 수 있습니다. 다음 방법으로 활동 보고서 정보를 사용할 수 있습니다.

마지막으로 알려진 메시지 위치 판별

활동 기록을 위해 사용할 수 있는 메시지가 의도된 목적지에 도달하지 않은 경우, 큐 관리자 네트워크를 통해 라우트되는 메시지에 대해 생성되는 활동 보고서를 연구하여 마지막으로 알려진 메시지 위치를 판별할 수 있습니다.

큐 관리자 네트워크의 구성 문제 판별

활동 기록으로 사용할 수 있는 여러 메시지를 큐 관리자 네트워크로 송신할 수 있습니다. 각 메시지와 관련된 활동 보고서를 연구하면 예상된 라우트를 취하지 않았음이 분명해질 수 있습니다. 이런 일이 발생할 수 있는 여러 이유가 있습니다. 예를 들어 강제로 메시지가 다른 라우트를 취하도록 하면서 채널이 중지될 수 있습니다. 이런 상황에서는 시스템 관리자가 큐 관리자 네트워크에 문제가 있는지 여부를 판별하고, 있는 경우 정정할 수 있습니다.

참고: IBM MQ 표시 라우트 애플리케이션을 사용하여 활동 기록을 라우트 추적 메시지와 결합하여 사용할 수 있습니다.

활동 보고서 형식

활동 보고서는 메시지를 위해 활동을 수행하는 애플리케이션에서 생성된 PCF 메시지입니다. 활동 보고서는 다음과 같이 메시지 디스크립터 및 메시지 데이터를 포함하는 표준 IBM MQ 보고 메시지입니다.

메시지 디스크립터

- MQMD 구조

메시지 데이터

- 임베드된 PCF 헤더(MQEPH)
- 활동 보고서 메시지 데이터

활동 보고서 메시지 데이터는 활동 PCF 그룹으로 구성되며, 라우트 추적 메시지로 생성된 경우 *TraceRoute* PCF 그룹으로 구성됩니다.

관련 참조

[MQMD - 메시지 디스크립터](#)

[MQEPH - 임베드된 PCF 헤더](#)

활동 기록 제어

큐 관리자 레벨에서 활동 기록을 사용 가능하게 하십시오. 전체 큐 관리자 네트워크를 사용 가능하게 설정하려면 활동 기록을 위해 네트워크의 모든 큐 관리자를 개별적으로 사용 가능하게 설정하십시오. 추가 큐 관리자를 사용 가능하게 하면 추가 활동 보고서가 생성됩니다.

이 태스크 정보

큐 관리자를 통해 라우트될 때 메시지에 대한 활동 보고서를 생성하려면 활동 보고서를 요청하는 메시지를 정의하고, 활동 기록을 위해 큐 관리자를 사용 가능하게 하고, 메시지에 대해 활동을 수행하는 애플리케이션이 활동 보고서를 생성할 수 있는지 확인하십시오.

메시지가 큐 관리자를 통해 라우트될 때 메시지에 대한 활동 보고서가 생성되기를 원하지 않는 경우, 활동 기록에 대해 큐 관리자를 사용 불가능하게 설정하십시오.

프로시저

1. 메시지에 대한 활동 보고서를 요청하십시오.

- a) 메시지의 메시지 디스크립터에서, *Report* 필드에 *MQRO_ACTIVITY*를 지정하십시오.
- b) 메시지의 메시지 디스크립터에서, *ReplyToQ* 필드에 응답 대상 큐의 이름을 지정하십시오.

경고: 활동 기록을 위해 큐 관리자 네트워크의 모든 메시지를 사용 가능하게 하는 것은 피하십시오. 활동 기록을 위해 사용 가능한 메시지는 그 대신 생성되는 많은 활동 보고서를 가질 수 있습니다. 큐 관리자 네트워크의 모든 메시지가 활동 기록에 대해 사용 가능한 경우, 큐 관리자 네트워크 트래픽이 허용할 수 없는 레벨로 증가할 수 있습니다.

2. 활동 기록에 대해 큐 관리자를 사용 가능 또는 사용 불가능하게 하십시오.

매개변수 *ACTIVREC*를 지정한 *MQSC* 명령 *ALTER QMGR*을 사용하여 큐 관리자 속성의 값을 변경하십시오. 가능한 값은 다음과 같습니다.

MSG

큐 관리자가 활동 기록에 대해 사용 가능합니다. 생성되는 모든 활동 보고서가 메시지의 메시지 디스크립터에 지정된 응답 대상 큐로 전달됩니다. 이는 기본값입니다.

큐

큐 관리자가 활동 기록에 대해 사용 가능합니다. 생성되는 모든 활동 보고서가 로컬 시스템 큐 *SYSTEM.ADMIN.ACTIVITY.QUEUE*로 전달됩니다. 시스템 큐는 또한 활동 보고서를 공용 큐로 전달하는데 사용될 수 있습니다.

DISABLED

큐 관리자가 활동 기록에 대해 사용 불가능합니다. 이 큐 관리자의 범위에 있는 중에는 활동 보고서가 생성되지 않습니다.

예를 들어, 활동 기록에 대해 큐 관리자를 사용 가능하게 하고 생성되는 모든 활동 보고서가 로컬 시스템 큐 *SYSTEM.ADMIN.ACTIVITY.QUEUE*로 전달되도록 지정하려면 다음 *MQSC* 명령을 사용하십시오.

```
ALTER QMGR ACTIVREC(Queue)
```

알아두기: *ACTIVREC* 큐 관리자 속성을 수정할 때 채널이 재시작될 때까지 실행 중인 *MCA*가 변경을 감지하지 않습니다.

3. 애플리케이션이 *MCA*가 메시지의 활동 보고서를 생성하는지 여부를 판별하기 위해 사용하는 것과 동일한 알고리즘을 사용하는지 확인하십시오.

- a) 메시지가 활동 보고서가 생성되도록 요청했는지 확인하십시오.
- b) 메시지가 현재 상주하는 큐 관리자가 활동 기록에 대해 사용 가능한지 확인하십시오.
- c) *ACTIVREC* 큐 관리자 속성에 의해 판별되는 큐에 활동 보고서를 넣으십시오.

활동 보고서를 위한 공용 큐 설정

보고서가 로컬 시스템 큐로 전달될 때 특정 메시지와 관련된 활동 보고서의 위치를 판별하려면 단일 노드에서 공용 큐를 사용하는 것이 더 효율적입니다.

시작하기 전에

활동 기록을 위해 큐 관리자를 사용 가능하게 하고 생성되는 모든 활동 보고서가 로컬 시스템 큐 *SYSTEM.ADMIN.ACTIVITY.QUEUE*로 전달되도록 지정하려면 **ACTIVREC** 매개변수를 설정하십시오.

이 태스크 정보

큐 관리자 네트워크의 많은 큐 관리자가 로컬 시스템 큐로 활동 보고서를 전달하도록 설정되는 경우, 특정 메시지와 관련된 활동 보고서의 위치를 판별하는 것은 시간 소모적일 수 있습니다. 그렇지 않으면 공용 큐를 호스팅하는 큐 관리자인 단일 노드를 사용하십시오. 큐 관리자 네트워크의 모든 큐 관리자가 이 공용 큐로 활동 보고서를 전달할 수 있습니다. 공용 큐 사용의 이점은 큐 관리자가 메시지에 지정된 응답 대상 큐로 활동 보고서를 전달할 필요가 없고 메시지와 관련된 활동 보고서의 위치를 판별할 때 하나의 큐만 조회한다는 점입니다.

공용 큐를 설정하려면 다음 단계를 수행하십시오.

프로시저

1. 큐 관리자를 단일 노드로 선택 또는 정의하십시오.
2. 단일 노드에서 공용 큐에 사용할 큐를 선택 또는 정의하십시오.
3. 활동 보고서가 공용 큐로 전달될 모든 큐 관리자에서 로컬 시스템 큐 SYSTEM.ADMIN.ACTIVITY.QUEUE를 리모트 큐 정의로 다시 정의하십시오.
 - a) 단일 노드의 이름을 리모트 큐 관리자 이름으로 지정하십시오.
 - b) 공용 큐의 이름을 리모트 큐 이름으로 지정하십시오.

메시지 라우트 정보 판별

메시지 라우트를 판별하려면 수집된 활동 보고서로부터 정보를 얻으십시오. 필수 정보를 판별하고 활동 보고서를 순서대로 정렬할 수 있기에 충분한 활동 보고서가 응답 대상 큐에 있는지 판별하십시오.

이 태스크 정보

활동 보고서가 응답 대상 큐에 놓여지는 순서가 반드시 활동이 수행된 순서와 상관되지 않습니다. 라우트 추적 메시지에 대해 생성되지 않은 경우 활동 보고서를 수동으로 정렬해야 하며, 라우트 추적 메시지를 위해 생성되는 경우에는 IBM MQ 표시 라우트 애플리케이션을 사용하여 활동 보고서를 정렬할 수 있습니다.

필요한 정보를 얻기 위해 응답 대상 큐에 충분한 활동 보고서가 있는지 판별하십시오.

프로시저

1. 활동 보고서의 ID와 원래 메시지를 비교하여 응답 대상 큐의 모든 관련 활동 보고서를 식별하십시오. 활동 보고서가 원래 메시지와 상관될 수 있도록 원래 메시지의 보고서 옵션을 설정해야 합니다.
2. 응답 대상 큐의 식별된 활동 보고서를 정렬하십시오.
활동 보고서의 다음 매개변수를 사용할 수 있습니다.

OperationType

수행되는 조작의 유형으로 현재 활동 보고서 직전 또는 직후에 생성된 활동 보고서를 판별할 수도 있습니다.

예를 들어, 활동 보고서에서 MCA가 전송 큐에서 채널을 따라 메시지를 송신했음을 상세히 설명합니다. 활동 보고서에서 상술되는 마지막 조작은 `send`의 `OperationType`을 갖고 메시지가 채널 CH1을 사용하여 목적지 큐 관리자 QM1으로 송신되었음을 상술합니다. 이것은 메시지에 수행된 다음 활동이 큐 관리자 QM1에서 발행했으며 채널 CH1의 `receive` 조작으로 시작했음을 의미합니다. 이 정보를 사용하여 다음 활동 보고서가 존재하고 획득되었다고 가정할 때 다음 활동 보고서를 식별할 수 있습니다.

OperationDate 및 **OperationTime**

각 활동 보고서에서 조작의 날짜 및 시간으로부터 활동의 일반적인 순서를 판별할 수 있습니다.

경고: 큐 관리자 네트워크의 모든 큐 관리자가 시스템 시계를 동기화하지 않은 경우, 날짜 및 시간순으로 정렬하는 것이 활동 보고서가 올바른 순서라고 보증하지 않습니다. 순서를 수동으로 설정해야 합니다.

활동 보고서의 순서는 메시지가 큐 관리자 네트워크를 통해 잡은 라우트 또는 부분 라우트를 나타냅니다.

3. 정렬된 활동 보고서의 활동 정보로부터 필요한 정보를 얻으십시오.
메시지에 관한 정보를 충분하지 않은 경우, 추가 활동 보고서를 획득할 수도 있습니다.

추가 활동 보고서 검색

메시지 라우트를 판별하기 위해서는 수집된 활동 보고서로부터 충분한 정보가 사용 가능해야 합니다. 메시지가 지정한 응답 대상 큐로부터 메시지 관련 활동 보고서를 검색하지만 필요한 정보가 없는 경우 추가 활동 보고서를 찾으십시오.

이 태스크 정보

추가 활동 보고서의 위치를 판별하려면 다음 단계를 수행하십시오.

프로시저

1. 활동 보고서를 공용 큐로 전달하는 큐 관리자 네트워크의 모든 큐 관리자의 경우, 원래 메시지의 *MsgId*와 일치하는 *CorrelId*를 갖는 공용 큐에서 활동 보고서를 검색하십시오.
2. 공용 큐로 활동 보고서를 전달하지 않는 큐 관리자 네트워크의 모든 큐 관리자의 경우, 다음과 같이 활동 보고서를 검색하십시오.
 - a) 기존 활동 보고서를 조사하여 메시지가 라우트된 큐 관리자를 식별하십시오.
 - b) 이들 큐 관리자에 대해 활동 기록에 대해 사용 가능한 큐 관리자를 식별하십시오.
 - c) 이들 큐 관리자에 대해, 지정된 응답 대상 큐로 활동 보고서를 리턴하지 않은 모든 큐 관리자를 식별하십시오.
 - d) 식별한 각 큐 관리자에 대해, 시스템 큐 SYSTEM.ADMIN.ACTIVITY.QUEUE를 검사하고 원래 메시지의 *MsgId*와 일치하는 *CorrelId*를 갖는 모든 활동 보고서를 검색하십시오.
 - e) 시스템 큐에서 활동 보고서를 찾지 못하는 경우, 큐 관리자 데드-레터 큐(있는 경우)를 확인하십시오. 활동 보고서는 보고서 옵션 MQRO_DEAD_LETTER_Q가 설정되는 경우 데드-레터 큐로만 전달될 수 있습니다.
3. 획득한 모든 활동 보고서를 순서대로 배열하십시오.
그러면 활동 보고서의 순서는 메시지가 잡은 라우트 또는 부분 라우트를 나타냅니다.
4. 정렬된 활동 보고서의 활동 정보로부터 필요한 정보를 얻으십시오.
일부 상황에서는 기록된 활동 정보가 지정된 응답 대상 큐, 공용 큐 또는 시스템 큐에 접근할 수 없습니다.

활동 정보가 획득되지 않는 상황

메시지를 위해 수행되는 활동의 전체 순서를 판별하려면 모든 활동과 관련된 정보가 획득되어야 합니다. 임의의 활동에 관련된 정보가 기록되지 않았거나 획득되지 않은 경우 활동의 부분적인 순서만 판별할 수 있습니다.

다음 상황에서 활동 정보가 기록되지 않습니다.

- 메시지가 IBM WebSphere® MQ 6.0 이전의 큐 관리자에 의해 처리됩니다.
- 메시지가 활동 기록에 대해 사용 가능하지 않은 큐 관리자에 의해 처리됩니다.
- 메시지를 처리할 것으로 예상한 애플리케이션이 실행 중이 아닙니다.

다음 상황에서는 기록된 활동 정보가 지정된 응답 대상 큐에 도달할 수 없습니다.

- 응답 대상 큐로 활동 보고서를 라우트하도록 정의된 채널이 없습니다.
- 활동 보고서를 응답 대상 큐로 라우트할 채널이 실행 중이 아닙니다.
- 활동 보고서를 다시 응답 대상 큐가 상주하는 큐 관리자(큐 관리자 알리어스)로 라우트할 리모트 큐 정의가 정의되지 않았습니다.
- 원래 메시지를 생성한 사용자가 큐 관리자 알리어스에 대한 열기 또는 Put 권한이 없습니다.
- 원래 메시지를 생성한 사용자가 응답 대상 큐에 대한 열기 또는 Put 권한이 없습니다.
- 응답 대상 큐가 put 금지되었습니다.

다음 상황에서는 기록된 활동 정보가 시스템 큐 또는 공용 큐에 도달할 수 없습니다.

- 공용 큐가 사용되며 활동 보고서를 공용 큐로 라우트하도록 정의된 채널이 없는 경우.
- 공용 큐가 사용되며 활동 보고서를 공용 큐로 라우트할 채널이 실행 중이 아닌 경우.
- 공용 큐가 사용되며 시스템 큐가 올바르게 정의된 경우.
- 원래 메시지를 생성한 사용자가 시스템 큐에 대한 열기 또는 Put 권한이 없습니다.
- 시스템 큐가 put 금지되었습니다.
- 공용 큐가 사용되며 원래 메시지를 생성한 사용자가 공용 큐에 대한 열기 또는 Put 권한이 없는 경우.
- 공용 큐가 사용되며 공용 큐가 put 금지된 경우.

이런 상황에서는 활동 보고서가 보고서 옵션 MQRO_DISCARD_MSG가 지정되지 않은 경우 활동 보고서가 거부되는 큐 관리자에서 데드-레터 큐가 정의된 경우 활동 보고서를 데드-레터 큐에서 검색할 수 있습니다. 활동 보고서는 활동 보고서가 생성된 원래 메시지가 메시지 디스크립터의 보고서 필드에서

MQRO_PASS_DISCARD_AND_EXPIRY 및 MQRO_DISCARD_MSG가 둘 다 지정된 경우에만 이 보고서 옵션이 지정됩니다.

라우트 추적 메시징

라우트 추적 메시징은 라우트 추적 메시지를 사용하여 메시지에 대한 활동 정보를 기록하는 기술입니다. 라우트 추적 메시징은 큐 관리자 네트워크로 라우트 추적 메시지 송신을 포함합니다.

라우트 추적 메시징이 큐 관리자 네트워크를 통해 라우트될 때 활동 정보가 기록됩니다. 이 활동 정보에는 활동을 수행한 애플리케이션, 수행된 시기, 활동의 일부로 수행된 조작에 관한 정보가 포함됩니다. 라우트 추적 메시징을 사용하여 기록된 정보를 다음 목적으로 사용할 수 있습니다.

메시지의 마지막 알려진 위치 판별

메시지가 의도된 목적지에 도달하지 않는 경우, 라우트 추적 메시지에 대해 기록된 활동 정보를 사용하여 메시지의 마지막으로 알려진 위치를 판별할 수 있습니다. 라우트 추적 메시지는 동일한 라우트를 따르기 위해 원래 메시지와 동일한 대상 목적지를 갖는 큐 관리자 네트워크로 송신됩니다. 활동 정보는 라우트 추적 메시지의 메시지 데이터에 누적되거나 활동 보고서를 사용하여 기록될 수 있습니다. 라우트 추적 메시지가 원래 메시지와 동일한 라우트를 따를 확률을 늘리기 위해 원래 메시지를 흉내내도록 라우트 추적 메시지를 수정할 수 있습니다.

큐 관리자 네트워크의 구성 문제 판별

라우트 추적 메시지는 큐 관리자 네트워크로 송신되고 활동 정보가 기록됩니다. 라우트 추적 메시지에 대해 기록되는 활동 정보를 연구하여 라우트 추적 메시지가 예상 라우트를 따르지 않았음이 명백해질 수 있습니다. 이런 일이 발생할 수 있는 여러 이유가 있습니다. 예를 들어 채널이 비활성이어서 메시지가 다른 라우트를 취하도록 강제할 수 있습니다. 이런 상황에서는 시스템 관리자가 큐 관리자 네트워크에 문제가 있는지 여부를 판별하고, 있는 경우 정정할 수 있습니다.

IBM MQ 표시 라우트 애플리케이션을 사용하여 라우트 추적 메시지를 구성, 생성하고 큐 관리자 네트워크로 보낼 수 있습니다.

경고: 라우트 추적 메시지를 분배 목록에 넣는 경우 결과는 정의되지 않습니다.

관련 개념

114 페이지의 『라우트 추적 메시지 참조』

이 페이지를 사용하여 라우트 추적 메시지 형식을 개괄합니다. 라우트 추적 메시지 데이터는 라우트 추적 메시지가 유발한 활동을 설명하는 매개변수를 포함합니다.

활동 정보가 기록되는 방법

라우트 추적 메시징을 사용하면 라우트 추적 메시지의 메시지 데이터에 활동 정보를 기록하거나 활동 보고서를 사용할 수 있습니다. 그렇지 않으면 두 기술을 모두 사용할 수 있습니다.

라우트 추적 메시지의 메시지 데이터에 활동 정보 누적

라우트 추적 메시지가 큐 관리자 네트워크를 통해 라우트될 때, 라우트 추적 메시지를 위해 수행되는 활동에 관한 정보가 라우트 추적 메시지의 메시지 데이터에 누적될 수 있습니다. 활동 정보는 *Activity* PCF 그룹에 저장됩니다. 라우트 추적 메시지를 위해 수행되는 모든 활동의 경우 *Activity* PCF 그룹이 라우트 추적 메시지의 메시지 데이터에 있는 PCF 블록의 끝에 기록됩니다.

추가 활동 정보는 *TraceRoute* PCF 그룹이라는 PCF 그룹에서 라우트 추적 메시지에 기록됩니다. 추가 활동 정보는 이 PCF 그룹에 저장되며, 기록된 활동의 순서를 판별하는 데 사용될 수 있습니다. 이 기술은 *TraceRoute* PCF 그룹의 *Accumulate* 매개변수에 의해 제어됩니다.

활동 보고서를 사용한 활동 정보 기록

라우트 추적 메시지가 큐 관리자 네트워크를 통해 라우트될 때, 라우트 추적 메시지를 위해 수행되는 모든 활동에 대한 활동 보고서가 생성될 수 있습니다. 활동 정보는 *Activity* PCF 그룹에 저장됩니다. 라우트 추적 메시지를 위해 수행되는 모든 활동에 대해 *Activity* PCF 그룹을 포함하는 활동 보고서가 생성됩니다. 라우트 추적 메시지에 대한 활동 기록은 다른 메시지와 동일한 방식으로 작업합니다.

라우트 추적 메시지에 대해 생성되는 활동 보고서는 다른 메시지에 대해 생성되는 메시지에 비교하여 추가 활동 정보를 포함합니다. 추가 정보는 *TraceRoute* PCF 그룹에서 리턴됩니다. *TraceRoute* PCF 그룹에 들어있는 정보는 활동 보고서가 생성된 시간부터만 정확합니다. 추가 정보를 사용하여 라우트 추적 메시지를 위해 수행되는 활동의 순서를 판별할 수 있습니다.

기록된 활동 정보 획득

라우트 추적 메시지가 그의 의도된 목적지에 도달했거나 제거될 때, 활동 정보를 획득하기 위해 사용하는 방법은 해당 정보가 기록된 방법에 따라 다릅니다.

시작하기 전에

활동 정보에 익숙하지 않은 경우 [64 페이지의 『활동 정보가 기록되는 방법』](#)의 내용을 참조하십시오.

이 태스크 정보

라우트 추적 메시지가 그의 의도된 목적지에 도달했거나 제거된 후 활동 정보를 획득하려면 다음 방법을 사용하십시오.

프로시저

- 라우트 추적 메시지를 검색하십시오.

TraceRoute PCF 그룹의 *Deliver* 매개변수가 라우트 추적 메시지가 도착 시 대상 큐에 배치되는지 아니면 제거되는지 여부를 제어합니다. 라우트 추적 메시지가 대상 큐로 전달되는 경우 이 큐에서 라우트 추적 메시지를 검색할 수 있습니다. 그러면 IBM MQ 표시 라우트 애플리케이션을 사용하여 활동 정보를 표시할 수 있습니다.

활동 정보가 라우트 추적 메시지의 메시지 데이터에 누적되도록 요청하려면 *TraceRoute* PCF 그룹의 *Accumulate* 매개변수를 `MQROUTE_ACCUMULATE_IN_MSG`로 설정하십시오.

- 라우트 추적 응답 메시지를 사용하십시오.

라우트 추적 메시지가 의도된 목적지에 도달하거나 라우트 추적 메시지가 큐 관리자 네트워크에서 더 이상 라우트될 수 없을 때, 라우트 추적 응답 메시지가 생성될 수 있습니다. 라우트 추적 응답 메시지에는 라우트 추적 메시지의 모든 활동 정보의 복제가 들어 있으며, 지정된 응답 대상 큐 또는 시스템 큐 `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`로 전달됩니다. IBM MQ 표시 라우트 애플리케이션을 사용하여 활동 정보를 표시할 수 있습니다.

라우트 추적 응답 메시지를 요청하려면 *TraceRoute* PCF 그룹의 *Accumulate* 매개변수를 `MQROUTE_ACCUMULATE_AND_REPLY`로 설정하십시오.

- 활동 보고서를 사용하십시오.

라우트 추적 메시지에 대한 활동 보고서가 생성되는 경우, 활동 정보를 획득할 수 있기 전에 활동 보고서를 찾아야 합니다. 그런 다음 활동 순서를 판별하기 위해 활동 보고서를 정렬해야 합니다.

라우트 추적 메시징 제어

해당 큐 관리자의 범위에 있는 애플리케이션이 라우트 추적 메시지에 활동 정보를 쓸 수 있도록 큐 관리자 레벨에서 라우트 추적 메시징을 사용 가능하게 하십시오. 전체 큐 관리자 네트워크를 사용 가능하게 설정하려면 라우트 추적 메시징을 위해 네트워크의 모든 큐 관리자를 개별적으로 사용 가능하게 설정하십시오. 추가 큐 관리자를 사용 가능하게 하면 추가 활동 보고서가 생성됩니다.

시작하기 전에

활동 보고서를 사용하여 라우트 추적 메시지에 대한 활동 정보를 기록 중인 경우 [60 페이지의 『활동 기록 제어』](#)의 내용을 참조하십시오.

이 태스크 정보

큐 관리자를 통해 라우트될 때 라우트 추적 메시지에 대한 활동 정보를 기록하려면 다음 단계를 수행하십시오.

프로시저

- 활동 정보가 라우트 추적 메시지에 대해 기록될 방법을 정의하십시오.

[67 페이지의 『라우트 추적 메시지 생성 및 구성』](#)의 내용을 참조하십시오.

- 라우트 추적 메시지에 활동 정보를 누적하려는 경우 큐 관리자가 라우트 추적 메시지에 대해 사용 가능한지 확인하십시오.

- 라우트 추적 메시지에 활동 정보를 누적하려는 경우, 라우트 추적 메시지에 활동을 수행하는 애플리케이션이 라우트 추적 메시지의 메시지 데이터에 활동 정보를 쓸 수 있는지 확인하십시오.

관련 개념

67 페이지의 『라우트 추적 메시지 생성 및 구성』

라우트 추적 메시지는 특정 메시지 디스크립터와 메시지 데이터 부분으로 구성됩니다. 라우트 추적 메시지를 생성하려면 메시지를 수동으로 작성하거나 IBM MQ 표시 라우트 애플리케이션을 사용하십시오.

관련 태스크

60 페이지의 『활동 기록 제어』

큐 관리자 레벨에서 활동 기록을 사용 가능하게 하십시오. 전체 큐 관리자 네트워크를 사용 가능하게 설정하려면 활동 기록을 위해 네트워크의 모든 큐 관리자를 개별적으로 사용 가능하게 설정하십시오. 추가 큐 관리자를 사용 가능하게 하면 추가 활동 보고서가 생성됩니다.

라우트 추적 메시지에 대한 큐 관리자 사용 가능

큐 관리자가 라우트 추적 메시지에 대해 사용 가능 또는 사용 불가능한지 여부를 제어하려면 큐 관리자 속성 ROUTEREC를 사용하십시오.

큐 관리자 속성의 값을 변경하려면 ROUTEREC 매개변수를 지정한 MQSC 명령 ALTER QMGR을 사용하십시오. 가능한 값은 다음 값 중 하나입니다.

MSG

큐 관리자가 라우트 추적 메시징에 사용 가능합니다. 큐 관리자의 범위 안에 있는 애플리케이션은 라우트 추적 메시징에 활동 정보를 쓸 수 있습니다.

TraceRoute PCF 그룹의 *Accumulate* 매개변수가 MQROUTE_ACCUMULATE_AND_REPLY로 설정되고 라우트 추적 메시징에 수행될 다음 활동이

- 제거됨
- 로컬 큐(대상 큐 또는 데드-레터 큐)에 넣어짐
- 라우트 추적 메시징에 대해 수행되는 총 활동 수가 TraceRoute PCF 그룹의 *MaxActivities* 매개변수 값을 초과하게 됨

라우트 추적 응답 메시징이 생성되고, 라우트 추적 메시징의 메시지 디스크립터에서 지정되는 응답 대상 큐로 전달됩니다.

큐

큐 관리자가 라우트 추적 메시징에 사용 가능합니다. 큐 관리자의 범위 안에 있는 애플리케이션은 라우트 추적 메시징에 활동 정보를 쓸 수 있습니다.

TraceRoute PCF 그룹의 *Accumulate* 매개변수가 MQROUTE_ACCUMULATE_AND_REPLY로 설정되고 라우트 추적 메시징에 수행될 다음 활동이

- 제거됨
- 로컬 큐(대상 큐 또는 데드-레터 큐)에 넣어짐
- 라우트 추적 메시징에 대해 수행되는 총 활동 수가 TraceRoute PCF 그룹의 *MaxActivities* 매개변수 값을 초과하게 됨

라우트 추적 응답 메시징이 생성되고, 로컬 시스템 큐 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE로 전달됩니다.

DISABLED

큐 관리자가 라우트 추적 메시징에 사용 불가능합니다. 활동 정보가 라우트 추적 메시징에 누적되지 않지만, TraceRoute PCF 그룹이 이 큐 관리자의 범위에 있는 동안 업데이트될 수 있습니다.

예를 들어, 라우트 추적 메시징에 대해 큐 관리자를 사용 불가능하게 설정하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR ROUTEREC(DISABLED)
```

알아두기: ROUTEREC 큐 관리자 속성을 수정할 때, 실행 중인 MCA는 채널이 재시작될 때까지 변경을 감지하지 못합니다.

라우트 추적 메시징에 대해 애플리케이션 사용 가능
사용자 애플리케이션에 대해 라우트 추적 메시징을 사용 가능하게 설정하려면 MCA(message channel agent)가 사용하는 알고리즘을 사용자 알고리즘의 기반으로 하십시오.

시작하기 전에

라우트 추적 메시지의 형식에 익숙하지 않은 경우 [114 페이지의 『라우트 추적 메시지 참조』](#)의 내용을 참조하십시오.

이 태스크 정보

MCA(Message channel agent)는 라우트 추적 메시징에 대해 사용 가능합니다. 사용자 애플리케이션을 라우트 추적 메시징에 대해 사용 가능하게 설정하려면 MCA가 사용하는 알고리즘에서 다음 단계를 사용하십시오.

프로시저

1. 처리되는 메시지가 라우트 추적 메시지인지 여부를 판별하십시오.
메시지가 라우트 추적 메시지의 형식을 준수하지 않는 경우, 메시지는 라우트 추적 메시지로 처리되지 않습니다.
2. 활동 정보가 기록되는지 여부를 판별하십시오.
수행되는 활동의 세부사항 레벨이 *Detail* 매개변수에 의해 지정되는 세부사항 레벨보다 낮지 않은 경우, 활동 정보는 특정 상황에서 기록됩니다. 이 정보는 라우트 추적 메시지가 누적을 요청하고 큐 관리자가 라우트 추적 메시징에 대해 사용 가능한 경우 또는 라우트 추적 메시지가 활동 보고서를 요청하고 큐 관리자가 활동 기록에 대해 사용 가능한 경우에만 기록됩니다.
 - 활동 정보가 기록될 경우 *RecordedActivities* 매개변수를 증분하십시오.
 - 활동 정보가 기록되지 않을 경우 *UnrecordedActivities* 매개변수를 증분하십시오.
3. 라우트 추적 메시지에 수행되는 활동의 총 수가 *MaxActivities* 매개변수의 값을 초과하는지 판별하십시오.
활동의 총 수는 *RecordedActivities*, *UnrecordedActivities* 및 *DiscontinuityCount*의 합입니다.
활동의 총 수가 *MaxActivities*를 초과하는 경우 메시지를 MQFB_MAX_ACTIVITIES 피드백과 함께 거부하십시오.
4. *Accumulate*의 값이 MQROUTE_ACCUMULATE_IN_MSG 또는 MQROUTE_ACCUMULATE_AND_REPLY로 설정되고 큐 관리자가 라우트 추적 메시징에 대해 사용 가능한 경우, 라우트 추적 메시지의 메시지 데이터에 있는 PCF 블록의 끝에 Activity PCF 그룹을 쓰십시오.
5. 라우트 추적 메시지를 로컬 큐로 전달하십시오.
 - *Deliver* 매개변수가 MQROUTE_DELIVER_NO로 지정되는 경우 라우트 추적 메시지를 MQFB_NOT_DELIVERED 피드백과 함께 거부하십시오.
 - *Deliver* 매개변수가 MQROUTE_DELIVER_YES로 지정되는 경우 라우트 추적 메시지를 로컬 큐로 전달하십시오.
6. 다음 조건이 모두 참인 경우 라우트 추적 응답 메시지를 생성하십시오.
 - 라우트 추적 메시지가 로컬 큐로 전달되거나 거부되었습니다.
 - *Accumulate* 매개변수의 값이 MQROUTE_ACCUMULATE_AND_REPLY입니다.
 - 큐 관리자가 라우트 추적 메시징에 대해 사용 가능합니다.라우트 추적 응답 메시지는 ROUTEREC 큐 관리자 속성에 의해 판별되는 큐에 넣어집니다.
7. 라우트 추적 메시지가 활동 보고서를 요청했고 큐 관리자가 활동 기록에 대해 사용 가능한 경우 활동 보고서를 생성하십시오.
활동 보고서는 ACTIVREC 큐 관리자 속성에 의해 판별되는 큐에 넣어집니다.

라우트 추적 메시지 생성 및 구성

라우트 추적 메시지는 특정 메시지 디스크립터와 메시지 데이터 부분으로 구성됩니다. 라우트 추적 메시지를 생성하려면 메시지를 수동으로 작성하거나 IBM MQ 표시 라우트 애플리케이션을 사용하십시오.

라우트 추적 메시지는 다음 부분으로 구성됩니다.

메시지 디스크립터

MQFMT_ADMIN 또는 MQFMT_EMBEDDED_PCF로 설정된 *Format* 필드를 갖는 MQMD 구조입니다.

메시지 데이터

다음 조합 중 하나입니다.

- *Format*이 MQFMT_ADMIN으로 설정된 경우 PCF 헤더(MQCFH)와 라우트 추적 메시지
- *Format*이 MQFMT_EMBEDDED_PCF로 설정된 경우 임베드된 PCF 헤더(MQEPH), 라우트 추적 메시지 데이터 및 추가 사용자 지정 메시지 데이터

라우트 추적 메시지 데이터는 *TraceRoute* PCF 그룹과 하나 이상의 *Activity* PCF 그룹으로 구성됩니다.

수동 생성

라우트 추적 메시지를 수동으로 생성할 때 *Activity* PCF 그룹은 필수가 아닙니다. *Activity* PCF 그룹은 MCA 또는 사용자 작성 애플리케이션이 그 대신에 활동을 수행할 때 라우트 추적 메시지의 메시지 데이터에 기록됩니다.

IBM MQ 표시 라우트 애플리케이션

라우트 추적 메시지를 구성 및 생성하고 큐 관리자 네트워크에 넣으려면 IBM MQ 표시 라우트 애플리케이션인 **dspmqrte**를 사용하십시오. 메시지 디스크립터의 *Format* 매개변수를 MQFMT_ADMIN으로 설정하십시오. IBM MQ 표시 라우트 애플리케이션에 의해 생성된 라우트 추적 메시지에 사용자 데이터를 추가할 수 없습니다.

제한사항: dspmqrte 는 IBM MQ for z/OS 큐 관리자에서 발행할 수 없습니다. 라우트 추적 메시지가 라우트되는 첫 번째 큐 관리자가 이 유형의 큐 관리자가 되도록 하려면 선택적 매개변수 *-c*를 사용하여 클라이언트로 큐 관리자에 연결하십시오.

원래 메시지 흉내내기

라우트 추적 메시지를 사용하여 다른 메시지가 큐 관리자 네트워크를 통해 취한 라우트를 판별할 때, 라우트 추적 메시지가 원래 메시지를 더 가깝게 흉내낼수록 라우트 추적 메시지가 원래 메시지와 동일한 라우트를 따를 기회가 더 큽니다.

다음 메시지 특성이 큐 관리자 네트워크 내에서 메시지가 전달되는 장소에 영향을 미칠 수 있습니다.

Priority

우선순위는 메시지의 메시지 디스크립터에서 지정할 수 있습니다.

Persistence

지속은 메시지의 메시지 디스크립터에서 지정할 수 있습니다.

만기

만기는 메시지의 메시지 디스크립터에서 지정할 수 있습니다.

보고서 옵션

보고서 옵션은 메시지의 메시지 디스크립터에서 지정할 수 있습니다.

메시지 크기

메시지의 크기를 흉내내기 위해 추가 데이터를 메시지의 메시지 데이터에 기록할 수 있습니다. 이 목적의 경우 추가 메시지 데이터는 의미가 없을 수 있습니다.

팁: IBM MQ 표시 라우트 애플리케이션은 메시지 크기를 지정할 수 없습니다.

메시지 데이터

일부 큐 관리자 네트워크는 콘텐츠 기반 라우팅을 사용하여 메시지가 전달되는 장소를 판별합니다. 이 경우 라우트 추적 메시지의 메시지 데이터가 원래 메시지의 메시지 데이터를 흉내내도록 작성되어야 합니다.

팁: IBM MQ 표시 라우트 애플리케이션은 메시지 데이터를 지정할 수 없습니다.

TraceRoute PCF 그룹

TraceRoute PCF 제어 그룹의 속성이 라우트 추적 메시지의 작동을 제어합니다. *TraceRoute* PCF 그룹은 모든 라우트 추적 메시지의 메시지 데이터에 있습니다.

다음 표는 MCA가 인식하는 *TraceRoute* 그룹의 매개변수를 나열합니다. 73 페이지의 『추가 활동 정보』에 설명된 대로 사용자 작성 애플리케이션이 인식하도록 작성된 경우 매개변수를 더 추가할 수 있습니다.

표 12. TraceRoute PCF 그룹

매개변수	유형
TraceRoute	MQCFGR
세부사항	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
누적	MQCFIN
Forward	MQCFIN
전달	MQCFIN

TraceRoute PCF 그룹에 있는 각 매개변수의 설명은 다음과 같습니다.

Detail

기록되는 활동 정보의 세부사항 레벨을 지정합니다. 가능한 값은 다음 값 중 하나입니다.

MQROUTE_DETAIL_LOW

사용자 애플리케이션이 수행한 활동만 기록됩니다.

MQROUTE_DETAIL_MEDIUM

MQROUTE_DETAIL_LOW에서 지정된 활동이 기록됩니다. 또한 MCA가 수행한 활동이 기록됩니다.

MQROUTE_DETAIL_HIGH

MQROUTE_DETAIL_LOW 및 MQROUTE_DETAIL_MEDIUM에서 지정되는 활동이 기록됩니다. MCA는 이 세부사항 레벨에서 어떤 추가 활동 정보도 기록하지 않습니다. 이 옵션은 추가 활동 정보를 기록할 사용자 애플리케이션에만 사용할 수 있습니다. 예를 들어, 사용자 애플리케이션이 특정 메시지 특성을 고려하여 메시지가 택하는 라우트를 판별하는 경우, 라우팅 논리에 관한 정보가 이 세부사항 레벨에 포함될 수 있습니다.

RecordedActivities

라우트 추적 메시지를 위해 수행되는 기록된 활동의 수를 지정합니다. 활동은 그에 관한 정보가 라우트 추적 메시지에 작성된 경우 또는 활동 보고서가 생성된 경우에 기록된 것으로 간주됩니다. 모든 기록된 활동에 대해 *RecordedActivities*가 하나씩 증분합니다.

UnrecordedActivities

라우트 추적 메시지를 위해 수행되는 기록되지 않은 활동의 수를 지정합니다. 활동은 라우트 추적 메시징에 대해 사용 가능한 애플리케이션이 누적하지 않거나 관련 활동 정보가 활동 보고서에 기록되지 않는 경우 기록되지 않는 것으로 간주됩니다.

라우트 추적 메시지를 위해 수행되는 활동은 다음 상황에서 기록되지 않습니다.

- 수행되는 활동의 세부사항 레벨이 *Detail* 매개변수에 의해 지정되는 세부사항 레벨보다 낮습니다.
- 라우트 추적 메시지가 활동 보고서를 요청하지만 누적은 요청하지 않으며, 큐 관리자가 활동 기록에 대해 사용 가능하지 않습니다.
- 라우트 추적 메시지가 누적을 요청하지만 활동 보고서는 요청하지 않으며, 큐 관리자가 라우트 추적 메시징에 대해 사용 가능하지 않습니다.
- 라우트 추적 메시지가 누적과 활동 보고서를 둘 다 요청하고, 큐 관리자가 활동 기록 및 라우트 추적 메시징에 대해 사용 가능하지 않습니다.
- 라우트 추적 메시지가 누적이나 활동 보고서를 요청하지 않습니다.

모든 기록되지 않는 활동에 대해 *UnrecordedActivities* 매개변수가 하나씩 증분합니다.

DiscontinuityCount

라우트 추적 메시지가 라우트 추적 메시징에 대해 사용 가능하지 않은 애플리케이션으로 큐 관리자를 통해 라우트된 횟수를 지정합니다. 이 값은 큐 관리자에 의해 증분됩니다. 값이 0보다 큰 경우 부분 메시지 라우트만 판별할 수 있습니다.

MaxActivities

라우트 추적 메시지를 위해 수행될 수 있는 활동의 최대 수를 지정합니다.

활동의 총 수는 *RecordedActivities*, *UnrecordedActivities* 및 *DiscontinuityCount*의 합입니다. 활동의 총 수는 *MaxActivities*의 값을 초과해서는 안됩니다.

*MaxActivities*의 값은 다음과 같을 수 있습니다.

양수

활동의 최대 수입니다.

활동의 최대 수가 초과되는 경우 라우트 추적 메시지는 *MQFB_MAX_ACTIVITIES* 피드백과 함께 거부됩니다. 이것은 라우트 추적 메시지가 무한 루프에 갇힌 경우 무한히 전송되지 않도록 막을 수 있습니다.

MQROUTE_UNLIMITED_ACTIVITIES

무제한 숫자의 활동이 라우트 추적 메시지를 위해 수행될 수 있습니다.

Accumulate

활동 정보를 누적하는 데 사용되는 방법을 지정합니다. 가능한 값은 다음 값 중 하나입니다.

MQROUTE_ACCUMULATE_IN_MSG

큐 관리자가 라우트 추적 메시징에 대해 사용 가능한 경우, 활동 정보는 라우트 추적 메시지의 메시지 데이터에 누적됩니다.

이 값이 지정되는 경우 라우트 추적 메시지 데이터는 다음으로 구성됩니다.

- *TraceRoute* PCF 그룹.
- 0개 이상의 *Activity* PCF 그룹.

MQROUTE_ACCUMULATE_AND_REPLY

큐 관리자가 라우트 추적 메시징에 대해 사용 가능한 경우 활동 정보가 라우트 추적 메시지의 메시지 데이터에 누적되고 다음 중 하나가 발생하는 경우 라우트 추적 응답 메시지가 생성됩니다.

- IBM MQ 큐 관리자가 라우트 추적 메시징을 제거합니다.
- IBM MQ 큐 관리자가 라우트 추적 메시지를 로컬 큐(대상 큐나 데드-레터 큐)에 넣습니다.
- 라우트 추적 메시징에 대해 수행되는 활동의 수가 *MaxActivities*의 값을 초과합니다.

이 값이 지정되는 경우 라우트 추적 메시지 데이터는 다음으로 구성됩니다.

- *TraceRoute* PCF 그룹.
- 0개 이상의 *Activity* PCF 그룹.

MQROUTE_ACCUMULATE_NONE

라우트 추적 메시지의 메시지 데이터에 활동 정보가 누적되지 않습니다.

이 값이 지정되는 경우 라우트 추적 메시지 데이터는 다음으로 구성됩니다.

- *TraceRoute* PCF 그룹.

Forward

라우트 추적 메시지가 전달될 수 있는 위치를 지정합니다. 가능한 값은 다음과 같습니다.

MQROUTE_FORWARD_IF_SUPPORTED

라우트 추적 메시지는 *TraceRoute* 그룹의 *Deliver* 매개변수의 값을 존중하는 큐 관리자로서만 전달됩니다.

MQROUTE_FORWARD_ALL

라우트 추적 메시지가 *Deliver* 매개변수의 값이 존중되는지 여부와 상관없이 모든 큐 관리자로 전달됩니다.

큐 관리자는 라우트 추적 메시지를 리모트 큐 관리자로 전달할지 여부를 판별할 때 다음 알고리즘을 사용합니다.

1. 리모트 큐 관리자가 라우트 추적 메시징을 지원할 수 있는지 여부를 판별합니다.
 - 리모트 큐 관리자가 라우트 추적 메시징을 지원할 수 있는 경우 알고리즘은 [71 페이지의 『4』](#) 단계에서 계속합니다.
 - 리모트 큐 관리자가 라우트 추적 메시징을 지원할 수 없는 경우 알고리즘은 [70 페이지의 『2』](#) 단계에서 계속합니다.
2. *TraceRoute* 그룹의 *Deliver* 매개변수가 *MQROUTE_DELIVER_REJ_UNSUP_MASK* 비트 마스크의 인식되지 않는 전달 옵션을 포함하는지 판별합니다.

- 인식되지 않는 전달 옵션이 발견된 경우 라우트 추적 메시지는 MQFB_UNSUPPORTED_DELIVERY 피드백과 함께 거부됩니다.
 - 인식되지 않는 전달 옵션이 발견되지 않는 경우 알고리즘은 71 페이지의 『3』 단계에서 계속됩니다.
3. 라우트 추적 메시지의 *TraceRoute* PCF 그룹에 있는 *Deliver* 매개변수의 값을 판별합니다.
- *Deliver*가 MQRROUTE_DELIVER_YES로 지정되는 경우, 라우트 추적 메시지는 리모트 큐 관리자로 전달됩니다.
 - *Deliver*가 MQRROUTE_DELIVER_NO로 지정되는 경우, 알고리즘이 71 페이지의 『4』 단계에서 계속됩니다.
4. *TraceRoute* 그룹의 *Forward* 매개변수가 MQRROUTE_FORWARDING_REJ_UNSUP_MASK 비트 마스크에 인식되지 않는 전달 옵션을 포함하는지 판별합니다.
- 인식되지 않는 전달 옵션이 발견된 경우 라우트 추적 메시지는 MQFB_UNSUPPORTED_FORWARDING 피드백과 함께 거부됩니다.
 - 인식되지 않는 전달 옵션이 발견되지 않는 경우 알고리즘은 71 페이지의 『5』 단계에서 계속됩니다.
5. 라우트 추적 메시지의 *TraceRoute* PCF 그룹에 있는 *Forward* 매개변수의 값을 판별합니다.
- *Forward*가 MQRROUTE_FORWARD_IF_SUPPORTED로 지정되는 경우, 라우트 추적 메시지는 MQFB_NOT_FORWARDED 피드백과 함께 거부됩니다.
 - *Forward*가 MQRROUTE_FORWARD_ALL로 지정되는 경우, 라우트 추적 메시지는 리모트 큐 관리자로 전달될 수 있습니다.

Deliver

라우트 추적 메시지가 그의 의도된 목적지에 도달하는 경우 취할 조치를 지정합니다. 사용자 작성 애플리케이션이 라우트 추적 메시지를 그의 대상 큐에 위치시키기 전에 이 속성을 확인해야 합니다. 가능한 값은 다음 값 중 하나입니다.

MQRROUTE_DELIVER_YES

라우트 추적 메시지가 도달하면 대상 큐에 메시지를 넣습니다. 대상 큐에 Get 조작을 수행하는 모든 애플리케이션이 라우트 추적 메시지를 검색할 수 있습니다.

MQRROUTE_DELIVER_NO

도착 시, 라우트 추적 메시지가 대상 큐로 전달되지 않습니다. 해당 보고서 옵션에 따라 메시지가 처리됩니다.

라우트 추적 응답 메시지에 대한 공용 큐 설정

보고서가 로컬 시스템 큐로 전달될 때 특정 메시지와 관련된 라우트 추적 응답 메시지의 위치를 판별하려면 단일 노드에서 공용 큐를 사용하는 것이 더 효율적입니다.

시작하기 전에

큐 관리자를 라우트 추적 메시징에 대해 사용 가능하게 하고 생성된 모든 라우트 추적 응답 메시지가 로컬 시스템 큐 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE로 전달되도록 지정하려면 **ROUTEREC** 매개변수를 설정하십시오.

이 태스크 정보

큐 관리자 네트워크의 많은 큐 관리자가 로컬 시스템 큐로 라우트 추적 응답 메시지를 전달하도록 설정되는 경우, 특정 메시지와 관련된 라우트 추적 응답 메시지의 위치를 판별하는 것은 시간 소비적일 수 있습니다. 그렇지 않으면 공용 큐를 호스팅하는 큐 관리자인 단일 노드를 사용하십시오. 큐 관리자 네트워크의 모든 큐 관리자가 이 공용 큐로 라우트 추적 응답 메시지를 전달할 수 있습니다. 공용 큐 사용의 이점은 큐 관리자가 메시지에 지정된 응답 대상 큐로 라우트 추적 응답 메시지를 전달할 필요가 없고 메시지와 관련된 라우트 추적 응답 메시지의 위치를 판별할 때 하나의 큐만 조회한다는 점입니다.

공용 큐를 설정하려면 다음 단계를 수행하십시오.

프로시저

1. 큐 관리자를 단일 노드로 선택 또는 정의하십시오.
2. 단일 노드에서 공용 큐에 사용할 큐를 선택 또는 정의하십시오.

3. 라우트 추적 응답 메시지를 공용 큐로 전달하는 모든 큐 관리자에서 로컬 시스템 큐 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE를 리모트 큐 정의로 다시 정의하십시오.
 - a) 단일 노드의 이름을 리모트 큐 관리자 이름으로 지정하십시오.
 - b) 공용 큐의 이름을 리모트 큐 이름으로 지정하십시오.

기록된 정보 획득 및 사용

라우트 추적 메시지에 대한 기록된 활동 정보를 획득하려면 다음 기술 중 하나를 사용하십시오.

활동 정보가 획득되지 않는 상황이 라우트 추적 응답 메시지에도 적용됨을 참고하십시오.

라우트 추적 메시지가 활동 기록 및 라우트 추적 메시징 모두에 대해 사용 불가능한 큐 관리자에 의해 처리될 때는 활동 정보가 기록되지 않습니다.

라우트 추적 응답 메시지에서 정보 획득
활동 정보를 획득하려면 라우트 추적 응답 메시지를 찾습니다. 그런 다음 메시지를 검색하고 활동 정보를 분석하십시오.

이 태스크 정보

라우트 추적 응답 메시지의 위치를 아는 경우에만 라우트 추적 응답 메시지에서부터 활동 정보를 획득할 수 있습니다. 다음과 같이 메시지를 찾고 활동 정보를 처리하십시오.

프로시저

1. 라우트 추적 메시지의 메시지 디스크립터에서 지정된 응답 대상 큐를 확인하십시오. 라우트 추적 응답 메시지가 응답 대상 큐에 없는 경우 다음 위치를 검사하십시오.
 - 라우트 추적 메시지의 대상 큐 관리자에 있는 로컬 시스템 큐 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
 - 라우트 추적 응답 메시지에 대한 공용 큐를 설정한 경우, 해당 공용 큐
 - 큐 관리자 네트워크에 있는 다른 모든 큐 관리자의 로컬 시스템 큐 SYSTEM.ADMIN.TRACE.ROUTE.QUEUE. 이것은 라우트 추적 메시지가 데드-레터 큐에 넣어졌거나 최대 활동 수를 초과한 경우에 발생할 수 있습니다.
2. 라우트 추적 응답 메시지를 검색하십시오.
3. IBM MQ 표시 라우트 애플리케이션을 사용하여 기록된 활동 정보를 표시하십시오.
4. 활동 정보를 연구하여 필요한 정보를 얻으십시오.

라우트 추적 메시지에서 정보 획득
활동 정보를 획득하려면 라우트 추적 메시지를 찾는데, 이것은 *TraceRoute* PCF 그룹에서 적절한 매개변수를 가져야 합니다. 그런 다음 메시지를 검색하고 활동 정보를 분석하십시오.

이 태스크 정보

라우트 추적 메시지의 위치를 알고 있고 *TraceRoute* PCF 그룹의 *Accumulate* 매개변수가 MQRROUTE_ACCUMULATE_IN_MSG 또는 MQRROUTE_ACCUMULATE_AND_REPLY로 지정된 경우에만 라우트 추적 메시지에서 활동 정보를 획득할 수 있습니다.

라우트 추적 메시지가 대상 큐로 전달되기 위해서는 *TraceRoute* PCF 그룹의 *Deliver* 매개변수가 MQRROUTE_DELIVER_YES로 지정되어야 합니다.

프로시저

1. 대상 큐를 검사하십시오. 라우트 추적 메시지가 대상 큐에 없는 경우, 활동 기록에 대해 사용 가능한 라우트 추적 메시지를 사용하여 라우트 추적 메시지를 찾으려고 시도할 수 있습니다. 생성된 활동 보고서를 사용하여 라우트 추적 메시지의 마지막으로 알려진 위치를 판별하십시오.
2. 라우트 추적 메시지를 검색하십시오.
3. IBM MQ 표시 라우트 애플리케이션을 사용하여 기록된 활동 정보를 표시하십시오.
4. 활동 정보를 연구하여 필요한 정보를 얻으십시오.

활동 보고서에서 정보 획득

활동 정보를 획득하려면 활동 보고서를 찾습니다. 이 활동 보고서는 메시지 디스크립터에서 지정되는 보고서 옵션을 가져야 합니다. 그런 다음 활동 보고서를 검색하고 활동 정보를 분석하십시오.

이 태스크 정보

활동 보고서의 위치를 알고 있고 보고서 옵션 MQRO_ACTIVITY가 라우트 추적 메시지의 메시지 디스크립터에서 지정된 경우에만 활동 보고서에서 활동 정보를 획득할 수 있습니다.

프로시저

1. 라우트 추적 메시지에 대해 생성된 활동 보고서를 찾아서 정렬하십시오.
활동 보고서를 찾았을 때 수동으로 정렬하거나 IBM MQ 표시 라우트 애플리케이션을 사용하여 활동 정보를 자동으로 정렬하고 표시할 수 있습니다.
2. 활동 정보를 연구하여 필요한 정보를 얻으십시오.

추가 활동 정보

라우트 추적 메시지가 큐 관리자 네트워크를 통해 라우트될 때, 사용자 애플리케이션은 라우트 추적 메시지 또는 활동 보고서의 메시지 데이터에 *Activity* 그룹을 쓸 때 하나 이상의 추가 PCF 매개변수를 포함하여 추가 정보를 기록할 수 있습니다.

추가 활동 정보는 시스템 관리자가 라우트 추적 메시지가 취한 라우트 또는 해당 라우트를 사용한 이유를 식별하는 데 도움이 될 수 있습니다.

IBM MQ 표시 라우트 애플리케이션을 사용하여 라우트 추적 메시지에 대해 기록된 정보를 표시하는 경우, 각 매개변수의 매개변수 ID가 IBM MQ 표시 라우트 애플리케이션에 의해 인식되지 않는 경우 모든 추가 PCF 매개변수는 숫자 ID만을 갖고 표시될 수 있습니다. 매개변수 ID를 인식하려면 추가 정보가 다음 PCF 매개변수를 사용하여 기록되어야 합니다. 이들 PCF 매개변수를 *Activity* PCF 그룹의 적절한 위치에 포함하십시오.

GroupName

표 13. 그룹 이름	
설명	추가 정보를 지정하는 그룹화 매개변수
ID	MQGACF_VALUE_NAMING.
데이터 유형	MQCFGR
그룹의 매개변수	<i>ParameterName</i> <i>ParameterValue</i>

ParameterName

표 14. 매개변수 이름	
설명	<i>ParameterValue</i> 의 값을 컨텍스트에 넣는 IBM MQ 표시 라우트 애플리케이션이 표시할 이름을 포함합니다.
ID	MQCA_VALUE_NAME.
데이터 유형	MQCFST
PCF 그룹에 포함:	<i>GroupName</i> .
값:	표시될 이름.

ParameterValue

표 15. 매개변수 값	
설명	IBM MQ 표시 라우트 애플리케이션에 의해 표시될 값을 포함합니다.
ID:	추가 정보에 대한 PCF 구조 ID.
데이터 유형:	추가 정보에 대한 PCF 구조 데이터 유형.
PCF 그룹에 포함:	GroupName.
값:	표시될 값.

추가 활동 정보 기록의 예

다음 예는 사용자 애플리케이션이 라우트 추적 메시지를 위해 활동을 수행할 때 추가 정보를 기록할 수 있는 방법을 설명합니다. 두 예 모두에서, IBM MQ 표시 라우트 애플리케이션이 라우트 추적 메시지를 생성하고 리턴되는 활동 정보를 표시하는 데 사용됩니다.

추가 활동 정보 기록: 예 1

추가 활동 정보는 매개변수 ID가 IBM MQ 표시 라우트 애플리케이션에 의해 인식되지 않는 형식으로 사용자 애플리케이션에 의해 기록됩니다.

1. IBM MQ 표시 라우트 애플리케이션이 라우트 추적 메시지를 생성하고 큐 관리자 네트워크에 넣는 데 사용됩니다. 필요한 옵션이 다음을 요청하도록 설정됩니다.
 - 라우트 추적 메시지의 메시지 데이터에 활동 정보가 누적됩니다.
 - 도착 시 대상 큐에서 라우트 추적 메시지가 제거되고, 라우트 추적 응답 메시지가 생성되어 지정된 응답 대상 큐로 전달됩니다.
 - 라우트 추적 응답 메시지의 수령 시, IBM MQ 표시 라우트 애플리케이션이 누적된 활동 정보를 표시합니다. 라우트 추적 메시지는 큐 관리자 네트워크에 넣어집니다.
2. 라우트 추적 메시지가 큐 관리자 네트워크를 통해 라우팅되므로 라우트 추적 메시징에 사용되는 사용자 애플리케이션은 메시지에 관한 낮은 상세 활동을 수행합니다. 라우트 추적 메시지에 표준 활동 정보를 쓰는 것 외에, 사용자 애플리케이션은 Activity 그룹의 끝에 다음 PCF 매개변수를 씁니다.

ColorValue

```
ID
65536
데이터 유형
MQCFST
값
'Red'
```

이 추가 PCF 매개변수는 수행된 활동에 대한 추가 정보를 제공하지만 매개변수 ID가 IBM MQ 표시 라우트 애플리케이션에서 인식되지 않는 형식으로 작성됩니다.

3. 라우트 추적 메시지가 대상 큐에 도달하고 라우트 추적 응답 메시지가 IBM MQ 표시 라우트 애플리케이션에 리턴됩니다. 추가 활동 정보는 다음과 같이 표시됩니다.

```
65536: 'Red'
```

IBM MQ 표시 라우트 애플리케이션은 PCF 매개변수의 매개변수 ID를 인식하지 않으며 숫자 값으로 표시합니다. 추가 정보의 컨텍스트가 명확하지 않습니다.

IBM MQ 표시 라우트 애플리케이션이 PCF 매개변수의 매개변수 ID를 인식하는 예에 대해서는 [74 페이지의 『추가 활동 정보 기록: 예 2』](#)의 내용을 참조하십시오.

추가 활동 정보 기록: 예 2

추가 활동 정보는 사용자 애플리케이션을 통해 매개변수 ID가 IBM MQ 표시 라우트 애플리케이션에서 인식되는 형식으로 기록됩니다.

1. IBM MQ 표시 라우트 애플리케이션은 74 페이지의 『추가 활동 정보 기록: 예 1』 에서와 동일한 방식으로 라우트 추적 메시지를 생성하여 큐 관리자 네트워크에 이 메시지를 넣는 데 사용됩니다.
2. 라우트 추적 메시지가 큐 관리자 네트워크를 통해 라우팅되므로 라우트 추적 메시징에 사용되는 사용자 애플리케이션은 메시지에 관한 낮은 상세 활동을 수행합니다. 사용자 애플리케이션은 라우트 추적 메시지에 표준 활동 정보를 쓰는 것 외에 활동 그룹의 끝에 다음 PCF 매개변수를 씁니다.

ColorInfo

표 16. 색상 정보	
설명	색상에 대한 정보를 지정하는 그룹화된 매개변수.
ID:	MQGACF_VALUE_NAMING.
데이터 유형:	MQCFGR.
그룹의 매개변수:	ColorName ColorValue

ColorName

표 17. 색상 이름	
설명	ColorValue 값을 컨텍스트에 넣는 IBM MQ 표시 라우트 애플리케이션이 표시할 이름을 포함합니다.
ID:	MQCA_VALUE_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	ColorInfo.
값:	'Color'

ColorValue

표 18. 색상 값	
설명	IBM MQ 표시 라우트 애플리케이션에 의해 표시될 값을 포함합니다.
ID:	65536.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	ColorInfo.
값:	'Red'


이러한 추가 PCF 매개변수는 수행된 활동에 대한 추가 정보를 제공합니다. 이 PCF 매개변수는 매개변수 ID가 IBM MQ 표시 라우트 애플리케이션에서 인식되는 형식으로 기록됩니다.

3. 라우트 추적 메시지가 대상 큐에 도달하고 라우트 추적 응답 메시지가 IBM MQ 표시 라우트 애플리케이션에 리턴됩니다. 추가 활동 정보는 다음과 같이 표시됩니다.

```
Color: 'Red'
```

IBM MQ 표시 라우트 애플리케이션은 추가 활동 정보의 값을 포함한 PCF 구조의 매개변수 ID에 해당 이름이 있음을 인식합니다. 숫자 값 대신 해당 이름이 표시됩니다.

IBM MQ 표시 라우트 애플리케이션

IBM MQ 표시 라우트 애플리케이션(**dspmqrte**)을 사용하여 명령행 인터페이스를 통해 라우트 추적 메시지 및 라우트 추적 메시지에 관련된 활동 정보에 대해 작업합니다.  IBM MQ 표시 라우트 애플리케이션

은 IBM MQ for z/OS에서 제공되지 않지만 분산 설치에서 실행하고 **dspmqrte** 명령을 실행할 때 **-c** 매개변수를 지정하여 IBM MQ for z/OS 큐 관리자에 클라이언트로 연결할 수 있습니다.

IBM MQ [dspmqrte](#) 라우트 표시 애플리케이션은 다음 용도로 사용할 수 있습니다.

- 라우트 추적 메시지를 구성 및 생성하여 큐 관리자 네트워크에 넣으려는 경우.

라우트 추적 메시지를 큐 관리자 네트워크에 넣어서 활동 정보를 수집하고 이를 사용하여 라우트 추적 메시지가 이용한 라우트를 판별할 수 있습니다. 다음과 같이 라우트 추적 메시지의 특성을 지정할 수 있습니다.

- 라우트 추적 메시지의 목적지.
- 라우트 추적 메시지가 다른 메시지를 가장하는 방식.
- 큐 관리자 네트워크를 통해 라우트할 때 라우트 추적 메시지의 핸들링 방식.
- 활동 정보를 기록하기 위해 활동 기록 또는 라우트 추적 메시지를 사용하는지 여부.

- 라우트 추적 메시지에 관련된 활동 정보의 순서화 및 표시.

IBM MQ 표시 라우트 애플리케이션이 라우트 추적 메시지를 큐 관리자 네트워크에 넣은 경우에는 관련된 활동 정보가 리턴된 직후에 정보를 순서화하고 표시할 수 있습니다. 또는 IBM MQ 표시 라우트 애플리케이션을 사용하여 이전에 생성된 라우트 추적 메시지에 관련한 활동 정보를 순서화하고 표시할 수 있습니다.

관련 참조

[dspmqrte](#)

라우트 추적 메시지에 대한 매개변수

이 페이지는 IBM MQ 표시 라우트 애플리케이션, **dspmqrte**가 큐 관리자 네트워크를 통해 라우팅되는 라우트 추적 메시지의 처리 방식을 포함한 특성을 판별하기 위해 제공하는 매개변수를 개괄합니다.

관련 참조

[dspmqrte](#)

큐 관리자 연결

이 페이지를 사용하여 IBM MQ 표시 라우트 애플리케이션이 연결하는 큐 관리자를 지정합니다.

-c

IBM MQ 표시 라우트 애플리케이션이 클라이언트 애플리케이션으로 연결하도록 지정합니다.

이 매개변수를 지정하지 않으면 IBM MQ 표시 라우트 애플리케이션이 클라이언트 애플리케이션으로 연결하지 않습니다.

-m QMgrName

IBM MQ 표시 라우트 애플리케이션에서 연결하는 큐 관리자의 이름. 이름은 최대 48자입니다.

이 매개변수를 지정하지 않으면 기본 큐 관리자가 사용됩니다.

대상 목적지

이 페이지를 사용하여 라우트 추적 메시지의 대상 목적지를 지정합니다.

-q TargetQName

IBM MQ 표시 라우트 애플리케이션을 사용하여 라우트 추적 메시지를 큐 관리자 네트워크에 송신하는 경우 *TargetQName*이 대상 큐의 이름을 지정합니다.

-ts TargetTopicString

토픽 문자열을 지정합니다.

-qm TargetQMgr

대상 목적지를 규정하며 그러면 일반 큐 관리자 이름 분석이 적용됩니다. 대상 목적지는 *-q TargetQName* 또는 *-ts TargetTopicString* 로 지정됩니다.

이 매개변수를 지정하지 않으면 IBM MQ 표시 라우트 애플리케이션이 연결된 큐 관리자가 대상 큐 관리자로 사용됩니다.

-o

대상 목적지가 지정된 목적지로 바인딩되지 않음을 지정합니다. 일반적으로 이 매개변수는 라우트 추적 메시지를 클러스터에 넣을 때 사용합니다. 대상 목적지는 MQOO_BIND_NOT_FIXED 옵션으로 열립니다.

이 매개변수를 지정하지 않을 경우 대상 목적지가 특정 목적지로 바인딩됩니다.

발행 토픽

발행/구독 애플리케이션의 경우 이 페이지를 사용하여 발행할 IBM MQ 표시 라우트 애플리케이션에 대한 라우트 추적 메시지의 토픽 문자열을 지정합니다.

-ts TopicName

IBM MQ 표시 라우팅 애플리케이션이 라우트 추적 메시지를 발행할 토픽 문자열을 지정하고 이 애플리케이션을 토픽 모드에 넣습니다. 이 모드에서 애플리케이션은 발행 요청에서 생성된 모든 메시지를 추적합니다.

IBM MQ 표시 라우트 애플리케이션을 사용하여 발행 메시지에 대해 생성된 활동 보고서의 결과를 표시할 수도 있습니다.

메시지 가장

이 페이지를 사용하여 예를 들어, 원래 메시지가 의도한 목적지에 도달하지 않았을 때 메시지를 가장하도록 라우트 추적 메시지를 구성합니다.

라우트 추적 메시징의 한 가지 용도는 의도한 목적지에 도달하지 않은 메시지의 마지막 알려진 위치를 판별하도록 돕는 것입니다. IBM MQ 표시 라우트 애플리케이션은 원래 메시지를 가장하도록 라우트 추적 메시지를 구성하는 데 유용한 매개변수를 제공합니다. 메시지를 가장할 때 다음 매개변수를 사용할 수 있습니다.

-l Persistence

생성된 라우트 추적 메시지의 지속성을 지정합니다. *Persistence*에 가능한 값은 다음과 같습니다.

yes

생성된 라우트 추적 메시지가 지속적입니다. (MQPER_PERSISTENT).

아니오

생성된 라우트 추적 메시지가 지속적이지 않습니다. (MQPER_NOT_PERSISTENT).

q

생성된 라우트 추적 메시지가 *-q TargetQName* 또는 *-ts TargetTopicString*에 지정된 목적지로부터 지속 값을 상속합니다. (MQPER_PERSISTENCE_AS_Q_DEF).

리턴된 라우트 추적 응답 메시지 또는 보고 메시지도 원래 라우트 추적 메시지와 동일한 지속 값을 공유합니다.

*Persistence*가 예로 지정되면 *-rq ReplyToQ* 매개변수를 지정해야 합니다. 응답 대상 큐를 임시 동적 큐로 해석하지 않아야 합니다.

이 매개변수를 지정하지 않을 경우 생성된 라우트 추적 메시지는 지속적이지 않습니다.

-p Priority

라우트 추적 메시지의 우선순위를 지정합니다. *Priority*의 값은 0 이상이거나 MQPRI_PRIORITY_AS_Q_DEF입니다. MQPRI_PRIORITY_AS_Q_DEF는 *-q TargetQName* 또는 *-ts TargetTopicString*에 지정된 목적지에서 우선순위 값을 가져옴을 지정합니다.

이 매개변수를 지정하지 않으면 *-q TargetQName* 또는 *-ts TargetTopicString*에 지정된 목적지에서 우선순위 값을 가져옵니다.

-xs Expiry

라우트 추적 메시지의 만료 시간을 초 단위로 지정합니다.

이 매개변수를 지정하지 않으면 만기 시간이 60초로 지정됩니다.

-ro none |ReportOption

none

보고서 옵션이 설정되지 않도록 지정합니다.

ReportOption

라우트 추적 메시지에 대한 보고서 옵션을 지정합니다. 심표를 구분 기호로 사용하여 다중 보고서 옵션을 지정할 수 있습니다. *ReportOption*에 가능한 값은 다음과 같습니다.

활동

보고서 옵션 MQRO_ACTIVITY가 설정됩니다.

coa

보고서 옵션 MQRO_COA_WITH_FULL_DATA가 설정됩니다.

cod

보고서 옵션 MQRO_COD_WITH_FULL_DATA가 설정됩니다.

예외

보고서 옵션 MQRO_EXCEPTION_WITH_FULL_DATA가 설정됩니다.

expiration

보고서 옵션 MQRO_EXPIRATION_WITH_FULL_DATA가 설정됩니다.

discard

보고서 옵션 MQRO_DISCARD_MSG가 설정됩니다.

`-ro ReportOption` 또는 `-ro none` 가 지정되지 않은 경우 MQRO_ACTIVITY 및 MQRO_DISCARD_MSG 보고서 옵션이 지정됩니다.

IBM MQ 표시 라우트 애플리케이션은 라우트 추적 메시지에 사용자 데이터를 추가하도록 허용하지 않습니다. 라우트 추적 메시지에 사용자 데이터를 추가해야 하는 경우에는 라우트 추적 메시지를 수동으로 생성해야 합니다.

기록된 활동 정보

이 페이지를 사용하여 기록된 활동 정보를 리턴하는 데 사용되는 메소드를 지정한 후 이 메소드를 사용하여 라우트 추적 메시지가 이용한 라우트를 판별할 수 있습니다.

기록된 활동 정보는 다음과 같이 리턴될 수 있습니다.

- 활동 보고서에
- 라우트 추적 응답 메시지에
- 라우트 추적 메시지 자체에(대상 큐에 넣음)

`dspmqrte`를 사용할 때에는 기록된 활동 정보를 리턴하는 데 사용된 메소드가 다음 매개변수를 사용하여 판별됩니다.

-ro activity

활동 정보가 활동 보고서를 사용하여 리턴됨을 지정합니다. 기본적으로 활동 기록이 사용으로 설정됩니다.

-ac -ar

활동 정보가 라우트 추적 메시지에 누적되고 라우트 추적 응답 메시지가 생성됨을 지정합니다.

-ac

활동 정보가 라우트 추적 메시지 내에 누적되도록 지정합니다.

이 매개변수를 지정하지 않으면 활동 정보가 라우트 추적 메시지 내에 누적되지 않습니다.

-ar

다음 상황에서 모든 누적 활동 정보가 포함된 라우트 추적 응답 메시지가 생성되도록 요청합니다.

- IBM MQ 큐 관리자가 라우트 추적 메시지를 제거합니다.
- IBM MQ 큐 관리자가 라우트 추적 메시지를 로컬 큐(대상 큐나 데드-레터 큐)에 넣습니다.
- 라우트 추적 메시지에 수행한 활동의 수가 `-s Activities`에 지정된 값을 초과합니다.

-ac -d yes

활동 정보가 라우트 추적 메시지에 누적되고 라우트 추적 메시지가 도달하면 대상 큐에 메시지를 넣음을 지정합니다.

-ac

활동 정보가 라우트 추적 메시지 내에 누적되도록 지정합니다.

이 매개변수를 지정하지 않으면 활동 정보가 라우트 추적 메시지 내에 누적되지 않습니다.

-d yes

큐 관리자가 라우트 추적 메시지를 지원하지 않는 경우에도 도달 시, 라우트 추적 메시지가 대상 큐로 들어갑니다.

이 매개변수를 지정하지 않으면 라우트 추적 메시지를 대상 큐에 넣지 않습니다.

그런 다음 대상 큐에서 라우트 추적 메시지를 검색하여 기록된 활동 정보를 얻을 수 있습니다.

필요에 따라 다음 메소드를 결합할 수 있습니다.

또한 다음 매개변수를 사용하여 기록된 활동 정보의 세부사항 레벨을 지정할 수 있습니다.

-t Detail

기록한 활동을 지정합니다. *Detail*에 가능한 값은 다음과 같습니다.

낮음

사용자 정의 애플리케이션이 수행한 활동만 기록됩니다.

중간

낮음에 지정된 활동이 기록됩니다. 또한 발행 활동과 MCA가 수행한 활동이 기록됩니다.

높음

낮음 및 중간에 지정된 활동이 기록됩니다. MCA는 이 레벨의 세부사항에서 추가 활동 정보를 표시하지 않습니다. 이 옵션은 추가 활동 정보만 표시할 사용자 정의 애플리케이션에 사용 가능합니다. 예를 들어, 사용자 정의 애플리케이션이 특정 메시지 특성을 고려하여 메시지가 이용하는 라우트를 판별하면 라우팅 논리가 이 세부사항 레벨에 포함될 수 있습니다.

이 매개변수를 지정하지 않으면 중간 레벨 활동이 기록됩니다.

기본적으로 IBM MQ 표시 라우트 애플리케이션은 임시 동적 큐를 사용하여 리턴된 메시지를 저장합니다. IBM MQ 표시 라우트 애플리케이션이 종료되면 임시 동적 큐가 닫히고 메시지가 제거됩니다. IBM MQ 표시 라우트 애플리케이션의 현재 실행이 종료되어도 리턴된 메시지가 필요하다면 다음 매개변수를 사용하여 영구적 큐를 지정해야 합니다.

-rq ReplyToQ

라우트 추적 메시지에 대해 모든 응답이 송신되는 응답 대상 큐의 이름을 지정합니다. 라우트 추적 메시지가 지속적이거나 *-n* 매개변수가 지정된 경우, 임시 동적 큐가 아닌 응답 대상 큐가 지정되어야 합니다.

이 매개변수를 지정하지 않으면 시스템 기본 모델 큐인 SYSTEM.DEFAULT.MODEL.QUEUE를 사용하여 동적 응답 대상 큐가 작성됩니다.

-rqm ReplyToQMgr

응답 대상 큐가 상주하는 큐 관리자의 이름을 지정합니다. 이름은 최대 48자입니다.

이 매개변수를 지정하지 않으면 IBM MQ 표시 라우트 애플리케이션이 연결된 큐 관리자를 응답 대상 큐 관리자 로 사용합니다.

라우트 추적 메시지의 핸들링 방식

이 페이지를 사용하여 큐 관리자 네트워크를 통해 라우팅되는 라우트 추적 메시지를 어떻게 핸들링할지 제어합니다.

다음 매개변수는 큐 관리자 네트워크에서 라우트 추적 메시지를 라우팅할 수 있는 위치를 제한할 수 있습니다.

-d Deliver

라우트 추적 메시지가 도착 시 대상 큐로 전달되는지 여부를 지정합니다. *Deliver*에 가능한 값은 다음과 같습니다.

yes

큐 관리자가 라우트 추적 메시징을 지원하지 않는 경우에도 도달 시, 라우트 추적 메시지가 대상 큐로 들어갑니다.

아니오

도달 시 라우트 추적 메시지를 대상 큐로 넣지 않습니다.

이 매개변수를 지정하지 않으면 라우트 추적 메시지를 대상 큐에 넣지 않습니다.

-f Forward

라우트 추적 메시지가 전달될 수 있는 큐 관리자의 유형을 지정합니다. 큐 관리자가 리모트 큐 관리자에 메시지를 전달할지 여부를 판별하기 위해 사용하는 알고리즘 세부사항은 68 페이지의 『[TraceRoute PCF 그룹](#)』의 내용을 참조하십시오. *Forward*에 가능한 값은 다음과 같습니다.

모두

라우트 추적 메시지가 모든 큐 관리자에 전달됩니다.

경고: IBM WebSphere MQ 6.0 이전의 큐 관리자로 전달되는 경우에는 라우트 추적 메시지가 인식되지 않으며 *-d Deliver* 매개변수의 값에 상관없이 로컬 큐로 전달될 수 있습니다.

지원됨

TraceRoute PCF 그룹의 *Deliver* 매개변수를 존중할 큐 관리자에만 라우트 추적 메시지가 전달됩니다.

이 매개변수를 지정하지 않으면 *Deliver* 매개변수를 존중할 큐 관리자에만 라우트 추적 메시지가 전달됩니다.

다음 매개변수는 라우트 추적 메시지가 큐 관리자 네트워크에 무제한으로 남아 있지 못하게 할 수 있습니다.

-s *Activities*

라우트 추적 메시지가 제거되기 전에 이를 대신하여 수행될 수 있는 기록된 최대 활동 수를 지정합니다. 이는 라우트 추적 메시지가 무한 루프에 빠졌을 때 무제한으로 전달되지 않게 합니다. *Activities*의 값이 1 이상이거나 *MQROUTE_UNLIMITED_ACTIVITIES*입니다. *MQROUTE_UNLIMITED_ACTIVITIES*는 라우트 추적 메시지를 대신하여 무제한 수의 활동이 수행될 수 있음을 지정합니다.

이 매개변수를 지정하지 않으면 라우트 추적 메시지를 대신하여 무제한 수의 활동이 수행될 수 있습니다.

-xs *Expiry*

라우트 추적 메시지의 만료 시간을 초 단위로 지정합니다.

이 매개변수를 지정하지 않으면 만기 시간이 60초로 지정됩니다.

-xp *PassExpiry*

라우트 추적 메시지의 만기 시간의 라우트 추적 응답 메시지에 전달되는지 여부를 지정합니다. *PassExpiry*에 가능한 값은 다음과 같습니다.

yes

보고서 옵션 *MQRO_PASS_DISCARD_AND_EXPIRY*는 라우트 추적 메시지의 메시지 디스크립터에 지정할 수 있습니다.

라우트 추적 메시지에 대해 라우트 추적 응답 메시지나 활동 보고서가 생성되면 *MQRO_DISCARD* 보고서 옵션(지정된 경우) 및 남은 만기 시간이 전달됩니다.

이는 기본값입니다.

아니오

보고서 옵션 *MQRO_PASS_DISCARD_AND_EXPIRY*가 지정되지 않습니다.

라우트 추적 메시지에 대해 라우트 추적 응답 메시지가 생성된 경우 라우트 추적 메시지의 만기 시간 및 제거 옵션이 전달되지 **않습니다**.

이 매개변수를 지정하지 않으면 *MQRO_PASS_DISCARD_AND_EXPIRY*가 지정되지 않습니다.

-ro *discard*

MQRO_DISCARD_MSG 보고서 옵션을 지정합니다. 이 옵션은 라우트 추적 메시지가 큐 관리자 네트워크에 무제한으로 남아 있지 못하게 할 수 있습니다.

활동 정보 표시

IBM MQ 표시 라우트 애플리케이션은 큐 관리자 네트워크에 방금 넣은 라우트 추적 메시지에 대한 활동 정보를 표시하거나 이전에 생성한 라우트 추적 메시지에 대한 활동 정보를 표시할 수 있습니다. 사용자 작성 애플리케이션이 기록한 추가 정보를 표시할 수도 있습니다.

라우트 추적 메시지에 대해 리턴된 활동 정보의 표시 여부를 지정하려면 다음 매개변수를 지정하십시오.

-n

라우트 추적 메시지에 대해 리턴되는 활동 정보가 표시되지 않도록 지정합니다.

이 매개변수에 라우트 추적 응답 메시지 (*-ar*) 에 대한 요청 또는 (*-ro ReportOption*) 의 보고서 생성 옵션이 수반되는 경우, *-rq ReplyToQ* 를 사용하여 특정 (비모델) 응답 대상 큐를 지정해야 합니다. 기본적으로 활동 보고서 메시지만 요청됩니다.

라우트 추적 메시지를 지정된 대상 큐에 넣고 나면 라우트 추적 메시지의 메시지 ID를 포함하여 48자의 16진 문자열이 표시됩니다. 메시지 ID는 IBM MQ 표시 라우트 애플리케이션이 *-i CorrelId* 매개변수를 사용하여 나중에 라우트 추적 메시지에 대한 활동 정보를 표시할 때 사용할 수 있습니다.

이 매개변수를 지정하지 않으면 라우트 추적 메시지에 리턴된 활동 정보가 *-v* 매개변수로 지정된 양식에 표시됩니다.

큐 관리자 네트워크에 방금 넣은 라우트 추적 메시지에 대한 활동 정보를 표시할 때에는 다음 매개변수가 지정될 수 있습니다.

-w WaitTime

활동 보고서 또는 라우트 추적 응답 메시지가 지정된 응답 대상 큐에 리턴할 때까지 IBM MQ 표시 라우트 애플리케이션이 대기할 시간(초)을 지정합니다.

이 매개변수를 지정하지 않으면 대기 시간이 라우트 추적 메시지의 만료 시간에 60초를 더한 시간으로 지정됩니다.

이전에 누적된 활동 정보를 표시할 때에는 다음 매개변수를 설정해야 합니다.

-q TargetQName

IBM MQ 표시 라우트 애플리케이션을 사용하여 이전에 수집한 활동 정보를 보는 경우 *TargetQName*이 활동 정보가 저장된 큐의 이름을 지정합니다.

-i CorrelId

IBM MQ 표시 라우트 애플리케이션을 사용하여 이전에 누적된 활동 정보만 표시하는 경우 이 매개변수를 사용합니다. *-q TargetQName*에 지정된 큐에는 많은 활동 보고서와 라우트 추적 응답 메시지가 있을 수 있습니다. *CorrelId*는 라우트 추적 메시지에 관련된 활동 보고서, 라우트 추적 응답 메시지를 식별하는 데 사용됩니다. *CorrelId*에서 원래 라우트 추적 메시지의 메시지 ID를 지정하십시오.

*CorrelId*의 형식은 48자의 16진 문자열입니다.

다음 매개변수는 이전에 누적된 활동 정보를 표시할 때에나 라우트 추적 메시지에 대한 현재 활동 정보를 표시할 때 사용할 수 있습니다.

-b

IBM MQ 표시 라우트 애플리케이션이 메시지에 관련된 라우트 추적 응답 메시지나 활동 보고서를 찾아보기만 함을 지정합니다. 이는 나중에 활동 정보를 다시 표시할 수 있도록 합니다.

이 매개변수를 지정하지 않을 경우 IBM MQ 표시 라우트 애플리케이션은 메시지에 관련된 라우트 추적 응답 메시지나 활동 보고서를 파괴적으로 가져옵니다.

-v summary | all | none | outline DisplayOption

요약

라우트 추적 메시지가 라우트된 큐가 표시됩니다.

모두

사용 가능한 모든 정보가 표시됩니다.

없음

정보가 표시되지 않습니다.

outline DisplayOption

라우트 추적 메시지의 표시 옵션을 지정합니다. 쉼표를 구분 기호로 사용하여 다중 표시 옵션을 지정할 수 있습니다.

값을 제공하지 않으면 다음 사항이 표시됩니다.

- 애플리케이션 이름
- 각 조작의 유형
- 조작별 매개변수

*DisplayOption*에 가능한 값은 다음과 같습니다.

활동

Activity PCF 그룹의 모든 비PCF 그룹 매개변수가 표시됩니다.

ID

매개변수 ID MQBACF_MSG_ID 또는 MQBACF_CORREL_ID가 있는 값이 표시됩니다. 이 값은 *msgdelta*를 대체합니다.

메시지

Message PCF 그룹의 모든 비PCF 그룹 매개변수가 표시됩니다. 이 값이 지정된 경우 *msgdelta*를 지정할 수 없습니다.

msgdelta

마지막 조작 이후 변경된 Message PCF 그룹의 모든 비PCF 그룹 매개변수가 표시됩니다. 이 값이 지정된 경우 message를 지정할 수 없습니다.

operation

Operation PCF 그룹의 모든 비PCF 그룹 매개변수가 표시됩니다.

traceroute

TraceRoute PCF 그룹의 모든 비PCF 그룹 매개변수가 표시됩니다.

이 매개변수를 지정하지 않으면 메시지 라우트의 요약이 표시됩니다.

추가 정보 표시

라우트 추적 메시지는 큐 관리자 네트워크를 통해 라우팅되므로 사용자 작성 애플리케이션이 하나 이상의 추가 PCF 매개변수를 라우트 추적 메시지의 메시지 데이터나 활동 보고서의 메시지 데이터에 씌으로써 추가 정보를 기록할 수 있습니다. IBM MQ 표시 라우트 애플리케이션이 추가 정보를 읽기 쉬운 양식으로 표시하게 하려면 73 페이지의 『추가 활동 정보』에 설명된 대로 추가 정보를 특정 형식으로 기록해야 합니다.

IBM MQ 표시 라우트 애플리케이션 예

다음 예는 IBM MQ 표시 라우트 애플리케이션의 가능한 사용 방법을 보여줍니다. 각 예마다 두 개의 큐 관리자(QM1 및 QM2)가 두 가지 채널(QM2.TO.QM1 및 QM1.TO.QM2)을 통해 서로 연결되어 있습니다.

예 1 - 활동 보고서 요청

대상 큐에 전달된 라우트 추적 메시지에서 활동 정보를 표시합니다.

이 예에서, IBM MQ 표시 라우트 애플리케이션은 큐 관리자, QM1에 연결하고 라우트 추적 메시지를 생성하여 리모트 큐 관리자 QM2의 대상 큐, TARGET.Q에 전달하는 데 사용됩니다. 라우트 추적 응답 메시지가 라우팅될 때 활동 보고서가 요청되도록 필요한 보고서 옵션이 지정됩니다. 대상 큐에 도달하면 라우트 추적 메시지가 제거됩니다. 활동 보고서를 사용하여 IBM MQ 표시 라우트 애플리케이션에 리턴된 활동 정보가 순서대로 놓이고 표시됩니다.

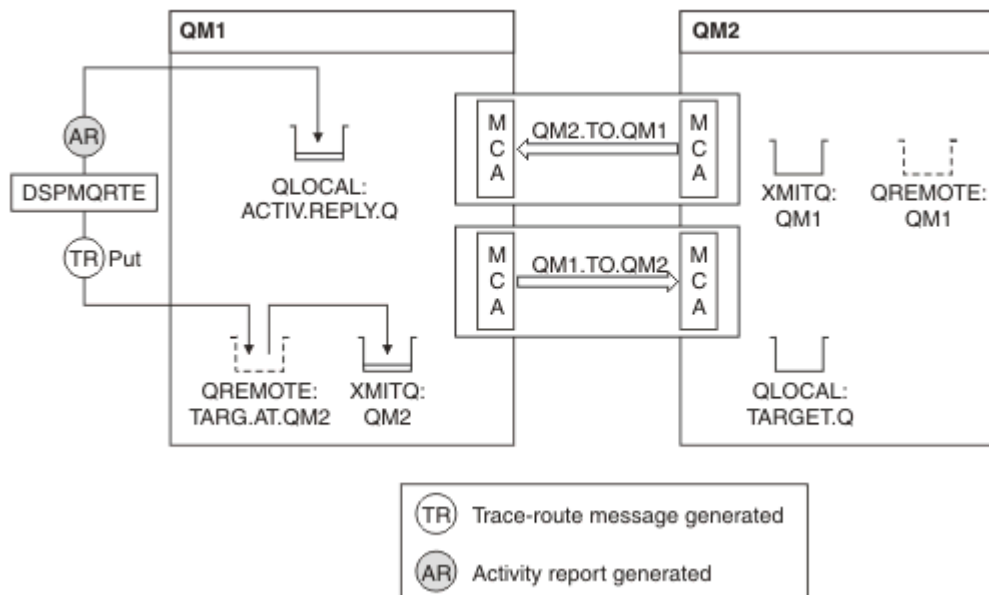


그림 9. 활동 보고서 요청, 다이어그램 1

- 각 큐 관리자(QM1 및 QM2)의 ACTIVREC 속성은 MSG로 설정됩니다.
- 다음 명령이 발행됩니다.

```
dspmqrite -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1은 IBM MQ 표시 라우트 애플리케이션이 연결하는 큐 관리자의 이름이고, TARG.AT.QM2는 대상 큐의 이름이며, ACTIV.REPLY.Q는 라우트 추적 메시지에 대한 모든 응답을 송신하도록 요청된 큐의 이름입니다.

지정되지 않은 모든 옵션은 기본값이 가정되지만 특히, -f 옵션(TraceRoute PCF 그룹의 Deliver 매개변수를 존중하는 큐 관리자에만 라우트 추적 메시지가 전달됨), -d 옵션(라우트 추적 메시지가 도달하면 대상 큐에 메시지를 넣지 않음), -ro 옵션(MQRO_ACTIVITY 및 MQRO_DISCARD_MSG 보고서 옵션이 지정됨), -t 옵션(중간 세부사항 레벨 활동이 기록됨)에 유의하십시오.

- DSPMQRTE는 라우트 추적 메시지를 생성하여 리모트 큐 TARG.AT.QM2에 넣습니다.
- DSPMQRTE는 그런 다음 큐 관리자 QM1의 ACTIVREC 속성 값을 봅니다. 값이 MSG이므로 DSPMQRTE는 활동 보고서를 생성하여 응답 큐 ACTIV.REPLY.Q에 넣습니다.

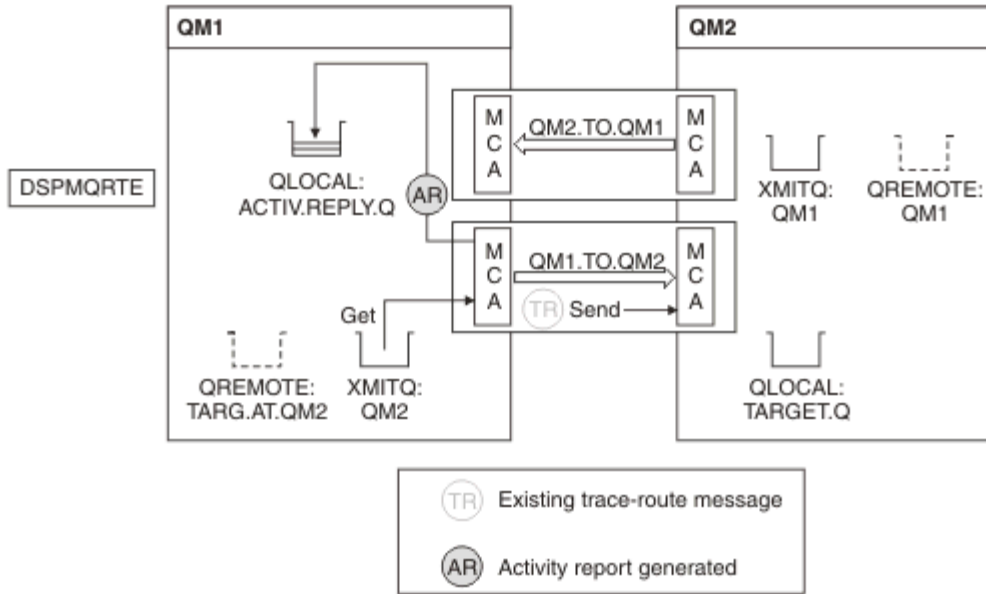


그림 10. 활동 보고서 요청, 다이어그램 2

- 송신 메시지 채널 에이전트(MCA)는 전송 큐에서 라우트 추적 메시지를 가져옵니다. 메시지가 라우트 추적 메시지이므로 MCA는 활동 정보를 기록하기 시작합니다.
- 큐 관리자(QM1)의 ACTIVREC 속성은 MSG이고 MQRO_ACTIVITY 옵션이 메시지 디스크립터의 Report 필드에 지정되어 있으므로 MCA는 활동 보고서를 나중에 생성합니다. TraceRoute PCF 그룹의 RecordedActivities 매개변수 값은 1씩 증분됩니다.
- MCA는 TraceRoute PCF 그룹의 MaxActivities 값이 초과되지 않았는지 검사합니다.
- 메시지가 QM2에 전달되기 전에 MCA는 전달(단계 70 페이지의 『1』, 71 페이지의 『4』, 71 페이지의 『5』)에 설명된 알고리즘에 따르며 메시지를 송신하도록 선택합니다.
- 그런 다음 MCA는 활동 보고서를 생성하여 응답 큐(ACTIV.REPLY.Q)에 넣습니다.

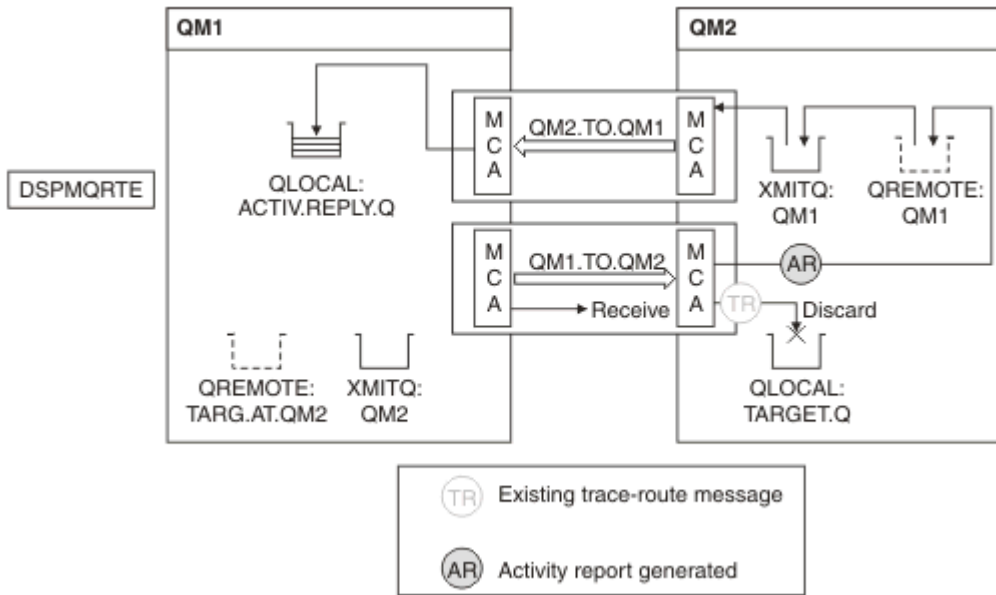


그림 11. 활동 보고서 요청, 다이어그램 3

- 수신 MCA는 채널에서 라우트 추적 메시지를 수신합니다. 메시지가 라우트 추적 메시지이므로 MCA는 활동에 대한 정보를 기록하기 시작합니다.
- 라우트 추적 메시지의 소스인 큐 관리자가 IBM WebSphere MQ 5.3.1 이전 버전인 경우 MCA는 TraceRoute PCF의 DiscontinuityCount 매개변수를 1만큼 증가시킵니다. 이 예는 여기에 해당하지 않습니다.
- 큐 관리자(QM2)의 ACTIVREC 속성은 MSG이고 MQRO_ACTIVITY 옵션이 지정되어 있으므로 MCA는 활동 보고서를 생성합니다. RecordedActivities 매개변수 값은 1씩 증분됩니다.
- 대상 큐가 로컬 큐이므로 TraceRoute PCF 그룹의 Deliver 매개변수 값에 따라 피드백 MQFB_NOT_DELIVERED와 함께 메시지가 제거됩니다.
- 그런 다음 MCA는 최종 활동 보고서를 생성하여 응답 큐에 넣습니다. 이는 큐 관리자 QM1과 연관된 전송 큐로 분석되며 큐 관리자 QM1(ACTIV.REPLY.Q)에 활동 보고서가 리턴됩니다.

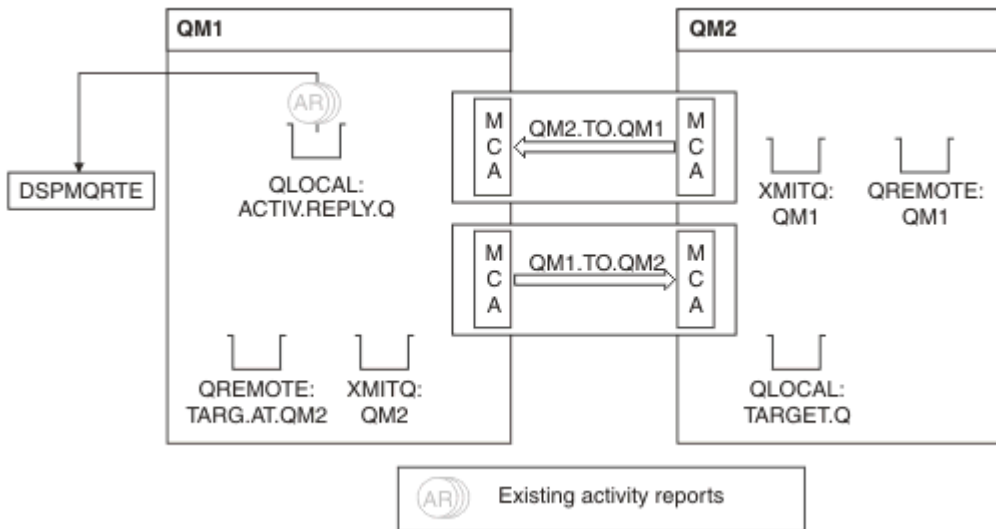


그림 12. 활동 보고서 요청, 다이어그램 4

- 그 동안, DSPMQRTE는 활동 보고서를 대기하며 응답 큐(ACTIV.REPLY.Q)에서 계속하여 MQGET을 수행했습니다. DSPMQRTE가 시작할 때 -w가 지정되지 않았으므로 최대 120초(라우트 추적 메시지의 만기 시간보다 60초 더) 동안 대기합니다.

- DSPMQRTE는 응답 큐에서 3개의 활동 보고서를 가져옵니다.
- 활동 보고서는 각 활동에 대한 TraceRoute PCF 그룹의 RecordedActivities, UnrecordedActivities, DiscontinuityCount 매개변수를 사용하여 순서화됩니다. 이 예에서 0이 아닌 유일한 값은 RecordedActivities 이므로 이 값이 실제로 사용되는 유일한 매개변수입니다.
- 제거 조작이 표시되면 프로그램이 즉시 종료합니다. 최종 조작이 제거였어도 피드백이 MQFB_NOT_DELIVERED이므로 넣기가 발생한 것처럼 처리됩니다.

표시되는 출력은 다음과 같습니다.

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

예 2 - 라우트 추적 응답 메시지 요청

라우트 추적 메시지를 생성하여 대상 큐에 전달합니다.

이 예에서, IBM MQ 표시 라우트 애플리케이션은 큐 관리자, QM1에 연결하고 라우트 추적 메시지를 생성하여 리모트 큐 관리자 QM2의 대상 큐, TARGET.Q에 전달하는 데 사용됩니다. 라우트 추적 메시지에 활동 정보가 누적 되도록 필요한 옵션이 지정됩니다. 대상 큐에 도달하면 라우트 추적 응답 메시지가 요청되고 라우트 추적 메시지는 제거됩니다.

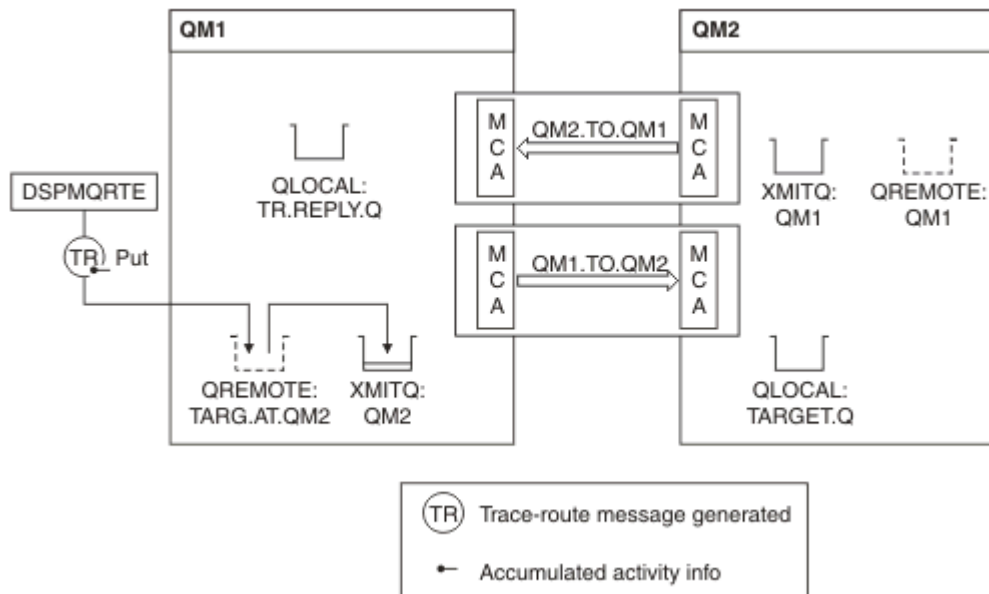


그림 13. 라우트 추적 응답 메시지 요청, 다이어그램 1

- 각 큐 관리자(QM1 및 QM2)의 ROUTEREC 속성은 MSG로 설정됩니다.
- 다음 명령이 발행됩니다.

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq TR.REPLY.Q -ac -ar -ro discard
```

QM1은 IBM MQ 표시 라우트 애플리케이션이 연결하는 큐 관리자의 이름이고, TARG.AT.QM2는 대상 큐의 이름이며, ACTIV.REPLY.Q는 라우트 추적 메시지에 대한 모든 응답을 송신하도록 요청된 큐의 이름입니다. -ac 옵션은 라우트 추적 메시지에 활동 정보가 누적되도록 지정하고, -ar 옵션은 -rq 옵션에 지정된 응답 대상 큐 (즉, TR.REPLY.Q)로 누적된 모든 활동을 송신하도록 지정합니다. -ro 옵션은 보고서 옵션 MQRO_DISCARD_MSG가 설정되도록 지정하며 이는 이 예에서 활동 보고서가 생성되지 않음을 의미합니다.

- DSPMQRTE는 대상 라우트에 메시지를 넣기 전에 라우트 추적 메시지에 활동 정보를 누적시킵니다. 이를 위해서는 큐 관리자 속성 ROUTEREC가 DISABLED이면 안됩니다.

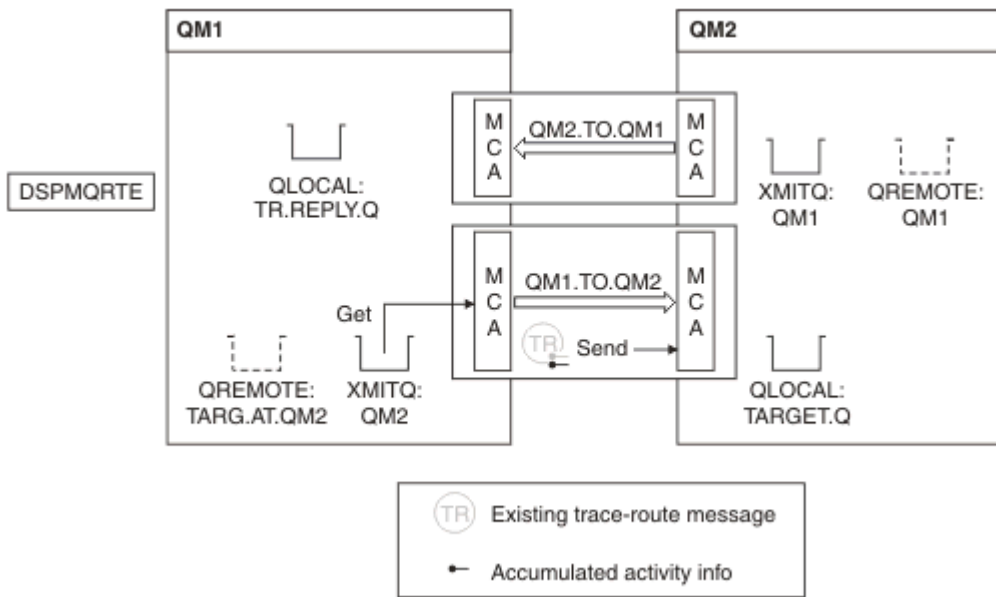


그림 14. 라우트 추적 응답 메시지 요청, 다이어그램 2

- 메시지가 라우트 추적 메시지이므로 송신 MCA는 활동에 대한 정보를 기록하기 시작합니다.
- QM1의 큐 관리자 속성 ROUTEREC가 DISABLED가 아니므로 MCA는 큐 관리자 QM2로 메시지가 전달되기 전에 활동 정보를 메시지 내에 누적시킵니다.

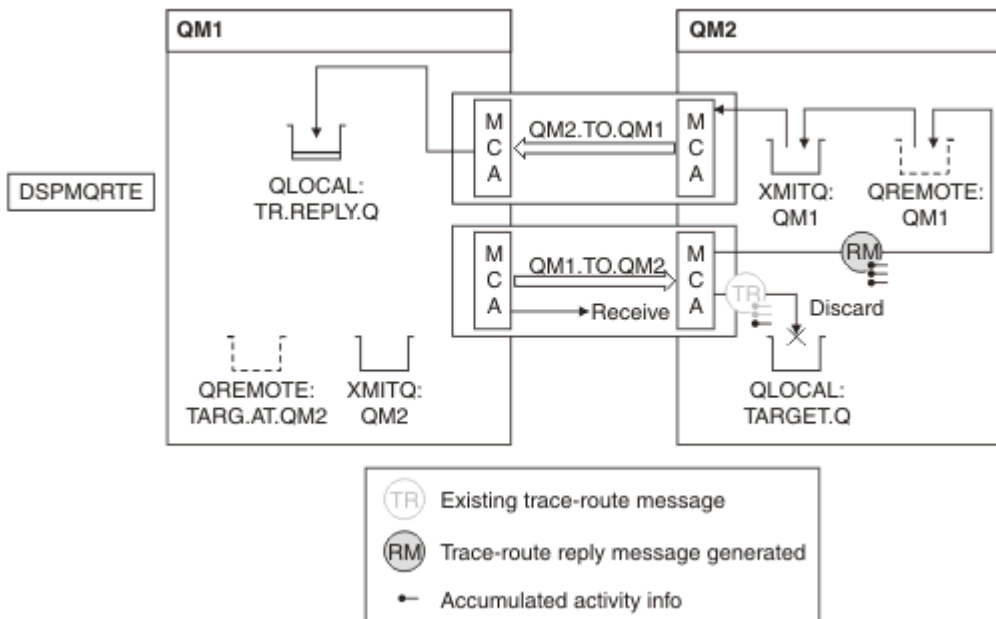


그림 15. 라우트 추적 응답 메시지 요청, 다이어그램 3

- 메시지가 라우트 추적 메시지이므로 수신 MCA는 활동에 대한 정보를 기록하기 시작합니다.
- QM2의 큐 관리자 속성 ROUTEREC가 DISABLED가 아니므로 MCA는 정보를 메시지 내에 누적시킵니다.
- 대상 큐가 로컬 큐이므로 TraceRoute PCF 그룹의 Deliver 매개변수 값에 따라 피드백 MQFB_NOT_DELIVERED와 함께 메시지가 제거됩니다.

- 이는 메시지에 대해 발생할 마지막 활동이며 QM1의 큐 관리자 속성 ROUTEREC가 DISABLED가 아니므로 MCA가 Accumulate 값에 따라 라우트 추적 응답 메시지를 생성합니다. ROUTEREC의 값이 MSG이므로 응답 메시지를 응답 큐에 넣습니다. 응답 메시지에는 라우트 추적 메시지의 누적된 모든 활동 정보가 포함됩니다.

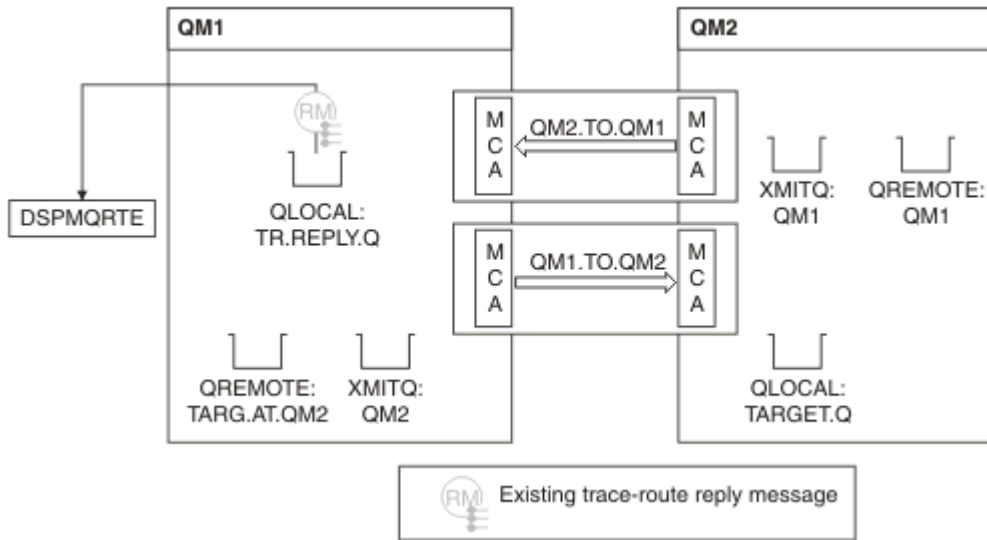


그림 16. 라우트 추적 응답 메시지 요청, 다이어그램 4

- 그 동안 DSPMQRTE는 라우트 추적 응답 메시지가 응답 큐에 리턴할 때까지 대기합니다. 메시지가 리턴되면 DSPMQRTE는 포함된 각 활동을 구문 분석하고 인쇄합니다. 최종 조작은 제거 조작입니다. 인쇄가 완료되면 DSPMQRTE가 종료합니다.

표시되는 출력은 다음과 같습니다.

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
TR.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

예 3 - 시스템 큐에 활동 보고서 전달

응답 큐가 아닌 다른 큐에 활동 보고서가 전달되는 시기를 감지하고 IBM MQ 표시 라우트 애플리케이션을 사용하여 다른 큐에서 활동 보고서를 읽습니다.

이 예는 QM2의 ACTIVREC 큐 관리자 속성 값이 이제 QUEUE로 설정된 점을 제외하면 82 페이지의 『예 1 - 활동 보고서 요청』과 동일합니다. 이를 적용하기 위해서는 채널 QM1.TO.QM2를 재시작해야 합니다.

이 예는 응답 대상 큐가 아닌 다른 큐에 활동 보고서가 전달되는 시기를 감지할 방법을 설명합니다. 일단 감지하면 IBM MQ 표시 라우트 애플리케이션을 사용하여 다른 큐에서 활동 보고서를 읽습니다.

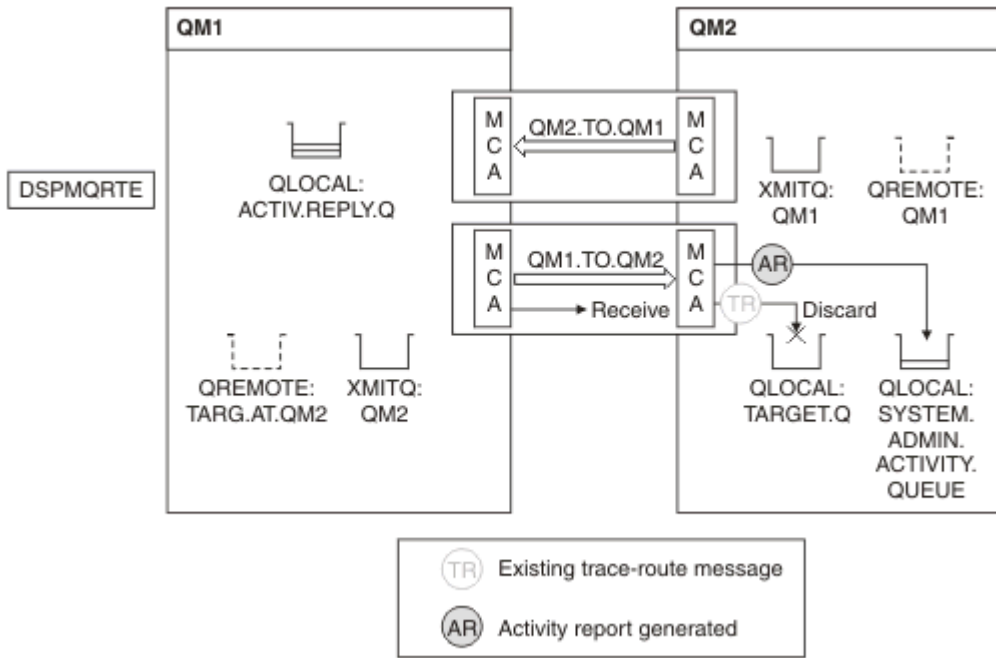


그림 17. 시스템 큐에 활동 보고서 전달, 다이어그램 1

- 메시지가 라우트 추적 메시지가므로 수신 MCA는 활동에 대한 정보를 기록하기 시작합니다.
- QM2의 ACTIVREC 큐 관리자 속성 값이 이제 QUEUE이므로 MCA는 활동 보고서를 생성하지만 응답 큐 (ACTIV.REPLY.Q)가 아닌 시스템 큐(SYSTEM.ADMIN.ACTIVITY.QUEUE)에 보고서를 넣습니다.

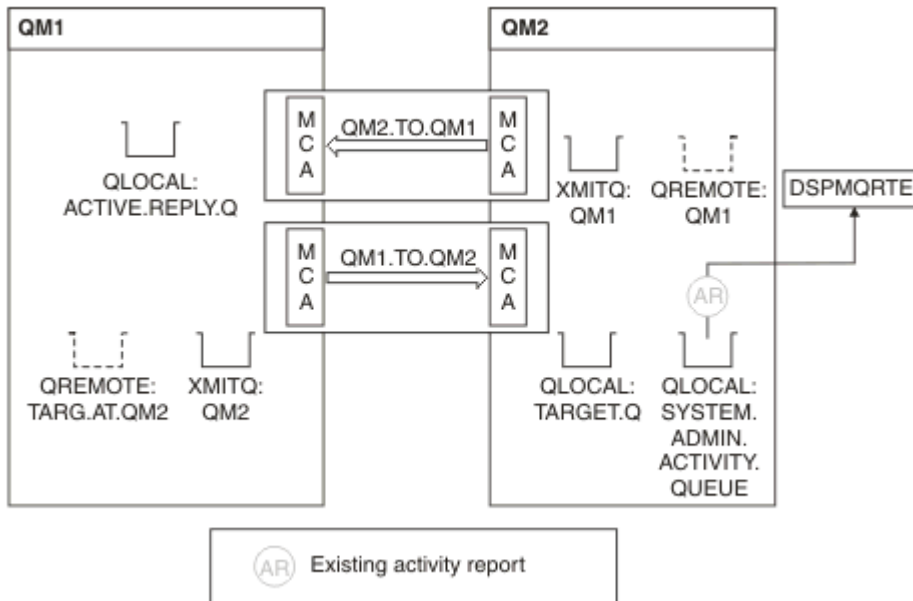


그림 18. 시스템 큐에 활동 보고서 전달, 다이어그램 2

- 그 동안 DSPMQRTE는 활동 보고서가 ACTIV.REPLY.Q에 도달하기를 대기했습니다. 두 개의 보고서만 도달합니다. DSPMQRTE는 라우트가 아직 완료되지 않은 것 같아서 계속하여 120초 동안 대기합니다.

표시되는 출력은 다음과 같습니다.

```

AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
ACTIV.REPLY.Q -v outline identifiers'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----

```

```

Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'

```

```

Operation:
  OperationType: Put

```

```

Message:

```

```

MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
-----

```

```

Activity:
  ApplName: 'cann\output\bin\runmqchl.EXE'

```

```

Operation:
  OperationType: Get

```

```

Message:

```

```

MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001505'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001502'

```

```

EmbeddedMQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'

```

```

Operation:
  OperationType: Send

```

```

Message:

```

```

MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  RemoteQMgrName: 'QM2'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Sender
  XmitQName: 'QM2'
-----

```

```

AMQ8652: DSPMQRTE command has finished.

```

- DSPMQRTE가 관찰한 마지막 조작이 송신이었으므로 채널은 실행 중입니다. 이제 우리는 큐 관리자 QM2(RemoteQMgrName에 식별된)에서 더 많은 활동 보고서를 수신하지 못한 이유를 파악해야 합니다.
- 시스템 큐에 활동 정보가 있는지 여부를 확인하기 위해 QM2에서 DSPMQRTE를 시작해서 더 많은 활동 보고서를 시도 및 수집하십시오. 다음 명령을 사용하여 DSPMQRTE를 시작하십시오.

```

dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline

```

여기서, 414D51204C41524745512020202020A3C9154220001502는 넣은 라우트 추적 메시지의 MsgId입니다.

- DSPMQRTE는 그런 다음 일련의 MQGET를 다시 수행하고 지정된 ID의 라우트 추적 메시지에 관련된 시스템 활동 큐의 응답을 대기합니다.
- DSPMQRTE는 표시된 활동 보고서를 하나 더 가져옵니다. DSPMQRTE는 이전 활동 보고서가 누락되었는지 판별하고 이를 알리는 메시지를 표시합니다. 하지만 우리는 이 라우트 파트에 대해 이미 알고 있습니다.

표시되는 출력은 다음과 같습니다.

```

AMQ8653: DSPMQRTE command started with options '-m QM2
        -q SYSTEM.ADMIN.ACTIVITY.QUEUE
        -i 414D51204C41524745512020202020A3C915420001502 -v outline'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  Activity information unavailable.
-----
Activity:
  ApplName: 'cann\output\bin\AMQRMPPA.EXE'

  Operation:
    OperationType: Receive
    QMgrName: 'QM2'
    RemoteQMGrName: 'QM1'
    ChannelName: 'QM1.TO.QM2'
    ChannelType: Receiver

  Operation:
    OperationType: Discard
    QMgrName: 'QM2'
    QName: 'TARGET.Q'
    Feedback: NotDelivered
-----
AMQ8652: DSPMQRTE command has finished.

```

- 이 활동 보고서는 라우트 정보가 이제 완료되었다고 표시합니다. 문제가 발생하지 않았습니다.
- 단지 라우트 정보가 사용 불가능하거나 DSPMQRTE가 모든 라우트를 표시할 수 없다는 이유에서, 이 사실이 메시지를 전달되지 않았음을 의미하는 것은 아닙니다. 예를 들어, 다른 큐 관리자의 큐 관리자 속성이 차이가 있거나 응답 큐가 응답을 다시 가져오도록 정의되지 않았을 수 있습니다.

예 4 - 채널 문제점 진단

라우트 추적 메시지가 대상 큐에 도달하지 않는 문제점을 진단합니다.

이 예에서, IBM MQ 표시 라우트 애플리케이션은 큐 관리자, QM1에 연결하고 라우트 추적 메시지를 생성한 후 리모트 큐 관리자 QM2의 대상 큐, TARGET.Q에 이 메시지를 전달하려 시도합니다. 이 예에서는 라우트 추적 메시지가 대상 큐에 도달하지 않습니다. 사용 가능한 활동 보고서를 사용하여 문제점을 진단합니다.

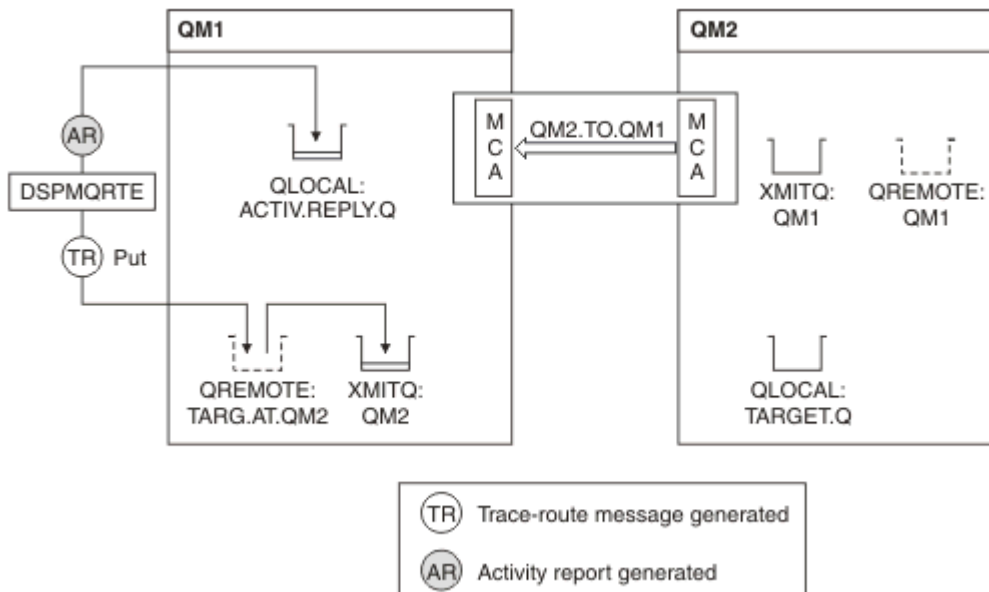


그림 19. 채널 문제점 진단

- 이 예에서는 채널 QM1.TO.QM2가 실행 중이지 않습니다.

- DSPMQRTE는 라우트 추적 메시지를 대상 큐에 넣고(예 1에서처럼) 활동 보고서를 생성합니다.
- 전송 큐(QM2)에서 메시지를 가져올 MCA가 없으므로 DSPMQRTE가 응답 큐에서 다시 가져오는 유일한 활동 보고서는 이 보고서 뿐입니다. 이번에는 라우트가 완료되지 않았다는 사실이 문제점을 표시합니다. 관리자는 ResolvedQName에 있는 전송 큐를 사용하여 전송 큐가 서비스를 제공하지 않은 원인을 조사할 수 있습니다. 표시되는 출력은 다음과 같습니다.

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

```
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'
```

```
Operation:
  OperationType: Put
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMGrName: 'QM2'
```

```
-----
AMQ8652: DSPMQRTE command has finished.
```

활동 보고서 참조

이 페이지를 사용하여 활동 보고서 메시지 형식을 개괄합니다. 활동 보고서 메시지 데이터는 활동을 설명하는 매개변수를 포함합니다.

활동 보고서 형식

활동 보고서는 메시지 디스크립터 및 메시지 데이터를 포함한 표준 IBM MQ 보고 메시지입니다. 활동 보고서는 메시지가 큐 관리자 네트워크를 통해 라우팅되었으므로 메시지에 관하여 활동을 수행한 애플리케이션이 생성하는 PCF 메시지입니다.

활동 보고서는 다음 정보를 포함합니다.

메시지 디스크립터

MQMD 구조

메시지 데이터

다음으로 구성되어 있습니다.

- 임베드된 PCF 헤더(MQEPH).
- 활동 보고서 메시지 데이터.

활동 보고서 메시지 데이터는 *Activity* PCF 그룹 및 라우트 추적 메시지에 대해 생성된 경우 *TraceRoute* PCF 그룹으로 이루어집니다.

92 페이지의 표 19은 특정 조건 하에서만 리턴되는 매개변수를 포함하여 이러한 보고서의 구조를 보여줍니다.

표 19. 활동 보고서 형식		
MQMD 구조	임베드된 PCF 헤더 MQEPH 구조	활동 보고서 메시지 데이터
구조 ID 구조 버전 보고서 옵션 메시지 유형 만기 시간 Feedback Encoding 코드화 문자 세트 ID 메시지 형식 Priority Persistence 메시지 ID 상관 ID 백아웃 수 응답 대상 큐 응답 대상 큐 관리자 사용자 ID 계정 토큰 애플리케이션 ID 데이터 애플리케이션 유형 애플리케이션 이름 넣은 날짜 넣은 시간 애플리케이션 원본 데이터 그룹 ID 메시지 순서 번호 오프셋 메시지 플래그 원래 길이	구조 ID 구조 버전 구조 길이 Encoding 코드화 문자 세트 ID 메시지 형식 플래그 PCF 헤더(MQCFH) 구조 유형 구조 길이 구조 버전 명령 ID 메시지 순서 번호 제어 옵션 완료 코드 이유 코드 매개변수 수	활동 활동 애플리케이션 이름 활동 애플리케이션 유형 활동 설명 Operation 조작 유형 조작 날짜 조작 시간 메시지 메시지 길이 MQMD ⁸ EmbeddedMQMD 큐 관리자 이름 큐 공유 그룹 이름 큐 이름 ¹ ² ³ ⁷ 해석된 큐 이름 ¹ ³ ⁷ 리모트 큐 이름 ³ ⁷ 리모크 큐 관리자 이름 ² ³ ⁴ ⁵ ⁷ 구독 레벨 ⁹ Subscription ID ⁹ 피드백 ² ¹⁰ 채널 이름 ⁴ ⁵ 채널 유형 ⁴ ⁵ 전송 큐 이름 ⁵ TraceRoute ⁶ 세부사항 기록된 활동 기록되지 않은 활동 중단 수 최대 활동 누적 전달

참고:

1. 가져오기 및 찾아보기 조작에 대해 리턴됩니다.
2. 제거 조작에 대해 리턴됩니다.
3. 넣기, 응답 넣기, 보고서 넣기 조작에 대해 리턴됩니다.
4. 수신 조작에 대해 리턴됩니다.

5. 송신 조작에 대해 리턴됩니다.
6. 라우트 추적 메시지에 대해 리턴됩니다.
7. 발행 활동 내에 포함된 토픽에 넣기 조작에 대해 리턴되지 않습니다.
8. 제외된 발행 조작에 대해 리턴되지 않습니다. 발행 및 제거된 발행 조작의 경우 매개변수의 서브셋을 포함하여 리턴됩니다.
9. 발행, 제거된 발행, 제외된 발행 조작에 대해 리턴됩니다.
10. 제거된 발행 및 제외된 발행 조작에 대해 리턴됩니다.

활동 보고서 MQMD(메시지 디스크립터)

이 페이지를 사용하여 활동 보고서의 MQMD 구조에 따라 포함된 값을 봅니다.

StrucId

구조 ID:

데이터 유형

MQCHAR4

값

MQMD_STRUC_ID.

Version

구조 버전 번호

데이터 유형

MLONG

값

원래 메시지 디스크립터에서 복사됩니다. 가능한 값은 다음과 같습니다.

MQMD_VERSION_1

모든 환경에서 지원되는 버전 1 메시지 디스크립터 구조

MQMD_VERSION_2

다음 환경에서 지원되는 버전-2 메시지 디스크립터 구조

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

및 해당 시스템에 연결된 모든 IBM MQ MQI clients.

Report

추가 보고 메시지에 대한 옵션

데이터 유형

MLONG

값

MQRO_PASS_DISCARD_AND_EXPIRY 또는 MQRO_DISCARD_MSG가 원래 메시지 디스크립터의 Report 필드에 지정된 경우:

MQRO_DISCARD

보고서를 목적지 큐에 전달할 수 없는 경우 보고서가 제거됩니다.

그렇지 않으면 다음을 수행하십시오.

MQRO_NONE

보고서가 필요하지 않습니다.

MsgType

메시지의 유형을 표시함

데이터 유형
MQLONG

값
MQMT_REPORT

Expiry

보고 메시지 수명

데이터 유형
MQLONG

값
원래 메시지 디스크립터의 *Report* 필드가 MQRO_PASS_DISCARD_AND_EXPIRY로 지정된 경우 원래 메시지의 남은 만기 시간이 사용됩니다.

그렇지 않으면 다음을 수행하십시오.

MQEI_UNLIMITED
보고서의 만기 시간이 없습니다.

Feedback

설명: 피드백 또는 이유 코드.

데이터 유형: MQLONG.

값: **MQFB_ACTIVITY**
활동 보고서.

Encoding

설명: 보고 메시지 데이터의 숫자 인코딩.

데이터 유형: MQLONG.

값: MQENC_NATIVE.

CodedCharSetId

설명: 보고 메시지 데이터의 문자 세트 ID.

데이터 유형: MQLONG.

값: 해당하는 값으로 설정됩니다.

Format

설명: 보고 메시지 데이터의 형식 이름.

데이터 유형: MQCHAR8.

값: **MQFMT_EMBEDDED_PCF**
임베드된 PCF 메시지.

Priority

설명: 보고 메시지 우선순위.

데이터 유형: MQLONG.

값: 원래 메시지 디스크립터에서 복사됩니다.

Persistence

설명: 보고 메시지 지속.

데이터 유형: MQLONG.

값: 원래 메시지 디스크립터에서 복사됩니다.

MsgId

설명: 메시지 ID.

데이터 유형: MQBYTE24.

값: 원래 메시지 디스크립터의 *Report* 필드가 MQRO_PASS_MSG_ID로 지정된 경우 원래 메시지의 메시지 ID가 사용됩니다.
그렇지 않은 경우에는 큐 관리자가 고유 값을 생성합니다.

CorrelId

설명: 상관 ID.

데이터 유형: MQBYTE24.

값: 원래 메시지 디스크립터의 *Report* 필드가 MQRO_PASS_CORREL_ID로 지정된 경우 원래 메시지의 상관 ID가 사용됩니다.
그렇지 않은 경우에는 원래 메시지에서 메시지 ID가 복사됩니다.

BackoutCount

설명: 백아웃 카운터입니다.

데이터 유형: MQLONG.

값: 0입니다.

ReplyToQ

설명: 응답 큐의 이름입니다.

데이터 유형: MQCHAR48.

값: 공백입니다.

ReplyToQMGr

설명: 응답 큐 관리자의 이름입니다.

데이터 유형: MQCHAR48.

값: 보고 메시지를 생성한 큐 관리자 이름.

UserIdentifier

설명: 보고 메시지를 생성한 애플리케이션의 사용자 ID.

데이터 유형: MQCHAR12.

값: 원래 메시지 디스크립터에서 복사됩니다.

AccountingToken

설명: 애플리케이션이 메시지의 결과로 완료된 작업에 청구할 수 있는 계정 토큰.

데이터 유형: MQBYTE32.

값: 원래 메시지 디스크립터에서 복사됩니다.

ApplIdentityData

설명: ID와 관련된 애플리케이션 데이터.

데이터 유형: MQCHAR32.
값: 원래 메시지 디스크립터에서 복사됩니다.

PutApplType

설명: 보고 메시지를 넣은 애플리케이션의 유형.
데이터 유형: MQLONG.
값: **MQAT_QMGR**
메시지를 생성한 큐 관리자.

PutApplName

설명: 보고 메시지를 넣은 애플리케이션의 이름.
데이터 유형: MQCHAR28.
값: 큐 관리자 이름 또는 보고 메시지를 생성한 MCA 이름의 처음 28바이트.

PutDate

설명: 메시지를 넣은 날짜입니다.
데이터 유형: MQCHAR8.
값: 큐 관리자가 생성한 그대로의 값입니다.

PutTime

설명: 메시지를 넣은 시간입니다.
데이터 유형: MQCHAR8.
값: 큐 관리자가 생성한 그대로의 값입니다.

ApplOriginData

설명: 원본과 관련된 애플리케이션 데이터입니다.
데이터 유형: MQCHAR4.
값: 공백입니다.

버전이 MQMD_VERSION_2이면 다음과 같은 추가 필드가 있습니다.

GroupId

설명: 실제 메시지가 속한 메시지 그룹 또는 논리 메시지를 식별합니다.
데이터 유형: MQBYTE24.
값: 원래 메시지 디스크립터에서 복사됩니다.

MsgSeqNumber

설명: 그룹 내의 논리 메시지의 순서 번호.
데이터 유형: MQLONG.
값: 원래 메시지 디스크립터에서 복사됩니다.

Offset

설명: 논리 메시지의 시작으로부터 실제 메시지의 데이터 오프셋입니다.

데이터 유형: MQLONG.
값: 원래 메시지 디스크립터에서 복사됩니다.

MsgFlags

설명: 메시지의 속성을 지정하거나 처리를 제어하는 메시지 플래그입니다.
데이터 유형: MQLONG.
값: 원래 메시지 디스크립터에서 복사됩니다.

OriginalLength

설명: 원래 메시지의 길이.
데이터 유형: MQLONG.
값: 원래 메시지 디스크립터에서 복사됩니다.

활동 보고서 MQEPH(임베드된 PCF 헤더)

이 페이지를 사용하여 활동 보고서의 MQEPH 구조에 따라 포함된 값을 봅니다.

MQEPH 구조는 활동 보고서의 메시지 데이터를 수반한 PCF 정보와 이를 따르는 애플리케이션 메시지 데이터 모두에 대한 설명을 포함합니다.

활동 보고서의 경우 MQEPH 구조가 다음 값을 포함합니다.

StrucId

설명: 구조 ID입니다.
데이터 유형: MQCHAR4.
값: MQEPH_STRUC_ID.

Version

설명: 구조 버전 번호입니다.
데이터 유형: MQLONG.
값: MQEPH_VERSION_1.

StrucLength

설명: 구조 길이.
데이터 유형: MQLONG.
값: 뒤따르는 PCF 매개변수 구조를 포함한 구조의 총 길이.

Encoding

설명: 마지막 PCF 매개변수 구조 다음에 오는 메시지 데이터의 숫자 인코딩.
데이터 유형: MQLONG.
값: 원래 애플리케이션 메시지 데이터의 데이터가 보고 메시지에 포함된 경우 원래 메시지 디스크립터의 *Encoding* 필드에서 값이 복사됩니다.
그렇지 않은 경우에는 0입니다.

CodedCharSetId

설명: 마지막 PCF 매개변수 구조 다음에 오는 메시지 데이터의 문자 세트 ID.

데이터 유형: MQLONG.
값: 원래 애플리케이션 메시지 데이터의 데이터가 보고 메시지에 포함된 경우 원래 메시지 디스크립터의 *CodedCharSetId* 필드에서 값이 복사됩니다.
그렇지 않은 경우에는 MQCCSI_UNDEFINED입니다.

Format

설명: 마지막 PCF 매개변수 구조 다음에 오는 메시지 데이터의 형식 이름.
데이터 유형: MQCHAR8.
값: 원래 애플리케이션 메시지 데이터의 데이터가 보고 메시지에 포함된 경우 원래 메시지 디스크립터의 *Format* 필드에서 값이 복사됩니다.
그렇지 않은 경우에는 MQFMT_NONE입니다.

Flags

설명: 구조의 속성을 지정하거나 처리를 제어하는 플래그.
데이터 유형: MQLONG.
값: **MQEPH_CCSSID_EMBEDDED**
문자 데이터를 포함한 매개변수의 문자 세트가 각 구조의 *CodedCharSetId* 필드 내에 개별적으로 지정됨을 지정합니다.

PCFHeader

설명: 프로그래밍 가능 명령 형식(PCF) 헤더
데이터 유형: MQCFH.
값: [98 페이지의 『활동 보고서 MQCFH\(PCF 헤더\)』](#)의 내용을 참조하십시오.

활동 보고서 MQCFH(PCF 헤더)

이 페이지를 사용하여 활동 보고서의 MQCFH 구조에 따라 포함된 PCF 값을 봅니다.

활동 보고서의 경우 MQCFH 구조가 다음 값을 포함합니다.

Type

설명: 보고서 메시지의 콘텐츠를 식별하는 구조 유형.
데이터 유형: MQLONG.
값: **MQCFT_REPORT**
메시지가 보고서입니다.

StrucLength

설명: 구조 길이.
데이터 유형: MQLONG.
값: **MQCFH_STRUC_LENGTH**
MQCFH 구조의 길이(바이트 단위).

Version

설명: 구조 버전 번호입니다.
데이터 유형: MQLONG.

값: MQCFH_VERSION_3

Command

설명: 명령 ID입니다. 메시지의 범주를 식별합니다.

데이터 유형: MQLONG.

값: **MQCMD_ACTIVITY_MSG**
메시지 활동.

MsgSeqNumber

설명: 메시지 순서 번호입니다. 관련 메시지 그룹 내의 메시지의 순서 번호입니다.

데이터 유형: MQLONG.

값: 1.

Control

설명: 제어 옵션입니다.

데이터 유형: MQLONG.

값: MQCFC_LAST.

CompCode

설명: 완료 코드.

데이터 유형: MQLONG.

값: MQCC_OK.

Reason

설명: 완료 코드를 규정하는 이유 코드입니다.

데이터 유형: MQLONG.

값: MQRC_NONE.

ParameterCount

설명: 매개변수 구조 수입니다. MQCFH 구조 뒤에 나오는 매개변수 구조의 수입니다. 그룹 구조(MQCFGR) 및 포함된 해당 매개변수 구조는 하나의 구조로만 계산됩니다.

데이터 유형: MQLONG.

값: 1 이상.

활동 보고서 메시지 데이터

이 페이지를 사용하여 활동 보고서 메시지의 *Activity* PCF 그룹에 포함된 매개변수를 봅니다. 일부 매개변수는 특정 조작이 수행되었을 때에만 리턴됩니다.

활동 보고서 메시지 데이터는 *Activity* PCF 그룹 및 라우트 추적 메시지에 대해 생성된 경우 *TraceRoute* PCF 그룹으로 이루어집니다. *Activity* PCF 그룹을 이 주제에서 자세히 설명합니다.

조작별 활동 보고서 메시지 데이터로 설명되는 일부 매개변수는 특정 조작이 수행되었을 때에만 리턴됩니다.

활동 보고서의 경우 활동 보고서 메시지 데이터에 다음 매개변수가 포함됩니다.

Activity

설명: 활동을 설명하는 그룹화된 매개변수.

ID: MQGACF_ACTIVITY.
 데이터 유형: MQCFGR.
 PCF 그룹에 포함: 없음
 PCF 그룹의 매개변수: *ActivityApplName*
 ActivityApplType
 ActivityDescription
 Operation
 TraceRoute

리턴됨: 항상.

ActivityApplName

설명: 활동을 수행한 애플리케이션의 이름.
 ID: MQCACF_APPL_NAME.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Activity*.
 최대 길이: MQ_APPL_NAME_LENGTH.
 리턴됨: 항상.

ActivityApplType

설명: 활동을 수행한 애플리케이션의 유형.
 ID: MQIA_APPL_TYPE.
 데이터 유형: MQCFIN.
 PCF 그룹에 포함: *Activity*.
 리턴됨: 항상.

ActivityDescription

설명: 애플리케이션이 수행한 활동에 대한 설명.
 ID: MQCACF_ACTIVITY_DESCRIPTION.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Activity*.
 최대 길이: 64
 리턴됨: 항상.

Operation

설명: 활동의 조작을 설명하는 그룹화된 매개변수.
 ID: MQGACF_OPERATION.
 데이터 유형: MQCFGR.
 PCF 그룹에 포함: *Activity*.

PCF 그룹의 매개변수:

- OperationType*
- OperationDate*
- OperationTime*
- Message*
- QMgrName*
- QSGName*

참고: 조작 유형에 따라 이 그룹의 추가 매개변수가 리턴됩니다. 이 추가 매개변수는 [조작별 활동 보고서 메시지 데이터](#)로 설명됩니다.

리턴됨: 활동의 조작별로 하나의 *Operation* PCF 그룹.

OperationType

설명: 수행한 조작의 유형.
 ID: MQIACF_OPERATION_TYPE.
 데이터 유형: MQCFIN.
 PCF 그룹에 포함: *Operation*.
 값: MQOPER_*.
 리턴됨: 항상.

OperationDate

설명: 조작이 수행된 날짜.
 ID: MQCACF_OPERATION_DATE.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Operation*.
 최대 길이: MQ_DATE_LENGTH.
 리턴됨: 항상.

OperationTime

설명: 조작이 수행된 시간.
 ID: MQCACF_OPERATION_TIME.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Operation*.
 최대 길이: MQ_TIME_LENGTH.
 리턴됨: 항상.

Message

설명: 활동을 유발한 메시지를 설명하는 그룹화된 매개변수.
 ID: MQGACF_MESSAGE.
 데이터 유형: MQCFGR.
 PCF 그룹에 포함: *Operation*.
 그룹의 매개변수: *MsgLength*
MQMD
EmbeddedMQMD

리턴됨: 항상, 제외된 발행 조작은 예외입니다.

MsgLength

설명: 활동이 발생하기 전, 활동을 유발한 메시지의 길이.
ID: MQIACF_MSG_LENGTH.
데이터 유형: MQCFIN.
PCF 그룹에 포함: *Message*.
리턴됨: 항상.

MQMD

설명: 활동을 유발한 메시지의 메시지 디스크립터에 관련한 그룹화된 매개변수.
ID: MQGACF_MQMD.
데이터 유형: MQCFGR.
PCF 그룹에 포함: *Message*.

그룹의 매개변수:

- StrucId*
- Version*
- Report*
- MsgType*
- Expiry*
- Feedback*
- Encoding*
- CodedCharSetId*
- Format*
- Priority*
- Persistence*
- MsgId*
- CorrelId*
- BackoutCount*
- ReplyToQ*
- ReplyToQMGr*
- UserIdentifier*
- AccountingToken*
- ApplIdentityData*
- PutApplType*
- PutApplName*
- PutDate*
- PutTime*
- ApplOriginData*
- GroupId*
- MsgSeqNumber*
- Offset*
- MsgFlags*
- OriginalLength*

리턴됨: 항상, 제외된 발행 조작은 예외입니다.

EmbeddedMQMD

설명: 전송 큐의 메시지 내에 임베드된 메시지 디스크립터를 설명하는 그룹화된 매개변수.

ID: MQGACF_EMBEDDED_MQMD.

데이터 유형: MQCFGR.

PCF 그룹에 포함: *Message*.

그룹의 매개변수:

- StrucId*
- Version*
- Report*
- MsgType*
- Expiry*
- Feedback*
- Encoding*
- CodedCharSetId*
- Format*
- Priority*
- Persistence*
- MsgId*
- CorrelId*
- BackoutCount*
- ReplyToQ*
- ReplyToQMgr*
- UserIdentifier*
- AccountingToken*
- ApplIdentityData*
- PutApplType*
- PutApplName*
- PutDate*
- PutTime*
- ApplOriginData*
- GroupId*
- MsgSeqNumber*
- Offset*
- MsgFlags*
- OriginalLength*

리턴됨: 큐가 전송 큐로 분석되는 가져오기 조작의 경우.

StrucId

설명: 구조 ID

ID: MQCACF_STRUC_ID.

데이터 유형: MQCFST.

PCF 그룹에 포함: *MQMD* 또는 *EmbeddedMQMD*.

최대 길이: 4.

리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

Version

설명: 구조 버전 번호입니다.

ID: MQIACF_VERSION.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

Report

설명: 보고 메시지의 옵션입니다.
ID: MQIACF_REPORT.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

MsgType

설명: 메시지 유형을 표시합니다.
ID: MQIACF_MSG_TYPE.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

Expiry

설명: 메시지 수명입니다.
ID: MQIACF_EXPIRY.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

Feedback

설명: 피드백 또는 이유 코드.
ID: MQIACF_FEEDBACK.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

Encoding

설명: 메시지 데이터의 숫자 인코딩입니다.
ID: MQIACF_ENCODING.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

CodedCharSetId

설명:	메시지 데이터의 문자 세트 ID.
ID:	MQIA_CODED_CHAR_SET_ID.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

Format

설명:	메시지 데이터의 형식 이름
ID:	MQCACH_FORMAT_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
최대 길이:	MQ_FORMAT_LENGTH.
리턴됨:	항상, 제외된 발행 조작은 예외입니다.

Priority

설명:	메시지 우선순위입니다.
ID:	MQIACF_PRIORITY.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	항상, 제외된 발행 조작은 예외입니다.

Persistence

설명:	메시지 지속성.
ID:	MQIACF_PERSISTENCE.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	항상, 제외된 발행 조작은 예외입니다.

MsgId

설명:	메시지 ID.
ID:	MQBACF_MSG_ID.
데이터 유형:	MQCFBS.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
최대 길이:	MQ_MSG_ID_LENGTH.
리턴됨:	항상, 제외된 발행 조작은 예외입니다.

CorrelId

설명:	상관 ID.
ID:	MQBACF_CORREL_ID.
데이터 유형:	MQCFBS.

PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_CORREL_ID_LENGTH.
리턴됨: 항상, 제외된 발행 조작은 예외입니다.

BackoutCount

설명: 백아웃 카운터입니다.
ID: MQIACF_BACKOUT_COUNT.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

ReplyToQ

설명: 응답 큐의 이름입니다.
ID: MQCACF_REPLY_TO_Q.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_Q_NAME_LENGTH.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

ReplyToQMgr

설명: 응답 큐 관리자의 이름입니다.
ID: MQCACF_REPLY_TO_Q_MGR.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_Q_MGR_NAME_LENGTH.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

UserIdentifier

설명: 메시지를 생성한 애플리케이션의 사용자 ID.
ID: MQCACF_USER_IDENTIFIER.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_USER_ID_LENGTH.
리턴됨: 항상, 제외된 발행 조작은 예외입니다.

AccountingToken

설명: 애플리케이션이 메시지의 결과로 완료된 작업에 청구할 수 있는 계정 토큰.
ID: MQBACF_ACCOUNTING_TOKEN.
데이터 유형: MQCFBS.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_ACCOUNTING_TOKEN_LENGTH.

리턴됨: 항상, 제외된 발행 조작은 예외입니다.

ApplIdentityData

설명: ID와 관련된 애플리케이션 데이터.
ID: MQCACF_APPL_IDENTITY_DATA.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_APPL_IDENTITY_DATA_LENGTH.
리턴됨: 항상, 제외된 발행 조작은 예외입니다.

PutApplType

설명: 메시지를 넣는 애플리케이션의 유형입니다.
ID: MQIA_APPL_TYPE.
데이터 유형: MQCFIN.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

PutApplName

설명: 메시지를 넣는 애플리케이션의 이름입니다.
ID: MQCACF_APPL_NAME.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_APPL_NAME_LENGTH.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

PutDate

설명: 메시지를 넣은 날짜입니다.
ID: MQCACF_PUT_DATE.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_PUT_DATE_LENGTH.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

PutTime

설명: 메시지를 넣은 시간입니다.
ID: MQCACF_PUT_TIME.
데이터 유형: MQCFST.
PCF 그룹에 포함: MQMD 또는 *EmbeddedMQMD*.
최대 길이: MQ_PUT_TIME_LENGTH.
리턴됨: 항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

ApplOriginData

설명:	원본과 관련된 애플리케이션 데이터입니다.
ID:	MQCACF_APPL_ORIGIN_DATA.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
최대 길이:	MQ_APPL_ORIGIN_DATA_LENGTH.
리턴됨:	항상, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작은 예외입니다.

GroupId

설명:	실제 메시지가 속한 메시지 그룹 또는 논리 메시지를 식별합니다.
ID:	MQBACF_GROUP_ID.
데이터 유형:	MQCFBS.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
최대 길이:	MQ_GROUP_ID_LENGTH.
리턴됨:	<i>Version</i> 이 MQMD_VERSION_2로 지정된 경우, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작의 경우 리턴되지 않습니다.

MsgSeqNumber

설명:	그룹 내의 논리 메시지의 순서 번호.
ID:	MQIACH_MSG_SEQUENCE_NUMBER.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	<i>Version</i> 이 MQMD_VERSION_2로 지정된 경우, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작의 경우 리턴되지 않습니다.

Offset

설명:	논리 메시지의 시작으로부터 실제 메시지의 데이터 오프셋입니다.
ID:	MQIACF_OFFSET.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	<i>Version</i> 이 MQMD_VERSION_2로 지정된 경우, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작의 경우 리턴되지 않습니다.

MsgFlags

설명:	메시지의 속성을 지정하거나 처리를 제어하는 메시지 플래그입니다.
ID:	MQIACF_MSG_FLAGS.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	<i>Version</i> 이 MQMD_VERSION_2로 지정된 경우, 제외된 발행 조작과 MQMD에서 발행 및 제거된 발행 조작의 경우 리턴되지 않습니다.

OriginalLength

설명:	원래 메시지의 길이.
ID:	MQIACF_ORIGINAL_LENGTH.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>MQMD</i> 또는 <i>EmbeddedMQMD</i> .
리턴됨:	<i>Version</i> 이 <i>MQMD_VERSION_2</i> 로 지정된 경우. 제외된 발행 조작과 <i>MQMD</i> 에서 발행 및 제거된 발행 조작의 경우 리턴되지 않습니다.

QMgrName

설명:	활동이 수행된 큐 관리자의 이름.
ID:	MQCA_Q_MGR_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_Q_MGR_NAME_LENGTH
리턴됨:	항상.

QSGName

설명:	활동이 수행된 큐 관리자가 속한 큐 공유 그룹의 이름.
ID:	MQCA_QSG_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_QSG_NAME_LENGTH
리턴됨:	IBM MQ for z/OS 큐 관리자에서 활동이 수행된 경우.

TraceRoute

설명:	라우트 추적 메시지의 속성을 지정하는 그룹화된 매개변수.
ID:	MQGACF_TRACE_ROUTE.
데이터 유형:	MQCFGR.
PCF 그룹에 포함:	<i>Activity</i> .
그룹의 매개변수:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
리턴됨:	라우트 추적 메시지에 관하여 활동이 수행된 경우.

TraceRoute PCF 그룹의 매개변수 값은 활동 보고서가 생성되었을 때 라우트 추적 메시지의 값입니다.

조작별 활동 보고서 메시지 데이터

이 페이지를 사용하여 *OperationType* 매개변수의 값에 따라 활동 보고서의 PCF 그룹 *Operation*에 리턴될 수 있는 추가 PCF 매개변수를 봅니다.

추가 매개변수는 다음 조작 유형에 따라 다릅니다.

가져오기/찾아보기(MQOPER_GET/MQOPER_BROWSE)

가져오기/찾아보기(MQOPER_GET/MQOPER_BROWSE) 조작 유형(큐의 메시지를 가져오거나 찾아봄)에 대해 PCF 그룹 *Operation*에 리턴되는 추가 활동 보고서 메시지 데이터 매개변수.

QName

설명:	열린 큐의 이름.
ID:	MQCA_Q_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_Q_NAME_LENGTH
리턴됨:	항상.

ResolvedQName

설명:	열린 큐를 분석한 이름.
ID:	MQCACF_RESOLVED_Q_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_Q_NAME_LENGTH
리턴됨:	항상.

제거(MQOPER_DISCARD)

제거(MQOPER_DISCARD) 조작 유형(메시지가 제거됨)에 대해 PCF 그룹 *Operation*에 리턴되는 추가 활동 보고서 메시지 데이터 매개변수.

Feedback

설명:	메시지를 제거하는 이유.
ID:	MQIACF_FEEDBACK.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	<i>Operation</i> .
리턴됨:	항상.

QName

설명:	열린 큐의 이름.
ID:	MQCA_Q_NAME.
데이터 유형:	MQCFST.
최대 길이:	MQ_Q_NAME_LENGTH
PCF 그룹에 포함:	<i>Operation</i> .
리턴됨:	큐에 넣는 데 실패해서 메시지가 제거된 경우.

RemoteQMgrName

설명:	메시지가 예정된 큐 관리자의 이름.
ID:	MQCA_REMOTE_Q_MGR_NAME.

데이터 유형: MQCFST.
 최대 길이: MQ_Q_MGR_NAME_LENGTH
 PCF 그룹에 포함: Operation.
 리턴됨: Feedback의 값이 MQFB_NOT_FORWARDED인 경우.

발행/제거된 발행/제외된 발행(MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/
 MQOPER_EXCLUDED_PUBLISH)
 발행/제거된 발행/제외된 발행(MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/
 MQOPER_EXCLUDED_PUBLISH) 조작 유형(발행/구독 메시지가 전달, 제거 또는 제외됨)에 대해 PCF 그룹
 Operation에 리턴되는 추가 활동 보고서 메시지 데이터 매개변수.

SubId

설명: 구독 ID.
 ID: MQBACF_SUB_ID.
 데이터 유형: MQCFBS.
 PCF 그룹에 포함: Operation.
 리턴됨: 항상.

SubLevel

설명: 구독 레벨.
 ID: MQIACF_SUB_LEVEL.
 데이터 유형: MQCFIN.
 PCF 그룹에 포함: Operation.
 리턴됨: 항상.

Feedback

설명: 메시지를 제거하는 이유.
 ID: MQIACF_FEEDBACK.
 데이터 유형: MQCFIN.
 PCF 그룹에 포함: Operation.
 리턴됨: 메시지가 구독자에게 전달되지 않았거나 구독자가 제외되었기 때문에 메시지가 전달
 되지 않아서 메시지가 제거된 경우.

발행 조작 MQOPER_PUBLISH는 특정 구독자에게 전달된 메시지에 대한 정보를 제공합니다. 이 조작은 연관된
 넣기 조작에 설명된 메시지에서 변경되었을 수 있는 향후 메시지의 요소를 설명합니다. 넣기 조작과 마찬가지로,
 메시지 그룹 MQGACF_MESSAGE 및 그룹 내 MQMD 그룹 MQGACF_MQMD를 포함합니다. 그러나 이 MQMD 그
 룬에는 구독자가 대체할 수 있는 *Format, Priority, Persistence, MsgId, CorrelId, UserIdentifier,*
AccountingToken, ApplIdentityData 필드만 포함됩니다.

구독자의 *SubId* 및 *SubLevel*은 조작 정보에 포함됩니다. *SubID*를 MQCMD_INQUIRE_SUBSCRIBER PCF 명령
 과 함께 사용하여 구독자에 대한 다른 모든 속성을 검색할 수 있습니다.

제거된 발행 조작 MQOPER_DISCARDED_PUBLISH는 메시지가 포인트-투-포인트 메시징으로 전달되지 않을
 때 사용되는 제거 조작과 유사합니다. 로컬 목적지에 메시지를 전달하지 않도록 명시적으로 요청되었고 이 구독
 자가 로컬 목적지를 지정하는 경우 구독자에게 메시지가 전달되지 않습니다. 큐가 가득 찼을 때처럼 목적지 큐에
 메시지가 도달하는 데 문제점이 있는 경우에도 메시지가 전달되지 않는 것으로 고려합니다.

제거된 발행 조작의 정보는 발행 조작에 대해 동일하며 메시지가 전달되지 않은 이유를 제공하는 *Feedback* 필드가 추가됩니다. 이 피드백 필드는 MQOPER_DISCARD 조작에 공통된 MQFB_* 또는 MQRC_* 값을 포함합니다. 발행 제외와 대조적으로, 발행을 제거하는 이유는 넣기를 제거하는 이유와 동일합니다.

구독자가 구독 중인 토픽이 연관된 넣기 조작의 토픽과 일치하지만 토픽에 넣고 있는 메시지와 다른 선택 기준이 일치하지 않아서 구독자에게 메시지가 전달되지 않았기 때문에, 제외된 발행 조작 MQOPER_EXCLUDED_PUBLISH는 메시지 전달을 고려한 구독자에 대한 정보를 제공합니다. 제거된 발행 조작에서처럼, *Feedback* 필드는 이 구독이 제외된 원인에 대한 정보를 제공합니다. 하지만 제거된 발행 조작과 달리, 이 구독자에 대한 메시지가 생성되지 않아서 메시지 관련 정보가 제공되지 않습니다.

넣기/응답 넣기/보고서 넣기(MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT)

넣기/응답 넣기/보고서 넣기(MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT) 조작 유형(메시지, 응답 메시지 또는 보고서 메시지를 큐에 넣음)에 대해 PCF 그룹 *Operation*에 리턴되는 추가 활동 보고서 메시지 데이터 매개변수.

QName

설명:	열린 큐의 이름.
ID:	MQCA_Q_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_Q_NAME_LENGTH
리턴됨:	항상, 발행 활동 내에 포함된 토픽에 대한 넣기 조작인 경우에 리턴되지 않는 한 가지 예외가 있습니다.

ResolvedQName

설명:	열린 큐를 분석한 이름.
ID:	MQCACF_RESOLVED_Q_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_Q_NAME_LENGTH
리턴됨:	열린 큐를 분석할 수 있을 때, 발행 활동 내에 포함된 토픽에 대한 넣기 조작인 경우 리턴되지 않습니다.

RemoteQName

설명:	리모트 큐 관리자에 알려진 열린 큐의 이름.
ID:	MQCA_REMOTE_Q_NAME.
데이터 유형:	MQCFST.
PCF 그룹에 포함:	<i>Operation</i> .
최대 길이:	MQ_Q_NAME_LENGTH
리턴됨:	열린 큐가 리모트 큐인 경우, 발행 활동 내에 포함된 토픽에 대한 넣기 조작인 경우 리턴되지 않습니다.

RemoteQMgrName

설명:	리모트 큐가 정의된 리모트 큐 관리자의 이름.
ID:	MQCA_REMOTE_Q_MGR_NAME.
데이터 유형:	MQCFST.

PCF 그룹에 포함: *Operation.*
 최대 길이: MQ_Q_MGR_NAME_LENGTH
 리턴됨: 열린 큐가 리모트 큐인 경우. 발행 활동 내에 포함된 토픽에 대한 넣기 조작인 경우 리턴되지 않습니다.

TopicString

설명: 메시지를 넣고 있는 전체 토픽 문자열.
 ID: MQCA_TOPIC_STRING.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Operation.*
 리턴됨: 발행 활동 내에 포함된 토픽에 대한 넣기 조작인 경우.

Feedback

설명: 데드-레터 큐에 메시지를 넣는 이유.
 ID: MQIACF_FEEDBACK.
 데이터 유형: MQCFIN.
 PCF 그룹에 포함: *Operation.*
 리턴됨: 데드-레터 큐에 메시지를 넣은 경우.

수신(MQOPER_RECEIVE)

수신(MQOPER_RECEIVE) 조작 유형(채널에 메시지가 수신됨)에 대해 PCF 그룹 *Operation*에 리턴되는 추가 활동 보고서 메시지 데이터 매개변수.

ChannelName

설명: 메시지가 수신된 채널의 이름.
 ID: MQCACH_CHANNEL_NAME.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Operation.*
 최대 길이: MQ_CHANNEL_NAME_LENGTH
 리턴됨: 항상.

ChannelType

설명: 메시지가 수신된 채널의 유형.
 ID: MQIACH_CHANNEL_TYPE.
 데이터 유형: MQCFIN.
 PCF 그룹에 포함: *Operation.*
 리턴됨: 항상.

RemoteQMgrName

설명: 메시지가 수신된 큐 관리자의 이름.
 ID: MQCA_REMOTE_Q_MGR_NAME.
 데이터 유형: MQCFST.
 PCF 그룹에 포함: *Operation.*

최대 길이: MQ_Q_MGR_NAME_LENGTH
리턴됨: 항상.

송신(MQOPER_SEND)

송신(MQOPER_SEND) 조작 유형(채널에서 메시지가 송신됨)에 대해 PCF 그룹 *Operation*에 리턴되는 추가 활동 보고서 메시지 데이터 매개변수.

ChannelName

설명: 메시지가 송신된 채널의 이름.
ID: MQCACH_CHANNEL_NAME.
데이터 유형: MQCFST.
PCF 그룹에 포함: *Operation*.
최대 길이: MQ_CHANNEL_NAME_LENGTH.
리턴됨: 항상.

ChannelType

설명: 메시지가 송신된 채널의 유형.
ID: MQIACH_CHANNEL_TYPE.
데이터 유형: MQCFIN.
PCF 그룹에 포함: *Operation*.
리턴됨: 항상.

XmitQName

설명: 메시지를 검색한 전송 큐.
ID: MQCACH_XMIT_Q_NAME.
데이터 유형: MQCFST.
PCF 그룹에 포함: *Operation*.
최대 길이: MQ_Q_NAME_LENGTH.
리턴됨: 항상.

RemoteQMgrName

설명: 메시지가 송신된 리모트 큐 관리자의 이름.
ID: MQCA_REMOTE_Q_MGR_NAME.
데이터 유형: MQCFST.
PCF 그룹에 포함: *Operation*.
최대 길이: MQ_Q_MGR_NAME_LENGTH
리턴됨: 항상.

라우트 추적 메시지 참조

이 페이지를 사용하여 라우트 추적 메시지 형식을 개괄합니다. 라우트 추적 메시지 데이터는 라우트 추적 메시지가 유발한 활동을 설명하는 매개변수를 포함합니다.

라우트 추적 메시지 형식

라우트 추적 메시지는 메시지 디스크립터 및 메시지 데이터를 포함한 표준 IBM MQ 메시지입니다. 메시지 데이터는 큐 관리자 네트워크를 통해 라우팅되었기 때문에 라우트 추적 메시지에 대해 수행된 활동에 대한 정보를 포함합니다.

라우트 추적 메시지는 다음 정보를 포함합니다.

메시지 디스크립터

MQFMT_ADMIN 또는 MQFMT_EMBEDDED_PCF로 설정된 *Format* 필드를 갖는 MQMD 구조입니다.

메시지 데이터

다음 중 하나로 이루어집니다.

- *Format*이 MQFMT_ADMIN으로 설정된 경우 PCF 헤더(MQCFH)와 라우트 추적 메시지 데이터 또는
- *Format*이 MQFMT_EMBEDDED_PCF로 설정된 경우 임베드된 PCF 헤더(MQEPH), 라우트 추적 메시지 데이터, 추가 사용자 지정 메시지 데이터.

IBM MQ 표시 라우트 애플리케이션을 사용하여 라우트 추적 메시지를 생성할 때에는 *Format*이 MQFMT_ADMIN으로 설정됩니다.

라우트 추적 메시지 데이터의 콘텐츠는 다음과 같이 *TraceRoute* PCF 그룹의 *Accumulate* 매개변수로 판별됩니다.

- *Accumulate*가 MQROUTE_ACCUMULATE_NONE으로 설정된 경우 라우트 추적 메시지 데이터에 *TraceRoute* PCF 그룹이 포함됩니다.
- *Accumulate*가 MQROUTE_ACCUMULATE_IN_MSG 또는 MQROUTE_ACCUMULATE_AND_REPLY로 설정된 경우에는 라우트 추적 메시지 데이터에 *TraceRoute* PCF 그룹과 0개 이상의 *Activity* PCF 그룹이 포함됩니다.

[116 페이지의 표 20](#)은 라우트 추적 메시지의 구조를 보여줍니다.

표 20. 라우트 추적 메시지 형식		
MQMD 구조	임베드된 PCF 헤더 MQEPH 구조	라우트 추적 메시지 데이터
구조 ID 구조 버전 보고서 옵션 메시지 유형 만기 시간 Feedback Encoding 코드화 문자 세트 ID 메시지 형식 Priority Persistence 메시지 ID 상관 ID 백아웃 수 응답 대상 큐 응답 대상 큐 관리자 사용자 ID 계정 토큰 애플리케이션 ID 데이터 애플리케이션 유형 애플리케이션 이름 넣은 날짜 넣은 시간 애플리케이션 원본 데이터 그룹 ID 메시지 순서 번호 오프셋 메시지 플래그 원래 길이	구조 ID 구조 버전 구조 길이 Encoding 코드화 문자 세트 ID 메시지 형식 플래그 PCF 헤더(MQCFH) 구조 유형 구조 길이 구조 버전 명령 ID 메시지 순서 번호 제어 옵션 완료 코드 이유 코드 매개변수 수	TraceRoute 세부사항 기록된 활동 기록되지 않은 활동 중단 수 최대 활동 누적 전달

라우트 추적 메시지 MQMD(메시지 디스크립터)

이 페이지를 사용하여 라우트 추적 메시지의 MQMD 구조에 따라 포함된 값을 봅니다.

StrucId

설명: 구조 ID입니다.
 데이터 유형: MQCHAR4.
 값: MQMD_STRUC_ID.

Version

설명: 구조 버전 번호입니다.
 데이터 유형: MQLONG.
 값: **MQMD_VERSION_1.**

Report

설명: 보고 메시지의 옵션입니다.
 데이터 유형: MQLONG.

값: 요구사항에 따라 설정됩니다. 공통 보고서 옵션은 다음과 같습니다.

MQRO_DISCARD_MSG

메시지가 로컬 큐에 도달하면 제거됩니다.

MQRO_PASS_DISCARD_AND_EXPIRY

모든 응답(활동 보고서 또는 라우트 추적 응답 메시지)에 보고서 옵션 MQRO_DISCARD_MSG가 설정되고 남은 만기가 전달됩니다. 이를 통해 응답이 큐 관리자 네트워크에 무제한으로 남아 있지 않습니다.

MsgType

설명: 메시지의 유형.

데이터 유형: MQLONG.

값: TraceRoute 그룹의 *Accumulate* 매개변수가 MQROUTE_ACCUMULATE_AND_REPLY로 지정되어 있으면 메시지 유형이 MQMT_REQUEST입니다.

그렇지 않으면 다음을 수행하십시오.

MQMT_DATAGRAM.

Expiry

설명: 메시지 수명입니다.

데이터 유형: MQLONG.

값: 요구사항에 따라 설정됩니다. 이 매개변수는 라우트 추적 메시지가 큐 관리자 네트워크에 무제한으로 남지 않도록 하는 데 사용할 수 있습니다.

Feedback

설명: 피드백 또는 이유 코드.

데이터 유형: MQLONG.

값: **MQFB_NONE.**

Encoding

설명: 메시지 데이터의 숫자 인코딩입니다.

데이터 유형: MQLONG.

값: 해당하는 값으로 설정됩니다.

CodedCharSetId

설명: 메시지 데이터의 문자 세트 ID.

데이터 유형: MQLONG.

값: 해당하는 값으로 설정됩니다.

Format

설명: 메시지 데이터의 형식 이름

데이터 유형: MQCHAR8.

값: **MQFMT_ADMIN**
관리 메시지. *TraceRoute* PCF 그룹 다음에 사용자 데이터가 없습니다.

MQFMT_EMBEDDED_PCF
임베드된 PCF 메시지. *TraceRoute* PCF 그룹 다음에 사용자 데이터가 있습니다.

Priority

설명: 메시지 우선순위입니다.
데이터 유형: MQLONG.
값: 요구사항에 따라 설정됩니다.

Persistence

설명: 메시지 지속성.
데이터 유형: MQLONG.
값: 요구사항에 따라 설정됩니다.

MsgId

설명: 메시지 ID.
데이터 유형: MQBYTE24.
값: 요구사항에 따라 설정됩니다.

CorrelId

설명: 상관 ID.
데이터 유형: MQBYTE24.
값: 요구사항에 따라 설정됩니다.

BackoutCount

설명: 백아웃 카운터입니다.
데이터 유형: MQLONG.
값: 0입니다.

ReplyToQ

설명: 응답 큐의 이름입니다.
데이터 유형: MQCHAR48.
값: 요구사항에 따라 설정됩니다.

*MsgType*이 MQMT_REQUEST로 설정된 경우나 *Report*에 보고서 생성 옵션이 설정된 경우에는 이 매개변수가 공백이면 안됩니다.

ReplyToQMgr

설명: 응답 큐 관리자의 이름입니다.
데이터 유형: MQCHAR48.
값: 요구사항에 따라 설정됩니다.

UserIdentifier

설명: 메시지를 생성한 애플리케이션의 사용자 ID.
데이터 유형: MQCHAR12.
값: 일반적으로 설정됩니다.

AccountingToken

설명: 애플리케이션이 메시지의 결과로 완료된 작업에 청구할 수 있는 계정 토큰.
데이터 유형: MQBYTE32.
값: 일반적으로 설정됩니다.

ApplIdentityData

설명: ID와 관련된 애플리케이션 데이터.
데이터 유형: MQCHAR32.
값: 일반적으로 설정됩니다.

PutApplType

설명: 메시지를 넣는 애플리케이션의 유형입니다.
데이터 유형: MQLONG.
값: 일반적으로 설정됩니다.

PutApplName

설명: 메시지를 넣는 애플리케이션의 이름입니다.
데이터 유형: MQCHAR28.
값: 일반적으로 설정됩니다.

PutDate

설명: 메시지를 넣은 날짜입니다.
데이터 유형: MQCHAR8.
값: 일반적으로 설정됩니다.

PutTime

설명: 메시지를 넣은 시간입니다.
데이터 유형: MQCHAR8.
값: 일반적으로 설정됩니다.

ApplOriginData

설명: 원본과 관련된 애플리케이션 데이터입니다.
데이터 유형: MQCHAR4.
값: 일반적으로 설정됩니다.

라우트 추적 메시지 MQEPH(임베드된 PCF 헤더)

이 페이지를 사용하여 라우트 추적 메시지의 MQEPH 구조에 따라 포함된 값을 봅니다.

MQEPH 구조는 라우트 추적 메시지의 메시지 데이터를 수반한 PCF 정보와 이를 따르는 애플리케이션 메시지 데이터 모두에 대한 설명을 포함합니다. MQEPH 구조는 TraceRoute PCF 그룹 다음에 추가 사용자 메시지 데이터가 있는 경우에만 사용됩니다.

라우트 추적 메시지의 경우 MQEPH 구조가 다음 값을 포함합니다.

StrucId

설명: 구조 ID입니다.
데이터 유형: MQCHAR4.
값: MQEPH_STRUC_ID.

Version

설명: 구조 버전 번호입니다.
데이터 유형: MQLONG.
값: MQEPH_VERSION_1.

StrucLength

설명: 구조 길이.
데이터 유형: MQLONG.
값: 뒤따르는 PCF 매개변수 구조를 포함한 구조의 총 길이.

Encoding

설명: 마지막 PCF 매개변수 구조 다음에 오는 메시지 데이터의 숫자 인코딩.
데이터 유형: MQLONG.
값: 메시지 데이터의 인코딩.

CodedCharSetId

설명: 마지막 PCF 매개변수 구조 다음에 오는 메시지 데이터의 문자 세트 ID.
데이터 유형: MQLONG.
값: 메시지 데이터의 문자 세트.

Format

설명: 마지막 PCF 매개변수 구조 다음에 오는 메시지 데이터의 형식 이름.
데이터 유형: MQCHAR8.
값: 메시지 데이터의 형식 이름.

Flags

설명: 구조의 속성을 지정하거나 처리를 제어하는 플래그.
데이터 유형: MQLONG.
값: **MQEPH_NONE**
플래그가 지정되지 않습니다.
MQEPH_CCSID_EMBEDDED
문자 데이터를 포함한 매개변수의 문자 세트가 각 구조의 *CodedCharSetId* 필드 내에 개별적으로 지정됨을 지정합니다.

PCFHeader

설명:	프로그래밍 가능 명령 형식(PCF) 헤더
데이터 유형:	MQCFH.
값:	121 페이지의 『라우트 추적 메시지 MQCFH(PCF 헤더)』의 내용을 참조하십시오.

라우트 추적 메시지 MQCFH(PCF 헤더)

이 페이지를 사용하여 라우트 추적 메시지의 MQCFH 구조에 따라 포함된 PCF 값을 봅니다.

라우트 추적 메시지의 경우 MQCFH 구조가 다음 값을 포함합니다.

Type

설명:	메시지의 콘텐츠를 식별하는 구조 유형입니다.
데이터 유형:	MQLONG.
값:	MQCFT_TRACE_ROUTE 메시지가 라우트 추적 메시지입니다.

StrucLength

설명:	구조 길이.
데이터 유형:	MQLONG.
값:	MQCFH_STRUC_LENGTH MQCFH 구조의 길이(바이트 단위).

Version

설명:	구조 버전 번호입니다.
데이터 유형:	MQLONG.
값:	MQCFH_VERSION_3

Command

설명:	명령 ID입니다. 메시지의 범주를 식별합니다.
데이터 유형:	MQLONG.
값:	MQCMD_TRACE_ROUTE 라우트 추적 메시지.

MsgSeqNumber

설명:	메시지 순서 번호입니다. 관련 메시지 그룹 내의 메시지의 순서 번호입니다.
데이터 유형:	MQLONG.
값:	1.

Control

설명:	제어 옵션입니다.
데이터 유형:	MQLONG.
값:	MQCFC_LAST.

CompCode

설명: 완료 코드.
데이터 유형: MQLONG.
값: MQCC_OK.

Reason

설명: 완료 코드를 규정하는 이유 코드입니다.
데이터 유형: MQLONG.
값: MQRC_NONE.

ParameterCount

설명: 매개변수 구조 수입니다. MQCFH 구조 뒤에 나오는 매개변수 구조의 수입니다. 그룹 구조(MQCFGR) 및 포함된 해당 매개변수 구조는 하나의 구조로만 계산됩니다.
데이터 유형: MQLONG.
값: 1 이상.

라우트 추적 메시지 데이터

이 페이지를 사용하여 라우트 추적 메시지 데이터의 *TraceRoute* PCF 그룹을 구성하는 매개변수를 봅니다.

라우트 추적 메시지 데이터의 콘텐츠는 *TraceRoute* PCF 그룹의 *Accumulate* 매개변수에 따라 다릅니다. 라우트 추적 메시지 데이터는 *TraceRoute* PCF 그룹과 0개 이상의 *Activity* PCF 그룹으로 구성됩니다. *TraceRoute* PCF 그룹을 이 주제에서 자세히 설명합니다. *Activity* PCF 그룹의 세부사항은 관련 정보를 참조하십시오.

라우트 추적 메시지 데이터는 다음 매개변수를 포함합니다.

TraceRoute

설명: 라우트 추적 메시지의 속성을 지정하는 그룹화된 매개변수. 라우트 추적 메시지의 경우 일부 매개변수를 변경하여 처리 방식을 제어할 수 있습니다.
ID: MQGACF_TRACE_ROUTE.
데이터 유형: MQCFGR.
PCF 그룹에 포함: 없음
그룹의 매개변수: *Detail*
RecordedActivities
UnrecordedActivities
DiscontinuityCount
MaxActivities
Accumulate
Forward
Deliver

Detail

설명: 활동에 대해 기록할 세부사항 레벨.
ID: MQIACF_ROUTE_DETAIL.
데이터 유형: MQCFIN.
PCF 그룹에 포함: *TraceRoute*.

값:	MQROUTE_DETAIL_LOW 사용자 작성 애플리케이션이 수행한 활동이 기록됩니다.
	MQROUTE_DETAIL_MEDIUM MQROUTE_DETAIL_LOW에 지정된 활동이 기록됩니다. 또한 MCA가 수행한 활동이 기록됩니다.
	MQROUTE_DETAIL_HIGH MQROUTE_DETAIL_LOW 및 MQROUTE_DETAIL_MEDIUM에 지정된 활동이 기록됩니다. MCA는 이 세부사항 레벨에서 어떤 추가 활동 정보도 기록하지 않습니다. 이 옵션은 추가 활동 정보를 기록할 사용자 작성 애플리케이션에만 사용 가능합니다.

RecordedActivities

설명:	정보가 기록된 라우트 추적 메시지가 유발한 활동의 수.
ID:	MQIACF_RECORDED_ACTIVITIES.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.

UnrecordedActivities

설명:	정보가 기록되지 않은 라우트 추적 메시지가 유발한 활동의 수.
ID:	MQIACF_UNRECORDED_ACTIVITIES.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.

DiscontinuityCount

설명:	라우트 추적 메시지를 지원하지 않는 큐 관리자에서 라우트 추적 메시지가 수신된 횟수.
ID:	MQIACF_DISCONTINUITY_COUNT.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.

MaxActivities

설명:	처리가 중지되기 전 라우트 추적 메시지에 포함시킬 수 있는 활동의 최대 수입니다.
ID:	MQIACF_MAX_ACTIVITIES.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.
값:	양수 활동의 최대 수입니다.

MQROUTE_UNLIMITED_ACTIVITIES
무제한 수의 활동입니다.

Accumulate

설명:	활동 정보가 라우트 추적 메시지 내에 누적되는지 여부 및 라우트 추적 메시지를 제거하거나 비전송 큐에 넣기 전에 누적된 활동 정보를 포함한 응답 메시지가 생성되는지 여부를 지정합니다.
ID:	MQIACF_ROUTE_ACCUMULATION.

데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.
값:	<p>MQROUTE_ACCUMULATE_NONE 라우트 추적 메시지의 메시지 데이터에 활동 정보가 누적되지 않습니다.</p> <p>MQROUTE_ACCUMULATE_IN_MSG 라우트 추적 메시지의 메시지 데이터에 활동 정보가 누적됩니다.</p> <p>MQROUTE_ACCUMULATE_AND_REPLY 활동 정보가 라우트 추적 메시지의 메시지 데이터에 누적되고 라우트 추적 응답 메시지가 생성됩니다.</p>

Forward

설명:	라우트 추적 메시지를 전달할 수 있는 큐 관리자를 지정합니다. 리모트 큐 관리자에 메시지를 전달할지 여부를 판별할 때 큐 관리자는 <u>전달</u> 에 설명된 알고리즘을 사용합니다.
ID:	MQIACF_ROUTE_FORWARDING.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.
값:	<p>MQROUTE_FORWARD_IF_SUPPORTED 라우트 추적 메시지는 <i>TraceRoute</i> 그룹의 <i>Deliver</i> 매개변수의 값을 존중하는 큐 관리자로만 전달됩니다.</p> <p>MQROUTE_FORWARD_ALL 라우트 추적 메시지가 <i>Deliver</i> 매개변수의 값이 존중되는지 여부와 상관없이 모든 큐 관리자로 전달됩니다.</p>

Deliver

설명:	라우트 추적 메시지가 목적지 큐에 정상적으로 도달하는 경우에 수행할 조치를 지정합니다.
ID:	MQIACF_ROUTE_DELIVERY.
데이터 유형:	MQCFIN.
PCF 그룹에 포함:	TraceRoute.
값:	<p>MQROUTE_DELIVER_YES 라우트 추적 메시지가 도달하면 대상 큐에 메시지를 넣습니다. 대상 큐에서 파괴적 가져오기를 수행하는 애플리케이션은 라우트 추적 메시지를 수신할 수 있습니다.</p> <p>MQROUTE_DELIVER_NO 라우트 추적 메시지가 도달하면 메시지를 제거합니다.</p>

라우트 추적 응답 메시지 참조

이 페이지를 사용하여 라우트 추적 응답 메시지 형식을 개괄합니다. 라우트 추적 응답 메시지 데이터는 데이터가 생성된 라우트 추적 메시지의 라우트 추적 메시지 데이터 복제입니다.

라우트 추적 응답 메시지 형식

라우트 추적 응답 메시지는 메시지 디스크립터 및 메시지 데이터를 포함한 표준 IBM MQ 메시지입니다. 메시지 데이터는 큐 관리자 네트워크를 통해 라우팅되었기 때문에 라우트 추적 메시지에 대해 수행된 활동에 대한 정보를 포함합니다.

라우트 추적 응답 메시지는 다음 정보를 포함합니다.

메시지 디스크립터

MQMD 구조

메시지 데이터

PCF 헤더(MQCFH) 및 라우트 추적 메시지 데이터

라우트 추적 응답 메시지 데이터는 하나 이상의 Activity PCF 그룹으로 구성됩니다.

라우트 추적 메시지가 대상 큐에 도달하면 라우트 추적 메시지의 활동 정보 사본을 포함한 라우트 추적 응답 메시지가 생성될 수 있습니다. 라우트 추적 응답 메시지는 응답 대상 큐나 시스템 큐로 전달됩니다.

125 페이지의 표 21은 특정 조건 하에서만 리턴되는 매개변수를 포함하여, 라우트 추적 응답 메시지의 구조를 보여줍니다.

표 21. 라우트 추적 응답 메시지 형식		
MQMD 구조	PCF 헤더 MQCFH 구조	라우트 추적 응답 메시지 데이터
구조 ID 구조 버전 보고서 옵션 메시지 유형 만기 시간 Feedback Encoding 코드화 문자 세트 ID 메시지 형식 Priority Persistence 메시지 ID 상관 ID 백아웃 수 응답 대상 큐 응답 대상 큐 관리자 사용자 ID 계정 토큰 애플리케이션 ID 데이터 애플리케이션 유형 애플리케이션 이름 넣은 날짜 넣은 시간 애플리케이션 원본 데이터 그룹 ID 메시지 순서 번호 오프셋 메시지 플래그 원래 길이	PCF 헤더(MQCFH) 구조 유형 구조 길이 구조 버전 명령 ID 메시지 순서 번호 제어 옵션 완료 코드 이유 코드 매개변수 수	활동 활동 애플리케이션 이름 활동 애플리케이션 유형 활동 설명 Operation 조작 유형 조작 날짜 조작 시간 메시지 메시지 길이 MQMD EmbeddedMQMD 큐 관리자 이름 큐 공유 그룹 이름 큐 이름 1 2 3 분석된 큐 이름 1 3 리모트 큐 이름 3 리모트 큐 관리자 이름 2 3 4 5 피드백 2 채널 이름 4 5 채널 유형 4 5 전송 큐 이름 5 TraceRoute 세부사항 기록된 활동 기록되지 않은 활동 중단 수 최대 활동 누적 전달
참고: 1. 가져오기 및 찾아보기 조작에 대해 리턴됩니다. 2. 제거 조작에 대해 리턴됩니다. 3. 넣기, 응답 넣기, 보고서 넣기 조작에 대해 리턴됩니다. 4. 수신 조작에 대해 리턴됩니다. 5. 송신 조작에 대해 리턴됩니다.		

라우트 추적 응답 메시지 MQMD(메시지 디스크립터)

이 페이지를 사용하여 라우트 추적 응답 메시지의 MQMD 구조에 따라 포함된 값을 봅니다.

라우트 추적 응답 메시지의 경우 MQMD 구조는 [활동 보고서 메시지 디스크립터](#)에 설명된 매개변수를 포함합니다. 라우트 추적 응답 메시지 디스크립터의 일부 매개변수 값은 다음과 같이 활동 보고서 메시지 디스크립터와는 차이가 있습니다.

MsgType

설명:	메시지의 유형.
데이터 유형:	MQLONG.
값:	MQMT_REPLY

Feedback

설명:	피드백 또는 이유 코드.
데이터 유형:	MQLONG.
값:	MQFB_NONE

Encoding

설명:	메시지 데이터의 숫자 인코딩입니다.
데이터 유형:	MQLONG.
값:	라우트 추적 메시지 디스크립터에서 복사됩니다.

CodedCharSetId

설명:	메시지 데이터의 문자 세트 ID.
데이터 유형:	MQLONG.
값:	라우트 추적 메시지 디스크립터에서 복사됩니다.

Format

설명:	메시지 데이터의 형식 이름
데이터 유형:	MQCHAR8.
값:	MQFMT_ADMIN 관리 메시지.

라우트 추적 응답 메시지 MQCFH(PCF 헤더)

이 페이지를 사용하여 라우트 추적 응답 메시지의 MQCFH 구조에 따라 포함된 PCF 값을 봅니다.

라우트 추적 응답 메시지의 PCF 헤더(MQCFH)는 라우트 추적 메시지의 경우와 동일합니다.

라우트 추적 응답 메시지 데이터

라우트 추적 응답 메시지 데이터는 데이터가 생성된 라우트 추적 메시지의 라우트 추적 메시지 데이터 복제입니다.

라우트 추적 응답 메시지 데이터는 하나 이상의 *Activity* 그룹을 포함합니다. 매개변수는 99 페이지의 『[활동 보고서 메시지 데이터](#)』에 설명되어 있습니다.

회계 및 통계 메시지

큐 관리자는 IBM MQ 애플리케이션으로 수행되는 MQI 조작에 대한 정보를 기록하거나 IBM MQ 시스템에서 발생하는 활동에 대한 정보를 기록하기 위해 회계 및 통계 메시지를 생성합니다.

회계 메시지

회계 메시지는 IBM MQ 애플리케이션으로 수행된 MQI 조작에 관한 정보를 기록하는 데 사용됩니다. [127 페이지의 『회계 메시지』](#)의 내용을 참조하십시오.

통계 메시지

통계 메시지는 IBM MQ 시스템에서 발생하는 활동에 대한 정보를 기록하는 데 사용됩니다. [130 페이지의 『통계 메시지』](#)의 내용을 참조하십시오. 통계 메시지에 기록된 일부 활동은 내부 큐 관리자 조작과 관련됩니다.

z/OS 여기서 설명한 대로 계정 메시지 및 통계 메시지는 IBM MQ for z/OS에서 사용할 수 없지만, 동일한 기능을 SMF(System Management Facility)를 통해 사용할 수 있습니다.

회계 및 통계 메시지는 두 개의 시스템 큐 중 하나에 전달됩니다. 사용자 애플리케이션은 이 시스템 큐에서 메시지를 검색하여 다음과 같은 다양한 용도로 기록된 정보를 사용할 수 있습니다.

- 애플리케이션 자원 사용을 설명합니다.
- 애플리케이션 활동을 기록합니다.
- 용량 계획.
- 큐 관리자 네트워크에서 문제점을 감지합니다.
- 큐 관리자 네트워크에서 문제점의 원인을 판별하는 데 도움을 줍니다.
- 큐 관리자 네트워크의 효율성을 향상시킵니다.
- 큐 관리자 네트워크의 실행에 익숙해집니다.
- 큐 관리자 네트워크가 올바르게 실행 중인지 확인합니다.

관련 개념

[299 페이지의 『Using System Management Facility』](#)

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

회계 메시지

회계 메시지는 IBM MQ 애플리케이션으로 수행된 MQI 조작에 관한 정보를 기록합니다. 회계 메시지는 많은 PCF 구조를 포함한 PCF 메시지입니다.

애플리케이션이 큐 관리자로부터 연결이 끊어지면 회계 메시지가 생성되어 시스템 회계 큐 (SYSTEM.ADMIN.ACCOUNTING.QUEUE)로 전달됩니다. 장기 실행 IBM MQ 애플리케이션은 다음의 경우 중간 회계 메시지가 생성됩니다.

- 연결이 설정된 이후의 시간이 구성된 간격을 초과할 때.
- 마지막 중간 회계 메시지 이후의 시간이 구성된 간격을 초과할 때.

회계 메시지는 다음 범주에 있습니다.

MQI 회계 메시지

MQI 회계 메시지는 큐 관리자에 대한 연결을 사용하여 수행된 MQI 호출에 관련한 정보를 포함합니다.

큐 회계 메시지

큐 회계 메시지는 큐 관리자에 대한 연결을 사용하여 수행된 MQI 호출 수에 관련한 큐별로 그룹화된 정보를 포함합니다.

각 큐 회계 메시지는 최대 100개의 레코드를 포함할 수 있으며, 모든 레코드가 특정 큐에 대해 애플리케이션으로 수행된 활동에 관련되어 있습니다.

회계 메시지는 로컬 큐에 대해서만 기록됩니다. 애플리케이션이 알리어스 큐에 대해 MQI 호출을 수행하면 기본 큐에 대해 회계 데이터가 기록되고 리모트 큐의 경우에는 전송 큐에 대해 회계 데이터가 기록됩니다.

참고: 정보가 IBM MQ 애플리케이션이 수행하는 MQI 조작과 관련되므로, 조작이 애플리케이션에 의해 해당 큐에 직접적으로 있지 않는 한 정보에는 스트리밍 큐와 관련된 조작이 포함되지 않습니다.

관련 참조

[144 페이지의 『MQI 회계 메시지 데이터』](#)

이 페이지를 사용하여 MQI 회계 메시지의 구조를 봅니다.

155 페이지의 『큐 회계 메시지 데이터』

이 페이지를 사용하여 큐 회계 메시지의 구조를 봅니다.

회계 메시지 형식

회계 메시지는 메시지 디스크립터와 메시지 데이터로 이루어진 PCF 필드 세트로 구성됩니다.

메시지 디스크립터

- 회계 메시지 MQMD(메시지 디스크립터)

회계 메시지 데이터

- 회계 메시지 MQCFH(PCF 헤더)
- 항상 리턴되는 회계 메시지 데이터
- 사용 가능한 경우에 리턴되는 회계 메시지 데이터

회계 메시지 MQCFH(PCF 헤더)는 회계 데이터가 기록된 간격 및 애플리케이션에 대한 정보를 포함합니다.

회계 메시지 데이터는 회계 정보를 저장하는 PCF 매개변수로 구성됩니다. 회계 메시지의 콘텐츠는 다음과 같은 메시지 범주에 따라 다릅니다.

MQI 회계 메시지

MQI 회계 메시지 데이터는 많은 PCF 매개변수로 구성됩니다(PCF 그룹은 없음).

큐 회계 메시지

큐 회계 메시지 데이터는 많은 PCF 매개변수로 구성되며 범위는 1 ~ 100 *QAccountingData* PCF 그룹입니다.

회계 데이터를 수집한 모든 큐에 *QAccountingData* PCF 그룹이 하나씩 있습니다. 애플리케이션이 100개를 초과한 큐에 액세스하면 여러 회계 메시지가 생성됩니다. 각 메시지마다 MQCFH(PCF 헤더)의 *SeqNumber* 가 그에 맞춰 업데이트되고, 순서의 마지막 메시지에는 MQCFH의 *Control* 매개변수가 MQCFC_LAST로 지정됩니다.

회계 정보 콜렉션

큐와 큐 관리자 속성을 사용하여 회계 정보 콜렉션을 제어합니다. MQCONNX 옵션을 사용하여 연결 레벨에서 콜렉션을 제어할 수도 있습니다.

MQI 회계 정보의 콜렉션 제어

큐 관리자 속성 ACCTMQI를 사용하여 MQI 회계 정보 콜렉션을 제어합니다.

이 속성의 값을 변경하려면 MQSC 명령, ALTER QMGR을 사용하고 매개변수 ACCTMQI를 지정하십시오. 회계 메시지는 회계가 사용된 후에 시작한 연결에 대해서만 생성됩니다. **ACCTMQI** 매개변수의 가능한 값은 다음과 같습니다.

켜짐

모든 큐 관리자 연결에 대한 MQI 회계 정보를 수집합니다.

OFF

MQI 회계 정보를 수집하지 않습니다. 이는 기본값입니다.

예를 들어, MQI 회계 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR ACCTMQI(ON)
```

큐 회계 정보

큐 속성 ACCTQ와 큐 관리자 속성 ACCTQ를 사용하여 큐 회계 정보의 콜렉션을 제어합니다.

큐 속성의 값을 변경하려면 MQSC 명령, ALTER QLOCAL을 사용하고 매개변수 **ACCTQ**를 지정하십시오. 회계 메시지는 회계가 사용된 후에 시작한 연결에 대해서만 생성됩니다. 이 값에 대한 변경사항은 속성 변경 후 발생하는 큐 관리자에 대한 연결에만 유효하다는 점을 참고하십시오.

큐 속성 ACCTQ의 가능한 값은 다음과 같습니다.

켜짐

이 큐에 대한 큐 회계 정보는 큐를 여는 모든 큐 관리자 연결에 대해 수집됩니다.

OFF

이 큐에 대한 큐 회계 정보를 수집하지 않습니다.

QMGR

이 큐에 대한 큐 회계 정보 콜렉션은 큐 관리자 속성 ACCTQ의 값에 따라 제어됩니다. 이는 기본값입니다.

큐 관리자 속성의 값을 변경하려면 MQSC 명령, ALTER QMGR을 사용하고 매개변수 ACCTQ를 지정하십시오. 큐 관리자 속성 ACCTQ의 가능한 값은 다음과 같습니다.

켜짐

큐 속성 ACCTQ가 QMGR로 설정된 큐에 대해서 큐 회계 정보를 수집합니다.

OFF

큐 속성 ACCTQ가 QMGR로 설정된 큐에 대해 큐 회계 정보를 수집하지 않습니다. 이는 기본값입니다.

NONE

큐 속성 ACCTQ와 무관하게 모든 큐에 대한 큐 회계 정보 콜렉션이 사용 불가능합니다.

큐 관리자 속성, ACCTQ가 NONE으로 설정된 경우에는 큐 속성 ACCTQ와 무관하게 모든 큐에 대한 큐 회계 정보 콜렉션이 사용 안함으로 설정됩니다.

예를 들어, 큐 Q1에 대한 회계 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

큐 속성 ACCTQ를 QMGR로 지정하는 모든 큐에 대한 회계 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR ACCTQ(ON)
```

회계 정보 콜렉션 제어를 위한 MQCONNX 옵션

MQCONNX 호출의 **ConnectOpts** 매개변수를 사용하여 큐 관리자 속성 ACCTMQI 및 ACCTQ의 유효 값을 대체해서, 연결 레벨에서 MQI와 큐 회계 정보 모두의 콜렉션을 수정합니다.

ConnectOpts 매개변수의 가능한 값은 다음과 같습니다.

MQCNO_ACCOUNTING_MQI_ENABLED

큐 관리자 속성 ACCTMQI의 값이 OFF로 지정된 경우 MQI 회계가 이 연결에 사용됩니다. 이는 큐 관리자 속성 ACCTMQI를 ON으로 지정한 경우와 동등합니다.

큐 관리자 속성 ACCTMQI의 값이 OFF로 지정되면 이 속성이 적용되지 않습니다.

MQCNO_ACCOUNTING_MQI_DISABLED

큐 관리자 속성 ACCTMQI의 값이 ON으로 지정된 경우 MQI 회계가 이 연결에 사용 불가능합니다. 이는 큐 관리자 속성 ACCTMQI를 OFF로 지정한 경우와 동등합니다.

큐 관리자 속성 ACCTMQI의 값이 ON으로 지정되면 이 속성이 적용되지 않습니다.

MQCNO_ACCOUNTING_Q_ENABLED

큐 관리자 속성 ACCTQ의 값이 OFF로 지정된 경우 큐 회계가 이 연결에 사용됩니다. ACCTQ가 QMGR로 지정된 모든 큐가 큐 회계에 대해 사용됩니다. 이는 큐 관리자 속성 ACCTQ를 ON으로 지정한 경우와 동등합니다.

큐 관리자 속성 ACCTQ의 값이 OFF로 지정되면 이 속성이 적용되지 않습니다.

MQCNO_ACCOUNTING_Q_DISABLED

큐 관리자 속성 ACCTQ의 값이 ON으로 지정된 경우 큐 회계가 이 연결에 사용 불가능합니다. 이는 큐 관리자 속성 ACCTQ를 OFF로 지정한 경우와 동등합니다.

큐 관리자 속성 ACCTQ의 값이 ON으로 지정되면 이 속성이 적용되지 않습니다.

이러한 대체는 기본적으로 사용 불가능합니다. 이 대체를 사용하려면 큐 관리자 속성 ACCTCONO를 ENABLED로 설정하십시오. 개별 연결에 대한 회계 대체를 사용하려면 다음 MQSC 명령을 사용하십시오.

ALTER QMGR ACCTCONO(ENABLED)

애플리케이션 연결이 끊어질 때의 회계 메시지 생성

회계 메시지는 큐 관리자로부터의 애플리케이션 연결이 끊어질 때 생성됩니다. 장기 실행 IBM MQ 애플리케이션의 경우에는 중간 회계 메시지도 기록됩니다.

회계 메시지는 애플리케이션 연결이 끊어질 때 다음 중 한 방식으로 생성됩니다.

- 애플리케이션이 MQDISC 호출을 발행함
- 큐 관리자가 애플리케이션이 종료한 사실을 인식함

장기 실행 IBM MQ 애플리케이션은 연결이 설정된 이후 또는 기록된 마지막 중간 회계 메시지 이후의 간격이 구성된 간격을 초과할 때 중간 회계 메시지가 기록됩니다. 큐 관리자 속성, ACCTINT는 중간 회계 메시지가 자동으로 기록되기 시작하는 시간(초)을 지정합니다. 회계 메시지는 애플리케이션이 큐 관리자와 상호작용할 때에만 생성되므로, MQI 요청을 실행하지 않은 채로 오랜 기간 큐 관리자에 연결되어 있는 애플리케이션은 회계 간격이 완료된 후 첫 번째 MQI 요청이 실행될 때까지 회계 메시지를 생성하지 않습니다.

기본 회계 간격은 1800초(30분)입니다. 예를 들어, 회계 간격을 900초(15분)로 변경하려면 다음 MQSC 명령을 사용하십시오.

ALTER QMGR ACCTINT(900)

통계 메시지

통계 메시지는 IBM MQ 시스템에서 발생하는 활동에 대한 정보를 기록합니다. 통계 메시지는 많은 PCF 구조를 포함한 PCF 메시지입니다.

통계 메시지는 구성된 간격으로 시스템 큐(SYSTEM.ADMIN.STATISTICS.QUEUE)에 전달됩니다.

통계 메시지는 다음 범주에 있습니다.

MQI 통계 메시지

MQI 통계 메시지는 구성된 간격에 수행된 MQI 호출의 수에 관련한 정보를 포함합니다. 예를 들어, 큐 관리자가 실행한 MQI 호출의 수가 정보에 포함될 수 있습니다.

큐 통계 메시지

큐 통계 메시지는 구성된 간격 중 큐의 활동에 관련된 정보를 포함합니다. 큐에 넣고 큐에서 검색한 메시지 수와 큐에서 처리된 총 바이트 수가 정보에 포함됩니다.

각 큐 통계 메시지는 통계가 수집된 큐별 활동에 각 레코드가 관련되어 있는 최대 100개의 레코드를 포함할 수 있습니다.

통계 메시지는 로컬 큐에 대해서만 기록됩니다. 애플리케이션이 알리어스 큐에 대해 MQI 호출을 수행하면 기본 큐에 대해 통계 데이터가 기록되고 리모트 큐의 경우에는 전송 큐에 대해 통계 데이터가 기록됩니다.

채널 통계 메시지

채널 통계 메시지는 구성된 간격 중 채널의 활동에 관련된 정보를 포함합니다. 예를 들어, 채널에서 전송된 메시지 수나 채널에서 전송된 바이트 수와 같은 정보일 수 있습니다.

각 채널 통계 메시지는 통계가 수집된 채널별 활동에 각 레코드가 관련되어 있는 최대 100개의 레코드를 포함합니다.

참고: 정보가 IBM MQ 시스템에서 발생하는 활동과 관련되므로 정보에는 스트리밍 큐와 관련된 조작이 포함됩니다.

관련 참조

131 페이지의 『MQI 통계 콜렉션 제어』

큐 관리자 속성 STATMQI를 사용하여 MQI 통계 정보 콜렉션을 제어합니다.

131 페이지의 『큐 통계 콜렉션 제어』

큐 속성 STATQ와 큐 관리자 속성 STATQ를 사용하여 큐 통계 정보의 콜렉션을 제어합니다.

132 페이지의 『채널 통계 콜렉션 제어』

채널 속성 STATCHL을 사용하여 채널 통계 정보 콜렉션을 제어합니다. 큐 관리자 속성을 사용하여 정보 콜렉션을 제어할 수도 있습니다.

통계 메시지 형식

통계 메시지는 메시지 디스크립터와 메시지 데이터로 이루어진 PCF 필드 세트 구성됩니다.

메시지 디스크립터

- 통계 메시지 MQMD(메시지 디스크립터)

회계 메시지 데이터

- 통계 메시지 MQCFH(PCF 헤더)
- 항상 리턴되는 통계 메시지 데이터
- 사용 가능한 경우에 리턴되는 통계 메시지 데이터

통계 메시지 MQCFH(PCF 헤더)는 통계 데이터가 기록된 간격에 대한 정보를 포함합니다.

통계 메시지 데이터는 통계 정보를 저장하는 PCF 매개변수로 구성됩니다. 통계 메시지의 콘텐츠는 다음과 같은 메시지 범주에 따라 다릅니다.

MQI 통계 메시지

MQI 통계 메시지 데이터는 많은 PCF 매개변수로 구성됩니다(PCF 그룹은 없음).

큐 통계 메시지

큐 통계 메시지 데이터는 많은 PCF 매개변수로 구성되며 범위는 1 ~ 100 *QStatisticsData* PCF 그룹입니다.

간격의 활성화된 모든 큐에 *QStatisticsData* PCF 그룹이 하나씩 있습니다. 간격 중에 100개를 초과한 큐가 활성화되면 여러 통계 메시지가 생성됩니다. 각 메시지마다 MQCFH(PCF 헤더)의 *SeqNumber*가 그에 맞춰 업데이트되고, 순서의 마지막 메시지에는 MQCFH의 *Control* 매개변수가 MQCFC_LAST로 지정됩니다.

채널 통계 메시지

채널 통계 메시지 데이터는 많은 PCF 매개변수로 구성되며 범위는 1 ~ 100 *ChlStatisticsData* PCF 그룹입니다.

간격의 활성화된 모든 채널에 *ChlStatisticsData* PCF 그룹이 하나씩 있습니다. 간격 중에 100개를 초과한 채널이 활성화되면 여러 통계 메시지가 생성됩니다. 각 메시지마다 MQCFH(PCF 헤더)의 *SeqNumber*가 그에 맞춰 업데이트되고, 순서의 마지막 메시지에는 MQCFH의 *Control* 매개변수가 MQCFC_LAST로 지정됩니다.

통계 정보 콜렉션

큐, 큐 관리자, 채널 속성을 사용하여 통계 정보 콜렉션을 제어합니다.

MQI 통계 콜렉션 제어

큐 관리자 속성 STATMQI를 사용하여 MQI 통계 정보 콜렉션을 제어합니다.

이 속성의 값을 변경하려면 MQSC 명령, ALTER QMGR을 사용하고 매개변수 **STATMQI**를 지정하십시오. 통계 메시지는 통계 콜렉션이 사용 가능하게 된 후에 열린 큐에 대해서만 생성됩니다. **STATMQI** 매개변수의 가능한 값은 다음과 같습니다.

켜짐

모든 큐 관리자 연결에 대한 MQI 통계 정보를 수집합니다.

OFF

MQI 통계 정보를 수집하지 않습니다. 이는 기본값입니다.

예를 들어, MQI 통계 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR STATMQI(ON)
```

큐 통계 콜렉션 제어

큐 속성 STATQ와 큐 관리자 속성 STATQ를 사용하여 큐 통계 정보의 콜렉션을 제어합니다.

개별 큐나 여러 큐에 대해서 큐 통계 정보 콜렉션을 사용 가능 또는 불가능하게 할 수 있습니다. 개별 큐를 제어하려면 큐 속성 STATQ를 설정하십시오. 큐 관리자 속성 STATQ를 사용하여 큐 관리자 레벨에서 큐 통계 정보 콜렉

션을 사용 또는 사용 안함으로 설정합니다. 큐 속성 STATQ가 QMGR 값으로 지정된 모든 큐의 경우에는 큐 관리자 레벨에서 큐 통계 정보 콜렉션이 제어됩니다.

큐 통계는 통계 콜렉션이 사용 가능하게 된 후에 열린 IBM MQ MQI 오브젝트 핸들을 사용하는 조작에 대해서만 증분됩니다.

큐 통계 메시지는 이전 기간에 통계 데이터가 수집된 큐에 대해서만 생성됩니다.

동일한 큐에는 여러 오브젝트 핸들을 통해 여러 넣기 조작과 가져오기 조작이 있을 수 있습니다. 일부 오브젝트 핸들은 통계 콜렉션이 사용 가능하게 되기 전에 열렸을 수 있지만 다른 핸들은 그 후에 열렸습니다. 따라서 큐 통계가 모든 넣기 조작 및 가져오기 조작이 아닌 이들 일부 조작의 활동을 기록할 수 있습니다.

큐 통계가 모든 애플리케이션의 활동을 기록하게 하려면 모니터링 중인 단일 또는 다중 큐의 오브젝트 핸들을 닫고 새 오브젝트 핸들을 열어야 합니다. 이를 위한 최상의 방법은 통계 콜렉션을 사용 가능하게 한 후 모든 애플리케이션을 종료하고 다시 시작하는 것입니다.

큐 속성 STATQ의 값을 변경하려면 MQSC 명령, ALTER QLOCAL을 사용하고 매개변수 STATQ를 지정하십시오. 큐 속성 STATQ의 가능한 값은 다음과 같습니다.

켜짐

큐 통계 정보는 큐를 여는 모든 큐 관리자 연결에 대해 수집됩니다.

OFF

이 큐에 대한 큐 통계 정보를 수집하지 않습니다.

QMGR

이 큐에 대한 큐 통계 정보 콜렉션은 큐 관리자 속성, STATQ의 값에 따라 제어됩니다. 이는 기본값입니다.

큐 관리자 속성 STATQ의 값을 변경하려면 MQSC 명령, ALTER QMGR을 사용하고 매개변수 STATQ를 지정하십시오. 큐 관리자 속성 STATQ의 가능한 값은 다음과 같습니다.

켜짐

큐 속성 STATQ가 QMGR로 설정된 큐에 대해서 큐 통계 정보를 수집합니다.

OFF

큐 속성 STATQ가 QMGR로 설정된 큐에 대해 큐 통계 정보를 수집하지 않습니다. 이는 기본값입니다.

NONE

큐 속성 STATQ와 무관하게 모든 큐에 대한 큐 통계 정보 콜렉션이 사용 불가능합니다.

큐 관리자 속성 STATQ가 NONE으로 설정된 경우에는 큐 속성 STATQ와 무관하게 모든 큐에 대한 큐 통계 정보 콜렉션이 사용 안함으로 설정됩니다.

예를 들어, 큐 Q1에 대한 통계 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QLOCAL(Q1) STATQ(ON)
```

큐 속성 STATQ를 QMGR로 지정하는 모든 큐에 대한 통계 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR STATQ(ON)
```

채널 통계 콜렉션 제어

채널 속성 STATCHL을 사용하여 채널 통계 정보 콜렉션을 제어합니다. 큐 관리자 속성을 사용하여 정보 콜렉션을 제어할 수도 있습니다.

개별 채널이나 여러 채널에 대해서 채널 통계 정보 콜렉션을 사용 가능 또는 불가능하게 할 수 있습니다. 개별 채널을 제어하려면 채널 통계 정보 콜렉션을 사용 또는 사용하지 않도록 채널 속성 STATCHL을 설정해야 합니다. 많은 채널을 함께 제어하려면 큐 관리자 속성 STATCHL을 사용하여 큐 관리자 레벨에서 채널 통계 정보 콜렉션을 사용 가능 또는 사용 불가능으로 설정할 수 있습니다. 채널 속성 STATCHL이 값 QMGR을 사용하여 지정된 모든 채널에 대해 채널 통계 정보 콜렉션이 큐 관리자 레벨에서 제어됩니다.

자동으로 정의된 클러스터 송신자 채널은 IBM MQ 오브젝트가 아니므로 채널 오브젝트와 동일한 방식으로 속성을 보유하지 않습니다. 자동으로 정의된 클러스터 송신자 채널을 제어하려면 큐 관리자 속성 STATACLS를 사용

하십시오. 이 속성은 큐 관리자 내의 자동으로 정의된 클러스터 송신자 채널이 채널 통계 정보 콜렉션에 대해 사용 가능 또는 불가능한지 여부를 판별합니다.

채널 통계 정보 콜렉션을 세 가지 모니터링 레벨인 낮음, 중간 또는 높음 중 하나로 설정할 수 있습니다. 오브젝트 레벨 또는 큐 관리자 레벨에서 모니터링 레벨을 설정할 수 있습니다. 사용할 레벨의 선택은 시스템에 따라 다릅니다. 통계 정보 데이터를 수집하려면 계산 비용이 상대적으로 비싼 몇 가지 지시사항이 필요할 수 있으므로, 채널 통계 정보 콜렉션의 영향을 줄이기 위해 중간 및 낮음 모니터링 옵션은 항상 데이터를 수집하기 보다는 정기적으로 데이터 샘플을 측정합니다. [133 페이지의 표 22](#)에서는 채널 통계 정보 콜렉션에 사용 가능한 레벨을 요약합니다.

표 22. 채널 통계 정보 콜렉션의 세부사항 레벨		
레벨	설명	사용법
낮음	일정한 간격으로 작은 데이터 샘플을 측정합니다.	높은 볼륨의 메시지를 처리하는 오브젝트의 경우.
중간	일정한 간격으로 데이터 샘플을 측정합니다.	대부분의 오브젝트의 경우.
높음	일정한 간격으로 모든 데이터를 측정합니다.	초당 몇 개의 메시지만을 처리하는 오브젝트의 경우(최신 정보가 중요함).

채널 속성 STATCHL의 값을 변경하려면 MQSC 명령, ALTER CHANNEL을 사용하고 매개변수 STATCHL을 지정하십시오.

큐 관리자 속성 STATCHL의 값을 변경하려면 MQSC 명령, ALTER QMGR을 사용하고 매개변수 STATCHL을 지정하십시오.

큐 관리자 속성 STATACLS의 값을 변경하려면 MQSC 명령, ALTER QMGR을 사용하고 매개변수 STATACLS를 지정하십시오.

채널 속성, STATCHL의 가능한 값은 다음과 같습니다.

LOW

채널 통계 정보가 낮은 세부사항 레벨로 수집됩니다.

MEDIUM

채널 통계 정보가 중간 세부사항 레벨로 수집됩니다.

HIGH

채널 통계 정보가 높은 세부사항 레벨로 수집됩니다.


OFF

이 채널에 대한 채널 통계 정보가 수집되지 않습니다.

QMGR

채널 속성이 QMGR로 설정됩니다. 이 채널에 대한 통계 정보 콜렉션은 큐 관리자 속성, STATCHL의 값으로 제어됩니다.

이는 기본값입니다.

 z/OS ~에 z/OS 시스템에서 이 매개변수를 활성화하면 선택한 값에 관계없이 통계 데이터 수집이 활성화됩니다. LOW, MEDIUM 또는 HIGH를 지정해도 결과에는 차이가 없습니다. 채널 회계 레코드를 수집하려면 이 매개변수를 사용해야 합니다.

큐 관리자 속성, STATCHL의 가능한 값은 다음과 같습니다.

LOW

채널 속성 STATCHL이 QMGR로 설정된 모든 채널에 대한 채널 통계 정보가 낮은 세부사항 레벨로 수집됩니다.

MEDIUM

채널 속성 STATCHL이 QMGR로 설정된 모든 채널에 대한 채널 통계 정보가 중간 세부사항 레벨로 수집됩니다.

HIGH

채널 속성 STATCHL이 QMGR로 설정된 모든 채널에 대한 채널 통계 정보가 높은 세부사항 레벨로 수집됩니다.

OFF

채널 속성 STATCHL이 QMGR로 설정된 모든 채널에 대한 채널 통계 정보가 수집되지 않습니다. 이는 기본값입니다.

NONE

채널 속성 STATCHL과 무관하게 모든 채널에 대한 채널 통계 정보 콜렉션이 사용 불가능합니다.

z/OS ~에z/OS 시스템에서 이 매개변수를 활성화하면 선택한 값에 관계없이 통계 데이터 수집이 활성화됩니다. LOW, MEDIUM 또는 HIGH를 지정해도 결과에는 차이가 없습니다.채널 회계 레코드를 수집하려면 이 매개변수를 사용해야 합니다.

큐 관리자 속성, STATACLS의 가능한 값은 다음과 같습니다.

LOW

자동으로 정의된 클러스터 송신자 채널에 대한 통계 정보가 낮은 세부사항 레벨로 수집됩니다.

MEDIUM

자동으로 정의된 클러스터 송신자 채널에 대한 통계 정보가 중간 세부사항 레벨로 수집됩니다.

HIGH

자동으로 정의된 클러스터 송신자 채널에 대한 통계 정보가 높은 세부사항 레벨로 수집됩니다.

OFF

자동으로 정의된 클러스터 송신자 채널에 대한 통계 정보가 수집되지 않습니다.

QMGR

자동으로 정의된 클러스터 송신자 채널에 대한 통계 정보 콜렉션은 큐 관리자 속성, STATCHL의 값으로 제어됩니다.

이는 기본값입니다.

z/OS ~에z/OS 시스템에서 이 매개변수를 활성화하면 선택한 값에 관계없이 통계 데이터 수집이 활성화됩니다. LOW, MEDIUM 또는 HIGH를 지정해도 결과에는 차이가 없습니다.채널 회계 레코드를 수집하려면 이 매개변수를 사용해야 합니다.

예를 들어, 송신자 채널 QM1.TO.QM2에 대한 통계 정보 콜렉션을 중간 세부사항 레벨로 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

채널 속성 STATCHL을 QMGR로 지정하는 모든 채널에 대한 통계 정보 콜렉션을 중간 세부사항 레벨로 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR STATCHL(MEDIUM)
```

모든 자동으로 정의된 클러스터 송신자 채널에 대한 통계 정보 콜렉션을 중간 세부사항 레벨로 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR STATACLS(MEDIUM)
```

통계 메시지 생성

통계 메시지는 구성된 간격에, 큐 관리자가 제어된 방식으로 시스템 종료할 때에 생성됩니다.

구성된 간격은 통계 메시지 생성 사이의 간격(초)을 지정하는 STATINT 큐 관리자 속성으로 제어됩니다. 기본 통계 간격은 1800초(30분)입니다. 통계 간격을 변경하려면 MQSC 명령 ALTER QMGR을 사용하고 **STATINT** 매개 변수를 지정하십시오. 예를 들어, 통계 간격을 900초(15분)로 변경하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR STATINT(900)
```

통계 콜렉션 간격이 만기되려 하기 전에 현재 수집한 통계 데이터를 통계 큐에 쓰려면 다음 MQSC 명령 RESET QMGR TYPE(STATISTICS)를 사용하십시오. 이 명령을 발행하면 통계 큐에 수집한 통계 데이터가 기록되고 새 통계 데이터 콜렉션 간격이 시작됩니다.

회계 및 통계 정보 표시

회계 및 통계 메시지에 기록된 정보를 사용하려면 **amqsmn** 샘플 프로그램과 같은 애플리케이션을 실행하여 기록된 정보를 적합한 형식으로 변환하십시오.

회계 및 통계 메시지는 시스템 회계 및 통계 큐에 기록됩니다. **amqsmn**은 IBM MQ와 함께 제공되는 샘플 프로그램으로, 회계 및 통계 큐에서 메시지를 처리하여 화면에 읽기 쉬운 양식으로 정보를 표시합니다.

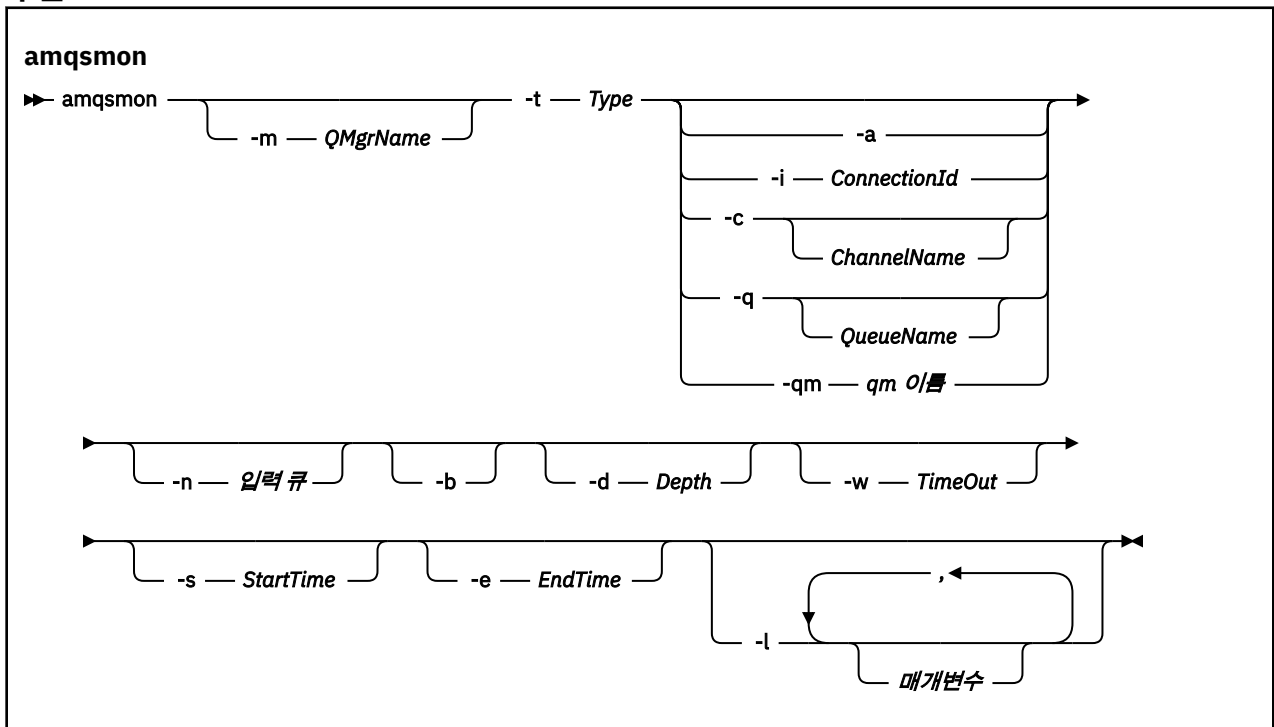
amqsmn은 샘플 프로그램이므로 회계 또는 통계 메시지를 처리할 자신만의 애플리케이션을 쓰기 위한 템플릿으로 제공된 소스 코드를 사용하거나 **amqsmn** 소스 코드를 자신의 특정 요구사항에 맞게 수정할 수 있습니다.

amqsmn(형식화된 모니터링 정보 표시)

amqsmn 샘플 프로그램을 사용하여 회계 및 통계 메시지 내에 포함된 정보를 읽기 쉬운 형식으로 표시합니다.

amqsmn 프로그램은 계정 큐 SYSTEM.ADMIN.ACCOUNTING.QUEUE입니다. 통계 큐 SYSTEM.ADMIN.STATISTICS.QUEUE입니다.

구문



필수 매개변수

-t Type

처리할 메시지의 유형. Type을 다음 중 하나로 지정하십시오.

회계

회계 레코드가 처리됩니다. `-n` 매개변수를 사용하여 읽을 특정 큐를 선택하지 않은 경우 시스템 큐 `SYSTEM.ADMIN.ACCOUNTING.QUEUE` 에서 메시지를 읽습니다.

통계

통계 레코드가 처리됩니다. 시스템 큐에서 메시지를 읽습니다. `SYSTEM.ADMIN.STATISTICS.QUEUE`

V 9.4.0 `-n` 매개변수를 사용하여 읽을 특정 큐를 선택하지 않은 경우.

선택적 매개변수

-m QMgrName

회계 또는 통계 메시지를 처리할 큐 관리자의 이름.

이 매개변수를 지정하지 않으면 기본 큐 관리자가 사용됩니다.

-a

MQI 레코드만을 포함한 프로세스 메시지.

MQI 레코드만 표시합니다. MQI 레코드를 포함하지 않은 메시지는 메시지를 읽는 큐에 항상 남습니다.

-q QueueName

*QueueName*은 선택적 매개변수입니다.

*QueueName*을 제공하지 않을 경우: 큐 회계와 큐 통계 레코드만 표시합니다.

*QueueName*을 제공할 경우: *QueueName*에 지정된 큐에 대한 큐 회계와 큐 통계 레코드만 표시합니다.

`-b` 가 지정되지 않으면 레코드가 발생한 계정 및 통계 메시지가 제거됩니다. 계정 및 통계 메시지에는 다른 큐의 레코드도 포함될 수 있으므로 `-b` 가 지정되지 않은 경우 표시되지 않은 레코드를 제거할 수 있습니다.

-c ChannelName

*ChannelName*은 선택적 매개변수입니다.

*ChannelName*을 제공하지 않을 경우: 채널 통계 레코드만 표시합니다.

*ChannelName*을 제공할 경우: *ChannelName*에 지정된 채널의 채널 통계 레코드만 표시합니다.

`-b` 가 지정되지 않으면 레코드가 발생한 통계 메시지가 제거됩니다. 통계 메시지에는 다른 채널의 레코드도 포함될 수 있으므로 `-b` 가 지정되지 않으면 표시되지 않은 레코드를 제거할 수 있습니다.

이 매개변수는 통계 메시지만 표시할 때(`-t statistics`) 사용 가능합니다.

V 9.4.0 -qm qm 이름

이 매개변수는 선택사항이지만 지정하는 경우 큐 관리자 이름을 제공해야 합니다.

지정된 큐 관리자가 생성한 계정 또는 통계 메시지만 큐에서 읽도록 지정합니다. 일반적으로 여러 큐 관리자의 메시지가 중앙 큐 관리자로 전달된 경우에만 사용됩니다.

-i ConnectionId

*ConnectionId*에 지정된 연결 ID와 관련이 있는 레코드만 표시합니다.

이 매개변수는 회계 메시지만 표시할 때(`-t accounting`) 사용 가능합니다.

`-b` 가 지정되지 않으면 레코드가 발생한 통계 메시지가 제거됩니다. 통계 메시지에는 다른 채널의 레코드도 포함될 수 있으므로 `-b` 가 지정되지 않으면 표시되지 않은 레코드를 제거할 수 있습니다.

V 9.4.0 -n 입력 큐

계정 또는 통계 메시지를 읽을 큐입니다. 이를 지정하지 않으면 선택한 `-t <type>` 에 대한 디폴트 큐가 사용 됩니다.

-b

메시지를 찾아봅니다.

메시지는 비파괴적으로 검색됩니다.

-d Depth

처리 가능한 최대 메시지 수.

이 매개변수를 지정하지 않으면 무제한 수의 메시지가 처리될 수 있습니다.

-w TimeOut

메시지가 사용 가능하게 될 때까지 대기할 최대 시간(초).

이 매개변수를 지정하지 않을 경우 처리할 메시지가 더 이상 없으면 `amqsmn`이 종료됩니다.

-s StartTime

지정된 `StartTime` 후에만 넣은 프로세스 메시지.

`StartTime`은 `yyyy-mm-dd hh.mm.ss` 형식으로 지정됩니다. 시간 없이 날짜가 지정되면 시간은 기본적으로 지정된 날짜에 `00.00.00`으로 설정됩니다. 시간은 GMT입니다.

이 매개변수를 지정하지 않을 경우에 미치는 영향은 [참고 1](#)을 참조하십시오.

-e EndTime

지정된 `EndTime` 이전에만 넣은 프로세스 메시지.

`EndTime`은 `yyyy-mm-dd hh.mm.ss` 형식으로 지정됩니다. 시간 없이 날짜가 지정되면 시간은 기본적으로 지정된 날짜에 `23.59.59`으로 설정됩니다. 시간은 GMT입니다.

이 매개변수를 지정하지 않을 경우에 미치는 영향은 [참고 1](#)을 참조하십시오.

-l Parameter

처리된 레코드에서 선택한 필드만 표시합니다. `Parameter`는 쉼표로 구분된 정수 값 목록으로, 각 정수 값이 필드의 숫자 상수에 매핑됩니다(`amqsmn` 예 5 참조).

이 매개변수를 지정하지 않으면 사용 가능한 모든 필드가 표시됩니다.

참고:

1. `-s StartTime` 또는 `-e EndTime`을 지정하지 않을 경우 처리 가능한 메시지가 넣기 시간까지 제한되지 않습니다.

amqsmn(형식화된 모니터링 정보 표시) 예

이 페이지를 사용하여 `amqsmn`(형식화된 모니터링 정보 표시) 샘플 프로그램 실행 예를 봅니다.

1. 속성에 대한 설명은 [175 페이지의 『큐 통계 메시지 데이터』](#)의 내용을 참조하십시오.

다음 명령은 큐 관리자 `saturn.queue.manager`의 모든 MQI 통계 메시지를 표시합니다.

```
amqsmn -m saturn.queue.manager -t statistics -a
```

이 명령의 출력은 다음과 같습니다.

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
```

```

OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0

```

2. 다음 명령은 큐 관리자 saturn.queue.manager의 LOCALQ 큐에 대한 모든 큐 통계 메시지를 표시합니다.

```
amqsmon -m saturn.queue.manager -t statistics -q LOCALQ
```

이 명령의 출력은 다음과 같습니다.

```

RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics:
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  QueueDefinitionType: Local
  QMinDepth: 0
  QMaxDepth: 18
  AverageQueueTime: [29827281, 0]
  PutCount: [26, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [88, 0]
  GetCount: [18, 0]
  GetBytes: [52, 0]
  GetFailCount: 0
  BrowseCount: [0, 0]
  BrowseBytes: [0, 0]
  BrowseFailCount: 1
  NonQueuedMsgCount: 0
  ExpiredMsgCount: 0
  PurgedMsgCount: 0

```

3. 다음 명령은 큐 관리자 saturn.queue.manager에서 2005년 4월 30일 15:30 이후에 기록된 모든 통계 메시지를 표시합니다.

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

이 명령의 출력은 다음과 같습니다.

```

RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'

```

```

IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0]
...
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics: 0
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...
QueueStatistics: 1
  QueueName: 'SAMPLEQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...

```

4. 속성에 대한 설명은 155 페이지의 『큐 회계 메시지 데이터』의 내용을 참조하십시오.

다음 명령은 큐 관리자 saturn.queue.manager에서 2005년 4월 30일에 기록된 모든 회계 메시지를 표시합니다.

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

이 명령의 출력은 다음과 같습니다.

```

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

```

```
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
SeqNumber: 0
ApplicationName: 'runmqsc'
ApplicationPid: 8615
ApplicationTid: 1
...
```

5. 다음 명령은 회계 큐를 찾아보고 MQI 회계 정보가 사용 가능한 모든 애플리케이션의 애플리케이션 이름과 연결 ID를 표시합니다.

```
amqsmom -m saturn.queue.manager -t accounting -b -l 7006,3024
```

이 명령의 출력은 다음과 같습니다.

```
MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e80322'
ApplicationName: 'WebSphere MQ\bin\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ea0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e90322'
ApplicationName: 's\IBM\MQ_4\bin64\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ef0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e60322'
ApplicationName: 's\IBM\MQ_4\bin64\runmqsc.exe'
QueueAccounting: 0

5 Records Processed
```

이 주제에서 사용된 변수를 찾는 방법에 대한 세부사항은 [141 페이지의 『ApplicationName\(3024\) 및 ConnectionId \(7006\)의 맵핑 찾기』](#)의 내용을 참조하십시오.

amqsmom 명령에서 **Type** 매개변수에 대해 통계를 선택한 경우 변수를 찾는 방법에 대한 세부사항은 [141 페이지의 『AvgTimeOnQ\(703\) 및 QmaxDepth\(739\)에 대한 맵핑 찾기』](#)의 내용을 참조하십시오.

ApplicationName(3024) 및 ConnectionId (7006)의 맵핑 찾기

이 주제를 사용하여 회계 데이터 모니터링을 위해 예제에서 사용되는 변수를 찾는 방법에 대해 설명합니다.

요약:

- 3024는 **ApplicationName**인 MQCACF_APPL_NAME을 의미합니다.
- 7006은 **ConnectionId**인 MQBACF_CONNECTION_ID를 의미합니다.

두 가지 프로시저를 수행하는 데 필요한 맵핑을 찾으려면 다음을 수행하십시오.

1. **amqsmon** 명령에서 **Type** 매개변수에 대해 *accounting* 을 선택한 속성 156 페이지의 『ApplicationName』 및 156 페이지의 『ConnectionId』에 대한 설명을 보려면 155 페이지의 『큐 회계 메시지 데이터』를 방문하십시오.

각 경우 **Identifier** 속성을 찾으십시오.

ApplicationName의 경우 **Identifier**은(는) MQCACF_APPL_NAME이고 **ConnectionId**의 경우는 MQBACF_CONNECTION_ID입니다.

2. 141 페이지의 『1』 단계에서 찾은 ID를 검색하십시오.

상수 섹션으로 이동하고 MQCACF *(명령 형식 문자 매개변수 유형) 목록을 찾을 때까지 아래로 스크롤하십시오. MQCACF_APPL_NAME을 찾으면 3024 값이 표시됩니다.

마찬가지로, MQBACF *(명령 형식 바이트 매개변수 유형) 목록을 찾으십시오. MQBACF_CONNECTION_ID를 찾으면 7006 값이 표시됩니다.

AvgTimeOnQ(703) 및 QmaxDepth(739)에 대한 맵핑 찾기

이 주제를 사용하여 통계 데이터 모니터링을 위해 예제에 사용된 변수를 찾는 방법을 설명합니다.

요약:

- 703은 **AvgTimeOnQ**인 MQIAMO64_AVG_Q_TIME을 의미합니다.
- 739는 **QMaxDepth**인 MQIAMO_Q_MAX_DEPTH를 의미합니다.

두 가지 프로시저를 수행하는 데 필요한 맵핑을 찾으려면 다음을 수행하십시오.

1. **amqsmon** 명령에서 **Type** 매개변수에 대해 통계 를 선택한 대로 179 페이지의 『AvgTimeOnQ』 및 178 페이지의 『QMaxDepth』 속성에 대한 설명을 보려면 175 페이지의 『큐 통계 메시지 데이터』를 방문하십시오.

각 경우 **Identifier** 속성을 찾으십시오.

AvgTimeOnQ의 경우 **Identifier**은(는) MQIAMO64_AVG_Q_TIME이고, **QMaxDepth**은(는) MQIAMO_Q_MAX_DEPTH입니다.

2. 141 페이지의 『1』 단계에서 찾은 ID를 검색하십시오.

상수 섹션으로 이동하여 MQIAMO *(명령 형식 정수 모니터링 매개변수 유형) 목록을 찾을 때까지 아래로 스크롤하십시오. MQIAMO_Q_MAX_DEPTH를 찾으면 739 값이 표시됩니다.

마찬가지로, MQIAMO64 *(명령 형식 64비트 정수 모니터링 매개변수 유형) 목록을 찾으십시오. MQIAMO64_AVG_Q_TIME을 찾으면 703 값이 표시됩니다.

회계 및 통계 메시지 참조

이 페이지를 사용하여 회계 및 통계 메시지의 형식과 이 메시지에 리턴되는 정보를 개괄합니다.

회계 및 통계 메시지는 메시지 디스크립터 및 메시지 데이터를 포함한 표준 IBM MQ 메시지입니다. 메시지 데이터는 IBM MQ 애플리케이션이 수행한 MQI 조작에 대한 정보나 IBM MQ 시스템에서 발생하는 활동에 대한 정보를 포함합니다.

메시지 디스크립터

- MQMD 구조

메시지 데이터

- PCF 헤더(MQCFH)
- 항상 리턴되는 회계 또는 통계 메시지 데이터
- 사용 가능한 경우에 리턴되는 회계 또는 통계 메시지 데이터

회계 및 통계 메시지 형식

이 페이지를 MQI 회계 메시지의 구조 예로 사용하십시오.

표 23. MQI 회계 메시지 구조		
MQMD 구조	회계 메시지 헤더 MQCFH 구조	MQI 계정 메시지 데이터 ¹
구조 ID 구조 버전 보고서 옵션 메시지 유형 만기 시간 피드백 코드 Encoding 코드화 문자 세트 ID 메시지 형식 메시지 우선순위 Persistence 메시지 ID 상관 ID 백아웃 수 응답 대상 큐 응답 대상 큐 관리자 사용자 ID 계정 토큰 애플리케이션 ID 데이터 애플리케이션 유형 애플리케이션 이름 넣은 날짜 넣은 시간 애플리케이션 원본 데이터 그룹 ID 메시지 순서 번호 오프셋 메시지 플래그 원래 길이	구조 유형 구조 길이 구조 버전 명령 ID 메시지 순서 번호 제어 옵션 완료 코드 이유 코드 매개변수 수	큐 관리자 간격 시작 날짜 간격 시작 시간 간격 종료 날짜 간격 종료 시간 명령 레벨 연결 ID 순서 번호 애플리케이션 이름 애플리케이션 프로세스 ID 애플리케이션 스레드 ID 사용자 ID 연결 날짜 연결 시간 연결 이름 채널 이름 연결 해제 날짜 연결 해제 시간 연결 해제 유형 열기 수 열기 실패 수 닫기 수 닫기 실패 수 넣기 수 넣기 실패 수 Put1 수 Put1 실패 수 넣기 바이트 가져오기 수 가져오기 실패 수 가져오기 바이트 찾아보기 수 찾아보기 실패 수 찾아보기 바이트 커밋 수 커밋 실패 수 백아웃 수 조회 수 조회 실패 수 설정 수 설정 실패 수

표 23. MQI 회계 메시지 구조 (계속)

MQMD 구조	회계 메시지 헤더 MQCFH 구조	MQI 계정 메시지 데이터 ¹
<p>참고:</p> <p>1. 표시된 매개변수는 MQI 회계 메시지에 대해 리턴된 매개변수입니다. 실제 회계 또는 통계 메시지 데이터는 메시지 범주에 따라 다릅니다.</p>		

회계 및 통계 메시지 MQMD(메시지 디스크립터)

이 페이지를 사용하여 회계 및 통계 메시지의 메시지 디스크립터와 이벤트 메시지의 메시지 디스크립터 간 차이를 이해합니다.

회계 및 통계 메시지의 메시지 디스크립터의 매개변수와 값은 다음을 제외하면, 이벤트 메시지의 메시지 디스크립터에서와 동일합니다.

Format

설명: 메시지 데이터의 형식 이름입니다.

데이터 유형: MQCHAR8.

값: **MQFMT_ADMIN**
관리 메시지.

회계 및 통계 메시지의 메시지 디스크립터에 포함된 일부 매개변수는 메시지를 생성한 큐 관리자가 제공하는 수정된 데이터를 포함합니다.

MQMD는 메시지를 넣는 큐 관리자의 이름(28자에서 잘림)과 회계 또는 통계 큐에 메시지를 넣은 날짜 및 시간도 지정합니다.

회계 및 통계 메시지의 메시지 데이터

회계 및 통계 메시지의 메시지 데이터는 PCF 명령 조회 및 응답에 사용되는 프로그래밍 가능 명령 형식(PCF)을 기반으로 합니다. 회계 및 통계 메시지의 메시지 데이터는 PCF 헤더(MQCFH)와 회계 또는 통계 보고서로 구성됩니다.

회계 및 통계 메시지 MQCFH(PCF 헤더)

회계 및 통계 메시지의 메시지 헤더는 MQCFH 구조입니다. 회계 및 통계 메시지의 메시지 헤더의 매개변수와 값은 다음을 제외하면, 이벤트 메시지의 메시지 헤더에서와 동일합니다.

Command

설명: 명령 ID입니다. 회계 또는 통계 메시지 범주를 식별합니다.

데이터 유형: MQLONG.

값: **MQCMD_ACCOUNTING_MQI**
MQI 회계 메시지.

MQCMD_ACCOUNTING_Q
큐 회계 메시지.

MQCMD_STATISTICS_MQI
MQI 통계 메시지.

MQCMD_STATISTICS_Q
큐 통계 메시지.

MQCMD_STATISTICS_CHANNEL
채널 통계 메시지.

Version

설명:	구조 버전 번호입니다.
데이터 유형:	MQLONG.
값:	MQCFH_VERSION_3 회계 및 통계 메시지 버전-3.

회계 및 통계 메시지 데이터

회계 및 통계 메시지 데이터의 콘텐츠는 다음과 같이 회계 또는 통계 메시지의 범주에 따라 다릅니다.

MQI 회계 메시지

MQI 회계 메시지 데이터는 많은 PCF 매개변수로 구성됩니다(PCF 그룹은 없음).

큐 회계 메시지

큐 회계 메시지 데이터는 많은 PCF 매개변수로 구성되며 범위는 1 ~ 100 *QAccountingData* PCF 그룹입니다.

MQI 통계 메시지

MQI 통계 메시지 데이터는 많은 PCF 매개변수로 구성됩니다(PCF 그룹은 없음).

큐 통계 메시지

큐 통계 메시지 데이터는 많은 PCF 매개변수로 구성되며 범위는 1 ~ 100 *QStatisticsData* PCF 그룹입니다.

채널 통계 메시지

채널 통계 메시지 데이터는 많은 PCF 매개변수로 구성되며 범위는 1 ~ 100 *ChlStatisticsData* PCF 그룹입니다.

MQI 회계 메시지 데이터

이 페이지를 사용하여 MQI 회계 메시지의 구조를 봅니다.

메시지 이름:	MQI 회계 메시지.
플랫폼:	IBM MQ for z/OS를 제외한 모두.
시스템 큐:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

설명:	큐 관리자의 이름
ID:	MQCA_Q_MGR_NAME
데이터 유형:	MQCFST
최대 길이:	MQ_Q_MGR_NAME_LENGTH
리턴됨:	항상

IntervalStartDate

설명:	모니터링 기간의 시작 날짜
ID:	MQCAMO_START_DATE
데이터 유형:	MQCFST
최대 길이:	MQ_DATE_LENGTH
리턴됨:	항상

IntervalStartTime

설명:	모니터링 기간의 시작 시간
ID:	MQCAMO_START_TIME

데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상

IntervalEndDate

설명: 모니터링 기간의 종료 날짜
ID: MQCAMO_END_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH
리턴됨: 항상

IntervalEndTime

설명: 모니터링 기간의 종료 시간
ID: MQCAMO_END_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상

CommandLevel

설명: 큐 관리자 명령 레벨
ID: MQIA_COMMAND_LEVEL
데이터 유형: MQCFIN
리턴됨: 항상

ConnectionId

설명: IBM MQ 연결의 연결 ID
ID: MQBACF_CONNECTION_ID
데이터 유형: MQCFBS
최대 길이: MQ_CONNECTION_ID_LENGTH
리턴됨: 항상

SeqNumber

설명: 순서 번호. 이 값은 장기 실행 연결에 대한 각 후속 레코드마다 증분됩니다.
ID: MQIACF_SEQUENCE_NUMBER
데이터 유형: MQCFIN
리턴됨: 항상

ApplicationName

설명: 애플리케이션의 이름. 이 필드의 콘텐츠는 메시지 디스크립터의 *PutAppIName* 필드 콘텐츠에 해당합니다.
ID: MQCACF_APPL_NAME

데이터 유형: MQCFST
최대 길이: MQ_APPL_NAME_LENGTH
리턴됨: 항상

ApplicationPid

설명: 애플리케이션의 운영 체제 프로세스 ID
ID: MQIACF_PROCESS_ID
데이터 유형: MQCFIN
리턴됨: 항상

ApplicationTid

설명: 애플리케이션 연결의 IBM MQ 스레드 ID.
ID: MQIACF_THREAD_ID
데이터 유형: MQCFIN
리턴됨: 항상

UserId

설명: 애플리케이션의 사용자 ID 컨텍스트
ID: MQCACF_USER_IDENTIFIER
데이터 유형: MQCFST
최대 길이: MQ_USER_ID_LENGTH
리턴됨: 항상

ConnDate

설명: MQCONN 조작 날짜
ID: MQCAMO_CONN_DATE
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 사용 가능할 때

ConnTime

설명: MQCONN 조작 시간
ID: MQCAMO_CONN_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 사용 가능할 때

ConnName

설명: 클라이언트 연결의 연결 이름
ID: MQCACH_CONNECTION_NAME
데이터 유형: MQCFST

최대 길이: MQ_CONN_NAME_LENGTH
리턴됨: 사용 가능할 때

ChannelName

설명: 클라이언트 연결의 채널 이름
ID: MQCACH_CHANNEL_NAME
데이터 유형: MQCFST
최대 길이: MQ_CHANNEL_NAME_LENGTH
리턴됨: 사용 가능할 때

RemoteProduct

설명: DISPLAY CHSTATUS의 **RPRODUCT** 필드에 표시된 클라이언트 연결에 대한 원격 제품 ID
ID: MQCACH_REMOTE_PRODUCT
데이터 유형: MQCFST
최대 길이: MQ_REMOTE_PRODUCT_LENGTH
리턴됨: 사용 가능할 때

RemoteVersion

설명: DISPLAY CHSTATUS의 **RVERSION** 필드에 표시된 클라이언트 연결을 위한 원격 제품 버전
ID: MQCACH_REMOTE_VERSION
데이터 유형: MQCFST
최대 길이: MQ_REMOTE_VERSION_LENGTH
리턴됨: 사용 가능할 때

DiscDate

설명: MQDISC 조작 날짜
ID: MQCAMO_DISC_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH
리턴됨: 사용 가능할 때

DiscTime

설명: MQDISC 조작 시간
ID: MQCAMO_DISC_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 사용 가능할 때

DiscType

설명: 연결 해제 유형

ID: MQIAMO_DISC_TYPE
 데이터 유형: MQCFIN
 값: 가능한 값은 다음과 같습니다.
MQDISCONNECT_NORMAL
 애플리케이션의 요청으로
MQDISCONNECT_IMPLICIT
 비정상 애플리케이션 종료
MQDISCONNECT_Q_MGR
 큐 관리자가 끊은 연결

리턴됨: 사용 가능할 때

OpenCount

설명: MQOPEN에 대한 호출을 실행하거나 MQPUT1 verb를 사용하여 열린 오브젝트의 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_OPENS
 데이터 유형: MQCFIL
 리턴됨: 사용 가능할 때

OpenFailCount

설명: 실패한 오브젝트 열기 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_OPENS_FAILED
 데이터 유형: MQCFIL
 리턴됨: 사용 가능할 때

CloseCount

설명: 닫힌 오브젝트 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_CLOSES
 데이터 유형: MQCFIL
 리턴됨: 사용 가능할 때

CloseFailCount

설명: 실패한 오브젝트 닫기 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_CLOSES_FAILED
 데이터 유형: MQCFIL
 리턴됨: 사용 가능할 때

PutCount

설명: MQPUT1 호출을 사용하여 넣은 메시지를 제외하고, 큐에 정상적으로 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_PUTS

데이터 유형: MQCFIL
리턴됨: 사용 가능할 때

PutFailCount

설명: 실패한 메시지 넣기 시도 수
ID: MQIAMO_PUTS_FAILED
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때

Put1Count

설명: MQPUT1 호출을 사용하여 큐에 성공적으로 넣은 지속 및 비지속 메시지 수. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.
ID: MQIAMO_PUT1S
데이터 유형: MQCFIL
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

Put1FailCount

설명: MQPUT1 호출을 사용하여 메시지를 넣으려는 실패한 시도 수
ID: MQIAMO_PUT1S_FAILED
데이터 유형: MQCFIN
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

PutBytes

설명: 지속 및 비지속 메시지에 대해 넣기 호출을 사용하여 쓴 바이트 수. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.
ID: MQIAMO64_PUT_BYTES
데이터 유형: MQCFIL64
리턴됨: 사용 가능할 때

GetCount

설명: 지속 및 비지속 메시지에 대한 성공한 파괴적 MQGET 호출의 수. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.
ID: MQIAMO_GETS
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때

GetFailCount

설명: 실패한 파괴적 MQGET 호출 수
ID: MQIAMO_GETS_FAILED
데이터 유형: MQCFIN

리턴됨: 사용 가능할 때

GetBytes

설명: 지속 및 비지속 메시지의 수신된 총 바이트 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_GET_BYTES

데이터 유형: MQCFIL64

리턴됨: 사용 가능할 때

BrowseCount

설명: 지속 및 비지속 메시지에 대한 성공한 비파괴적 MQGET 호출의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_BROWSES

데이터 유형: MQCFIL

리턴됨: 사용 가능할 때

BrowseFailCount

설명: 실패한 비파괴 MQGET 호출 수

ID: MQIAMO_BROWSES_FAILED

데이터 유형: MQCFIN

리턴됨: 사용 가능할 때

BrowseBytes

설명: 지속 및 비지속 메시지를 찾아본 총 바이트 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_BROWSE_BYTES

데이터 유형: MQCFIL64

리턴됨: 사용 가능할 때

CommitCount

설명: 성공한 트랜잭션 수. 이 수는 연결된 애플리케이션이 내재적으로 커밋한 트랜잭션을 포함합니다. 미해결 작업이 없는 커밋 요청이 이 수에 포함됩니다.

ID: MQIAMO_COMMITS

데이터 유형: MQCFIN

리턴됨: 사용 가능할 때

CommitFailCount

설명: 실패한 트랜잭션 완료 시도 수

ID: MQIAMO_COMMITS_FAILED

데이터 유형: MQCFIN

리턴됨: 사용 가능할 때

BackCount

설명: 비정상 연결 해제로 인한 암시적 백아웃을 포함하여, 처리된 백아웃 수
ID: MQIAMO_BACKOUTS
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때

InqCount

설명: 조회에 성공한 오브젝트 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다(참조 [참고 1](#) 참조).
ID: MQIAMO_INQS
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때

InqFailCount

설명: 실패한 오브젝트 조회 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다(참조 [참고 1](#) 참조).
ID: MQIAMO_INQS_FAILED
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때

SetCount

설명: 성공한 MQSET 호출 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다(참조 [참고 1](#) 참조).
ID: MQIAMO_SETS
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때

SetFailCount

설명: 실패한 MQSET 호출 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다(참조 [참고 1](#) 참조).
ID: MQIAMO_SETS_FAILED
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때

SubCountDur

설명: 지속 가능한 구독을 작성, 변경 또는 재개한 구독 요청의 수. 조작 유형별로 색인화된 값 배열입니다.
0 = 작성된 구독 수
1 = 변경된 구독 수
2 = 재개한 구독 수
ID: MQIAMO_SUBS_DUR
데이터 유형: MQCFIL

리턴됨: 사용 가능할 때.

SubCountNDur

설명: 지속 불가능한 구독을 작성, 변경 또는 재개한 구독 요청의 수. 조작 유형별로 색인화된 값 배열입니다.

0 = 작성된 구독 수

1 = 변경된 구독 수

2 = 재개한 구독 수

ID: MQIAMO_SUBS_NDUR

데이터 유형: MQCFIL

리턴됨: 사용 가능할 때.

SubFailCount

설명: 실패한 구독 요청 수.

ID: MQIAMO_SUBS_FAILED

데이터 유형: MQCFIN

리턴됨: 사용 가능할 때.

UnsubCountDur

설명: 지속적 구독에 대한 성공적인 구독 해제 요청의 수입입니다. 조작 유형별로 색인화된 값 배열입니다.

0 - 구독이 닫혔지만 제거되지 않음

1 - 구독이 닫히고 제거됨

ID: MQIAMO_UNSUBS_DUR

데이터 유형: MQCFIL

리턴됨: 사용 가능할 때.

UnsubCountNDur

설명: 지속적 구독에 대한 성공적인 구독 해제 요청의 수입입니다. 조작 유형별로 색인화된 값 배열입니다.

0 - 구독이 닫혔지만 제거되지 않음

1 - 구독이 닫히고 제거됨

ID: MQIAMO_UNSUBS_NDUR

데이터 유형: MQCFIL

리턴됨: 사용 가능할 때.

UnsubFailCount

설명: 실패한 구독 해제 요청 수.

ID: MQIAMO_UNSUBS_FAILED

데이터 유형: MQCFIN

리턴됨: 사용 가능할 때.

SubRqCount

설명: 성공한 MQSUBRQ 요청 수.
ID: MQIAMO_SUBRQS
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

SubRqFailCount

설명: 실패한 MQSUB 요청 수.
ID: MQIAMO_SUBRQS_FAILED
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

CBCCount

설명: 성공한 MQCB 요청 수. 조작 유형별로 색인화된 값 배열입니다.
0 - 콜백이 작성 또는 변경됨
1 - 콜백이 제거됨
2 - 콜백이 재개됨
3 - 콜백이 일시중단됨
ID: MQIAMO_CBS
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

CBFailCount

설명: 실패한 MQCB 요청 수.
ID: MQIAMO_CBS_FAILED
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

CtlCount

설명: 성공한 MQCTL 요청 수. 조작 유형별로 색인화된 값 배열입니다.
0 - 연결이 시작됨
1 - 연결이 중지됨
2 - 연결이 재개됨
3 - 연결이 일시중단됨
ID: MQIAMO_CTLS
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때.

CtlFailCount

설명: 실패한 MQCTL 요청 수.

ID: MQIAMO_CTL5_FAILED
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

StatCount

설명: 성공한 MQSTAT 요청 수.
ID: MQIAMO_STATS.
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

StatFailCount

설명: 실패한 MQSTAT 요청 수.
ID: MQIAMO_STATS_FAILED
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

PutTopicCount

설명: MQPUT1 호출을 사용하여 넣은 메시지를 제외하고, 토픽에 정상적으로 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

참고: 토픽으로 분석되며 큐 알리어스를 사용하여 넣은 메시지가 이 값에 포함됩니다.

ID: MQIAMO_TOPIC_PUTS
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때.

PutTopicFailCount

설명: 토픽에 메시지를 넣으려는 실패한 시도 수.
ID: MQIAMO_TOPIC_PUTS_FAILED
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

Put1TopicCount

설명: MQPUT1 호출을 사용하여 토픽에 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

참고: 토픽으로 분석되며 큐 알리어스를 사용하여 넣은 메시지가 이 값에 포함됩니다.

ID: MQIAMO_TOPIC_PUT1S
데이터 유형: MQCFIL
리턴됨: 사용 가능할 때.

Put1TopicFailCount

설명: MQPUT1 호출을 사용하여 토픽에 메시지를 넣으려는 실패한 시도 수.
ID: MQIAMO_TOPIC_PUT1S_FAILED

데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

PutTopicBytes

설명: 발행 조작으로 분석된 지속 및 비지속 메시지에 대해 넣기 호출을 사용하여 쓴 바이트 수. 이 수는 구독자에게 전달된 결과적인 바이트 수가 아니라, 애플리케이션이 넣은 바이트 수입니다. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_TOPIC_PUT_BYTES
데이터 유형: MQCFIL64
리턴됨: 사용 가능할 때.

큐 회계 메시지 데이터

이 페이지를 사용하여 큐 회계 메시지의 구조를 봅니다.

메시지 이름:	큐 회계 메시지.
플랫폼:	IBM MQ for z/OS를 제외한 모두.
시스템 큐:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

설명: 큐 관리자의 이름
ID: MQCA_Q_MGR_NAME
데이터 유형: MQCFST
최대 길이: MQ_Q_MGR_NAME_LENGTH
리턴됨: 항상

IntervalStartDate

설명: 모니터링 기간의 시작 날짜
ID: MQCAMO_START_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH
리턴됨: 항상

IntervalStartTime

설명: 모니터링 기간의 시작 시간
ID: MQCAMO_START_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상

IntervalEndDate

설명: 모니터링 기간의 종료 날짜
ID: MQCAMO_END_DATE

데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH
리턴됨: 항상

IntervalEndTime

설명: 모니터링 기간의 종료 시간
ID: MQCAMO_END_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상

CommandLevel

설명: 큐 관리자 명령 레벨
ID: MQIA_COMMAND_LEVEL
데이터 유형: MQCFIN
리턴됨: 항상

ConnectionId

설명: IBM MQ 연결의 연결 ID
ID: MQBACF_CONNECTION_ID
데이터 유형: MQCFBS
최대 길이: MQ_CONNECTION_ID_LENGTH
리턴됨: 항상

SeqNumber

설명: 순서 번호. 이 값은 장기 실행 연결에 대한 각 후속 레코드마다 증분됩니다.
ID: MQIACF_SEQUENCE_NUMBER
데이터 유형: MQCFIN
리턴됨: 항상

ApplicationName

설명: 애플리케이션의 이름. 이 필드의 콘텐츠는 메시지 디스크립터의 PutApplName 필드 콘텐츠에 해당합니다.
ID: MQCACF_APPL_NAME
데이터 유형: MQCFST
최대 길이: MQ_APPL_NAME_LENGTH
리턴됨: 항상

ApplicationPid

설명: 애플리케이션의 운영 체제 프로세스 ID
ID: MQIACF_PROCESS_ID
데이터 유형: MQCFIN

리턴됨: 항상

ApplicationTid

설명: 애플리케이션 연결의 IBM MQ 스레드 ID.

ID: MQIACF_THREAD_ID

데이터 유형: MQCFIN

리턴됨: 항상

UserId

설명: 애플리케이션의 사용자 ID 컨텍스트

ID: MQCACF_USER_IDENTIFIER

데이터 유형: MQCFST

최대 길이: MQ_USER_ID_LENGTH

리턴됨: 항상

ChannelName

설명: 클라이언트 연결의 채널 이름

ID: MQCACH_CHANNEL_NAME

데이터 유형: MQCFST

최대 길이: MQ_CHANNEL_NAME_LENGTH

리턴됨: 사용 가능할 때

ConnName

설명: 클라이언트 연결의 연결 이름

ID: MQCACH_CONNECTION_NAME

데이터 유형: MQCFST

최대 길이: MQ_CONN_NAME_LENGTH

리턴됨: 사용 가능할 때

ObjectCount

설명: 회계 데이터가 기록된 간격에 액세스한 큐의 수. 이 값은 메시지에 포함된 *QAccountingData* PCF 그룹 수로 설정됩니다.

ID: MQIAMO_OBJECT_COUNT

데이터 유형: MQCFIN

리턴됨: 항상

QAccountingData

설명: 큐에 대한 회계 세부사항을 지정하는 그룹화된 매개변수.

ID: MQGACF_Q_ACCOUNTING_DATA

데이터 유형: MQCFGR

그룹의 매개변수:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- OpenCount*
- OpenDate*
- OpenTime*
- CloseDate*
- CloseTime*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- PutMinBytes*
- PutMaxBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- GetMinBytes*
- GetMaxBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- BrowseMinBytes*
- BrowseMaxBytes*
- TimeOnQMin*
- TimeOnQAvg*
- TimeOnQMax*

리턴됨: 항상

QName

설명: 큐의 이름.
 ID: MQCA_Q_NAME
 데이터 유형: MQCFST
 PCF 그룹에 포함: *QAccountingData*
 최대 길이: MQ_Q_NAME_LENGTH
 리턴됨: 사용 가능할 때

CreateDate

설명: 큐가 작성된 날짜
 ID: MQCA_CREATION_DATE
 데이터 유형: MQCFST
 PCF 그룹에 포함: *QAccountingData*
 최대 길이: MQ_DATE_LENGTH

리턴됨: 사용 가능할 때

CreateTime

설명: 큐가 작성된 시간
ID: MQCA_CREATION_TIME
데이터 유형: MQCFST
PCF 그룹에 포함: QAccountingData
최대 길이: MQ_TIME_LENGTH
리턴됨: 사용 가능할 때

QType

설명: 큐의 유형
ID: MQIA_Q_TYPE
데이터 유형: MQCFIN
PCF 그룹에 포함: QAccountingData
값: MQQT_LOCAL
리턴됨: 사용 가능할 때

QDefinitionType

설명: 큐 정의 유형
ID: MQIA_DEFINITION_TYPE
데이터 유형: MQCFIN
PCF 그룹에 포함: QAccountingData
값: 가능한 값은 다음과 같습니다.

MQQDT_PREDEFINED
MQQDT_PERMANENT_DYNAMIC
MQQDT_TEMPORARY_DYNAMIC

리턴됨: 사용 가능할 때

OpenCount

설명: MQOPEN에 대한 호출을 실행하거나 MQPUT1 verb를 사용하여, 이 큐가 이 간격으로 애플리케이션에서 열린 횟수.
ID: MQIAMO_OPENS
데이터 유형: MQCFIL
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

OpenDate

설명: 이 레코딩 간격에 처음으로 큐를 연 날짜. 이 간격이 시작할 때 큐가 이미 열려 있으면 이 값은 큐가 원래 열린 날짜를 반영합니다.
ID: MQCAMO_OPEN_DATE
데이터 유형: MQCFST

PCF 그룹에 포함: *QAccountingData*
리턴됨: 사용 가능할 때

OpenTime

설명: 이 레코딩 간격에 처음으로 큐를 연 시간. 이 간격이 시작할 때 큐가 이미 열려 있으면 이 값은 큐가 원래 열린 시간을 반영합니다.

ID: MQCAMO_OPEN_TIME
데이터 유형: MQCFST
PCF 그룹에 포함: *QAccountingData*
리턴됨: 사용 가능할 때

CloseDate

설명: 이 레코딩 간격에 마지막으로 큐를 닫은 날짜. 큐가 여전히 열려 있으면 값이 리턴되지 않습니다.

ID: MQCAMO_CLOSE_DATE
데이터 유형: MQCFST
PCF 그룹에 포함: *QAccountingData*
리턴됨: 사용 가능할 때

CloseTime

설명: 이 레코딩 간격에 마지막으로 큐를 닫은 시간. 큐가 여전히 열려 있으면 값이 리턴되지 않습니다.

ID: MQCAMO_CLOSE_TIME
데이터 유형: MQCFST
PCF 그룹에 포함: *QAccountingData*
리턴됨: 사용 가능할 때

PutCount

설명: MQPUT1 호출을 제외하고, 큐에 넣은 지속 및 비지속 메시지의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_PUTS
데이터 유형: MQCFIL
PCF 그룹에 포함: *QAccountingData*
리턴됨: 사용 가능할 때

PutFailCount

설명: MQPUT1 호출을 제외한, 실패한 메시지 넣기 시도 수

ID: MQIAMO_PUTS_FAILED
데이터 유형: MQCFIN
PCF 그룹에 포함: *QAccountingData*
리턴됨: 사용 가능할 때

Put1Count

설명: MQPUT1 호출을 사용하여 큐에 성공적으로 넣은 지속 및 비지속 메시지 수. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_PUT1S

데이터 유형: MQCFIL

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

Put1FailCount

설명: MQPUT1 호출을 사용하여 메시지를 넣으려는 실패한 시도 수

ID: MQIAMO_PUT1S_FAILED

데이터 유형: MQCFIN

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

PutBytes

설명: 지속 및 비지속 메시지의 넣은 총 바이트 수. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_PUT_BYTES

데이터 유형: MQCFIL64

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

PutMinBytes

설명: 큐에 있는 지속 및 비지속 메시지의 최소 크기. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_PUT_MIN_BYTES

데이터 유형: MQCFIL

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

PutMaxBytes

설명: 큐에 있는 지속 및 비지속 메시지의 최대 크기. 이 매개 변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_PUT_MAX_BYTES

데이터 유형: MQCFIL

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

GeneratedMsgCount

설명:	생성된 메시지의 수. 생성되는 메시지는 다음과 같습니다. <ul style="list-style-type: none">• 큐 용량 높음 이벤트• 큐 용량 낮음 이벤트
ID:	MQIAMO_GENERATED_MSGS
데이터 유형:	MQCFIN
PCF 그룹에 포함:	<i>QAccountingData</i>
리턴됨:	사용 가능할 때

GetCount

설명:	지속 및 비지속 메시지에 대한 성공한 파괴적 MQGET 호출의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2 를 보십시오.
ID:	MQIAMO_GETS
데이터 유형:	MQCFIL
PCF 그룹에 포함:	<i>QAccountingData</i>
리턴됨:	사용 가능할 때

GetFailCount

설명:	실패한 파괴적 MQGET 호출 수
ID:	MQIAMO_GETS_FAILED
데이터 유형:	MQCFIN
PCF 그룹에 포함:	<i>QAccountingData</i>
리턴됨:	사용 가능할 때

GetBytes

설명:	지속 및 비지속 메시지에 대한 파괴적 MQGET 호출의 읽은 바이트 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2 를 보십시오.
ID:	MQIAMO64_GET_BYTES
데이터 유형:	MQCFIL64
PCF 그룹에 포함:	<i>QAccountingData</i>
리턴됨:	사용 가능할 때

GetMinBytes

설명:	큐에서 검색한 지속 및 비지속 메시지의 최소 크기. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2 를 보십시오.
ID:	MQIAMO_GET_MIN_BYTES
데이터 유형:	MQCFIL
PCF 그룹에 포함:	<i>QAccountingData</i>
리턴됨:	사용 가능할 때

GetMaxBytes

설명: 큐에서 검색한 지속 및 비지속 메시지의 최대 크기. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_GET_MAX_BYTES

데이터 유형: MQCFIL

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

BrowseCount

설명: 지속 및 비지속 메시지에 대한 성공한 비파괴적 MQGET 호출의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_BROWSES

데이터 유형: MQCFIL

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

BrowseFailCount

설명: 실패한 비파괴 MQGET 호출 수

ID: MQIAMO_BROWSES_FAILED

데이터 유형: MQCFIN

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

BrowseBytes

설명: 지속 메시지를 리턴한 비파괴 MQGET 호출의 읽은 바이트 수.

ID: MQIAMO64_BROWSE_BYTES

데이터 유형: MQCFIL64

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

BrowseMinBytes

설명: 큐에서 찾아본 지속 및 비지속 메시지의 최소 크기. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_BROWSE_MIN_BYTES

데이터 유형: MQCFIL

PCF 그룹에 포함: *QAccountingData*

리턴됨: 사용 가능할 때

BrowseMaxBytes

설명: 큐에서 찾아본 지속 및 비지속 메시지의 최대 크기. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_BROWSE_MAX_BYTES

데이터 유형: MQCFIL
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

TimeOnQMin

설명: 파괴적 검색 이전에 지속 및 비지속 메시지가 큐에 머문 최단 시간(마이크로초). 동기 점 아래에서 메시지가 검색되도록 이 값은 가져오기 조작이 커밋되기 전의 시간을 포함하지 않습니다. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_Q_TIME_MIN
데이터 유형: MQCFIL64
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

TimeOnQAvg

설명: 파괴적 검색 이전에 지속 및 비지속 메시지가 큐에 머문 평균 시간(마이크로초). 동기 점 아래에서 메시지가 검색되도록 이 값은 가져오기 조작이 커밋되기 전의 시간을 포함하지 않습니다. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_Q_TIME_AVG
데이터 유형: MQCFIL64
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

TimeOnQMax

설명: 파괴적 검색 이전에 지속 및 비지속 메시지가 큐에 머문 최장 시간(마이크로초). 동기 점 아래에서 메시지가 검색되도록 이 값은 가져오기 조작이 커밋되기 전의 시간을 포함하지 않습니다. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO64_Q_TIME_MAX
데이터 유형: MQCFIL64
PCF 그룹에 포함: QAccountingData
리턴됨: 사용 가능할 때

CBCCount

설명: 성공한 MQCB 요청 수. 조작 유형별로 색인화된 값 배열입니다.

- 0 - 콜백이 작성 또는 변경됨
- 1 - 콜백이 제거됨
- 2 - 콜백이 재개됨
- 3 - 콜백이 일시중단됨

ID: MQIAMO_CBS
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

CBFailCount

설명:	실패한 MQCB 요청 수.
ID:	MQIAMO_CBS_FAILED
데이터 유형:	MQCFIN
리턴됨:	사용 가능할 때.

MQI 통계 메시지 데이터

이 페이지를 사용하여 MQI 통계 메시지의 구조를 봅니다.

메시지 이름:	MQI 통계 메시지.
플랫폼:	IBM MQ for z/OS를 제외한 모두.
시스템 큐:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

설명:	큐 관리자의 이름.
ID:	MQCA_Q_MGR_NAME.
데이터 유형:	MQCFST.
최대 길이:	MQ_Q_MGR_NAME_LENGTH.
리턴됨:	항상.

IntervalStartDate

설명:	모니터링 기간의 시작 날짜.
ID:	MQCAMO_START_DATE.
데이터 유형:	MQCFST.
최대 길이:	MQ_DATE_LENGTH
리턴됨:	항상.

IntervalStartTime

설명:	모니터링 기간의 시작 시간.
ID:	MQCAMO_START_TIME.
데이터 유형:	MQCFST.
최대 길이:	MQ_TIME_LENGTH
리턴됨:	항상.

IntervalEndDate

설명:	모니터링 기간의 종료 날짜.
ID:	MQCAMO_END_DATE.
데이터 유형:	MQCFST.
최대 길이:	MQ_DATE_LENGTH
리턴됨:	항상.

IntervalEndTime

설명: 모니터링 기간의 종료 시간.
ID: MQCAMO_END_TIME.
데이터 유형: MQCFST.
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상.

CommandLevel

설명: 큐 관리자 명령 레벨.
ID: MQIA_COMMAND_LEVEL.
데이터 유형: MQCFIN.
리턴됨: 항상.

ConnCount

설명: 큐 관리자에 대한 연결 수.
ID: MQIAMO_CONNS.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

ConnFailCount

설명: 실패한 연결 시도 수.
ID: MQIAMO_CONNS_FAILED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

ConnsMax

설명: 레코딩 간격의 최대 동시 연결 수.
ID: MQIAMO_CONNS_MAX.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

DiscCount

설명: 큐 관리자에 대한 연결 해제 수. 다음 상수에 따라 색인화된 정수 배열입니다.
• MQDISCONNECT_NORMAL
• MQDISCONNECT_IMPLICIT
• MQDISCONNECT_Q_MGR
ID: MQIAMO_DISCS.
데이터 유형: MQCFIL.
리턴됨: 사용 가능할 때.

OpenCount

설명: MQOPEN에 대한 호출을 실행하거나 MQPUT1 verb를 사용하여 열린 오브젝트의 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_OPENS.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

OpenFailCount

설명: 실패한 오브젝트 열기 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_OPENS_FAILED.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

CloseCount

설명: 닫힌 오브젝트 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_CLOSES.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

CloseFailCount

설명: 실패한 오브젝트 닫기 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_CLOSES_FAILED.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

InqCount

설명: 조회한 오브젝트 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_INQS.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

InqFailCount

설명: 실패한 오브젝트 조회 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다([참조 참고 1 참조](#)).

ID: MQIAMO_INQS_FAILED.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

SetCount

설명: 업데이트(SET)된 오브젝트 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다(참조 [참고 1](#) 참조).

ID: MQIAMO_SETS.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

SetFailCount

설명: 실패한 SET 시도 수. 이 매개변수는 오브젝트 유형별로 색인화된 정수 목록입니다(참조 [참고 1](#) 참조).

ID: MQIAMO_SETS_FAILED.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

PutCount

설명: MQPUT1 호출을 제외하고, 큐에 넣은 지속 및 비지속 메시지의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

ID: MQIAMO_PUTS.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

PutFailCount

설명: 실패한 메시지 넣기 시도 수.

ID: MQIAMO_PUTS_FAILED.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

Put1Count

설명: MQPUT1 요청을 사용하여 큐에 성공적으로 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 [노트 2](#) 참조).

ID: MQIAMO_PUT1S.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

Put1FailCount

설명: MQPUT1 요청을 사용하여 지속 및 비지속 메시지를 큐에 넣으려는 실패한 시도 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 [노트 2](#) 참조).

ID: MQIAMO_PUT1S_FAILED.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

PutBytes

설명:	넣기 요청을 사용하여 기록된 지속 및 비지속 메시지의 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 노트 2 참조).
ID:	MQIAMO64_PUT_BYTES.
데이터 유형:	MQCFIL64.
리턴됨:	사용 가능할 때.

GetCount

설명:	지속 및 비지속 메시지에 대한 성공한 파괴적 가져오기 요청의 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 노트 2 참조).
ID:	MQIAMO_GETS.
데이터 유형:	MQCFIL.
리턴됨:	사용 가능할 때.

GetFailCount

설명:	실패한 파괴적 가져오기 요청의 수.
ID:	MQIAMO_GETS_FAILED.
데이터 유형:	MQCFIN.
리턴됨:	사용 가능할 때.

GetBytes

설명:	지속 및 비지속 메시지에 대한 파괴적 가져오기 요청의 읽은 바이트 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 노트 2 참조).
ID:	MQIAMO64_GET_BYTES.
데이터 유형:	MQCFIL64.
리턴됨:	사용 가능할 때.

BrowseCount

설명:	지속 및 비지속 메시지에 대한 성공한 비파괴 가져오기 요청의 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 노트 2 참조).
ID:	MQIAMO_BROWSES.
데이터 유형:	MQCFIL.
리턴됨:	사용 가능할 때.

BrowseFailCount

설명:	실패한 비파괴 가져오기 요청의 수.
ID:	MQIAMO_BROWSES_FAILED.
데이터 유형:	MQCFIN.
리턴됨:	사용 가능할 때.

BrowseBytes

설명:	지속 및 비지속 메시지에 대한 비파괴 가져오기 요청의 읽은 바이트 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다(참조 노트 2 참조).
-----	---

ID: MQIAMO64_BROWSE_BYTES.
데이터 유형: MQCFIL64.
리턴됨: 사용 가능할 때.

CommitCount

설명: 완료한 트랜잭션 수. 이 수는 애플리케이션 연결 해제를 통해 내재적으로 커밋된 트랜잭션 및 미해결 작업이 없는 커밋 요청을 포함합니다.
ID: MQIAMO_COMMITS.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

CommitFailCount

설명: 실패한 트랜잭션 완료 시도 수.
ID: MQIAMO_COMMITS_FAILED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

BackCount

설명: 비정상 연결 해제로 인한 암시적 백아웃을 포함하여, 처리된 백아웃 수.
ID: MQIAMO_BACKOUTS.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

ExpiredMsgCount

설명: 검색 가능하게 되기 전에 만기되어서 제거된 지속 및 비지속 메시지의 수.
ID: MQIAMO_MSGS_EXPIRED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

PurgeCount

설명: 큐가 지워진 횟수.
ID: MQIAMO_MSGS_PURGED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

SubCountDur

설명: 지속 가능한 구독을 작성, 변경 또는 재개한 구독 요청의 수. 조작 유형별로 색인화된 값 배열입니다.
0 = 작성된 구독 수
1 = 변경된 구독 수
2 = 재개한 구독 수
ID: MQIAMO_SUBS_DUR.

데이터 유형: MQCFIL
리턴됨: 사용 가능할 때.

SubCountNDur

설명: 지속 불가능한 구독을 작성, 변경 또는 재개한 구독 요청의 수. 조작 유형별로 색인화된 값 배열입니다.

0 = 작성된 구독 수
1 = 변경된 구독 수
2 = 재개한 구독 수

ID: MQIAMO_SUBS_NDUR.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

SubFailCount

설명: 실패한 구독 요청 수.

ID: MQIAMO_SUBS_FAILED.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

UnsubCountDur

설명: 지속적 구독에 대한 성공적인 구독 해제 요청의 수입니다. 조작 유형별로 색인화된 값 배열입니다.

0 - 구독이 단혔지만 제거되지 않음
1 - 구독이 단히고 제거됨

ID: MQIAMO_UNSUBS_DUR.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

UnsubCountNDur

설명: 지속 불가능한 구독에 대한 성공한 구독 해제 요청 수. 조작 유형별로 색인화된 값 배열입니다.

0 - 구독이 단혔지만 제거되지 않음
1 - 구독이 단히고 제거됨

ID: MQIAMO_UNSUBS_NDUR.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

UnsubFailCount

설명: 실패한 구독 해제 요청 수.

ID: MQIAMO_UNSUBS_FAILED.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

SubRqCount

설명: 성공한 MQSUBRQ 요청 수.

ID: MQIAMO_SUBRQS

데이터 유형: MQCFIN

리턴됨: 사용 가능할 때.

SubRqFailCount

설명: 실패한 MQSUBRQ 요청 수.

ID: MQIAMO_SUBRQS_FAILED.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

CBCount

설명: 성공한 MQCB 요청 수. 조작 유형별로 색인화된 값 배열입니다.

0 - 콜백이 작성 또는 변경됨

1 - 콜백이 제거됨

2 - 콜백이 재개됨

3 - 콜백이 일시중단됨

ID: MQIAMO_CBS.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

CBFailCount

설명: 실패한 MQCB 요청 수.

ID: MQIAMO_CBS_FAILED.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

CtlCount

설명: 성공한 MQCTL 요청 수. 조작 유형에 의해 색인화된 값의 배열입니다.

0 - 연결이 시작됨

1 - 연결이 중지됨

2 - 연결이 재개됨

3 - 연결이 일시중단됨

ID: MQIAMO_CTL.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

CtlFailCount

설명: 실패한 MQCTL 요청 수.
ID: MQIAMO_CTLs_FAILED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

StatCount

설명: 성공한 MQSTAT 요청 수.
ID: MQIAMO_STATS.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

StatFailCount

설명: 실패한 MQSTAT 요청 수.
ID: MQIAMO_STATS_FAILED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

SubCountDurHighWater

설명: 시간 간격 중의 지속 가능한 구독 수에 대한 고수위 표시. SUBTYPE별로 색인화된 값 배열입니다.
0 - 시스템의 모든 지속 가능 구독에 대한 고수위 표시
1 - 지속 가능 애플리케이션 구독(MQSUBTYPE_API)에 대한 고수위 표시
2 - 지속 가능 관리 구독(MQSUBTYPE_ADMIN)에 대한 고수위 표시
3 - 지속 가능 프록시 구독(MQSUBTYPE_PROXY)에 대한 고수위 표시
ID: MQIAMO_SUB_DUR_HIGHWATER
데이터 유형: MQCFIL.
리턴됨: 사용 가능할 때.

SubCountDurLowWater

설명: 시간 간격 중의 지속 가능한 구독 수에 대한 저수위 표시. SUBTYPE별로 색인화된 값 배열입니다.
0 - 시스템의 모든 지속 가능 구독에 대한 저수위 표시
1 - 지속 가능 애플리케이션 구독(MQSUBTYPE_API)에 대한 저수위 표시
2 - 지속 가능 관리 구독(MQSUBTYPE_ADMIN)에 대한 저수위 표시
3 - 지속 가능 프록시 구독(MQSUBTYPE_PROXY)에 대한 저수위 표시
ID: MQIAMO_SUB_DUR_LOWWATER
데이터 유형: MQCFIL.
리턴됨: 사용 가능할 때.

SubCountNDurHighWater

설명: 시간 간격 중의 지속 불가능한 구독 수에 대한 고수위 표시. SUBTYPE별로 색인화된 값 배열입니다.

- 0 - 시스템의 모든 지속 불가능 구독에 대한 고수위 표시
- 1 - 지속 불가능 애플리케이션 구독(MQSUBTYPE_API)에 대한 고수위 표시
- 2 - 지속 불가능 관리 구독(MQSUBTYPE_ADMIN)에 대한 고수위 표시
- 3 - 지속 불가능 프록시 구독(MQSUBTYPE_PROXY)에 대한 고수위 표시

ID: MQIAMO_SUB_NDUR_HIGHWATER

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

SubCountNDurLowWater

설명: 시간 간격 중의 지속 불가능한 구독 수에 대한 저수위 표시. SUBTYPE별로 색인화된 값 배열입니다.

- 0 - 시스템의 모든 지속 불가능 구독에 대한 저수위 표시
- 1 - 지속 불가능 애플리케이션 구독(MQSUBTYPE_API)에 대한 저수위 표시
- 2 - 지속 불가능 관리 구독(MQSUBTYPE_ADMIN)에 대한 저수위 표시
- 3 - 지속 불가능 프록시 구독(MQSUBTYPE_PROXY)에 대한 저수위 표시

ID: MQIAMO_SUB_NDUR_LOWWATER

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

PutTopicCount

설명: MQPUT1 호출을 사용하여 넣은 메시지를 제외하고, 토픽에 정상적으로 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

참고: 토픽으로 분석되며 큐 알리언스를 사용하여 넣은 메시지가 이 값에 포함됩니다.

ID: MQIAMO_TOPIC_PUTS.

데이터 유형: MQCFIL.

리턴됨: 사용 가능할 때.

PutTopicFailCount

설명: 토픽에 메시지를 넣으려는 실패한 시도 수.

ID: MQIAMO_TOPIC_PUTS_FAILED.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

Put1TopicCount

설명: MQPUT1 호출을 사용하여 토픽에 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. [참조 노트 2](#)를 보십시오.

참고: 토픽으로 분석되며 큐 알리언스를 사용하여 넣은 메시지가 이 값에 포함됩니다.

ID: MQIAMO_TOPIC_PUT1S.
데이터 유형: MQCFIL.
리턴됨: 사용 가능할 때.

Put1TopicFailCount

설명: MQPUT1 호출을 사용하여 토픽에 메시지를 넣으려는 실패한 시도 수.
ID: MQIAMO_TOPIC_PUT1S_FAILED.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

PutTopicBytes

설명: 발행 조작으로 분석된 지속 및 비지속 메시지에 대해 넣기 호출을 사용하여 쓴 바이트 수. 이 수는 구독자에게 전달된 결과적인 바이트 수가 아니라, 애플리케이션이 넣은 바이트 수입니다(이 값에 대해 PublishMsgBytes 참조). 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2를 보십시오.
ID: MQIAMO64_TOPIC_PUT_BYTES.
데이터 유형: MQCFIL64.
리턴됨: 사용 가능할 때.

PublishMsgCount

설명: 시간 간격 중 구독에 전달된 메시지의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2를 보십시오.
ID: MQIAMO64_PUBLISH_MSG_COUNT
데이터 유형: MQCFIL.
리턴됨: 사용 가능할 때.

PublishMsgBytes

설명: 시간 간격 중 구독에 전달된 바이트의 수. 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2를 보십시오.
ID: MQIAMO64_PUBLISH_MSG_BYTES
데이터 유형: MQCFIL64.
리턴됨: 사용 가능할 때.

큐 통계 메시지 데이터

이 페이지를 사용하여 큐 통계 메시지의 구조를 봅니다.

메시지 이름:	큐 통계 메시지.
플랫폼:	IBM MQ for z/OS를 제외한 모두.
시스템 큐:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

설명: 큐 관리자의 이름
ID: MQCA_Q_MGR_NAME
데이터 유형: MQCFST

최대 길이: MQ_Q_MGR_NAME_LENGTH
리턴됨: 항상

IntervalStartDate

설명: 모니터링 기간의 시작 날짜
ID: MQCAMO_START_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH
리턴됨: 항상

IntervalStartTime

설명: 모니터링 기간의 시작 시간
ID: MQCAMO_START_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상

IntervalEndDate

설명: 모니터링 기간의 종료 날짜
ID: MQCAMO_END_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH
리턴됨: 항상

IntervalEndTime

설명: 모니터링 기간의 종료 시간
ID: MQCAMO_END_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상

CommandLevel

설명: 큐 관리자 명령 레벨
ID: MQIA_COMMAND_LEVEL
데이터 유형: MQCFIN
리턴됨: 항상

ObjectCount

설명: 통계 데이터가 기록된 간격에 액세스한 큐 오브젝트 수. 이 값은 메시지에 포함된 QStatisticsData PCF 그룹 수로 설정됩니다.
ID: MQIAMO_OBJECT_COUNT
데이터 유형: MQCFIN

리턴됨: 항상

QStatisticsData

설명: 큐에 대한 통계 세부사항을 지정하는 그룹화된 매개변수.

ID: MQGACF_Q_STATISTICS_DATA

데이터 유형: MQCFGR

그룹의 매개변수:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- QMinDepth*
- QMaxDepth*
- AvgTimeOnQ*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- NonQueuedMsgCount*
- ExpiredMsgCount*
- PurgeCount*

리턴됨: 항상

QName

설명: 큐의 이름.

ID: MQCA_Q_NAME

데이터 유형: MQCFST

최대 길이: MQ_Q_NAME_LENGTH

리턴됨: 항상

CreateDate

설명: 큐가 작성된 날짜

ID: MQCA_CREATION_DATE

데이터 유형: MQCFST

최대 길이: MQ_DATE_LENGTH

리턴됨: 항상

CreateTime

설명:	큐가 작성된 시간
ID:	MQCA_CREATION_TIME
데이터 유형:	MQCFST
최대 길이:	MQ_TIME_LENGTH
리턴됨:	항상

QType

설명:	큐의 유형
ID:	MQIA_Q_TYPE
데이터 유형:	MQCFIN
값:	MQOT_LOCAL
리턴됨:	항상

QDefinitionType

설명:	큐 정의 유형
ID:	MQIA_DEFINITION_TYPE
데이터 유형:	MQCFIN
값:	가능한 값은 다음과 같습니다. <ul style="list-style-type: none">• MQQDT_PREDEFINED• MQQDT_PERMANENT_DYNAMIC• MQQDT_TEMPORARY_DYNAMIC
리턴됨:	사용 가능할 때

QMinDepth

설명:	모니터링 기간의 최소 큐 용량
ID:	MQIAMO_Q_MIN_DEPTH
데이터 유형:	MQCFIN
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

QMaxDepth

설명:	모니터링 기간의 최대 큐 용량
ID:	MQIAMO_Q_MAX_DEPTH
데이터 유형:	MQCFIN
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

AvgTimeOnQ

설명:	모니터링 기간 중 큐에서 파괴적으로 검색한 메시지의 평균 대기 시간(마이크로초). 이 매개변수는 지속성 값에 의해 색인화되는 정수 목록입니다. 참조 노트 2 를 보십시오.
ID:	MQIAMO64_AVG_Q_TIME
데이터 유형:	MQCFIL64
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

PutCount

설명:	MQPUT1 요청을 제외하고, 큐에 넣은 지속 및 비지속 메시지의 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다. 참조 참고 2 를 참조하십시오.
ID:	MQIAMO_PUTS
데이터 유형:	MQCFIL
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

PutFailCount

설명:	큐에 메시지를 넣으려는 실패한 시도 수.
ID:	MQIAMO_PUTS_FAILED
데이터 유형:	MQCFIN
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

Put1Count

설명:	MQPUT1 호출을 사용하여 큐에 성공적으로 넣은 지속 및 비지속 메시지 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다. 참조 참고 2 를 참조하십시오.
ID:	MQIAMO_PUT1S
데이터 유형:	MQCFIL
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

Put1FailCount

설명:	MQPUT1 호출을 사용하여 메시지를 넣으려는 실패한 시도 수
ID:	MQIAMO_PUT1S_FAILED
데이터 유형:	MQCFIN
PCF 그룹에 포함:	<i>QStatisticsData</i>
리턴됨:	사용 가능할 때

PutBytes

설명:	큐에 대한 넣기 요청의 기록된 바이트 수
ID:	MQIAMO64_PUT_BYTES

데이터 유형: MQCFIL64
PCF 그룹에 포함: *QStatisticsData*
리턴됨: 사용 가능할 때

GetCount

설명: 지속 및 비지속 메시지에 대한 성공한 파괴적 가져오기 요청의 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다. [참조 참고 2](#)를 참조하십시오.

ID: MQIAMO_GETS
데이터 유형: MQCFIL
PCF 그룹에 포함: *QStatisticsData*
리턴됨: 사용 가능할 때

GetFailCount

설명: 실패한 파괴적 가져오기 요청의 수

ID: MQIAMO_GETS_FAILED
데이터 유형: MQCFIN
PCF 그룹에 포함: *QStatisticsData*
리턴됨: 사용 가능할 때

GetBytes

설명: 지속 및 비지속 메시지에 대한 파괴적 넣기 요청의 읽은 바이트 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다. [참조 참고 2](#)를 참조하십시오.

ID: MQIAMO64_GET_BYTES
데이터 유형: MQCFIL64
PCF 그룹에 포함: *QStatisticsData*
리턴됨: 사용 가능할 때

BrowseCount

설명: 지속 및 비지속 메시지에 대한 성공한 비파괴 가져오기 요청의 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다. [참조 참고 2](#)를 참조하십시오.

ID: MQIAMO_BROWSES
데이터 유형: MQCFIL
PCF 그룹에 포함: *QStatisticsData*
리턴됨: 사용 가능할 때

BrowseFailCount

설명: 실패한 비파괴 가져오기 요청의 수

ID: MQIAMO_BROWSES_FAILED
데이터 유형: MQCFIN
PCF 그룹에 포함: *QStatisticsData*
리턴됨: 사용 가능할 때

BrowseBytes

설명:	지속 및 비지속 메시지에 대한 비파괴 가져오기 요청의 읽은 바이트 수. 이 매개변수는 지속 값별로 색인화된 정수 목록입니다. 참조 참고 2 를 참조하십시오.
ID:	MQIAMO64_BROWSE_BYTES
데이터 유형:	MQCFIL64
PCF 그룹에 포함:	QStatisticsData
리턴됨:	사용 가능할 때

NonQueuedMsgCount

설명:	큐를 무시하고 대기 중인 애플리케이션에 직접 전송된 메시지의 수. 큐 무시는 특정 상황에만 발생할 수 있습니다. 이 수는 애플리케이션이 대기하는 횟수가 아닌, IBM MQ가 큐를 무시할 수 있는 횟수를 나타냅니다.
ID:	MQIAMO_MSGS_NOT_QUEUED
데이터 유형:	MQCFIN
PCF 그룹에 포함:	QStatisticsData
리턴됨:	사용 가능할 때

ExpiredMsgCount

설명:	검색 가능하게 되기 전에 만기되어서 제거된 지속 및 비지속 메시지의 수.
ID:	MQIAMO_MSGS_EXPIRED
데이터 유형:	MQCFIN
PCF 그룹에 포함:	QStatisticsData
리턴됨:	사용 가능할 때

PurgeCount

설명:	제거된 메시지의 수.
ID:	MQIAMO_MSGS_PURGED
데이터 유형:	MQCFIN
PCF 그룹에 포함:	QStatisticsData
리턴됨:	사용 가능할 때

채널 통계 메시지 데이터

이 페이지를 사용하여 채널 통계 메시지의 구조를 봅니다.

메시지 이름:	채널 통계 메시지.
플랫폼:	IBM MQ for z/OS를 제외한 모두.
시스템 큐:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

설명:	큐 관리자 이름.
ID:	MQCA_Q_MGR_NAME.
데이터 유형:	MQCFST.

최대 길이: MQ_Q_MGR_NAME_LENGTH.
리턴됨: 항상.

IntervalStartDate

설명: 모니터링 기간의 시작 날짜.
ID: MQCAMO_START_DATE.
데이터 유형: MQCFST.
최대 길이: MQ_DATE_LENGTH.
리턴됨: 항상.

IntervalStartTime

설명: 모니터링 기간의 시작 시간.
ID: MQCAMO_START_TIME.
데이터 유형: MQCFST.
최대 길이: MQ_TIME_LENGTH.
리턴됨: 항상.

IntervalEndDate

설명: 모니터링 기간의 종료 날짜
ID: MQCAMO_END_DATE.
데이터 유형: MQCFST.
최대 길이: MQ_DATE_LENGTH.
리턴됨: 항상.

IntervalEndTime

설명: 모니터링 기간의 종료 시간
ID: MQCAMO_END_TIME.
데이터 유형: MQCFST.
최대 길이: MQ_TIME_LENGTH
리턴됨: 항상.

CommandLevel

설명: 큐 관리자 명령 레벨.
ID: MQIA_COMMAND_LEVEL.
데이터 유형: MQCFIN.
리턴됨: 항상.

ObjectCount

설명: 통계 데이터가 기록된 간격에 액세스한 채널 오브젝트의 수. 이 값은 메시지에 포함된 ChIStatisticsData PCF 그룹 수로 설정됩니다.
ID: MQIAMO_OBJECT_COUNT
데이터 유형: MQCFIN.

리턴됨: 항상.

ChlStatisticsData

설명: 채널에 대한 통계 세부사항을 지정하는 그룹화된 매개변수.

ID: MQGACF_CHL_STATISTICS_DATA.

데이터 유형: MQCFGR.

그룹의 매개변수:

- ChannelName*
- ChannelType*
- RemoteQmgr*
- ConnectionName*
- MsgCount*
- TotalBytes*
- NetTimeMin*
- NetTimeAvg*
- NetTimeMax*
- ExitTimeMin*
- ExitTimeAvg*
- ExitTimeMax*
- FullBatchCount*
- IncplBatchCount*
- AverageBatchSize*
- PutRetryCount*

리턴됨: 항상.

ChannelName

설명: 채널의 이름.

ID: MQCACH_CHANNEL_NAME.

데이터 유형: MQCFST.

최대 길이: MQ_CHANNEL_NAME_LENGTH.

리턴됨: 항상.

ChannelType

설명: 채널 유형.

ID: MQIACH_CHANNEL_TYPE.

데이터 유형: MQCFIN.

값: 가능한 값은 다음과 같습니다.

MQCHT_SENDER

송신자 채널.

MQCHT_SERVER

서버 채널.

MQCHT_RECEIVER

수신자 채널.

MQCHT_REQUESTER

요청자 채널.

MQCHT_CLUSRCVR

클러스터 수신자 채널.

MQCHT_CLUSSDR

클러스터 송신자 채널.

리턴됨: 항상.

RemoteQmgr

설명: 리모트 큐 관리자의 이름.
ID: MQCA_REMOTE_Q_MGR_NAME.
데이터 유형: MQCFST.
최대 길이: MQ_Q_MGR_NAME_LENGTH
리턴됨: 사용 가능할 때.

ConnectionName

설명: 리모트 큐 관리자의 연결 이름.
ID: MQCACH_CONNECTION_NAME.
데이터 유형: MQCFST
최대 길이: MQ_CONN_NAME_LENGTH
리턴됨: 사용 가능할 때.

MsgCount

설명: 송신 또는 수신된 지속 및 비지속 메시지의 수.
ID: MQIAMO_MSGS.
데이터 유형: MQCFIN
리턴됨: 사용 가능할 때.

TotalBytes

설명: 지속 및 비지속 메시지의 송신 또는 수신된 바이트 수.
ID: MQIAMO64_BYTES.
데이터 유형: MQCFIN64.
리턴됨: 사용 가능할 때.

NetTimeMin

설명: 레코딩 간격에 측정된 최단 기록 채널 라운드트립(마이크로초).

ID: MQIAMO_NET_TIME_MIN.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

NetTimeAvg

설명: 레코딩 간격에 측정된 평균 기록 채널 라운드트립(마이크로초).
ID: MQIAMO_NET_TIME_AVG.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

NetTimeMax

설명: 레코딩 간격에 측정된 최장 기록 채널 라운드트립(마이크로초).
ID: MQIAMO_NET_TIME_MAX.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

ExitTimeMin

설명: 레코딩 간격에 사용자 엑시트를 실행하는 데 소모한 최단 기록 시간(마이크로초).
ID: MQIAMO_EXIT_TIME_MIN.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

ExitTimeAvg

설명: 레코딩 간격에 사용자 엑시트를 실행하는 데 소모한 평균 기록 시간(마이크로초). 마이크로초로 측정됩니다.
ID: MQIAMO_EXIT_TIME_AVG.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

ExitTimeMax

설명: 레코딩 간격에 사용자 엑시트를 실행하는 데 소모한 최장 기록 시간(마이크로초). 마이크로초로 측정됩니다.
ID: MQIAMO_EXIT_TIME_MAX.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

FullBatchCount

설명: 채널 속성 BATCHSZ 또는 BATCHLIM의 값에 도달했기 때문에 전송되었으며 채널에서 처리한 배치 수.
ID: MQIAMO_FULL_BATCHES.
데이터 유형: MQCFIN.
리턴됨: 사용 가능할 때.

IncmplBatchCount

설명: 채널 속성 BATCHSZ 또는 BATCHLIM 의 값에 도달하지 않고 송신된 채널에 의해 처리된 배치 수입니다.

ID: MQIAMO_INCOMPLETE_BATCHES.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

AverageBatchSize

설명: 채널에서 처리된 배치의 평균 배치 크기.

ID: MQIAMO_AVG_BATCH_SIZE.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

PutRetryCount

설명: 시간 간격 중 메시지 넣기에 실패하여 재시도 루프에 들어간 횟수.

ID: MQIAMO_PUT_RETRIES.

데이터 유형: MQCFIN.

리턴됨: 사용 가능할 때.

참조 참고

이 페이지를 사용하여 회계 및 통계 메시지 구조에 대한 설명이 참조하는 참고를 봅니다.

다음 메시지 데이터 설명이 참고를 참조합니다.

- [144 페이지의 『MQI 회계 메시지 데이터』](#)
- [155 페이지의 『큐 회계 메시지 데이터』](#)
- [165 페이지의 『MQI 통계 메시지 데이터』](#)
- [175 페이지의 『큐 통계 메시지 데이터』](#)
- [181 페이지의 『채널 통계 메시지 데이터』](#)

1. 이 매개변수는 IBM MQ 오브젝트와 관련이 있습니다. 이 매개변수는 다음 상수로 색인화된 값(MQCFIL 또는 MQCFIL64) 배열입니다.

오브젝트 유형	값 컨텍스트
MQOT_Q (1)	큐 오브젝트에 관련된 값을 포함합니다.
MQOT_NAMELIST (2)	이름 목록 오브젝트에 관련된 값을 포함합니다.
MQOT_PROCESS (3)	프로세스 오브젝트에 관련된 값을 포함합니다.
MQOT_Q_MGR (5)	큐 관리자 오브젝트에 관련된 값을 포함합니다.
MQOT_CHANNEL (6)	채널 오브젝트에 관련된 값을 포함합니다.
MQOT_AUTH_INFO (7)	인증 정보 오브젝트에 관련된 값을 포함합니다.
MQOT_TOPIC (8)	토픽 오브젝트에 관련된 값을 포함합니다.

참고: 13 MQCFIL 또는 MQCFIL64 값의 배열이 리턴되지만 나열된 값만 의미가 있습니다.

2. 이 매개변수는 IBM MQ 메시지와 관련이 있습니다. 이 매개변수는 다음 상수로 색인화된 값(MQCFIL 또는 MQCFIL64) 배열입니다.

표 25. 지속 값별로 색인화된 배열	
상수	값
1	비지속 메시지에 대한 값을 포함합니다.
2	지속 메시지에 대한 값을 포함합니다.

참고: 각 배열에 대한 색인은 0에서 시작하므로 색인 1은 배열의 두 번째 행을 나타냅니다. 이 테이블에 나열되지 않은 배열의 요소는 회계 또는 통계 정보를 포함하지 않습니다.

애플리케이션 활동 추적

애플리케이션 활동 추적은 큐 관리자에 연결된 애플리케이션의 작동에 대한 자세한 정보를 생성합니다. 애플리케이션의 작동을 추적하고 IBM MQ 자원과 상호작용할 때 애플리케이션에 사용된 매개변수의 상세 보기를 제공합니다. 애플리케이션이 발행한 MQI 호출의 시퀀스도 표시합니다.

이벤트 모니터링, 메시지 모니터링, 회계 및 통계 메시지, 실시간 모니터링이 제공하는 정보 이상의 자세한 정보가 필요할 때 애플리케이션 활동 추적을 사용하십시오.

참고: 활동 추적은 각 애플리케이션에 대해 IBM MQ 연결에서 생성됩니다. 따라서 계정 메시지가 사용 가능한 경우 활동 추적 조작은 각 애플리케이션에 대한 MQI 계정 정보로 계수됩니다.

IBM MQ에서는 애플리케이션 활동 추적 데이터를 수집하는 두 가지 방법을 지원합니다.

- 애플리케이션 활동 추적 정보의 중앙 수집 - 애플리케이션 활동 추적을 수집한 후 활동 추적 PCF 메시지를 시스템 큐 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에 기록하여 읽습니다.
- 활동 추적 데이터 구독 - 특수 IBM MQ 시스템 토픽에 기록됩니다.

활동 추적은 IBM MQ for z/OS에서 지원되지 않습니다.

ALW 이제 IBM MQ가 지원하는 대부분의 프로그래밍 언어에서 애플리케이션 이름을 지정할 수 있으므로, 자세한 정보는 지원되는 프로그래밍 언어로 애플리케이션 이름 지정을 참조하십시오.

애플리케이션 활동 추적 정보의 중앙 수집 구성

애플리케이션 활동 추적 메시지는 PCF 메시지입니다. 구성 파일을 사용하여 활동 추적을 구성합니다. 애플리케이션 활동 추적 정보의 중앙 수집을 구성하려면 ACTVTRC 큐 관리자 속성을 설정하십시오. MQCONNX 옵션을 사용하여 연결 레벨에서 또는 활동 추적 구성 파일을 사용하여 애플리케이션 스탠자 레벨에서 이 설정을 대체할 수 있습니다.

이 태스크 정보

활동 추적 메시지는 MQMD 구조 즉, PCF(MQCFH) 헤더 구조와 뒤이은 여러 PCF 매개변수로 이루어집니다. PCF 매개변수 다음에는 일련의 ApplicationTraceData PCF 그룹이 있습니다. 이 PCF 그룹은 애플리케이션에 큐 관리자에 연결된 동안에 수행하는 MQI 조작에 대한 정보를 수집합니다. mqat.ini라는 구성 파일을 사용하여 활동 추적을 구성합니다.

애플리케이션 활동 추적 정보의 수집 여부를 제어하려면 하나 이상의 다음 설정을 구성합니다.

1. ACTVTRC 큐 관리자 속성.
2. ACTVCONO 설정(MQCONNX에 전달된 MQCNO 구조의).
3. 활동 추적 구성 파일 mqat.ini의 애플리케이션에 일치하는 스탠자.

이전 순서가 중요합니다. ACTVTRC 속성은 ACTVCONO 설정으로 대체되고, 이 설정은 mqat.ini 파일의 설정으로 대체됩니다.

다르게 명시되지 않는 한, 각 조작이 완료된 후에 추적 항목이 기록됩니다. 이 항목은 먼저 시스템 큐 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에 기록된 후 애플리케이션과 큐 관리자의 연결이 끊어질 때 애플리케이션 활동 추적 메시지에 기록됩니다. 장기 실행 애플리케이션의 경우 다음 이벤트가 발생하면 중간 이벤트가 기록됩니다.

- 연결 수명이 정의된 제한시간 값에 도달합니다.

- 조작 수가 지정된 수에 도달합니다.
- 메모리의 수집된 데이터 양이 큐의 허용된 최대 메시지 길이에 도달합니다.

ActivityInterval 매개변수를 사용하여 제한시간 값을 설정합니다. 조작 수는 **ActivityCount** 매개변수를 사용하여 설정합니다. 두 매개변수 모두 활동 추적 구성 파일 `mqat.ini`에 지정됩니다.

애플리케이션 활동 추적의 사용은 성능에 영향을 미칠 수 있습니다. **ActivityCount** 및 **ActivityInterval** 설정을 조정해서 오버헤드를 줄일 수 있습니다. 194 페이지의 『애플리케이션 활동 추적의 성능 영향 조정』의 내용을 참조하십시오.

애플리케이션 활동 추적 메시지의 콘텐츠를 보는 가장 단순한 방법은 194 페이지의 『amqsact 샘플 프로그램』을 사용하는 것입니다.

프로시저

1. 188 페이지의 『ACTVTRC를 설정하여 활동 추적 정보 콜렉션 제어』.
2. 188 페이지의 『MQCONNX를 설정하여 활동 추적 정보 콜렉션 제어』.
3. 189 페이지의 『mqat.ini을(를) 사용하여 활동 추적 작동 구성』.
4. 194 페이지의 『애플리케이션 활동 추적의 성능 영향 조정』.

ACTVTRC를 설정하여 활동 추적 정보 콜렉션 제어

큐 관리자 속성 ACTVTRC를 사용하여 MQI 애플리케이션 활동 추적 정보 콜렉션을 제어합니다.

이 태스크 정보

애플리케이션 활동 추적 메시지는 애플리케이션 활동 추적이 사용된 후에 시작한 연결에 대해서만 생성됩니다. **ACTVTRC** 매개변수의 가능한 값은 다음과 같습니다.

켜짐

API 활동 추적 콜렉션이 사용으로 설정됩니다.

OFF

API 활동 추적 콜렉션이 사용 안함으로 설정됩니다.

참고: **ACTVTRC** 설정은 큐 관리자 **ACTVCONO** 매개변수로 대체될 수 있습니다. **ACTVCONO** 매개변수를 **ENABLED**로 설정하면 **MQCNO** 구조에서 **Options** 필드를 사용하여 주어진 연결에 대한 **ACTVTRC** 설정을 대체할 수 있습니다. 188 페이지의 『MQCONNX를 설정하여 활동 추적 정보 콜렉션 제어』의 내용을 참조하십시오.

예

ACTVTRC 매개변수의 값을 변경하려면 MQSC 명령 ALTER QMGR을 사용합니다. 예를 들어, MQI 애플리케이션 활동 추적 정보 콜렉션을 사용하려면 다음 MQSC 명령을 사용하십시오.

```
ALTER QMGR ACTVTRC(ON)
```

다음에 수행할 작업

애플리케이션 활동 추적 메시지의 콘텐츠를 보는 가장 단순한 방법은 194 페이지의 『amqsact 샘플 프로그램』을 사용하는 것입니다.

애플리케이션 활동 추적의 사용은 성능에 영향을 미칠 수 있습니다. **ActivityCount** 및 **ActivityInterval** 설정을 조정해서 오버헤드를 줄일 수 있습니다. 194 페이지의 『애플리케이션 활동 추적의 성능 영향 조정』의 내용을 참조하십시오.

MQCONNX를 설정하여 활동 추적 정보 콜렉션 제어

큐 관리자 속성 **ACTVCONO**가 **ENABLED**로 설정되면 **MQCONNX** 호출에 **ConnectOpts** 매개변수를 사용하여 각 연결별로 애플리케이션 활동 보고서를 사용 가능 또는 불가능하게 할 수 있습니다. 이 옵션은 큐 관리자 속성 **ACTVTRC**에 정의된 활동 추적 작동을 대체하며 활동 추적 구성 파일 `mqat.ini`의 설정으로 대체될 수 있습니다.

프로시저

1. 큐 관리자 속성 **ACTVCONO**를 ENABLED로 설정하십시오.

참고: 애플리케이션이 **ConnectOpts** 매개변수를 사용하여 애플리케이션의 회계 작동을 수정하려 시도하며 QMGR 속성 **ACTVCONO**가 DISABLED로 설정된 경우에는, 애플리케이션에 오류가 리턴되지 않고 활동 추적 콜렉션이 큐 관리자 속성 또는 활동 추적 구성 파일 `mqat.ini`를 통해 정의됩니다.

2. MQCONNX 호출의 **ConnectOpts** 매개변수를 MQCNO_ACTIVITY_TRACE_ENABLED로 설정하십시오.

MQCONNX 호출의 **ConnectOpts** 매개변수의 가능한 값은 다음과 같습니다.

MQCNO_ACTIVITY_TRACE_DISABLED

연결에 대한 활동 추적이 사용 안함으로 설정됩니다.

MQCNO_ACTIVITY_TRACE_ENABLED

연결에 대한 활동 추적이 사용으로 설정됩니다.

참고: 애플리케이션이 MQCONNX에 대해 MQCNO_ACTIVITY_TRACE_ENABLED와 MQCNO_ACTIVITY_TRACE_DISABLED를 모두 선택하는 경우 호출이 실패하고 이유 코드 MQRC_OPTIONS_ERROR가 표시됩니다.

3. 이러한 활동 추적 설정이 활동 추적 구성 파일 `mqat.ini`의 설정으로 대체되지 않는지 확인하십시오.

[189 페이지의 『mqat.ini을\(를\) 사용하여 활동 추적 작동 구성』](#)의 내용을 참조하십시오.

다음에 수행할 작업

애플리케이션 활동 추적 메시지의 콘텐츠를 보는 가장 단순한 방법은 [194 페이지의 『amqsact 샘플 프로그램』](#)을 사용하는 것입니다.

애플리케이션 활동 추적의 사용은 성능에 영향을 미칠 수 있습니다. **ActivityCount** 및 **ActivityInterval** 설정을 조정해서 오버헤드를 줄일 수 있습니다. [194 페이지의 『애플리케이션 활동 추적의 성능 영향 조정』](#)의 내용을 참조하십시오.

mqat.ini을(를) 사용하여 활동 추적 작동 구성

활동 추적 작동은 `mqat.ini`(이)라는 구성 파일을 사용하여 구성합니다. 이 파일은 활동 추적 데이터의 보고 레벨 및 빈도를 정의하는 데 사용됩니다. 파일은 애플리케이션의 이름을 기반으로 활동 추적을 사용 가능 및 불가능하게 하는 규칙을 정의할 방법도 제공합니다.

이 태스크 정보

Linux **AIX** AIX and Linux 시스템에서는 `mqat.ini`(가) 큐 관리자 데이터 디렉토리에 있으며, 이는 `qm.ini` 파일과 동일한 위치입니다.

Windows Windows 시스템에서는 `mqat.ini`(가) 큐 관리자 데이터 디렉토리 `C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name`에 있습니다. 추적할 애플리케이션을 실행하는 사용자에게는 이 파일을 읽을 권한이 필요합니다.

참고: IBM WebSphere MQ 7.1 이하 버전에서 마이그레이션된 큐 관리자에는 `mqat.ini` 파일이 없습니다. 이러한 경우 `mqat.ini` 파일을 수동으로 작성해야 하며 파일에 660 권한이 설정되어야 합니다.

`mqat.ini` 파일이 수정되면 새로 작성된 IBM MQ 연결이 수정된 버전에 따라 처리됩니다. 큐 관리자 매개변수가 변경(예를 들어, `ALTER QMGR` 명령 후에)되지 않으면 기존 연결은 계속해서 이전 버전을 사용합니다.

이 파일은 `mq.s.ini` 및 `qm.ini` 파일과 동일한 스탠자 키 및 매개변수-값 쌍 형식을 따릅니다.

파일은 기본적으로 모든 활동 추적에 대한 활동 추적 데이터의 보고 레벨 및 빈도를 구성하는 단일 스탠자, **AllActivityTrace**로 이루어집니다.

여러 **ApplicationTrace** 스탠자가 파일에 포함될 수도 있습니다. 이들 각 스탠자는 연결의 애플리케이션 이름을 규칙에 일치시켜서 하나 이상의 연결의 추적 작동에 대한 규칙을 정의합니다.

AllActivityTrace 스탠자

단일 **AllActivityTrace** 스탠자는 대체되지 않는 한, 모든 IBM MQ 연결에 적용되는 활동 추적에 대한 설정을 정의합니다.

AllActivityTrace 스탠자의 개별 값은 **ApplicationTrace** 스탠자의 보다 구체적 정보로 대체될 수 있습니다.

둘 이상의 **AllActivityTrace** 스탠자가 지정되면 마지막 스탠자의 값이 사용됩니다. 선택된 **AllActivityTrace**에서 누락된 매개변수는 기본값을 사용합니다. 이전 **AllActivityTrace** 스탠자의 매개변수와 값은 무시됩니다.

AllActivityTrace 스탠자 아래에 다음 매개변수를 지정할 수 있습니다.

표 26. 활동 추적 구성 파일에 사용 가능한 매개변수/값 쌍		
이름	값(기본값은 굵게)	설명
ActivityInterval	0-999999999(1)	추적 메시지 간의 대략적 시간 간격(초). 이 간격에 연결을 통해 수행된 모든 활동은 단일 메시지에 기록됩니다. 값이 0이면 연결이 끊어질 때(또는 활동 수에 도달할 때) 추적 메시지가 기록됩니다.
ActivityCount	0-999999999(100)	추적 메시지 간의 MQI 또는 XA 조작 수. 값이 0이면 연결이 끊어질 때(또는 활동 간격이 경과되었을 때) 추적 메시지가 기록됩니다.
TraceLevel	LOW / MEDIUM / HIGH	각 조작에 대해 추적한 매개변수 세부사항의 양. 각 추적 레벨의 매개변수가 포함된 개별 조작 세부사항의 설명.
TraceMessageData	0 - 104 857 600(최대 100MB)	MQGET, MQPUT, MQPUT1, 콜백 조작에 대해 추적한 메시지 데이터의 양(바이트)
StopOnGetTraceMsg	ON / OFF	루프 발생 가능성으로 인해, 활동 추적 메시지도 처리 중인 애플리케이션을 활동 추적을 사용하여 추적하는 것은 바람직하지 않습니다.
SubscriptionDelivery	배치됨 / 즉시	하나 이상의 활동 추적 구독이 있을 때 ActivityInterval 및 ActivityCount 매개변수를 사용하는지 여부를 결정합니다. 이 매개변수를 IMMEDIATE로 설정하면 추적 데이터에 일치하는 구독이 있을 경우 ActivityInterval 및 ActivityCount 값이 유효값 1로 대체됩니다. 각 활동 추적 레코드가 동일한 연결의 다른 레코드와 배치되지 않고, 대신 지체하지 않고 구독으로 즉시 전달됩니다.

ApplicationTrace 스탠자

ApplicationTrace 스탠자는 애플리케이션 이름에 기초하여 추적하거나 추적하지 않을 IBM MQ 연결을 정의하는 규칙을 포함합니다. 선택적으로 글로벌 추적 레벨 및 빈도 설정을 대체하는 Allsettings 아래에 정의된 기본 작동.




이 스탠자는 연결 일치 규칙에 정의된 일치 규칙에 따라 스탠자가 특정 연결에 적용되는지 여부를 판별하는 데 사용되는 ApplName, ApplFunction, ApplClass 매개변수를 포함할 수 있습니다.

이 규칙이 일치하는 연결에 대해 활동 추적을 켜거나 끄지 여부를 판별하는 Trace 매개변수가 스탠자에 포함되어야 합니다.

OFF 규칙을 사용하여 특정 애플리케이션 이름에 대한 추적을 명시적으로 사용 불가능하게 하고 큐 관리자 또는 활동 추적 연결 옵션의 ACTVTRC 설정을 대체할 수 있습니다.

ApplicationTrace 스탠자 아래에 다음 매개변수를 지정할 수 있습니다.

표 27. 애플리케이션 추적 구성 파일에 사용 가능한 매개변수/값 쌍

이름	값(기본값은 굵게)	설명
추적	ON / OFF(필수 매개변수 - 기본값 없음)	활동 추적 스위치. 이 스위치는 현재 애플리케이션 스탠자 범위에 대한 활동 추적이 활성화되는지 여부를 판별하는 용도로 애플리케이션별 스탠자에 사용할 수 있습니다. 이 값은 큐 관리자의 ACTVTRC 및 ACTVCONO 설정을 대체함에 유의하십시오.
ApplName	문자열(필수 매개변수 - 기본값 없음)	이 값은 ApplicationTrace 스탠자가 적용되는 애플리케이션을 판별하는 데 사용됩니다. API 엑시트 컨텍스트 구조의 ApplName 값 (MQMD.PutApplName에 해당함)에 일치합니다. ApplName 값의 콘텐츠는 애플리케이션 환경에 따라 다릅니다. z/OS 이외의 플랫폼에서는 MQAXC.ApplName의 파일 이름 부분만 스탠자의 값에 일치합니다. 비교를 수행할 때 오른쪽 끝에 있는 경로 구분 기호 왼쪽의 문자는 무시됩니다.  z/OS 애플리케이션의 경우 전체 MQAXC.ApplName이 스탠자의 값에 일치합니다. ApplName 값의 끝에 단일 와일드카드 문자(*)를 사용하여 해당 지점 이후의 임의의 수의 문자를 일치시킬 수 있습니다. ApplName 값이 단일 와일드카드 문자(*)로 설정되면 ApplName 값은 모든 애플리케이션에 일치합니다.
  ApplFunction	문자열(기본값 *)	이 값은 ApplicationTrace 스탠자와 ApplName 값이 적용되는 애플리케이션 프로그램을 규정하는 데 사용됩니다. 스탠자는 선택적이지만 IBM i 큐 관리자에게만 유효합니다. ApplName 값의 끝에 단일 와일드카드 문자(*)를 사용하여 임의의 수의 문자를 일치시킬 수 있습니다. 예를 들어, ApplName = * 및 ApplFunction = AMQSPUTO 를 지정하는 ApplicationTrace 스탠자는 작업의 모든 AMQSPUTO 프로그램 호출에 적용됩니다.
ApplClass	USER / MCA / ALL	애플리케이션의 클래스. AppType 값을 IBM MQ 연결에 일치시키는 방식에 대한 설명은 다음 표를 참조하십시오.

다음 표는 연결 API 엑시트 컨텍스트 구조에서 **AppClass** 값을 **APICallerType** 및 **APIEnvironment** 필드에 일치시키는 방식을 보여줍니다.

표 28. **Appclass** 값과 **APICallerType** 및 **APIEnvironment** 필드에 이 값을 일치시키는 방식

APPLCLASS	API 호출자 유형:	API 환경:	설명
사용자	MQXACT_EXTERNAL	MQXE_OTHER	사용자 애플리케이션만 추적함
MCA	(임의의 값)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	클라이언트 및 채널(amqrmppa)

표 28. <i>Appclass</i> 값과 <i>APICallerType</i> 및 <i>APIEnvironment</i> 필드에 이 값을 일치시키는 방식 (계속)			
APPLCLASS	API 호출자 유형:	API 환경:	설명
모두	(임의의 값)	(임의의 값)	모든 연결을 추적함



주의: 클라이언트 사용자 애플리케이션에 대해 **APPLCLASS/MCA**를 사용해야 합니다. *USER*의 한 클래스로서 이들과 일치되지 않습니다.

예를 들어 **amqsputc** 샘플 애플리케이션을 추적하려면 다음 코드를 사용합니다.

```

ApplicationTrace:
App1Class=MCA                                # Application type
                                              # Values: (USER | MCA | INTERNAL | ALL)
                                              # Default: USER
App1Name=amqsputc    # Application name (may be wildcard)
                                              # (matched to app name without path)
                                              # Default: *
Trace=ON                                        # Activity trace switch for application
                                              # Values: ( ON | OFF )
                                              # Default: OFF
ActivityInterval=30    # Time interval between trace messages
                                              # Values: 0-99999999 (0=off)
                                              # Default: 0
ActivityCount=1        # Number of operations between trace msgs
                                              # Values: 0-99999999 (0=off)
                                              # Default: 0
TraceLevel=MEDIUM    # Amount of data traced for each operation
                                              # Values: LOW | MEDIUM | HIGH
                                              # Default: MEDIUM
TraceMessageData=1000 # Amount of message data traced
                                              # Values: 0-1000000000
                                              # Default: 0

```

큐 관리자가 작성될 때 생성된 기본 *mqat.ini*는 제공된 활동 추적 샘플, **amqsact**에 대한 활동 추적을 명시적으로 사용 불가능하게 하는 단일 규칙을 포함합니다.

연결 일치 규칙

큐 관리자는 연결에 사용할 스탠자 설정을 판별하기 위해 다음 규칙을 적용합니다.

1. **ApplicationTrace** 스탠자에도 값이 발생하며 스탠자가 2, 3, 4 지점에 설명된 연결에 대한 일치 기준을 이행하는 경우가 아니면, **AllActivityTrace** 스탠자에 지정된 값이 연결에 사용됩니다.
2. **App1Class**는 IBM MQ 연결 유형에 일치시킵니다. **App1Class**가 연결 유형에 일치하지 않으면 스탠자가 이 연결에 대해 무시됩니다.
3. 스탠자의 **App1Name** 값은 연결에 대한 API 엑시트 컨텍스트 구조(MQAXC)에서 **App1Name** 필드의 파일 이름 부분에 일치시킵니다.

파일 이름 부분은 마지막 경로 구분 기호(/ 또는 \) 문자의 오른쪽에 있는 문자에서 파생됩니다. **App1Name** 스탠자가 와일드카드(*)를 포함하면 와일드카드 왼쪽의 문자만을 연결 **App1Name**의 해당하는 수의 문자와 비교합니다.

예를 들어, 스탠자 값 "FRE*"가 지정되면 처음 세 개의 문자만 비교에 사용되므로 "path/FREEDOM" 및 "path\FREDDY"는 일치하지만 "path/FRIEND"는 일치하지 않습니다. 스탠자의 **App1Name** 값이 연결 **App1Name**에 일치하지 않는 경우에는 스탠자가 이 연결에 대해 무시됩니다.

4. 둘 이상의 스탠자가 연결의 **App1Name** 및 **App1Class**에 일치하면 **App1Name**이 가장 고유한 스탠자가 사용됩니다.

가장 고유한 **App1Name**은 연결의 **App1Name**에 일치시킬 가장 많은 문자를 사용하는 이름으로 정의됩니다.

예를 들어, ini 파일이 **App1Name = "FRE*"**인 스탠자와 **App1Name = "FREE*"**인 또 다른 스탠자를 포함하면 네 개의 문자가 일치하기 때문에 **App1Name = "FREE*"**인 스탠자가 **App1Name = "path/FREEDOM"**인 연결에 대한 최상의 일치로 선택됩니다(반면에 **App1Name = "FRE*"**는 세 개의 문자만 일치함).

5. 2, 3, 4 지점에 규칙을 적용한 후 연결의 연결 **App1Name** 및 **App1Class**에 일치하는 스탠자가 둘 이상인 경우에는 마지막 일치의 값이 사용되고 다른 모든 스탠자는 무시됩니다.

각 규칙의 기본 설정 대체

선택적으로 **AllActivityTrace** 스탠자 아래의 글로벌 추적 레벨 및 빈도 설정을 **ApplicationTrace** 스탠자에 일치하는 연결에 맞게 대체할 수 있습니다.

ApplicationTrace 스탠자 아래에 다음 매개변수를 설정할 수 있습니다. 설정하지 않으면 **AllActivityTrace** 스탠자 설정에서 값이 상속됩니다.

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**
- **StopOnTraceMsg**

mqat.ini 구문

mqat.ini 파일의 형식에 대한 구문 규칙은 다음과 같습니다.

- 해시 또는 세미콜론으로 시작하는 텍스트는 행의 끝까지 확장하는 주석으로 간주합니다.
- 첫 번째 유효(무주석) 행은 스탠자 키여야 합니다.
- 스탠자 키는 스탠자의 이름과 뒤이은 콜론으로 이루어집니다.
- 매개변수-값 쌍은 매개변수의 이름과 뒤이은 등호 부호 및 값으로 이루어집니다.
- 한 행에는 단일 매개변수-값 쌍만 표시할 수 있습니다. (매개변수-값이 다른 행으로 넘어가면 안됩니다.)
- 선두 및 후미 공백은 무시됩니다. 스탠자 이름, 매개변수 이름과 값 또는 매개변수/값 쌍 사이의 공백 수에는 제한이 없습니다. 행 바꾸기는 유효하며 무시되지 않습니다.
- 행의 최대 길이는 2048자입니다.
- 스탠자 키, 매개변수 이름, 상수 매개변수 값은 대소문자를 구분하지 않지만 변수 매개변수 값(*ApplName* 및 *DebugPath*)은 대소문자를 구분합니다.

애플리케이션 활동 추적 파일 예

다음 예는 활동 추적 ini 파일에 구성 데이터가 지정되는 방식을 보여줍니다.

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
ApplName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
ApplName=amqsact*
Trace=OFF
```

위의 **AllActivityTrace** 스탠자는 **ApplicationTrace** 규칙이나 큐 관리자 ACTVTRC 속성을 통해 또는 애플리케이션을 통해 프로그래밍 방식으로 사용 가능하게 해서 활동 추적이 사용 가능할 때 수행하는 방식을 정의합니다.

첫 번째 **ApplicationTrace** 스탠자는 이름이 "amqs"로 시작하며 추적 중인 애플리케이션을 통해 MQI 활동을 야기할 규칙을 정의합니다. 이 애플리케이션에 대해 생성되는 추적은 높은 세부사항 레벨이며 최대 1000바이트의 메시지 데이터를 포함합니다. 활동 간격 및 수 매개변수는 상속됩니다.

두 번째 **ApplicationTrace** 스탠자는 이름이 "amqsact"(활동 추적 샘플)로 시작하는 애플리케이션에 대한 추적을 끄는 규칙을 정의합니다. 이 규칙은 amqsact 애플리케이션에 대한 이전의 'on' 규칙을 대체하므로 결과적으로 이 애플리케이션을 추적하지 않습니다.

C 샘플 디렉토리(amqsact.c 파일과 동일한 디렉토리)에 있는 mqat.ini(이)라는 샘플도 예로 제공됩니다. IBM MQ의 이전 릴리스에서 마이그레이션된 큐 관리자의 경우 큐 관리자 데이터 디렉토리에 이 파일을 복사할 수 있습니다.

다음에 수행할 작업

애플리케이션 활동 추적의 사용은 성능에 영향을 미칠 수 있습니다. **ActivityCount** 및 **ActivityInterval** 설정을 조정해서 오버헤드를 줄일 수 있습니다. 194 페이지의 『애플리케이션 활동 추적의 성능 영향 조정』의 내용을 참조하십시오.

애플리케이션 활동 추적의 성능 영향 조정

애플리케이션 활동 추적을 사용하면 성능이 저하될 수 있습니다. 큐를 드레인하는 애플리케이션의 수를 늘리거나 mqat.ini에서 **ActivityInterval**, **ActivityCount**, **TraceLevel**을 조정해서 필요한 애플리케이션만 추적함으로써 이 영향을 감소시킬 수 있습니다.

이 태스크 정보

선택적으로 한 애플리케이션에 또는 모든 큐 관리자 애플리케이션에 애플리케이션 활동 추적을 사용하면 추가 메시징 활동이 유발되어 큐 관리자에 추가 스토리지 공간이 필요하게 됩니다. 워크로드가 높은 애플리케이션이나 SLA(Service Level Agreement)에 메시징 제공자로부터의 최소 응답이 시간이 요구되는 환경처럼 메시징 성능이 중요한 환경에서는, 애플리케이션 활동 추적을 수집하는 것이 적합하지 않거나 생성되는 추적 활동 메시지의 세부사항 또는 빈도를 조정해야 할 수 있습니다. mqat.ini 파일에서 **ActivityInterval**, **ActivityCount**, **TraceLevel**의 사전 설정된 값은 세부사항 및 성능의 기본 밸런스를 제공합니다. 하지만 시스템의 세밀한 기능 및 성능 요구사항에 맞게 이 값을 조정할 수 있습니다.

프로시저

- 필요한 애플리케이션만 추적하십시오.

mqat.ini에 ApplicationTrace 애플리케이션별 스탠자를 작성하거나 MQCONNX 호출에서 **MQCNO** 구조의 옵션 필드에 MQCNO_ACTIVITY_TRACE_ENABLED를 지정하도록 애플리케이션을 변경하여 이를 수행하십시오. 189 페이지의 『mqat.ini을(를) 사용하여 활동 추적 작동 구성』 and 188 페이지의 『MQCONNX를 설정하여 활동 추적 정보 콜렉션 제어』의 내용을 참조하십시오.

- 추적을 시작하기 전에 최소 하나의 애플리케이션이 실행 중이며 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에서 활동 추적 메시지 데이터를 검색할 준비가 되어 있는지 확인하십시오.

- 큐를 드레인하는 애플리케이션의 수를 늘려서 큐 용량을 가능한 낮게 유지하십시오.

- mqat.ini 파일에 **TraceLevel** 값을 설정하여 필요한 최소 데이터를 수집하십시오.

TraceLevel=LOW가 메시징 성능에 가장 적은 영향을 미칩니다. 189 페이지의 『mqat.ini을(를) 사용하여 활동 추적 작동 구성』의 내용을 참조하십시오.

- mqat.ini에서 **ActivityCount** 및 **ActivityInterval** 값을 조정하여 활동 추적 메시지가 생성되는 빈도를 조정하십시오.

여러 애플리케이션을 추적하는 경우에는 활동 추적 메시지가 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에서 제거 가능한 것보다 더 빠르게 생성되고 있는 것일 수 있습니다. 하지만 활동 추적 메시지가 생성되는 빈도를 줄이면 그와 동시에 큐 관리자에 필요한 스토리지 공간 및 큐에 기록될 때의 메시지 크기를 늘리는 것이기도 합니다.

다음에 수행할 작업

애플리케이션 활동 추적 메시지의 콘텐츠를 보는 가장 단순한 방법은 194 페이지의 『amqsact 샘플 프로그램』을 사용하는 것입니다.

amqsact 샘플 프로그램

amqsact는 애플리케이션 활동 추적 메시지를 형식화하며 IBM MQ와 함께 제공됩니다.

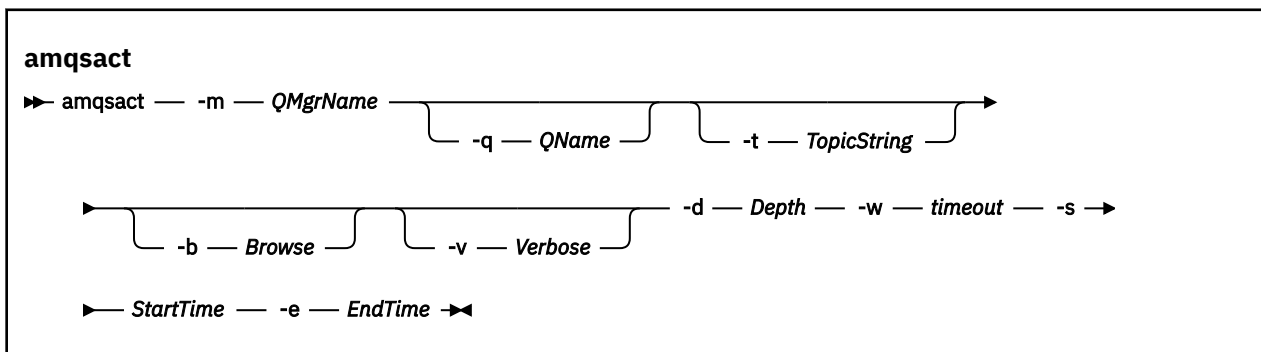
컴파일된 프로그램은 다음과 같은 샘플 디렉토리에 있습니다.

- **Linux** **AIX** AIX and Linux MQ_INSTALLATION_PATH/samp/bin의 경우
- **Windows** Windows MQ_INSTALLATION_PATH\tools\c\Samples\Bin의 경우

표시 모드

기본적으로 표시 모드에서 **amqsact**는 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE의 메시지를 처리합니다. 큐 이름이나 토픽 문자열을 지정하여 이 작동을 대체할 수 있습니다.

표시된 추적 기간을 제어하고 활동 추적 메시지를 표시 후 제거하거나 보유할지 여부를 지정할 수도 있습니다.



표시 모드의 필수 매개변수

-m *QMgrName*

큐 관리자의 이름.

-d *Depth*

표시할 레코드 수.

-w *Timeout*

대기 시간(초). 지정된 기간에 추적 메시지가 표시되지 않을 경우 **amqsact**가 종료됩니다.

-s *StartTime*

처리할 레코드의 시작 시간.

-e *EndTime*

처리할 레코드의 종료 시간.

표시 모드의 선택적 매개변수

-q *QName*

기본 큐 이름을 대체할 특정 큐를 지정하십시오.

-t *TopicString*

이벤트 토픽을 구독함

-b

레코드만 찾아봄

-v

상세 출력

표시 모드의 출력 예

amqsact를 큐 관리자 *TESTQM*에 사용하며 MQCONN API 호출에 대해 상세 출력합니다.

```
amqsact -m TESTQM -v
```

이전 명령의 출력 예는 다음과 같습니다.

```

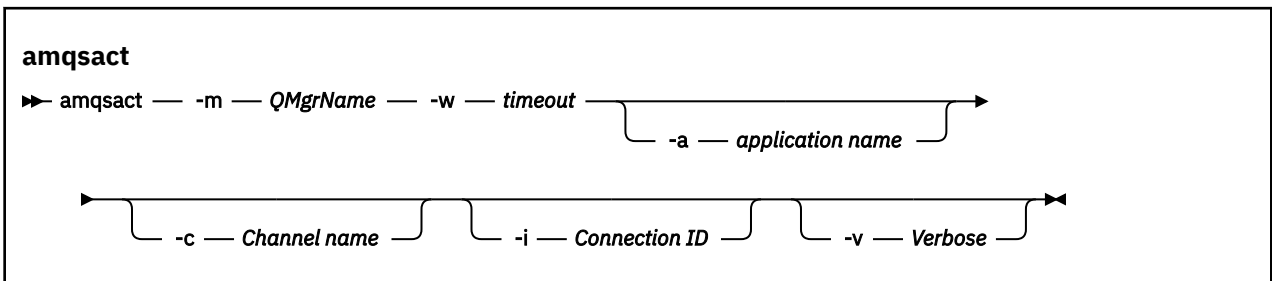
MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: B5F6 4251 2000 E601
QueueManager: 'TESTQM'
Host Name: 'ADMINIB-1VTJ6N1'
IntervalStartDate: '2014-03-15'
IntervalStartTime: '12:08:10'
IntervalEndDate: '2014-03-15'
IntervalEndTime: '12:08:10'
CommandLevel: 750
SeqNumber: 0
ApplicationName: 'IBM MQ_1\bin\amqsput.exe'
ApplicationType: MQAT_WINDOWS_7
ApplicationPid: 14076
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: '-'
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationTid: 1
OperationDate: '2014-03-15'
OperationTime: '12:08:10'
ConnectionId:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: FFFFFFFB5FFFFFFF6 4251 2000 FFFFFFFE601
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0

```

동적 모드

애플리케이션 이름, 채널 이름 또는 연결 ID를 **amqsact**에 대한 인수로 지정하여 동적 모드를 사용으로 설정할 수 있습니다. 이름에 와일드카드 문자를 사용할 수 있음에 유의하십시오.

동적 모드에서는, 시스템 토픽의 지속 불가능한 구독을 사용하여 샘플 시작 시에 활동 추적 데이터가 사용 가능하게 됩니다. **amqsact**가 중지되면 활동 추적 데이터 수집이 중지됩니다. 동적 모드의 **amqsact**에 대한 제한시간을 지정해야 합니다. 각 인스턴스가 활동 추적 데이터의 한 사본을 수신하는 **amqsact**의 다중 사본을 동시에 실행할 수 있습니다.



동적 모드의 필수 매개변수

-m *QMgrName*

큐 관리자의 이름.

-w *Timeout*

대기 시간(초). 지정된 기간에 추적 메시지가 표시되지 않을 경우 **amqsact**가 종료됩니다.

동적 모드의 선택적 매개변수

-a *Application name*

메시지를 수집할 애플리케이션 이름을 지정하십시오.

-c Channel name

메시지를 수집할 채널을 지정하십시오.

-i Connection ID

메시지를 수집할 연결을 지정하십시오.

-v

상세 출력

동적 모드의 출력 예

다음 명령은 텍스트 "amqs"로 시작하는 애플리케이션이 수행한 연결에 대한 활동 추적 메시지를 생성하고 표시합니다. 무활동 30초가 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

다음 명령은 QMGR1.TO.QMGR2 채널의 활동에 대한 활동 추적 메시지를 생성하고 표시합니다. 무활동 10초가 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

다음 명령은 CONN이 "6B576B5420000701"이고 EXTCONN이 "414D5143514D47523120202020202020"인 기존 IBM MQ 연결에 대한 활동의 상세 활동 추적 메시지를 생성하고 표시합니다. 무활동 1분이 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

애플리케이션 활동 추적 정보 구독

큐 관리자 레벨 구성을 통해 정보를 수집하는 대신 애플리케이션 활동 추적 정보를 동적으로 구독할 수 있습니다.

이 태스크 정보

애플리케이션 활동 추적은 애플리케이션 동작을 추적하고 애플리케이션이 IBM MQ 자원과 상호작용할 때 사용하는 매개변수의 세부사항 보기를 제공합니다. 애플리케이션이 발행한 MQI 호출의 시퀀스도 표시합니다.

시스템 큐에 추적 데이터를 기록하는 것 외에도 제품은 큐 관리자 레벨 구성을 통해 정보를 수집하지 않고 특수 IBM MQ 시스템 토픽에 기록된 활동 추적 데이터를 동적으로 구독하는 기능을 제공합니다.

IBM MQ 9.0에서 제품은 이 용도로 엑시트를 사용하지 않습니다. 이전에 애플리케이션 활동을 추적하기 위해 엑시트를 사용한 경우, 애플리케이션 활동 추적을 수집하기 위해 대체 메소드를 사용하도록 전환해야 합니다.

세션에서 메시지 생성자 작성 큐 관리자 또는 애플리케이션 속성을 설정할 필요가 없습니다. 그러나 큐 관리자 또는 애플리케이션에서 추적을 사용 불가능하게 설정하여 활동 추적의 차단을 이용하십시오.

프로시저

- [197 페이지의 『애플리케이션 활동 추적에 대한 구독』](#)
- [198 페이지의 『애플리케이션 활동 추적에 대한 구독 작성』](#)
- [199 페이지의 『amqsact를 사용하여 추적 메시지 보기』](#)
- [201 페이지의 『mqat.ini를 사용하여 추적 레벨 구성』](#)

애플리케이션 활동 추적에 대한 구독

IBM MQ 시스템 토픽을 구독하여 애플리케이션 활동 추적 정보를 수집할 수 있습니다.

추적할 활동을 나타내는 특수한 IBM MQ 시스템 토픽 문자열을 구독하십시오. 자동으로 구독하면 활동 추적 데이터 메시지를 생성하여 구독 목적지 큐에 발행합니다. 구독을 삭제하면 해당 구독에 대한 활동 추적 데이터의 생성이 중지됩니다.

발행시 다음 자원 중 하나에 대한 활동을 추적할 수 있습니다.

- 지정된 애플리케이션
- 지정된 IBM MQ 채널
- 기존 IBM MQ 연결

다른 토픽 문자열 또는 동일한 토픽 문자열이 포함된 여러 구독을 작성할 수 있습니다. 동일한 시스템 활동 추적 토픽 문자열을 포함하는 여러 구독을 작성할 경우 각 구독은 활동 추적 데이터 사본을 수신하고 이는 부정적인 성능 영향을 미칠 수 있습니다.

모든 레벨의 활동 추적을 사용 가능하게 하면 부정적인 성능 영향을 미칠 수 있습니다. 구독이 많을수록 또는 구독되는 자원이 많을수록 잠재적인 성능 오버헤드가 더 커집니다. 활동 추적 수집의 오버헤드를 최소화하기 위해 데이터가 메시지에 기록되고 애플리케이션 활동에서 구독으로 비동기로 전달됩니다. 자주 여러 조작이 단일 활동 추적 데이터 메시지에 기록됩니다. 비동기 조작의 경우 애플리케이션 조작과 조작을 기록하는 추적 데이터 수신 간에 지연이 발생할 수 있습니다.

애플리케이션 활동 추적에 대한 구독 작성

특정 토픽에 대한 구독을 작성하여 애플리케이션 활동 추적 데이터를 수집할 수 있습니다.

특정 시스템 토픽 문자열에 대한 구독이 작성된 경우 해당하는 활동 추적 PCF 데이터 메시지가 해당 구독에 자동으로 발행됩니다. 토픽 구독에 대한 자세한 정보는 [발행/구독 메시징](#)을 참조하십시오.

토픽 문자열의 형식은 다음과 같습니다.

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifier
```

여기서,

- *qmgr_name*은 추적 대상 애플리케이션이 연결되어 있는 큐 관리자를 지정합니다. *qmgr_name* 은 후미 공백 문자가 모두 제거되고 슬래시 (/) 문자가 앰퍼샌드 (&) 로 대체된 큐 관리자의 이름입니다. 문자.
- *resource_type*은 수집되는 자원 데이터 유형을 지정하며 다음 문자열 중 하나입니다.
 - *AppName*. 애플리케이션을 지정합니다. 요청은 애플리케이션 이름이 *resource_identifier*로 지정된 이름과 일치하는 모든 IBM MQ 연결을 구독합니다.
 - *ChannelName*. IBM MQ 채널을 지정합니다.
 - *ConnectionId*. IBM MQ 연결을 지정합니다.
- *resource_identifier*는 실제 자원을 식별합니다. 형식은 자원 유형에 따라 달라집니다.
 - 자원 유형이 *AppName*인 경우 *resource_identifier*는 큐 관리자에게 표시되는 애플리케이션 이름의 후행 부분(마지막 / 또는 \ 다음에 나오는 값)이며 후행 공백 문자는 제거되어 있습니다. 값은 API 엑시트 컨텍스트 구조(MQAXC)의 *AppName* 값과 일치합니다. MQSC 명령 **DISPLAY CONN**를 사용하면 연결의 *AppName* 이 APPLTAG값으로 리턴됩니다.
 - *ChannelName* 자원 유형의 경우 *resource_identifier*는 추적되는 채널의 이름입니다. 채널 이름이 SVRCONN 채널을 식별하는 경우 연결된 클라이언트의 모든 애플리케이션 활동이 추적됩니다. 채널 이름이 큐 관리자 채널에 대한 큐 관리자를 식별하는 경우 수신 및 발신 메시지가 추적됩니다. *resource_identifier* 는 모든 후미 공백 문자가 제거되고 '/' 문자가 '&' 로 대체된 채널 이름입니다. 문자.
 - *ConnectionId* 자원 유형의 경우 *resource_identifier*는 각 연결에 지정된 고유한 연결 ID입니다. 토픽 문자열의 연결 ID는 16진 문자열로 기록된 전체 24바이트 값입니다. 이 값은 MQSC 명령 **DISPLAY CONN**에서 리턴된 CONN 값과 EXTCNN이 연결된 것입니다.

*resource_identifier*에서 와일드카드를 사용하여 단일 구독의 여러 자원 ID를 비교할 수 있습니다. 와일드카드는 기본 토픽 스타일('#' 또는 '+') 또는 문자 스타일('*' 또는 '?')일 수 있습니다. 토픽 스타일 와일드카드를 사용할 경우 자원 이름 일부와 결합할 수 없습니다. 가능한 모든 애플리케이션, 채널 또는 연결을 비교하는 데만 사용할 수 있습니다. 와일드카드를 사용하면 생성되는 추적 데이터 수준이 증가하여 성능에 영향을 미칠 수 있습니다.

이 토픽 문자열에 구독하려면 "subscribe" 권한이 있어야 합니다. 시스템 토픽은 큐 관리자 토픽 트리에서 권한을 상속하지 않습니다. 토픽 트리의 \$SYS/MQ 지점보다 더 깊은 관리되는 토픽 오브젝트에 대한 액세스 권한이 사용자에게 부여되어야 합니다. 모든 \$SYS/MQ 토픽 문자열에 대한 액세스 권한을 부여해도 활동 추적에는 부여

하지 않기 때문에 SYSTEM.ADMIN.TOPIC에 대한 액세스 권한이 있으면 구독할 수 있습니다. 더 자세히 액세스를 제어하려면 트리의 더 깊은 부분, 가령 모든 활동 추적이나 예를 들면 특정 애플리케이션 이름 또는 채널 이름에 대해 새로 관리되는 토픽 오브젝트를 정의할 수 있습니다.

예

다음 예제에서는 Windows 시스템에서 실행 중인 amqsput 애플리케이션에 대한 토픽 문자열을 보여줍니다.

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqsputc.exe
```

다음 예제에서는 채널에 대한 토픽 문자열을 보여줍니다.

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

다음 예제에서는 연결에 대한 토픽 문자열을 보여줍니다.

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ConnectionId/  
414D5143514D4752312020202020206B576B5420000701
```

다음 예제에서는 큐 관리자 QMGR1의 모든 채널에 대한 데이터를 추적하는 구독을 작성하는 토픽 문자열을 보여줍니다.

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

다음 예제에서는 "amqs"로 시작하는 이름을 가진 애플리케이션에 대한 데이터를 추적하는 구독을 작성하는 토픽 문자열을 보여줍니다. "*" 와일드카드를 사용하려면 와일드카드 문자 모델을 사용하여 구독을 작성해야 합니다.

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqs*
```

관련 개념

269 페이지의 『모니터링 및 활동 추적에 대한 시스템 토픽』

큐 관리자 토픽 트리의 시스템 토픽은 자원 모니터링 (일부는 통계 메시지의 콘텐츠와 유사함) 및 애플리케이션 활동 추적을 이용하는 방법으로 사용됩니다.

amqsact를 사용하여 추적 메시지 보기

amqsact 프로그램을 사용하여 추적 메시지를 생성하고 볼 수 있습니다.

amqsact 프로그램은 IBM MQ 샘플입니다. 이 샘플을 사용하려면 클라이언트 연결 실행 파일 amqsactc를 사용해야 합니다. 컴파일된 프로그램은 다음과 같은 샘플 디렉토리에 있습니다.

- Linux 및 UNIX 플랫폼의 경우, MQ_INSTALLATION_PATH/samp/bin64
- Windows 플랫폼의 경우, MQ_INSTALLATION_PATH\tools\c\Samples\Bin64

amqsact를 다음 두 가지 방식으로 구성할 수 있습니다.

표시 모드

SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에 전달되는 활동 추적 데이터 메시지를 포맷 및 표시합니다.

동적 모드

amqsact를 실행하여 생성된 활동 추적을 표시하고 자원 세트에 대한 구독을 작성합니다.

표시 모드

기본적으로 표시 모드에서 amqsact는 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE의 메시지를 처리합니다. 큐 이름이나 토픽 문자열을 지정하여 이 작동을 대체할 수 있습니다. 애플리케이션 활동 추적 정보 수집에서 설명한 방법 중 하나를 사용하여 활동 추적을 활성화해야 합니다. 표시된 추적 기간을 제어하고 활동 추적 메시지를 표시 후 제거하거나 보유할지 여부를 지정할 수도 있습니다. 표시 모드에서 amqsact에는 다음 인수가 있습니다.

-m queue_manager_name

필수입니다. 추적 메시지가 수집되는 대상 큐 관리자를 지정하십시오.

-q queue_name

지정된 큐와 관련된 추적 메시지만 표시합니다.

-t topic_string

지정된 큐와 관련된 추적 메시지만 표시합니다.

-b

표시 후 보관되는 추적 메시지를 지정합니다.

-v

상세 모드에서 추적 메시지를 표시하십시오.

-d depth

표시할 메시지 수입니다.

-w timeout

제한시간을 지정하십시오. 지정된 기간에 추적 메시지가 표시되지 않을 경우 **amqsact**가 종료됩니다.

-s start_time

-e 인수와 함께 이 인수를 사용하여 기간을 지정하십시오. 특정 시점으로부터 추적 메시지가 표시됩니다.

-e end_time

-s 인수와 함께 이 인수를 사용하여 기간을 지정하십시오. 특정 시점으로부터 추적 메시지가 표시됩니다.

예를 들어, 다음 명령에서 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에 보관 중인 활동 추적 메시지를 표시하고 표시 후 메시지를 삭제합니다.

```
amqsact -m QMGR1
```

다음 명령은 지정된 큐 SUB.QUEUE에 활동 추적 메시지를 표시하고, 표시한 후 메시지를 삭제합니다. 새 메시지 없이 30초가 경과할 때까지 메시지가 계속 표시됩니다. 이 명령을 사용하여 활동 추적 시스템 토픽 문자열에 구독을 발행합니다.

```
amqsact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

다음 명령은 현재 지정된 20분 기간에 발생한 SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE에 보관된 활동 추적 데이터를 상세하게 표시합니다. 메시지가 표시 후에 큐에 남아 있습니다.

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

동적 모드

애플리케이션 이름, 채널 이름 또는 연결 ID를 **amqsact**에 대한 인수로 지정해서 동적 모드를 사용 가능하게 합니다. 이름에 와일드카드 문자를 사용할 수 있음에 유의하십시오. 동적 모드에서는, 시스템 토픽의 지속 불가능한 구독을 사용하여 샘플 시작 시에 활동 추적 데이터가 사용 가능하게 됩니다. **amqsact**가 중지되면 활동 추적 데이터 수집이 중지됩니다. 동적 모드의 **amqsact**에 대한 제한시간을 지정해야 합니다. 각 인스턴스가 활동 추적 데이터의 한 사본을 수신하는 **amqsact**의 다중 사본을 동시에 실행할 수 있습니다. 표시 모드에서 **amqsact**에는 다음 인수가 있습니다.

-m queue_manager_name

필수입니다. 추적 메시지가 수집되는 대상 큐 관리자를 지정하십시오.

-w timeout

필수입니다. 제한시간을 지정하십시오. 지정된 기간에 추적 메시지가 표시되지 않을 경우 **amqsact**가 종료됩니다.

-a application_name

메시지를 수집할 애플리케이션을 지정하십시오.

-c channel_name

메시지를 수집할 채널을 지정하십시오.

-i connection_id

메시지를 수집할 연결을 지정하십시오.

-v

상세 모드에서 추적 메시지를 표시하십시오.

예를 들어 다음 명령은 "amqsget.exe"로 이름 지정된 애플리케이션이 수행한 연결에 대한 활동 추적 메시지를 생성하고 표시합니다. 무활동 30초가 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

다음 명령은 텍스트 "amqs"로 시작하는 애플리케이션이 수행한 연결에 대한 활동 추적 메시지를 생성하고 표시합니다. 무활동 30초가 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

다음 명령은 QMGR1.TO.QMGR2 채널의 활동에 대한 활동 추적 메시지를 생성하고 표시합니다. 무활동 10초가 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

다음 명령은 channels.TO.QMGR2 채널의 활동에 대한 활동 추적 메시지를 생성하고 표시합니다. 무활동 10초가 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 10 -c #
```

다음 명령은 CONN이 "6B576B5420000701"이고 EXTCONN이 "414D5143514D47523120202020202020"인 기존 IBM MQ 연결에 대한 활동의 상세 활동 추적 메시지를 생성하고 표시합니다. 무활동 1분이 지나면 **amqsact** 프로그램이 종료되고 새 활동 추적 데이터가 생성되지 않습니다.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

mqat.ini를 사용하여 추적 레벨 구성

mqat.ini 구성 파일의 AllActivityTrace 스탠자에 대한 값을 설정하여 큐 관리자에 대한 추적 레벨을 구성합니다.

AllActivityTrace 스탠자에 대한 다음 값을 설정할 수 있습니다.

ActivityInterval

메시지 추적 간 시간 간격(초 단위)입니다. 활동 추적은 타이머 스레드를 사용하지 않으므로 시간이 경과된 정확한 순간에 추적 메시지가 기록되지 않습니다. 대신 시간 간격이 경과된 후 첫 번째 MQI 조작이 실행될 때 기록됩니다. 값이 0이면 연결이 끊어질 때(또는 활동 수에 도달할 때) 추적 메시지가 기록됩니다. 기본적으로 1로 설정됩니다.

ActivityCount

추적 메시지 사이의 MQI 조작의 수입니다. 값이 0이면 연결이 끊어질 때(또는 활동 간격이 경과되었을 때) 추적 메시지가 기록됩니다. 기본값은 100입니다.

TraceLevel

각 조작에서 추적하는 매개변수 세부사항의 양입니다. 각 추적 레벨의 매개변수가 포함된 개별 조작 세부사항의 설명. LOW, MEDIUM 또는 HIGH로 설정합니다. 기본값은 MEDIUM입니다.

TraceMessageData

MQGET, MQPUT, MQPUT1 및 콜백 조작에 대해 추적되는 메시지 데이터의 양(바이트 단위) 기본값은 0입니다.

StopOnGetTraceMsg

ON 또는 OFF로 설정할 수 있습니다. 기본값은 ON입니다.

SubscriptionDelivery

BATCHED 또는 IMMEDIATE로 설정할 수 있습니다. 하나 이상의 활동 추적 구독이 있을 때 ActivityInterval 및 ActivityCount 매개변수를 사용하는지 여부를 결정합니다. 이 매개변수를 IMMEDIATE로 설정하면 추적 데이터에 일치하는 구독이 있을 경우 ActivityInterval 및 ActivityCount 값이 유효값 1로 대체됩니다. 각 활동 추적 레코드가 동일한 연결의 다른 레코드와 배치되지 않고, 대신 지체하지 않고 구독으로 즉시 전달됩니다. IMMEDIATE 설정은 활동 추적 데이터 수집의 성능 오버헤드를 증가시킵니다. 기본 설정은 BATCHED입니다.

애플리케이션 활동 추적 메시지 참조

이 페이지를 사용하여 애플리케이션 활동 추적 메시지의 형식과 이 메시지에 리턴되는 정보를 개괄합니다.

애플리케이션 활동 추적 메시지는 메시지 디스크립터 및 메시지 데이터를 포함한 표준 IBM MQ 메시지입니다. 메시지 데이터는 IBM MQ 애플리케이션이 수행한 MQI 조작에 대한 정보나 IBM MQ 시스템에서 발생하는 활동에 대한 정보를 포함합니다.

메시지 디스크립터

- MQMD 구조

메시지 데이터

- PCF 헤더(MQCFH)
- 항상 리턴되는 애플리케이션 활동 추적 메시지 데이터
- 조작별 애플리케이션 활동 추적 메시지 데이터

애플리케이션 활동 추적 메시지 MQMD(메시지 디스크립터)

이 페이지를 사용하여 애플리케이션 활동 추적 메시지의 메시지 디스크립터와 이벤트 메시지의 메시지 디스크립터 간 차이를 이해합니다.

애플리케이션 활동 추적 메시지의 메시지 디스크립터의 매개변수와 값은 다음을 제외하면, 이벤트 메시지의 메시지 디스크립터에서와 동일합니다.

Format

설명:	메시지 데이터의 형식 이름입니다.
값:	MQFMT_ADMIN 관리 메시지.

CorrelId

설명:	상관 ID.
값:	애플리케이션의 ConnectionId로 초기화됨

MQCFH(PCF 헤더)

이 페이지를 사용하여 활동 추적 메시지의 MQCFH 구조에 따라 포함된 PCF 값을 봅니다.

활동 추적 메시지의 경우 MQCFH 구조가 다음 값을 포함합니다.

Type

설명:	메시지의 콘텐츠를 식별하는 구조 유형입니다.
데이터 유형:	MLONG.
값:	MQCFT_APP_ACTIVITY

StrucLength

설명:	MQCFH 구조의 길이(바이트 단위).
데이터 유형:	MLONG.

값: MQCFH_STRUC_LENGTH

Version

설명: 구조 버전 번호입니다.

데이터 유형: MQLONG.

값: MQCFH_VERSION_3

Command

설명: 명령 ID입니다. 이 필드는 메시지의 범주를 식별합니다.

데이터 유형: MQLONG.

값: MQCMD_ACTIVITY_TRACE

MsgSeqNumber

설명: 메시지 순서 번호입니다. 이 필드는 관련된 메시지 그룹 내에 있는 메시지의 순서 번호입니다.

데이터 유형: MQLONG.

값: 1

Control

설명: 제어 옵션입니다.

데이터 유형: MQLONG.

값: MQCFC_LAST.

CompCode

설명: 완료 코드.

데이터 유형: MQLONG.

값: MQCC_OK.

Reason

설명: 완료 코드를 규정하는 이유 코드입니다.

데이터 유형: MQLONG.

값: MQRC_NONE.

ParameterCount

설명: 매개변수 구조 수입니다. 이 필드는 MQCFH 구조를 따르는 매개변수 구조의 수입니다. 그룹 구조(MQCFGR) 및 포함된 해당 매개변수 구조는 하나의 구조로만 계산됩니다.

데이터 유형: MQLONG.

값: 1 이상

애플리케이션 활동 추적 메시지 데이터

PCF 헤더 바로 다음에는 활동 추적의 시간 간격을 설명하는 매개변수 세트가 있습니다. 이 매개변수는 메시지가 기록될 때 메시지의 순서도 표시합니다. 나중에 추가 정보가 추가될 수 있도록 헤더 다음의 필드 수와 순서와 보장되지 않습니다.

메시지 이름: 활동 추적 메시지.
시스템 큐: SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

QueueManager

설명: 큐 관리자 이름.
ID: MQCA_Q_MGR_NAME
데이터 유형: MQCFST
최대 길이: MQ_Q_MGR_NAME_LENGTH

QSGName



설명: 큐 관리자가 멤버인 큐 공유 그룹의 이름(z/OS에만 해당)
ID: MQCA_QSG_NAME
데이터 유형: MQCFST
최대 길이: MQ_Q_MGR_NAME_LENGTH

HostName

설명: 큐 관리자가 실행 중인 시스템의 호스트 이름.
ID: MQCACF_HOST_NAME
데이터 유형: MQCFST

IntervalStartDate

설명: 모니터링 기간의 시작 날짜.
ID: MQCAMO_START_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH

IntervalStartTime

설명: 모니터링 기간의 시작 시간.
ID: MQCAMO_START_TIME
데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH

IntervalEndDate

설명: 모니터링 기간의 종료 날짜.
ID: MQCAMO_END_DATE
데이터 유형: MQCFST
최대 길이: MQ_DATE_LENGTH

IntervalEndTime

설명: 모니터링 기간의 종료 시간.
ID: MQCAMO_END_TIME

데이터 유형: MQCFST
최대 길이: MQ_TIME_LENGTH

CommandLevel

설명: IBM MQ 명령 레벨.
ID: MQIA_COMMAND_LEVEL
데이터 유형: MQCFIN

SeqNumber

설명: 일반적으로 순서 번호 0. 이 값은 장기 실행 연결에 대한 각 후속 레코드마다 증분됩니다.
ID: MQIACF_SEQUENCE_NUMBER
데이터 유형: MQCFIN

ApplicationName

설명: 애플리케이션의 이름(프로그램 이름).
ID: MQCACF_APPL_NAME
데이터 유형: MQCFST
최대 길이: MQ_APPL_NAME_LENGTH

ApplClass

설명: 활동을 수행한 애플리케이션의 유형. 가능한 값: MQAT_*
ID: MQIA_APPL_TYPE
데이터 유형: MQCFIN

ApplicationPid

설명: 애플리케이션의 운영 체제 프로세스 ID.
ID: MQIACF_PROCESS_ID
데이터 유형: MQCFIN

UserId

설명: 애플리케이션의 사용자 ID 컨텍스트.
ID: MQCACF_USER_IDENTIFIER
데이터 유형: MQCFST
최대 길이: MQ_USER_ID_LENGTH

APICallerType

설명: 애플리케이션의 유형. 가능한 값: MQXACT_EXTERNAL 또는 MQXACT_INTERNAL.
ID: MQIACF_API_CALLER_TYPE
데이터 유형: MQCFIN

Environment

설명: 애플리케이션의 런타임 환경. 가능한 값: MQXE_*

ID: MQIACF_API_ENVIRONMENT
데이터 유형: MQCFIN

ChannelName

설명: 연결과 연관된 채널 이름. 이 매개변수는 환경 매개변수의 값이 MQXE_MCA 또는 MQXE_MCA_SVRCONN인 경우에만 리턴됩니다.

ID: MQCACH_CHANNEL_NAME
데이터 유형: MQCFST
최대 길이: MQ_CHANNEL_NAME_LENGTH

ConnectionName

설명: 연결과 연관된 네트워크 연결 이름. 이 매개변수는 환경 매개변수의 값이 MQXE_MCA 또는 MQXE_MCA_SVRCONN인 경우에만 리턴됩니다.

ID: MQCACH_CONNECTION_NAME
데이터 유형: MQCFST
최대 길이: MQ_CONN_NAME_LENGTH

ChannelType

설명: 연결과 연관된 채널의 유형. 이 매개변수는 환경 매개변수의 값이 MQXE_MCA 또는 MQXE_MCA_SVRCONN인 경우에만 리턴됩니다. 가능한 값: MQCHT_*

ID: MQIACH_CHANNEL_TYPE
데이터 유형: MQCFIN

RemoteProduct

설명: 연결과 연관된 원격 제품 ID. 이 매개변수는 환경 매개변수의 값이 MQXE_MCA 또는 MQXE_MCA_SVRCONN인 경우에만 리턴됩니다.

ID: MQCACH_REMOTE_PRODUCT
데이터 유형: MQCFST
최대 길이: MQ_REMOTE_PRODUCT_LENGTH

RemoteVersion

설명: 연결과 연관된 원격 제품 버전. 이 매개변수는 환경 매개변수의 값이 MQXE_MCA 또는 MQXE_MCA_SVRCONN인 경우에만 리턴됩니다.

ID: MQCACH_REMOTE_VERSION
데이터 유형: MQCFST
최대 길이: MQ_REMOTE_VERSION_LENGTH

FunctionName

설명: 초기 스레드에서 시작된 마지막 상위 레벨 함수의 이름입니다.

ID: MQCACF_APPL_FUNCTION
데이터 유형: MQCFST

FunctionType

설명:	초기 스크린에서 시작된 마지막 상위 레벨 함수의 유형입니다. 가능한 값: MQFUN_*
ID:	MQIACF_APPL_FUNCTION_TYPE
데이터 유형:	MQCFIN

Detail

설명:	연결에 대해 기록할 세부사항 레벨. 가능한 값: 1=LOW 2=MEDIUM 3=HIGH
ID:	MQIACF_TRACE_DETAIL
데이터 유형:	MQCFIN

TraceDataLength

설명:	이 연결에 대해 추적하는 메시지 데이터의 길이(바이트).
ID:	MQIACF_TRACE_DATA_LENGTH
데이터 유형:	MQCFIN

PointerSize

설명:	애플리케이션이 실행 중인 플랫폼의 포인터(2진 구조의 해석을 지원) 길이(바이트).
ID:	MQIACF_POINTER_SIZE
데이터 유형:	MQCFIN

Platform

설명:	큐 관리자가 실행 중인 플랫폼. 가능한 값: MQPL_*
ID:	MQIA_PLATFORM
데이터 유형:	MQCFIN

애플리케이션 활동 MQI 조작에 대한 변수 매개변수

애플리케이션 활동 데이터 MQCFGR 구조 다음에는 수행 중인 조작에 해당하는 PCF 매개변수 세트가 이어집니다. 각 조작에 대한 매개변수는 다음 절에 정의되어 있습니다.

추적 레벨은 추적이 포함시킬 매개변수에 필요한 추적 세분화 레벨을 표시합니다. 가능한 추적 레벨 값은 다음과 같습니다.

1. 낮음

애플리케이션에 "낮음", "중간" 또는 "높음" 활동 추적이 구성될 때 매개변수가 포함됩니다. 이 설정은 매개변수가 항상 조작에 대한 AppActivityData 그룹에 포함됨을 의미합니다. 애플리케이션이 수행하는 MQI 호출을 추적하고 호출이 성공했는지 여부를 확인하는 데에는 이 매개변수 세트면 충분합니다.

2. 중간

애플리케이션에 대해 "중간" 또는 "높음" 활동 추적이 구성되어 있을 때에만 조작의 AppActivityData 그룹에 매개변수가 포함됩니다. 이 매개변수 세트는 애플리케이션에 사용되는 큐와 토크 이름 같은 자원에 대한 정보를 추가합니다.

3. 높음

애플리케이션에 대해 "높음" 활동 추적이 구성되어 있을 때에만 조작의 AppActivityData 그룹에 매개변수가 포함됩니다. 이 매개변수 세트는 MQI 및 XA 함수에 전달되는 구조의 메모리 덤프를 포함합니다. 이러한 이유로, MQI 및 XA 호출에 사용되는 매개변수에 대한 자세한 정보를 포함합니다. 구조 메모리 덤프는 구조의 셸로우(shallow) 사본입니다. 참조 해제 포인터에 대한 잘못된 시도를 피하기 위해 구조의 포인터 값은 NULL로 설정됩니다.

참고: 덤프된 구조 버전이 애플리케이션에 사용된 버전과 반드시 동일하지는 않습니다. API 교차 엑시트, 활동 추적 코드 또는 큐 관리자를 통해 구조를 수정할 수 있습니다. 큐 관리자는 이후 버전에 구조를 수정할 수 있지만 구조의 이전 버전에는 결코 구조를 변경하지 않습니다. 이전 버전에 대해 변경할 경우 데이터 손실 위험이 있습니다.

MQBACK

애플리케이션이 MQBACK MQI 함수를 시작함

CompCode

설명:	조작 결과를 나타내는 완료 코드
PCF 매개변수:	MQIACF_COMP_CODE
추적 레벨:	1
유형	MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형	MQCFIN

QMgrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형	MQCFIN64

MQBEGIN

애플리케이션이 MQBEGIN MQI 함수를 시작함

CompCode

설명:	조작 결과를 나타내는 완료 코드
PCF 매개변수:	MQIACF_COMP_CODE
추적 레벨:	1
유형	MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형	MQCFIN

MQBO

설명:	MQBEGIN 옵션 구조. NULL 포인터가 MQBEGIN 호출에 사용되면 이 매개변수가 포함되지 않습니다.
PCF 매개변수:	MQBACF_MQBO_STRUCT
추적 레벨:	3
유형	MQCFBS
길이:	MQBO 구조의 길이(바이트).

QMGrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형	MQCFIN64

MQCALLBACK

애플리케이션이 MQCALLBACK MQI 함수를 시작함

ObjectHandle

설명:	오브젝트 핸들
PCF 매개변수:	MQIACF_HOBJ
추적 레벨:	1
유형	MQCFIN

CallType

설명:	함수가 호출된 이유. MQCBCT_* 값 중 하나
PCF 매개변수:	MQIACF_CALL_TYPE
추적 레벨:	1
유형	MQCFIN

MsgBuffer

설명:	메시지 데이터.
PCF 매개변수:	MQBACF_MESSAGE_DATA
추적 레벨:	1
유형	MQCFBS
길이:	길이는 APPTRACE 구성에 설정된 TRACEDATA() 매개변수로 제어됩니다. TRACEDATA=NONE이면 이 매개변수가 생략됩니다.

MsgLength

설명:	메시지의 길이 (MQCBC 구조의 DataLength 필드에서 가져옴).
PCF 매개변수:	MQIACF_MSG_LENGTH

추적 레벨: 1
유형 MQCFIN

HighResTime

설명: 1970년 1월 1일(UTC), 자정 이후의 조작 시간(마이크로초)
참고: 이 타이머의 정확도는 고해상도 타이머에 대한 플랫폼 지원에 따라 다릅니다.

PCF 매개변수: MQIAMO64_HIGHRES_TIME
추적 레벨: 2
유형 MQCFIN64

ReportOptions

설명: 보고 메시지의 옵션
PCF 매개변수: MQIACF_REPORT
추적 레벨: 2
유형 MQCFIN

MsgType

설명: 메시지의 유형
PCF 매개변수: MQIACF_MSG_TYPE
추적 레벨: 2
유형 MQCFIN

Expiry

설명: 메시지 수명
PCF 매개변수: MQIACF_EXPIRY
추적 레벨: 2
유형 MQCFIN

Format

설명: 메시지 데이터의 형식 이름
PCF 매개변수: MQCACH_FORMAT_NAME
추적 레벨: 2
유형 MQCFST
길이: MQ_FORMAT_LENGTH

Priority

설명: 메시지 우선순위
PCF 매개변수: MQIACF_PRIORITY
추적 레벨: 2
유형 MQCFIN

Persistence

설명:	메시지 지속성
PCF 매개변수:	MQIACF_PERSISTENCE
추적 레벨:	2
유형	MQCFIN

MsgId

설명:	메시지 ID
PCF 매개변수:	MQBACF_MSG_ID
추적 레벨:	2
유형	MQCFBS
길이:	MQ_MSG_ID_LENGTH

CorrelId

설명:	상관 ID
PCF 매개변수:	MQBACF_CORREL_ID
추적 레벨:	2
유형	MQCFBS
길이:	MQ_CORREL_ID_LENGTH

ObjectName

설명:	열린 오브젝트의 이름.
PCF 매개변수:	MQCACF_OBJECT_NAME
추적 레벨:	2
유형	MQCFST
길이:	MQ_Q_NAME_LENGTH

ResolvedQName

설명:	메시지를 검색한 큐의 로컬 이름.
PCF 매개변수:	MQCACF_RESOLVED_Q_NAME
추적 레벨:	2
유형	MQCFST
길이:	MQ_Q_NAME_LENGTH

ReplyToQueue

설명:	MQ_Q_NAME_LENGTH
PCF 매개변수:	MQCACF_REPLY_TO_Q
추적 레벨:	2
유형	MQCFST

ReplyToQMgr

설명:	MQ_Q_MGR_NAME_LENGTH
PCF 매개변수:	MQCACF_REPLY_TO_Q_MGR
추적 레벨:	2
유형	MQCFST

CodedCharSetId

설명:	메시지 데이터의 문자 세트 ID
PCF 매개변수:	MQIA_CODED_CHAR_SET_ID
추적 레벨:	2
유형	MQCFIN

Encoding

설명:	메시지 데이터의 숫자 인코딩입니다.
PCF 매개변수:	MQIACF_ENCODING
추적 레벨:	2
유형	MQCFIN

PutDate

설명:	MQ_PUT_DATE_LENGTH
PCF 매개변수:	MQCACF_PUT_DATE
추적 레벨:	2
유형	MQCFST

PutTime

설명:	MQ_PUT_TIME_LENGTH
PCF 매개변수:	MQCACF_PUT_TIME
추적 레벨:	2
유형	MQCFST

ResolvedQName

설명:	ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
PCF 매개변수:	MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨:	2
유형	MQCFST
길이:	MQ_Q_NAME_LENGTH.

ResObjectString

설명:	ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.
PCF 매개변수:	MQCACF_RESOLVED_OBJECT_STRING
추적 레벨:	2

유형 MQCFST
길이: 길이가 다양합니다.

ResolvedType

설명: ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수: MQIACF_RESOLVED_TYPE
추적 레벨: 2
유형 MQCFIN

PolicyName

설명: 이 메시지에 적용된 정책 이름.
참고: AMS 보호 메시지만 해당함
PCF 매개변수: MQCA_POLICY_NAME
추적 레벨: 2
유형 MQCFST
길이: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

설명: 전송 큐 헤더에 있는 메시지의 메시지 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함
PCF 매개변수: MQBACF_XQH_MSG_ID
추적 레벨: 2
유형 MQCFBS
길이: MQ_MSG_ID_LENGTH

XmitqCorrelId

설명: 전송 큐 헤더에 있는 메시지의 상관 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함
PCF 매개변수: MQBACF_XQH_CORREL_ID
추적 레벨: 2
유형 MQCFBS
길이: MQ_CORREL_ID_LENGTH

XmitqPutTime

설명: 전송 큐 헤더에 있는 메시지의 넣기 시간.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함
PCF 매개변수: MQCACF_XQH_PUT_TIME
추적 레벨: 2
유형 MQCFST

길이: MQ_PUT_TIME_LENGTH

XmitqPutDate

설명: 전송 큐 헤더에 있는 메시지의 넣기 날짜.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_DATE
추적 레벨: 2
유형 MQCFST
길이: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_Name
추적 레벨: 2
유형 MQCFST
길이: MQ_Q_NAME_LENGTH

XmitqRemoteQMGr

설명: 전송 큐 헤더에 있는 메시지의 메시지 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_MGR
추적 레벨: 2
유형 MQCFST
길이: MQ_MSG_ID_LENGTH

MsgDescStructure

설명: MQMD 구조. 버전 4 MQGMO를 사용하여 MQMD 대신 메시지 핸들을 리턴하도록 요청한 경우에는 이 매개변수가 생략됩니다.

PCF 매개변수: MQBACF_MQMD_STRUCT
추적 레벨: 3
유형 MQCFBS
길이: MQMD 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

GetMsgOptsStructure

설명: MQGMO 구조.

PCF 매개변수: MQBACF_MQGMO_STRUCT
추적 레벨: 3
유형 MQCFBS
길이: MQGMO 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

MQCBCContextStructure

설명:	MQCBC 구조.
PCF 매개변수:	MQBACF_MQCBC_STRUCT
추적 레벨:	3
유형	MQCFBS
길이:	MQCBC 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMGrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형	MQCFIN64

MQCB

애플리케이션이 콜백 MQI 함수 관리를 시작함

CallbackOperation

설명:	콜백 함수 관리 조작. MQOP_* 값 중 하나로 설정됩니다.
PCF 매개변수:	MQIACF_MQCB_OPERATION
추적 레벨:	1
유형	MQCFIN

CallbackType

설명:	콜백 함수의 유형(MQCBD 구조의 CallbackType 필드). MQCBT_* 값 중 하나로 설정됩니다.
PCF 매개변수:	MQIACF_MQCB_TYPE
추적 레벨:	1
유형	MQCFIN

CallbackOptions

설명:	콜백 옵션. MQCBDO_* 값 중 하나로 설정됩니다.
PCF 매개변수:	MQIACF_MQCB_OPTIONS
추적 레벨:	1
유형	MQCFIN

CallbackFunction

설명:	함수 호출로 시작된 경우 콜백 함수에 대한 포인터.
PCF 매개변수:	MQBACF_MQCB_FUNCTION
추적 레벨:	1
유형	MQCFBS

길이: MQPTR의 크기

CallbackName

설명: 동적 링크된 프로그램으로 시작된 경우 콜백 함수의 이름.
PCF 매개변수: MQCACF_MQCB_NAME
추적 레벨: 1
유형: MQCFST
길이: MQCHAR128의 크기

ObjectHandle

설명: 오브젝트 핸들
PCF 매개변수: MQIACF_HOBJ
추적 레벨: 1
유형: MQCFIN

MaxMsgLength

설명: 최대 메시지 길이. 정수 또는 특수 값 MQCBD_FULL_MSG_LENGTH로 설정됩니다.
PCF 매개변수: MQIACH_MAX_MSG_LENGTH
추적 레벨: 2
유형: MQCFIN

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 1
유형: MQCFIN

ResolvedQName

설명: ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH.

ResObjectString

설명: ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.

PCF 매개변수:	MQCACF_RESOLVED_OBJECT_STRING
추적 레벨:	2
유형	MQCFST
길이:	길이가 다양합니다.

ResolvedType

설명:	ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형	MQCFIN

Callback DescriptorStructure

설명:	MQCBD 구조. NULL MQCBC 값이 MQCB 호출에 전달된 경우 이 매개변수는 생략됩니다.
PCF 매개변수:	MQBACF_MQCBD_STRUCT
추적 레벨:	3
유형	MQCFBS
길이:	MQCBC 구조의 길이(바이트)

MsgDescStructure

설명:	MQMD 구조. NULL MQMD 값이 MQCB 호출에 전달된 경우 MsgDescStructure 매개변수는 생략됩니다.
PCF 매개변수:	MQBACF_MQMD_STRUCT
추적 레벨:	3
유형	MQCFBS
길이:	MQMD 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

GetMsgOptsStructure

설명:	MQGMO 구조. NULL MQGMO 값이 MQCB 호출에 전달된 경우 이 매개변수는 생략됩니다.
PCF 매개변수:	MQBACF_MQGMO_STRUCT
추적 레벨:	3
유형	MQCFBS
길이:	MQGMO 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMgrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION

추적 레벨: 2
유형 MQCFIN64

MQCLOSE
애플리케이션이 MQCLOSE MQI 함수를 시작함

ObjectHandle

설명: 오브젝트 핸들
PCF 매개변수: MQIACF_HOBJ
추적 레벨: 1
유형 MQCFIN

CloseOptions

설명: 닫기 옵션
PCF 매개변수: MQIACF_CLOSE_OPTIONS
추적 레벨: 1
유형 MQCFIN

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형 MQCFIN

Reason

설명: 조작의 이유 코드 결과
PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 1
유형 MQCFIN

ResolvedQName

설명: ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨: 2
유형 MQCFST
길이: MQ_Q_NAME_LENGTH.

ResObjectString

설명: ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.
PCF 매개변수: MQCACF_RESOLVED_OBJECT_STRING
추적 레벨: 2
유형 MQCFST
길이: 길이가 다양합니다.

ResolvedType

설명:	ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형	MQCFIN

QMgrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형	MQCFIN64

MQCMIT

애플리케이션이 MQCMIT MQI 함수를 시작함

CompCode

설명:	조작 결과를 나타내는 완료 코드
PCF 매개변수:	MQIACF_COMP_CODE
추적 레벨:	1
유형	MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형	MQCFIN

QMgrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형	MQCFIN64

MQCONN 및 MQCONNX

애플리케이션이 MQCONN 또는 MQCONNX MQI 함수를 시작함

ConnectionId

설명:	사용 가능한 경우 연결 ID 또는 그렇지 않으면 MQCONNID_NONE
PCF 매개변수:	MQBACF_CONNECTION_ID
추적 레벨:	1
유형:	MQCFBS
최대 길이:	MQ_CONNECTION_ID_LENGTH

QueueManagerName

설명:	MQCONN(X) 호출에 사용된 큐 관리자의 (분석되지 않은) 이름
PCF 매개변수:	MQCA_Q_MGR_NAME
추적 레벨:	1
유형:	MQCFST
최대 길이:	MQ_Q_MGR_NAME_LENGTH

CompCode

설명:	조작 결과를 나타내는 완료 코드
PCF 매개변수:	MQIACF_COMP_CODE
추적 레벨:	1
유형:	MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형:	MQCFIN

ConnectOptions

설명:	MQCNO_* 값에서 파생된 연결 옵션 참고: MQCONNX만 해당함
PCF 매개변수:	MQIACF_CONNECT_OPTIONS
추적 레벨:	2
유형:	MQCFIN

ConnectionOptionsStructure

설명:	MQCNO 구조. 참고: (MQCONNX만 해당함)
PCF 매개변수:	MQBACF_MQCNO_STRUCT
추적 레벨:	3
유형:	MQCFBS
최대 길이:	MQCNO 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

ChannelDefinitionStructure

설명:	MQCD 구조. 참고: 클라이언트 연결만 해당함
PCF 매개변수:	MQBACF_MQCD_STRUCT
추적 레벨:	3
유형:	MQCFBS
최대 길이:	MQCD 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMGrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형:	MQCFIN64

MQCTL

애플리케이션이 MQCTL MQI 함수를 시작함

CompCode

설명:	조작 결과를 나타내는 완료 코드
PCF 매개변수:	MQIACF_COMP_CODE
추적 레벨:	1
유형:	MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형:	MQCFIN

CtlOperation

설명:	MQOP_* 값 중 하나
PCF 매개변수:	MQIACF_CTL_OPERATION
추적 레벨:	1
유형:	MQCFIN

QMgrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
추적 레벨: 2
유형: MQCFIN64

MQDISC

애플리케이션이 MQDISC MQI 함수를 시작함

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 1
유형: MQCFIN

MQGET

애플리케이션이 MQGET MQI 함수를 시작함

ObjectHandle

설명: 오브젝트 핸들
PCF 매개변수: MQIACF_HOBJ
추적 레벨: 1
유형: MQCFIN

GetOptions

설명: MQGMO.Options의 가져오기 옵션
PCF 매개변수: MQIACF_GET_OPTIONS
추적 레벨: 1
유형: MQCFIN

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형: MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형:	MQCFIN

MsgBuffer

설명:	메시지 데이터. TRACEDATA=NONE이면 이 매개변수가 생략됩니다.
PCF 매개변수:	MQBACF_MESSAGE_DATA
추적 레벨:	1
유형:	MQCFBS
최대 길이:	길이는 APPTRACE 구성에 설정된 TRACEDATA() 매개변수로 제어됩니다. (추적 메시지에 MQIACF_TRACE_DATA_LENGTH로 포함됨).

MsgLength

설명:	메시지의 길이
PCF 매개변수:	MQIACF_MSG_LENGTH
추적 레벨:	1
유형:	MQCFIN

HighResTime

설명:	1970년 1월 1일(UTC), 자정 이후의 조작 시간(마이크로초) 참고: 이 타이머의 정확도는 고해상도 타이머에 대한 플랫폼 지원에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_HIGHRES_TIME
추적 레벨:	2
유형:	MQCFIN64

BufferLength

설명:	애플리케이션이 제공하는 버퍼의 길이
PCF 매개변수:	MQIACF_BUFFER_LENGTH
추적 레벨:	2
유형:	MQCFIN

ObjectName

설명:	열린 오브젝트의 이름
PCF 매개변수:	MQCACF_OBJECT_NAME
추적 레벨:	2
유형:	MQCFST
길이:	MQ_Q_NAME_LENGTH

ResolvedQName

설명:	메시지를 검색한 큐의 로컬 이름.
PCF 매개변수:	MQCACF_RESOLVED_Q_NAME
추적 레벨:	2
유형:	MQCFST
최대 길이:	MQ_Q_NAME_LENGTH

ReportOptions

설명:	메시지 보고 옵션
PCF 매개변수:	MQIACF_REPORT
추적 레벨:	2
유형:	MQCFIN

MsgType

설명:	메시지의 유형
PCF 매개변수:	MQIACF_MSG_TYPE
추적 레벨:	2
유형:	MQCFIN

Expiry

설명:	메시지 수명
PCF 매개변수:	MQIACF_EXPIRY
추적 레벨:	2
유형:	MQCFIN

Format

설명:	메시지 데이터의 형식 이름
PCF 매개변수:	MQCACH_FORMAT_NAME
추적 레벨:	2
유형:	MQCFST
최대 길이:	MQ_FORMAT_LENGTH

Priority

설명:	메시지 우선순위
PCF 매개변수:	MQIACF_PRIORITY
추적 레벨:	2
유형:	MQCFIN

Persistence

설명:	메시지 지속성
PCF 매개변수:	MQIACF_PERSISTENCE

추적 레벨: 2
유형: MQCFIN

MsgId

설명: 메시지 ID
PCF 매개변수: MQBACF_MSG_ID
추적 레벨: 2
유형: MQCFBS
최대 길이: MQ_MSG_ID_LENGTH

CorrelId

설명: 상관 ID
PCF 매개변수: MQBACF_CORREL_ID
추적 레벨: 2
유형: MQCFBS
최대 길이: MQ_CORREL_ID_LENGTH

ReplyToQueue

설명:
PCF 매개변수: MQCACF_REPLY_TO_Q
추적 레벨: 2
유형: MQCFST
최대 길이: MQ_Q_NAME_LENGTH

ReplyToQMgr

설명:
PCF 매개변수: MQCACF_REPLY_TO_Q_MGR
추적 레벨: 2
유형: MQCFST
최대 길이: MQ_Q_MGR_NAME_LENGTH

CodedCharSetId

설명: 메시지 데이터의 문자 세트 ID
PCF 매개변수: MQIA_CODED_CHAR_SET_ID
추적 레벨: 2
유형: MQCFIN

Encoding

설명: 메시지 데이터의 숫자 인코딩입니다.
PCF 매개변수: MQIACF_ENCODING
추적 레벨: 2

유형: MQCFIN

PutDate

설명:
PCF 매개변수: MQCACF_PUT_DATE
추적 레벨: 2
유형: MQCFST
최대 길이: MQ_PUT_DATE_LENGTH

PutTime

설명:
PCF 매개변수: MQCACF_PUT_TIME
추적 레벨: 2
유형: MQCFST
최대 길이: MQ_PUT_TIME_LENGTH

ResolvedQName

설명: ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH.

ResObjectString

설명: ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.
PCF 매개변수: MQCACF_RESOLVED_OBJECT_STRING
추적 레벨: 2
유형: MQCFST
길이: 길이가 다양합니다.

ResolvedType

설명: ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수: MQIACF_RESOLVED_TYPE
추적 레벨: 2
유형: MQCFIN

PolicyName

설명: 이 메시지에 적용된 정책 이름.
참고: AMS 보호 메시지만 해당함
PCF 매개변수: MQCA_POLICY_NAME

추적 레벨: 2
유형: MQCFST
길이: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

설명: 전송 큐 헤더에 있는 메시지의 메시지 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQBACF_XQH_MSG_ID
추적 레벨: 2
유형: MQCFBS
길이: MQ_MSG_ID_LENGTH

XmitqCorrelId

설명: 전송 큐 헤더에 있는 메시지의 상관 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQBACF_XQH_CORREL_ID
추적 레벨: 2
유형: MQCFBS
길이: MQ_CORREL_ID_LENGTH

XmitqPutTime

설명: 전송 큐 헤더에 있는 메시지의 넣기 시간.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_TIME
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_TIME_LENGTH

XmitqPutDate

설명: 전송 큐 헤더에 있는 메시지의 넣기 날짜.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_DATE
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_NAME

추적 레벨: 2
 유형: MQCFST
 길이: MQ_Q_NAME_LENGTH

XmitqRemoteQMGr

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 관리자 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_MGR
 추적 레벨: 2
 유형: MQCFST
 길이: MQ_Q_NAME_LENGTH

MsgDescStructure

설명: MQMD 구조.
 PCF 매개변수: MQBACF_MQMD_STRUCT
 추적 레벨: 3
 유형: MQCFBS
 최대 길이: MQMD 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

GetMsgOptsStructure

설명: MQGMO 구조.
 PCF 매개변수: MQBACF_MQGMO_STRUCT
 추적 레벨: 3
 유형: MQCFBS
 최대 길이: MQGMO 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMGrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
 추적 레벨: 2
 유형: MQCFIN64

MQINQ

애플리케이션이 MQINQ MQI 함수를 시작함

ObjectHandle

설명: 오브젝트 핸들
 PCF 매개변수: MQIACF_HOBJ
 추적 레벨: 1

유형: MQCFIN

CompCode

설명: 조작 결과를 나타내는 완료 코드

PCF 매개변수: MQIACF_COMP_CODE

추적 레벨: 1

유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과

PCF 매개변수: MQIACF_REASON_CODE

추적 레벨: 1

유형: MQCFIN

SelectorCount

설명: Selectors 배열에 제공되는 선택자 수.

PCF 매개변수: MQIACF_SELECTOR_COUNT

추적 레벨: 2

유형: MQCFIN

Selectors

설명: MQINQ가 값을 리턴해야 하는 속성의 목록(정수 또는 문자).

PCF 매개변수: MQIACF_SELECTORS

추적 레벨: 2

유형: MQCFIL

ResolvedQName

설명: ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.

PCF 매개변수: MQCACF_RESOLVED_Q_NAME

추적 레벨: 2

유형: MQCFST

최대 길이: MQ_Q_NAME_LENGTH

ResObjectString

설명: ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.

PCF 매개변수: MQCACF_RESOLVED_OBJECT_STRING

추적 레벨: 2

유형: MQCFST

최대 길이: 길이가 다양합니다.

ResolvedType

설명:	ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형:	MQCFIN

IntAttrCount

설명:	조회 조작이 리턴하는 정수 속성의 수
PCF 매개변수:	MQIACF_INTATTR_COUNT
추적 레벨:	3
유형:	MQCFIN

IntAttrs

설명:	조회 조작이 리턴하는 정수 속성 값. 이 매개변수는 MQINQ가 리턴할 때 IntAttrCount > 0인 경우에만 존재합니다.
PCF 매개변수:	MQIACF_INT_ATTRS
추적 레벨:	3
유형:	MQCFIL

CharAttrs

설명:	조회 조작이 리턴하는 문자 속성 값이 서로 연결됩니다. 이 매개변수는 MQINQ가 리턴할 때 CharAttrLength > 0인 경우에만 포함됩니다.
PCF 매개변수:	MQCACF_CHAR_ATTRS
추적 레벨:	3
유형:	MQCFST

QMgrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2
유형:	MQCFIN64

MQOPEN

애플리케이션이 MQOPEN MQI 함수를 시작함

ObjectType

설명:	MQOT.ObjectType에 전달된 오브젝트 유형
PCF 매개변수:	MQIACF_OBJECT_TYPE
추적 레벨:	1

유형: MQCFIN

ObjectName

설명: 큐 이름 분석을 시도하기 전에 MQI 호출에 전달된 오브젝트의 이름.

PCF 매개변수: MQCACF_OBJECT_NAME

추적 레벨: 1

유형: MQCFST

최대 길이: MQ_Q_NAME_LENGTH

ObjectQMgrName

설명: 큐 이름 분석을 시도하기 전에 MQI 호출에 전달된 오브젝트 큐 관리자의 이름.

PCF 매개변수: MQCACF_OBJECT_Q_MGR_NAME

추적 레벨: 1

유형: MQCFST

최대 길이: MQ_Q_MGR_NAME_LENGTH

ObjectHandle

설명: 오브젝트 핸들

PCF 매개변수: MQIACF_HOBJ

추적 레벨: 1

유형: MQCFIN

CompCode

설명: 조작 결과를 나타내는 완료 코드

PCF 매개변수: MQIACF_COMP_CODE

추적 레벨: 1

유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과

PCF 매개변수: MQIACF_REASON_CODE

추적 레벨: 1

유형: MQCFIN

OpenOptions

설명: 오브젝트를 여는 데 사용하는 옵션

PCF 매개변수: MQIACF_OPEN_OPTIONS

추적 레벨: 1

유형: MQCFIN

AlternateUserId

설명: MQOO_ALTERNATE_USER_AUTHORITY가 지정된 경우에만 포함됩니다.

PCF 매개변수: MQCACF_ALTERNATE_USERID
추적 레벨: 2
유형: MQCFST
최대 길이: MQ_USER_ID_LENGTH

RecsPresent

설명: 오브젝트 이름 레코드의 수가 표시됩니다. MQOD 버전 >= MQOD_VERSION_2인 경우에만 포함됩니다.
PCF 매개변수: MQIACF_RECS_PRESENT
추적 레벨: 1
유형: MQCFIN

KnownDestCount

설명: 열기에 성공한 로컬 큐 수. MQOD 버전 >= MQOD_VERSION_2인 경우에만 포함됩니다.
PCF 매개변수: MQIACF_KNOWN_DEST_COUNT
추적 레벨: 1
유형: MQCFIN

UnknownDestCount

설명: 열기에 성공한 리모트 큐 수. MQOD 버전 >= MQOD_VERSION_2인 경우에만 포함됩니다.
PCF 매개변수: MQIACF_UNKNOWN_DEST_COUNT
추적 레벨: 1
유형: MQCFIN

InvalidDestCount

설명: 열기에 실패한 큐 수. MQOD 버전 >= MQOD_VERSION_2인 경우에만 포함됩니다.
PCF 매개변수: MQIACF_INVALID_DEST_COUNT
추적 레벨: 1
유형: MQCFIN

DynamicQName

설명: MQOPEN 호출에 입력으로 전달된 동적 큐 이름.
PCF 매개변수: MQCACF_DYNAMIC_Q_NAME
추적 레벨: 2
유형: MQCFST
최대 길이: MQ_Q_NAME_LENGTH

ResolvedLocalQName ^{1 2}

설명: 이름 분석이 수행된 후의 로컬 큐 이름을 포함합니다. (예를 들어, 리모트 큐의 경우 전송 큐의 이름입니다.)
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME

추적 레벨: 2
 유형: MQCFST
 범위: MQOD.Version이 MQOD_VERSION_3 미만이면 MQOPEN 호출이 완료된 후의 MQOD.ObjectName 필드 값을 포함합니다. MQOD.Version이 MQOD_VERSION_3 이상이면 MQOD의 값이 포함됩니다. ResolvedQName 필드,
 최대 길이: MQ_Q_NAME_LENGTH

ResolvedLocalQMgrName ^{1 2}

설명: 이름 분석이 수행된 후의 로컬 큐 관리자 이름.
 PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_MGR
 추적 레벨: 2
 유형: MQCFST
 범위: MQOD 버전 >= MQOD_VERSION_3인 경우에만 해당됩니다.
 최대 길이: MQ_Q_MGR_NAME_LENGTH

ResolvedQName ^{1 2}

설명: 이름 분석이 수행된 후의 큐 이름.
 PCF 매개변수: MQCACF_RESOLVED_Q_NAME
 추적 레벨: 2
 유형: MQCFST
 범위: MQOD.Version이 MQOD_VERSION_3 미만이면 MQOPEN 호출이 완료된 후의 MQOD.ObjectName 필드 값을 포함합니다. MQOD.Version이 MQOD_VERSION_3 이상이면 MQOD의 값이 포함됩니다. ResolvedQName 필드,
 최대 길이: MQ_Q_NAME_LENGTH

ResolvedQMgrName ^{1 2}

설명: 이름 분석이 수행된 후의 큐 관리자 이름을 포함합니다. MQOD.Version이 MQOD_VERSION_3 미만이면 MQOD의 값이 포함됩니다. MQOPEN 호출 다음의 ObjectQMgrName 필드가 완료되었습니다. MQOD.Version이 MQOD_VERSION_3 이상이면 MQOD의 값이 포함됩니다. ResolvedQMgrName 필드
 PCF 매개변수: MQCACF_RESOLVED_Q_MGR
 추적 레벨: 2
 유형: MQCFST
 최대 길이: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

설명: 대체 보안 ID. MQOD.Version이 MQOD_VERSION_3 이상이고, MQOO_ALTERNATE_USER_AUTHORITY가 지정되어 있으며, MQOD.AlternateSecurityId가 MQSID_NONE과 동일하지 않을 때에만 표시됩니다.
 PCF 매개변수: MQBACF_ALTERNATE_SECURITYID
 추적 레벨: 2
 유형: MQCFBS
 최대 길이: MQ_SECURITY_ID_LENGTH

ObjectString

설명:	긴 오브젝트 이름. MQOD.Version이 MQOD_VERSION_4 이상이고 MQOD.ObjectString의 VSLength 필드가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 포함됩니다.
PCF 매개변수:	MQCACF_OBJECT_STRING
추적 레벨:	2
유형:	MQCFST
최대 길이:	길이가 다양합니다.

SelectionString

설명:	선택 문자열. MQOD.Version이 MQOD_VERSION_4 이상이고 VSLength 필드가 MQOD와 같은 경우에만 포함됩니다. SelectionString은 MQVS_NULL_TERMINATED이거나 0보다 큼니다.
PCF 매개변수:	MQCACF_SELECTION_STRING
추적 레벨:	2
유형:	MQCFST
최대 길이:	길이가 다양합니다.

ResObjectString

설명:	큐 관리자가 ObjectName 필드에 제공된 이름을 분석한 후의 긴 오브젝트 이름. MQOD.Version이 MQOD_VERSION_4 이상이고 VSLength가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 토픽 오브젝트를 참조하는 토픽 및 큐 알리어스에 대해 포함됩니다.
PCF 매개변수:	MQCACF_RESOLVED_OBJECT_STRING
추적 레벨:	2
유형:	MQCFST
최대 길이:	길이가 다양합니다.

ResolvedType

설명:	여는 해석된(기본) 오브젝트의 유형입니다. MQOD.Version이 MQOD_VERSION_4 이상인 경우에만 포함됩니다. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형:	MQCFIN

QMgrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2

애플리케이션 활동 분배 목록 PCF 그룹 헤더 구조

MQOPEN 함수가 분배 목록을 여는 경우 MQOPEN 매개변수는 RecsPresent에 번호 지정된 구조 수까지, 분배 목록의 각 큐에 대한 AppActivityDistList PCF 그룹을 하나씩 포함합니다. Ap-pActivityDistList PCF 그룹은 MQOR 및 MQRR 구조의 정보를 결합하여 큐 이름을 식별하고 큐에 대한 열기 조작의 결과를 표시합니다.

AppActivityDistList 그룹은 항상 다음 MQCFGR 구조로 시작합니다.

표 29. AppActivityDistList 그룹 MQCFGR 구조		
MQCFGR 필드	값	설명
유형	MQCFT_GROUP	
StrucLength	MQCFGR 구조의 길이(바이트)	
매개변수	MQGACF_APP_DIST_LIST	분배 목록 그룹 매개변수
ParameterCount	4	이 그룹 내에 포함된 MQCFGR 구조를 따르는 매개변수 구조의 수.

ObjectName

- 설명: 분배 목록 MQ_Q_NAME_LENGTH의 큐 이름. MQOR 구조가 제공된 경우에만 포함됩니다.
- PCF 매개변수: MQCACF_OBJECT_NAME
- 추적 레벨: 2
- 유형: MQCFST
- 길이: MQ_Q_NAME_LENGTH. MQOR 구조가 제공된 경우에만 포함됩니다.

ObjectQMgrName

- 설명: ObjectName에 이름 지정된 큐가 정의되어 있는 큐 관리자의 이름.
- PCF 매개변수: MQCACF_OBJECT_Q_MGR_NAME
- 추적 레벨: 2
- 유형: MQCFST
- 길이: MQ_Q_MGR_NAME_LENGTH. MQOR 구조가 제공된 경우에만 포함됩니다.

CompCode

- 설명: 이 오브젝트에 대한 열기 결과를 나타내는 완료 코드. MQRR 구조가 제공되고 MQOPEN의 이유 코드가 MQRC_MULTIPLE_REASONS인 경우에만 포함됩니다.
- PCF 매개변수: MQIACF_COMP_CODE
- 추적 레벨: 2
- 유형: MQCFIN

¹ 이 매개변수는 열린 오브젝트가 큐로 해석되고 MQOO_INPUT_*, MQOO_OUTPUT 또는 MQOO_BROWSE에 대해 큐가 열린 경우에만 포함됩니다.

² ResolvedLocalQName 매개변수는 ResolvedQName 매개변수와 다른 경우에만 포함됩니다.

Reason

설명:	이 오브젝트에 대한 열기 결과를 나타내는 이유 코드. MQRR 구조가 제공되고 MQOPEN의 이유 코드가 MQRC_MULTIPLE_REASONS인 경우에만 포함됩니다.
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	2
유형:	MQCFIN

MQPUT

애플리케이션이 MQPUT MQI 함수를 시작했습니다.

ObjectHandle

설명:	오브젝트 핸들
PCF 매개변수:	MQIACF_HOBJ
추적 레벨:	1
유형:	MQCFIN

PutOptions

설명:	MQPMO.Options의 넣기 옵션
PCF 매개변수:	MQIACF_PUT_OPTIONS
추적 레벨:	1
유형:	MQCFIN

CompCode

설명:	조작 결과를 나타내는 완료 코드
PCF 매개변수:	MQIACF_COMP_CODE
추적 레벨:	1
유형:	MQCFIN

Reason

설명:	조작의 이유 코드 결과
PCF 매개변수:	MQIACF_REASON_CODE
추적 레벨:	1
유형:	MQCFIN

MsgBuffer

설명:	메시지 데이터.
PCF 매개변수:	MQBACF_MESSAGE_DATA
추적 레벨:	1
유형:	MQCFBS
길이:	길이는 APPTRACE 구성에 설정된 TRACEDATA() 매개변수로 제어됩니다. TRACEDATA=NONE이면 이 매개변수가 생략됩니다.

MsgLength

설명:	메시지의 길이
PCF 매개변수:	MQIACF_MSG_LENGTH
추적 레벨:	1
유형:	MQCFIN

RecsPresent

설명:	레코드 또는 응답 레코드가 있는 메시지 넣기 수. MQPMO 버전 >= MQPMO_VERSION_2인 경우에만 포함됩니다.
PCF 매개변수:	MQIACF_RECS_PRESENT
추적 레벨:	1
유형:	MQCFIN

KnownDestCount

설명:	로컬 큐에 성공적으로 전송된 메시지 수
PCF 매개변수:	MQIACF_KNOWN_DEST_COUNT
추적 레벨:	1
유형:	MQCFIN

UnknownDestCount

설명:	리모트 큐에 성공적으로 전송된 메시지 수
PCF 매개변수:	MQIACF_UNKNOWN_DEST_COUNT
추적 레벨:	1
유형:	MQCFIN

InvalidDestCount

설명:	전송할 수 없는 메시지 수
PCF 매개변수:	MQIACF_INVALID_DEST_COUNT
추적 레벨:	1
유형:	MQCFIN

HighResTime

설명:	1970년 1월 1일(UTC), 자정 이후의 조작 시간(마이크로초) 참고: 이 타이머의 정확도는 고해상도 타이머에 대한 플랫폼 지원에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_HIGHRES_TIME
추적 레벨:	2
유형:	MQCFIN64

ObjectName

설명:	열린 오브젝트의 이름.
PCF 매개변수:	MQCACF_OBJECT_NAME

추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

ResolvedQName

설명: 큐 이름 분석이 수행된 후의 큐 이름.
PCF 매개변수: MQCACF_RESOLVED_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

ResolvedQMgrName

설명: 이름 분석이 수행된 후의 큐 관리자 이름.
PCF 매개변수: MQCACF_RESOLVED_Q_MGR
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_MGR_NAME_LENGTH

ResolvedLocalQName ³

설명: 이름 분석이 수행된 후의 로컬 큐 이름을 포함합니다.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨: 2
유형: MQCFST

ResolvedLocalQMgrName ³

설명: 이름 분석이 수행된 후의 로컬 큐 관리자 이름을 포함합니다.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_MGR
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_MGR_NAME_LENGTH

ReportOptions

설명: 메시지 보고 옵션
PCF 매개변수: MQIACF_REPORT
추적 레벨: 2
유형: MQCFIN

MsgType

설명: 메시지의 유형
PCF 매개변수: MQIACF_MSG_TYPE
추적 레벨: 2

유형: MQCFIN

Expiry

설명: 메시지 수명

PCF 매개변수: MQIACF_EXPIRY

추적 레벨: 2

유형: MQCFIN

Format

설명: 메시지 데이터의 형식 이름

PCF 매개변수: MQCACH_FORMAT_NAME

추적 레벨: 2

유형: MQCFST

길이: MQ_FORMAT_LENGTH

Priority

설명: 메시지 우선순위

PCF 매개변수: MQIACF_PRIORITY

추적 레벨: 2

유형: MQCFIN

Persistence

설명: 메시지 지속성

PCF 매개변수: MQIACF_PERSISTENCE

추적 레벨: 2

유형: MQCFIN

MsgId

설명: 메시지 ID

PCF 매개변수: MQBACF_MSG_ID

추적 레벨: 2

유형: MQCFBS

길이: MQ_MSG_ID_LENGTH

CorrelId

설명: 상관 ID

PCF 매개변수: MQBACF_CORREL_ID

추적 레벨: 2

유형: MQCFBS

길이: MQ_CORREL_ID_LENGTH

ReplyToQueue

설명:
PCF 매개변수: MQCACF_REPLY_TO_Q
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

ReplyToQMgr

설명:
PCF 매개변수: MQCACF_REPLY_TO_Q_MGR
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_MGR_NAME_LENGTH

CodedCharSetId

설명: 메시지 데이터의 문자 세트 ID
PCF 매개변수: MQIA_CODED_CHAR_SET_ID
추적 레벨: 2
유형: MQCFIN

Encoding

설명: 메시지 데이터의 숫자 인코딩입니다.
PCF 매개변수: MQIACF_ENCODING
추적 레벨: 2
유형: MQCFIN

PutDate

설명:
PCF 매개변수: MQCACF_PUT_DATE
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_DATE_LENGTH

PutTime

설명:
PCF 매개변수: MQCACF_PUT_TIME
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_TIME_LENGTH

ResolvedQName

설명:	ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
PCF 매개변수:	MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨:	2
유형:	MQCFST
길이:	MQ_Q_NAME_LENGTH.

ResObjectString

설명:	ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.
PCF 매개변수:	MQCACF_RESOLVED_OBJECT_STRING
추적 레벨:	2
유형:	MQCFST
길이:	길이가 다양합니다.

ResolvedType

설명:	ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형:	MQCFIN

PolicyName

설명:	이 메시지에 적용된 정책 이름. 참고: AMS 보호 메시지만 해당함
PCF 매개변수:	MQCA_POLICY_NAME
추적 레벨:	2
유형:	MQCFST
길이:	MQ_OBJECT_NAME_LENGTH

XmitqMsgId

설명:	전송 큐 헤더에 있는 메시지의 메시지 ID. 참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함
PCF 매개변수:	MQBACF_XQH_MSG_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_MSG_ID_LENGTH

XmitqCorrelId

설명:	전송 큐 헤더에 있는 메시지의 상관 ID. 참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함
PCF 매개변수:	MQBACF_XQH_CORREL_ID

추적 레벨: 2
유형: MQCFBS
길이: MQ_CORREL_ID_LENGTH

XmitqPutTime

설명: 전송 큐 헤더에 있는 메시지의 넣기 시간.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_TIME
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_TIME_LENGTH

XmitqPutDate

설명: 전송 큐 헤더에 있는 메시지의 넣기 날짜.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_DATE
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

XmitqRemoteQMgr

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 관리자 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_MGR
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

PutMsgOptsStructure

설명: MQPMO 구조.
PCF 매개변수: MQBACF_MQPMO_STRUCT
추적 레벨: 3

유형: MQCFBS
 길이: MQPMO 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMGrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
 추적 레벨: 2
 유형: MQCFIN64

MQPUT 애플리케이션 활동 분배 목록 PCF 그룹 헤더 구조

MQPUT 함수가 분배 목록에 넣기를 수행 중이면 MQPUT 매개변수가 AppActivityDistList PCF 그룹을 하나 포함합니다. 분배 목록의 각 큐에 대해서는 235 페이지의 『애플리케이션 활동 분배 목록 PCF 그룹 헤더 구조』의 내용을 참조하십시오. AppActivityDistList PCF 그룹은 MQPMR 및 MQRR 구조의 정보를 결합하여 PUT 매개변수를 식별하고 각 큐에 대한 PUT 조작의 결과를 표시합니다. MQPUT 조작의 경우 AppActivityDistList 그룹은 다음의 모든 또는 일부 매개변수를 포함합니다(이유 코드가 MQRC_MULTIPLE_REASONS이고 MQPMO.PutMsgRecFields 필드를 통해 다른 매개변수가 판별되면 CompCode 및 Reason 필드가 존재함).

CompCode

설명: 조작 결과를 나타내는 완료 코드. MQRR 구조가 제공되고 MQPUT의 이유 코드가 MQRC_MULTIPLE_REASONS인 경우에만 포함됩니다.

PCF 매개변수: MQIACF_COMP_CODE
 추적 레벨: 2
 유형: MQCFIN

Reason

설명: 이 오브젝트에 대한 넣기 결과를 나타내는 이유 코드. MQRR 구조가 제공되고 MQPUT의 이유 코드가 MQRC_MULTIPLE_REASONS인 경우에만 포함됩니다.

PCF 매개변수: MQIACF_REASON_CODE
 추적 레벨: 2
 유형: MQCFIN

MsgId

설명: 메시지 ID. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_MSG_ID가 있는 경우에만 포함됩니다.

PCF 매개변수: MQBACF_MSG_ID
 추적 레벨: 2
 유형: MQCFBS
 길이: MQ_MSG_ID_LENGTH

³ ResolvedLocalQName 매개변수는 ResolvedQName 매개변수와 다른 경우에만 포함됩니다.

CorrelId

설명:	상관 ID. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_CORREL_ID가 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_CORREL_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_CORREL_ID_LENGTH

GroupId

설명:	그룹 ID. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_GROUP_ID가 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_GROUP_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_GROUP_ID_LENGTH

Feedback

설명:	피드백. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_FEEDBACK이 있는 경우에만 포함됩니다.
PCF 매개변수:	MQIACF_FEEDBACK
추적 레벨:	2
유형:	MQCFIN

AccountingToken

설명:	AccountingToken. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_ACCOUNTING_TOKEN이 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_ACCOUNTING_TOKEN
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_ACCOUNTING_TOKEN_LENGTH.

MQPUT1

애플리케이션이 MQPUT1 MQI 함수를 시작함

ObjectType

설명:	MQOT.ObjectType에 전달된 오브젝트 유형
PCF 매개변수:	MQIACF_OBJECT_TYPE
추적 레벨:	1
유형:	MQCFIN

ObjectName

설명:	큐 이름 분석을 시도하기 전에 MQI 호출에 전달된 오브젝트의 이름.
PCF 매개변수:	MQCACF_OBJECT_NAME

추적 레벨: 1
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

ObjectQMgrName

설명: 큐 이름 분석을 시도하기 전에 MQI 호출에 전달된 오브젝트 큐 관리자의 이름.
PCF 매개변수: MQCACF_OBJECT_Q_MGR_NAME
추적 레벨: 1
유형: MQCFST
길이: MQ_Q_MGR_NAME_LENGTH

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 1
유형: MQCFIN

PutOptions

설명: MQPMO.Options의 넣기 옵션
PCF 매개변수: MQIACF_PUT_OPTIONS
추적 레벨: 1
유형: MQCFIN

AlternateUserId

설명: MQPMO_ALTERNATE_USER_AUTHORITY가 지정된 경우에만 포함됩니다.
PCF 매개변수: MQCACF_ALTERNATE_USERID
추적 레벨: 2
유형: MQCFST
길이: MQ_USER_ID_LENGTH

RecsPresent

설명: 오브젝트 이름 레코드의 수가 표시됩니다.
PCF 매개변수: MQIACF_RECS_PRESENT
추적 레벨: 1
유형: MQCFIN

KnownDestCount

설명:	열린 로컬 큐의 수
PCF 매개변수:	MQIACF_KNOWN_DEST_COUNT
추적 레벨:	1
유형:	MQCFIN

UnknownDestCount

설명:	열린 리모트 큐의 수
PCF 매개변수:	MQIACF_UNKNOWN_DEST_COUNT
추적 레벨:	1
유형:	MQCFIN

InvalidDestCount

설명:	여는 데 실패한 큐의 수
PCF 매개변수:	MQIACF_INVALID_DEST_COUNT
추적 레벨:	1
유형:	MQCFIN

MsgBuffer

설명:	메시지 데이터.
PCF 매개변수:	MQBACF_MESSAGE_DATA
추적 레벨:	1
유형:	MQCFBS
길이:	길이는 APPTRACE 구성에 설정된 TRACEDATA() 매개변수로 제어됩니다. TRACEDATA=NONE이면 이 매개변수가 생략됩니다.

MsgLength

설명:	메시지의 길이
PCF 매개변수:	MQIACF_MSG_LENGTH
추적 레벨:	1
유형:	MQCFIN

HighResTime

설명:	1970년 1월 1일(UTC), 자정 이후의 조작 시간(마이크로초) 참고: 이 타이머의 정확도는 고해상도 타이머에 대한 플랫폼 지원에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_HIGHRES_TIME
추적 레벨:	2
유형:	MQCFIN64

ResolvedQName

설명:	큐 이름 분석이 수행된 후의 큐 이름.
-----	-----------------------

PCF 매개변수: MQCACF_RESOLVED_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

ResolvedQMgrName

설명: 이름 분석이 수행된 후의 큐 관리자 이름.
PCF 매개변수: MQCACF_RESOLVED_Q_MGR
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_MGR_NAME_LENGTH

ResolvedLocalQName ⁴

설명: 이름 분석이 수행된 후의 로컬 큐 이름을 포함합니다.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨: 2
유형: MQCFST

ResolvedLocalQMgrName ⁴

설명: 이름 분석이 수행된 후의 로컬 큐 관리자 이름을 포함합니다.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_MGR
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

설명: 대체 보안 ID. MQOD.Version이 MQOD_VERSION_3 이상이고 MQOD.AlternateSecurityId가 MQSID_NONE과 동일하지 않을 때에만 표시됩니다.
PCF 매개변수: MQBACF_ALTERNATE_SECURITYID
추적 레벨: 2
유형: MQCFBS
길이: MQ_SECURITY_ID_LENGTH

ObjectString

설명: 긴 오브젝트 이름. MQOD.Version이 MQOD_VERSION_4 이상이고 MQOD.ObjectString의 VSLength 필드가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 포함됩니다.
PCF 매개변수: MQCACF_OBJECT_STRING
추적 레벨: 2
유형: MQCFST
길이: 길이가 다양합니다.

ResObjectString

설명:	큐 관리자가 ObjectName 필드에 제공된 이름을 분석한 후의 긴 오브젝트 이름. MQOD.Version이 MQOD_VERSION_4 이상이고 VSLength가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 토픽 오브젝트를 참조하는 토픽 및 큐 알리어스에 대해 포함됩니다.
PCF 매개변수:	MQCACF_RESOLVED_OBJECT_STRING
추적 레벨:	2
유형:	MQCFST
길이:	길이가 다양합니다.

ResolvedType

설명:	여는 해석된(기본) 오브젝트의 유형입니다. MQOD.Version이 MQOD_VERSION_4 이상인 경우에만 포함됩니다. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE 입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형:	MQCFIN

ReportOptions

설명:	메시지 보고 옵션
PCF 매개변수:	MQIACF_REPORT
추적 레벨:	2
유형:	MQCFIN

MsgType

설명:	메시지의 유형
PCF 매개변수:	MQIACF_MSG_TYPE
추적 레벨:	2
유형:	MQCFIN

Expiry

설명:	메시지 수명
PCF 매개변수:	MQIACF_EXPIRY
추적 레벨:	2
유형:	MQCFIN

Format

설명:	메시지 데이터의 형식 이름
PCF 매개변수:	MQCACH_FORMAT_NAME
추적 레벨:	2
유형:	MQCFST
길이:	MQ_FORMAT_LENGTH

Priority

설명:	메시지 우선순위
PCF 매개변수:	MQIACF_PRIORITY
추적 레벨:	2
유형:	MQCFIN

Persistence

설명:	메시지 지속성
PCF 매개변수:	MQIACF_PERSISTENCE
추적 레벨:	2
유형:	MQCFIN

MsgId

설명:	메시지 ID
PCF 매개변수:	MQBACF_MSG_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_MSG_ID_LENGTH

CorrelId

PCF 매개변수:	상관 ID
설명:	MQBACF_CORREL_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_CORREL_ID_LENGTH

ReplyToQueue

설명:	
PCF 매개변수:	MQCACF_REPLY_TO_Q
추적 레벨:	2
유형:	MQCFST
길이:	MQ_Q_NAME_LENGTH

ReplyToQMgr

설명:	
PCF 매개변수:	MQCACF_REPLY_TO_Q_MGR
추적 레벨:	2
유형:	MQCFST
길이:	MQCFST

CodedCharSetId

설명: 메시지 데이터의 문자 세트 ID
PCF 매개변수: MQIA_CODED_CHAR_SET_ID
추적 레벨: 2
유형: MQCFIN

Encoding

설명: 메시지 데이터의 숫자 인코딩입니다.
PCF 매개변수: MQIACF_ENCODING
추적 레벨: 2
유형: MQCFIN

PutDate

설명:
PCF 매개변수: MQCACF_PUT_DATE
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_DATE_LENGTH

PutTime

설명:
PCF 매개변수: MQCACF_PUT_TIME
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_TIME_LENGTH

PolicyName

설명: 이 메시지에 적용된 정책 이름.
참고: AMS 보호 메시지만 해당함
PCF 매개변수: MQCA_POLICY_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

설명: 전송 큐 헤더에 있는 메시지의 메시지 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함
PCF 매개변수: MQBACF_XQH_MSG_ID
추적 레벨: 2
유형: MQCFBS

길이: MQ_MSG_ID_LENGTH

XmitqCorrelId

설명: 전송 큐 헤더에 있는 메시지의 상관 ID.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQBACF_XQH_CORREL_ID
추적 레벨: 2
유형: MQCFBS
길이: MQ_CORREL_ID_LENGTH

XmitqPutTime

설명: 전송 큐 헤더에 있는 메시지의 넣기 시간.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_TIME
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_TIME_LENGTH

XmitqPutDate

설명: 전송 큐 헤더에 있는 메시지의 넣기 날짜.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_PUT_DATE
추적 레벨: 2
유형: MQCFST
길이: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

XmitqRemoteQMgr

설명: 전송 큐 헤더에 있는 메시지의 리모트 큐 관리자 목적지.
참고: 형식이 MQFMT_XMIT_Q_HEADER일 때에만 해당함

PCF 매개변수: MQCACF_XQH_REMOTE_Q_MGR
추적 레벨: 2
유형: MQCFST

길이: MQ_Q_NAME_LENGTH

PutMsgOptsStructure

설명: MQPMO 구조.
PCF 매개변수: MQBACF_MQPMO_STRUCT
추적 레벨: 3
유형: MQCFBS
길이: MQPMO 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMGrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
추적 레벨: 2
유형: MQCFIN64

MQPUT1 AppActivityDistList PCF 그룹 헤더 구조

MQPUT1 함수가 분배 목록에 넣기를 수행 중이면 변수 매개변수가 AppActivityDistList PCF 그룹을 하나 포함합니다. 분배 목록의 각 큐에 대해서는 235 페이지의 『애플리케이션 활동 분배 목록 PCF 그룹 헤더 구조』의 내용을 참조하십시오. AppActivityDistList PCF 그룹은 MQOR, MQPMR, MQRR 구조의 정보를 결합하여 PUT 매개변수를 식별하고 각 큐에 대한 PUT 조작의 결과를 표시합니다. MQPUT1 조작의 경우 AppActivityDistList 그룹은 다음의 모든 또는 일부 매개변수를 포함합니다(이유 코드가 MQRC_MULTIPLE_REASONS이고 MQPMO.PutMsgRecFields 필드를 통해 다른 매개변수가 판별되면 CompCode, Reason, ObjectName, ObjectQMGrName 필드가 존재함).

CompCode

설명: 이 오브젝트에 대한 넣기 결과를 나타내는 완료 코드. MQRR 구조가 제공되고 MQPUT1의 이유 코드가 MQRC_MULTIPLE_REASONS인 경우에만 포함됩니다.

PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 2
유형: MQCFIN

Reason

설명: 이 오브젝트에 대한 넣기 결과를 나타내는 이유 코드. MQRR 구조가 제공되고 MQPUT1의 이유 코드가 MQRC_MULTIPLE_REASONS인 경우에만 포함됩니다.

PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 2
유형: MQCFIN

⁴ ResolvedLocalQName 매개변수는 ResolvedQName 매개변수와 다른 경우에만 포함됩니다.

ObjectName

설명:	분배 목록의 큐 이름. MQOR 구조가 제공된 경우에만 포함됩니다.
PCF 매개변수:	MQCACF_OBJECT_NAME
추적 레벨:	2
유형:	MQCFST
길이:	MQ_Q_NAME_LENGTH

MsgId

설명:	메시지 ID. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_MSG_ID가 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_MSG_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_MSG_ID_LENGTH

CorrelId

설명:	상관 ID. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_CORREL_ID가 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_CORREL_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_CORREL_ID_LENGTH

GroupId

설명:	그룹 ID. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_GROUP_ID가 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_GROUP_ID
추적 레벨:	2
유형:	MQCFBS
길이:	MQ_GROUP_ID_LENGTH

Feedback

설명:	피드백. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_FEEDBACK이 있는 경우에만 포함됩니다.
PCF 매개변수:	MQIACF_FEEDBACK
추적 레벨:	2
유형:	MQCFIN

AccountingToken

설명:	AccountingToken. MQPMR 구조가 제공되고 PutMsgRecFields에 MQPMRF_ACCOUNTING_TOKEN이 있는 경우에만 포함됩니다.
PCF 매개변수:	MQBACF_ACCOUNTING_TOKEN

추적 레벨: 2
 유형: MQCFBS
 길이: MQ_ACCOUNTING_TOKEN_LENGTH.

MQSET

애플리케이션이 MQSET MQI 함수를 시작함

ObjectHandle

설명: 오브젝트 핸들
 PCF 매개변수: MQIACF_HOBJ
 추적 레벨: 1
 유형: MQCFIN

CompCode

설명: 조작 결과를 나타내는 완료 코드
 PCF 매개변수: MQIACF_COMP_CODE
 추적 레벨: 1
 유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
 PCF 매개변수: MQIACF_REASON_CODE
 추적 레벨: 1
 유형: MQCFIN

SelectorCount

설명: Selectors 배열에 제공되는 선택자 수.
 PCF 매개변수: MQIACF_SELECTOR_COUNT
 추적 레벨: 2
 유형: MQCFIN

Selectors

설명: MQSET을 통해 값을 업데이트하고 있는 속성의 목록(정수 또는 문자).
 PCF 매개변수: MQIACF_SELECTORS
 추적 레벨: 2
 유형: MQCFIL

ResolvedQName

설명: ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
 PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
 추적 레벨: 2
 유형: MQCFST
 길이: MQ_Q_NAME_LENGTH.

ResObjectString

설명:	ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.
PCF 매개변수:	MQCACF_RESOLVED_OBJECT_STRING
추적 레벨:	2
유형	MQCFST
길이:	길이가 다양합니다.

ResolvedType

설명:	ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수:	MQIACF_RESOLVED_TYPE
추적 레벨:	2
유형	MQCFIN

IntAttrCount

설명:	설정 조작으로 업데이트할 정수 속성의 수.
PCF 매개변수:	MQIACF_INTATTR_COUNT
추적 레벨:	3
유형:	MQCFIN

IntAttrs

설명:	정수 속성 값
PCF 매개변수:	MQIACF_INT_ATTRS
추적 레벨:	3
유형:	MQCFIL
범위:	이 매개변수는 IntAttrCount가 0보다 큰 경우에만 제공됩니다.

CharAttrs

설명:	설정 조작으로 업데이트할 문자 속성. 값이 서로 연결됩니다.
PCF 매개변수:	MQCACF_CHAR_ATTRS
추적 레벨:	3
유형:	MQCFST
범위:	이 매개변수는 CharAttrLength가 0보다 큰 경우에만 포함됩니다.

QMgrOpDuration

설명:	큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초). 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다. 참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.
PCF 매개변수:	MQIAMO64_QMGR_OP_DURATION
추적 레벨:	2

유형 MQCFIN64

MQSUB

애플리케이션이 MQSUB MQI 함수를 시작함

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 1
유형: MQCFIN

SubHandle

설명: 구독 핸들
PCF 매개변수: MQIACF_HSUB
추적 레벨: 1
유형: MQCFIN

ObjectHandle

설명: 오브젝트 핸들
PCF 매개변수: MQIACF_HOBJ
추적 레벨: 1
유형: MQCFIN

Options

설명: 구독 옵션
PCF 매개변수: MQIACF_SUB_OPTIONS
추적 레벨: 1
유형: MQCFIN

ObjectName

설명: 오브젝트의 이름.
PCF 매개변수: MQCACF_OBJECT_NAME
추적 레벨: 1
유형: MQCFST
길이: MQ_Q_NAME_LENGTH

ObjectString

설명:	긴 오브젝트 이름.
PCF 매개변수:	MQCACF_OBJECT_STRING
추적 레벨:	1
유형:	MQCFST
범위:	MQSD.ObjectString의 VSLength 필드가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 포함됩니다.
길이:	길이가 다양합니다.

AlternateUserId

설명:	
PCF 매개변수:	MQCACF_ALTERNATE_USERID
추적 레벨:	2
유형:	MQCFST
범위:	MQSO_ALTERNATE_USER_AUTHORITY가 지정된 경우에만 포함됩니다.
길이:	MQ_USER_ID_LENGTH

AlternateSecurityId

설명:	대체 보안 ID.
PCF 매개변수:	MQBACF_ALTERNATE_SECURITYID
추적 레벨:	2
유형:	MQCFBS
범위:	MQSO_ALTERNATE_USER_AUTHORITY가 지정되고 MQSD.AlternateSecurityId가 MQSID_NONE이 아닌 경우에만 존재합니다.
길이:	MQ_SECURITY_ID_LENGTH

SubName

설명:	구독 이름
PCF 매개변수:	MQCACF_SUB_NAME
추적 레벨:	2
유형:	MQCFST
범위:	MQSD.SubName의 VSLength 필드가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 포함됩니다.
길이:	길이가 다양합니다.

SubUserData

설명:	Subscription 사용자 데이터
PCF 매개변수:	MQCACF_SUB_USER_DATA
추적 레벨:	2
유형:	MQCFST
범위:	MQSD.SubName의 VSLength 필드가 MQVS_NULL_TERMINATED 또는 0보다 큰 경우에만 포함됩니다.

길이: 길이가 다양합니다.

SubCorrelId

설명: 구독 상관 ID
PCF 매개변수: MQBACF_SUB_CORREL_ID
추적 레벨: 2
유형: MQCFBS
길이: MQ_CORREL_ID_LENGTH

SelectionString

설명: 선택 문자열.
PCF 매개변수: MQCACF_SELECTION_STRING
추적 레벨: 2
유형: MQCFST
범위: MQSD의 VSLength 필드가 지정된 경우에만 포함됩니다. SelectionString은 MQVS_NULL_TERMINATED이거나 0보다 큼니다.
길이: 길이가 다양합니다.

ResolvedQName

설명: ResolvedType이 MQOT_Q일 때 ObjectHandle에 참조된 큐 이름.
PCF 매개변수: MQCACF_RESOLVED_LOCAL_Q_NAME
추적 레벨: 2
유형: MQCFST
길이: MQ_Q_NAME_LENGTH.

ResObjectString

설명: ResolvedType이 MQOT_TOPIC일 때 ObjectHandle에 참조된 오브젝트 이름.
PCF 매개변수: MQCACF_RESOLVED_OBJECT_STRING
추적 레벨: 2
유형: MQCFST
길이: 길이가 다양합니다.

ResolvedType

설명: ObjectHandle에 참조된 오브젝트의 유형. 가능한 값은 MQOT_Q, MQOT_TOPIC 또는 MQOT_NONE입니다.
PCF 매개변수: MQIACF_RESOLVED_TYPE
추적 레벨: 2
유형: MQCFIN

SubDescriptorStructure

설명: MQSD 구조.
PCF 매개변수: MQBACF_MQSD_STRUCT

추적 레벨: 3
 유형: MQCFBS
 길이: MQSD 구조의 길이(바이트).

QMGrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
 추적 레벨: 2
 유형: MQCFIN64

MQSUBRQ

애플리케이션이 MQSUBRQ MQI 함수를 시작함

CompCode

설명: 조작 결과를 나타내는 완료 코드
 PCF 매개변수: MQIACF_COMP_CODE
 추적 레벨: 1
 유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
 PCF 매개변수: MQIACF_REASON_CODE
 추적 레벨: 1
 유형: MQCFIN

SubHandle

설명: 구독 핸들
 PCF 매개변수: MQIACF_HSUB
 추적 레벨: 1
 유형: MQCFIN

SubOptions

설명: MQSB.Options의 하위 옵션
 PCF 매개변수: MQIACF_SUBRQ_OPTIONS
 추적 레벨: 2
 유형: MQCFIN

Action

설명: 구독 요청 조치(MQSR_*)
 PCF 매개변수: MQIACF_SUBRQ_ACTION

추적 레벨: 2
유형: MQCFIN

NumPubs

설명: 이 호출의 결과로 송신된(MQSB.NumPubs에서) 발행물 수
PCF 매개변수: MQIACF_NUM_PUBS
추적 레벨: 2
유형: MQCFIN

QMGrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
추적 레벨: 2
유형: MQCFIN64

MQSTAT
애플리케이션이 MQSTAT MQI 함수를 시작함

CompCode

설명: 조작 결과를 나타내는 완료 코드
PCF 매개변수: MQIACF_COMP_CODE
추적 레벨: 1
유형: MQCFIN

Reason

설명: 조작의 이유 코드 결과
PCF 매개변수: MQIACF_REASON_CODE
추적 레벨: 1
유형: MQCFIN

Type

설명: 요청 중인 상태 정보의 유형
PCF 매개변수: MQIACF_STATUS_TYPE
추적 레벨: 2
유형: MQCFIN

StatusStructure

설명: MQSTS 구조.
PCF 매개변수: MQBACF_MQSTS_STRUCT
추적 레벨: 3

유형: MQCFBS
 길이: MQSTS 구조의 바이트 길이(실제 크기는 구조 버전에 따라 다름)

QMGrOpDuration

설명: 큐 관리자 내의 대략적인 API 호출 지속 기간(마이크로초).
 큐 관리자 외부에서 소모된 시간은 지속 기간에 포함되지 않습니다. 예를 들어, IBM MQ 클라이언트로 사용한 시간은 포함되지 않습니다.
참고: 이 타이머의 정확도는 엔터프라이즈에서 사용하는 플랫폼에 따라 다릅니다.

PCF 매개변수: MQIAMO64_QMGR_OP_DURATION
 추적 레벨: 2
 유형: MQCFIN64

애플리케이션 활동 XA 조작에 대한 변수 매개변수

XA 조작은 MQ가 트랜잭션에 참여하도록 하기 위해 애플리케이션이 수행할 수 있는 API 호출입니다. 각 조작에 대한 매개변수는 다음 절에 정의되어 있습니다.

추적 레벨은 추적에 포함시킬 매개변수에 필요한 추적 세분화 레벨을 표시합니다. 가능한 추적 레벨 값은 다음과 같습니다.

1. 낮음

애플리케이션에 "낮음", "중간" 또는 "높음" 활동 추적이 구성될 때 매개변수가 포함됩니다. 이 설정은 매개변수가 항상 조작에 대한 AppActivityData 그룹에 포함됨을 의미합니다. 애플리케이션이 수행하는 MQI 호출을 추적하고 호출이 성공했는지 여부를 확인하는 데에는 이 매개변수 세트면 충분합니다.

2. 중간

애플리케이션에 대해 "중간" 또는 "높음" 활동 추적이 구성되어 있을 때에만 조작의 AppActivityData 그룹에 매개변수가 포함됩니다. 이 매개변수 세트는 애플리케이션에 사용되는 큐와 토픽 이름 같은 자원에 대한 정보를 추가합니다.

3. 높음

애플리케이션에 대해 "높음" 활동 추적이 구성되어 있을 때에만 조작의 AppActivityData 그룹에 매개변수가 포함됩니다. 이 매개변수 세트는 MQI 및 XA 함수에 전달되는 구조의 메모리 덤프를 포함합니다. 이러한 이유로, MQI 및 XA 호출에 사용되는 매개변수에 대한 자세한 정보를 포함합니다. 구조 메모리 덤프는 구조의 셸로우(shallow) 사본입니다. 참조 해제 포인터에 대한 잘못된 시도를 피하기 위해 구조의 포인터 값은 NULL로 설정됩니다.

참고: 덤프된 구조 버전이 애플리케이션에 사용된 버전과 반드시 동일하지는 않습니다. API 교차 엑시트, 활동 추적 코드 또는 큐 관리자를 통해 구조를 수정할 수 있습니다. 큐 관리자는 이후 버전에 구조를 수정할 수 있지만 구조의 이전 버전에는 결코 구조를 변경하지 않습니다. 이전 버전에 대해 변경할 경우 데이터 손실 위험이 있습니다.

AXREG

애플리케이션이 AXREG AX 함수를 시작함

XID

설명: XID 구조
 PCF 매개변수: MQBACF_XA_XID
 추적 레벨: 1
 유형: MQCFBS
 길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

AXUNREG

애플리케이션이 AXUNREG AX 함수를 시작함

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XACLOSE

애플리케이션이 XACLOSE AX 함수를 시작함

Xa_info

설명: 자원 관리자를 초기화하는 데 사용된 정보.

PCF 매개변수: MQCACF_XA_INFO
추적 레벨: 1
유형: MQCFST

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XACOMMIT

애플리케이션이 XACOMMIT AX 함수를 시작함

XID

설명: XID 구조
PCF 매개변수: MQBACF_XA_XID
추적 레벨: 1
유형: MQCFBS
길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XACOMPLETE

애플리케이션이 XACOMPLETE AX 함수를 시작함

Handle

설명: 비동기 조작에 대한 핸들
PCF 매개변수: MQIACF_XA_HANDLE
추적 레벨: 1
유형: MQCFIN

Retval

설명: 비동기 함수의 리턴 값
PCF 매개변수: MQIACF_XA_RETVAl
추적 레벨: 1
유형: MQCFINMQCFBS

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XAEND

애플리케이션이 XAEND AX 함수를 시작함

XID

설명: XID 구조

PCF 매개변수: MQBACF_XA_XID
추적 레벨: 1
유형: MQCFBS
길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XAFORGET

애플리케이션이 AXREG AX 함수를 시작함

XID

설명: XID 구조
PCF 매개변수: MQBACF_XA_XID
추적 레벨: 1
유형: MQCFBS
길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1

유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XAOPEN

애플리케이션이 XAOPEN AX 함수를 시작함

Xa_info

설명: 자원 관리자를 초기화하는 데 사용된 정보.
PCF 매개변수: MQCACF_XA_INFO
추적 레벨: 1
유형: MQCFST

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XAPREPARE

애플리케이션이 XAPREPARE AX 함수를 시작함

XID

설명: XID 구조
PCF 매개변수: MQBACF_XA_XID
추적 레벨: 1
유형: MQCFBS
길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XARECOVER

애플리케이션이 XARECOVER AX 함수를 시작함

Count

설명: XID 수
PCF 매개변수: MQIACF_XA_COUNT
추적 레벨: 1
유형: MQCFIN

XIDs

설명: XID 구조
참고: 모든 XID 구조에 하나씩, 최대 Count XID 수까지 이 PCF 매개변수의 여러 인스턴스가 있습니다.
PCF 매개변수: MQBACF_XA_XID
추적 레벨: 1
유형: MQCFBS
길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XAROLLBACK

애플리케이션이 XAROLLBACK AX 함수를 시작함

XID

설명: XID 구조
PCF 매개변수: MQBACF_XA_XID
추적 레벨: 1
유형: MQCFBS
길이: Sizeof(XID)

Rmid

설명: 자원 관리자 ID
PCF 매개변수: MQIACF_XA_RMID
추적 레벨: 1
유형: MQCFIN

Flags

설명: 플래그
PCF 매개변수: MQIACF_XA_FLAGS
추적 레벨: 1
유형: MQCFIN

XARetCode

설명: 리턴 코드
PCF 매개변수: MQIACF_XA_RETCODE
추적 레벨: 1
유형: MQCFIN

XASTART

애플리케이션이 XASTART AX 함수를 시작함

XID

설명:	XID 구조
PCF 매개변수:	MQBACF_XA_XID
추적 레벨:	1
유형:	MQCFBS
길이:	Sizeof(XID)

Rmid

설명:	자원 관리자 ID
PCF 매개변수:	MQIACF_XA_RMID
추적 레벨:	1
유형:	MQCFIN

Flags

설명:	플래그
PCF 매개변수:	MQIACF_XA_FLAGS
추적 레벨:	1
유형:	MQCFIN

XARetCode

설명:	리턴 코드
PCF 매개변수:	MQIACF_XA_RETCODE
추적 레벨:	1
유형:	MQCFIN

Multi 모니터링 및 활동 추적에 대한 시스템 토픽

큐 관리자 토픽 트리의 시스템 토픽은 자원 모니터링 (일부는 통계 메시지의 콘텐츠와 유사함) 및 애플리케이션 활동 추적을 이용하는 방법으로 사용됩니다.

큐 관리자 토픽 트리의 \$SYS/MQ 분기

각 큐 관리자의 토픽 트리에는 \$SYS/MQ 분기가 포함되어 있습니다. 큐 관리자는 이 분기에서 토픽 문자열로 발행합니다. 권한 부여된 사용자는 이 토픽 문자열에 등록하여 큐 관리자 및 큐 관리자의 활동에 대한 정보를 수신합니다. 이러한 시스템 토픽은 애플리케이션 활동 추적 및 모니터링에 사용됩니다. 자세한 정보는 [토픽 트리를 참조](#)하십시오.

\$SYS/MQ 분기의 루트는 SYSTEM.ADMIN.TOPIC 토픽 오브젝트로 표시됩니다. 토픽 트리의 \$SYS/MQ 분기는 다음의 방법으로 토픽 트리의 나머지에서 격리됩니다.

- 트리에서 \$SYS/MQ보다 위에 있는 지점에서 와일드카드 문자로 작성된 등록이 \$SYS/MQ 분기 내 토픽 문자열과 일치하지 않습니다. SYSTEM.ADMIN.TOPIC에 대한 와일드카드 조작은 "Block"으로 설정되어, 수정할 수 없습니다. 이러한 제한사항은 **runmqsc** 명령 DISPLAY TPSTATUS에서 와일드카드 문자를 사용하여 토픽 트리에서 노드를 표시하는 경우에도 적용됩니다. \$SYS/MQ 분기에서 토픽 노드를 보려면, \$SYS/MQ로 토픽 문자열을 시작하십시오. 예를 들어, 모든 노드를 보려면 \$SYS/MQ/#을 사용하십시오.
- \$SYS/MQ 토픽 트리를 사용하려면 \$SYS/MQ 또는 이보다 깊은 수준에서 권한을 부여해야 합니다. 토픽 문자열에 등록하기 위한 권한은 토픽 트리에서 토픽 문자열 또는 이보다 더 높은 수준에서 관리되는 토픽 오브젝트에 대해 부여되는 권한에 기반합니다. 루트(SYSTEM.BASE.TOPIC)에서 부여되는 권한은 모든 토픽 문자열에 사

용자 권한을 부여합니다. 그러나 \$SYS/MQ 분기의 경우 \$SYS/MQ보다 더 높은 수준에서 부여된 액세스는 \$SYS/MQ 토픽 문자열에 적용되지 않습니다.

- 토픽 트리의 \$SYS/MQ 분기는 트리에서 더 높은 수준으로 설정된 토픽 속성에서 격리됩니다. SYSTEM.ADMIN.TOPIC은 토픽 트리에서 더 높은 수준으로 정의된 토픽 오브젝트에서 속성을 상속하지 않습니다. 예를 들어, SYSTEM.BASE.TOPIC의 속성을 변경해도 \$SYS/MQ 분기의 동작에는 영향을 주지 않습니다.

\$SYS/MQ로 시작하는 모든 토픽 문자열은 IBM MQ에서 사용하도록 예약되어 있습니다. 이 토픽 문자열에는 다음과 같은 제한사항이 있습니다.

- 토픽 트리의 \$SYS/MQ 분기에서 멀티캐스트를 사용할 수 없습니다.
- 클러스터링은 \$SYS/MQ 분기에 대해 지원되지 않습니다.
- 프록시 등록 메커니즘은 "force"로 설정할 수 없습니다.
- 애플리케이션은 \$SYS/MQ 토픽 문자열로 발행할 수 없습니다.
- 발행 등록 범위의 기본값은 로컬 큐 관리자만으로 설정됩니다.
- 와일드카드 문자는 다음 지점에서 사용할 수 없습니다.
 - \$SYS/MQ/
 - \$SYS/MQ/INFO
 - \$SYS/MQ/INFO/QMGR
 - \$SYS/MQ/INFO/QMGR/queue_manager_name
 - \$SYS/MQ/INFO/QMGR/queue_manager_name/ActivityTrace

이 지점에서 와일드카드 문자를 사용하려고 하면 MQRC_ADMIN_TOPIC_STRING_ERROR의 이유로 등록에 실패합니다.

발행/구독을 통해 사용 가능한 성능 모니터링 API

단순 발행/구독 메커니즘과 amqsrua 샘플 애플리케이션 또는 자체 애플리케이션을 사용하여 큐 관리자 통계를 모니터링할 수 있습니다. 통계는 사용자가 자원을 모니터링하는 데 도움이 되는, \$SYS/MQ/INFO/QMGR 아래 시스템 토픽에 대해 발행됩니다. 이러한 통계는 amqsrua 샘플 애플리케이션을 실행하거나 **amqsrua**와 유사한 방식으로 자원 모니터링 시스템 토픽을 구독하는 애플리케이션을 작성하여 볼 수 있습니다.

관련 개념

187 페이지의 『애플리케이션 활동 추적』

애플리케이션 활동 추적은 큐 관리자에 연결된 애플리케이션의 작동에 대한 자세한 정보를 생성합니다. 애플리케이션의 작동을 추적하고 IBM MQ 자원과 상호작용할 때 애플리케이션에 사용된 매개변수의 상세 보기를 제공합니다. 애플리케이션이 발행한 MQI 호출의 시퀀스도 표시합니다.

amqsrua 명령을 사용하여 시스템 자원 사용 모니터링

amqsrua 명령을 사용하여 큐 관리자의 시스템 자원 사용과 관련한 성능 데이터를 조회할 수 있습니다.

이 태스크 정보

amqsrua 샘플 애플리케이션은 IBM MQ 모니터링 발행물을 이용하는 방법을 보여주고 큐 관리자가 발행하는 성능 데이터를 표시합니다. 이 데이터는 CPU, 메모리, 디스크 사용량에 대한 정보를 포함할 수 있습니다. 또한 STATMQI PCF 통계 데이터와 동등한 데이터를 볼 수도 있습니다. 데이터는 10초마다 발행되며, 명령을 실행하는 동안 데이터가 보고됩니다.


큐 관리자 이름만 사용하여 명령을 실행하고 대화식 스텝 스루를 통해 **CLASS**을(를) 선택한 후 각 단계에서 큐 관리자에 사용 가능한 **TYPE** 및 **object** 매개변수를 선택할 수 있습니다. 정보를 보려는 **CLASS**, **TYPE** 및 **object** 이름을 알고 있는 경우에는 **amqsrua** 명령을 실행할 때 해당 이름을 지정할 수 있습니다.

기본적으로 amqsrua 애플리케이션은 \$SYS/MQ/INFO/QMGR 토픽 트리 아래에서 큐 관리자가 발행한 통계를 검색합니다. 기타 컴포넌트 또는 애플리케이션이 비슷한 메커니즘을 사용하여 다른 토픽 시작점 아래에 발행할 수 있습니다. IBM MQ 9.1.0부터는 **-p** 매개변수를 사용하여 amqsrua가 Linux와 Windows에서 기타 컴포넌트에 대한 통계를 검색하는 위치를 지정할 수 있습니다.

- m**
큐 관리자 이름. 큐 관리자가 실행 중이어야 합니다. 큐 관리자 이름을 지정하지 않으면 기본 큐 관리자에 대한 통계가 표시됩니다.
- c**
CLASS 이름입니다. IBM MQ 자원 사용에 관한 발행물은 클래스와 연관됩니다. 클래스는 가용의 자원 사용 정보를 기술하는 메타데이터 트리의 최상위 레벨을 나타냅니다.
CPU
CPU 사용량에 대한 정보를 리턴합니다.
디스크
디스크 사용량에 대한 정보를 리턴합니다.
STATMQI
MQI 사용량에 대한 정보를 리턴합니다.
STATQ
큐당 MQI 사용량에 대한 정보를 리턴합니다.
STATAPP
지정된 애플리케이션의 사용량 통계에 대한 정보를 리턴합니다.
273 페이지의 『고유 자원 모니터링 애플리케이션 개발』 메타데이터에 대한 토픽 트리를 지정하는 방법에 대한 정보 및 애플리케이션 이름에 사용할 수 있는 문자는 지원되는 프로그래밍 언어로 애플리케이션 이름 사용을 참조하십시오.
참고: -o 옵션을 사용하여 STATAPP에 대해 모니터링 중인 애플리케이션 이름 및 STATQ의 큐 이름을 지정합니다.
- t**
TYPE 이름입니다. IBM MQ 자원 사용에 관한 발행물은 클래스 내의 유형과 연관됩니다. 각 발행물에는 class/type/element 정의를 찾고 결과 발행물을 처리할 수 있도록 하는 클래스 및 유형이 포함되어 있습니다. 큐 관리자 시작 시 class/type/element 설명이 메타데이터로 공개됩니다.
- o**
오브젝트 이름. 자원 사용 발행물은 일련의 PCF 요소로 구성되는 PCF 메시지입니다. 각 클래스 유형 쌍에 대해 공개되는 PCF 요소는 메타데이터로 소개됩니다. 리프는 요소를 처리할 수 있도록 각 요소를 설명하는 트리에 저장됩니다.
- p**
메타데이터 접두부입니다. **amqsrua**가 큐 관리자가 공개하는 통계를 검색할 수 있는 토픽 트리 시작 위치를 지정합니다. 기본 토픽 트리는 \$SYS/MQ/INFO/QMGR 이지만 다른 컴포넌트 또는 애플리케이션은 다른 토픽 트리 시작점에서 통계를 발행할 수 있습니다.
- n**
발행 수입니다. 명령이 끝나기 전에 리턴되는 보고서의 수를 지정할 수 있습니다. 약 10초 단위로 데이터가 공개되므로 50의 값을 입력하면 명령은 500초 동안 50개의 보고서를 리턴합니다. 이 매개변수를 지정하지 않으면 명령은 오류가 발생하거나 큐 관리자가 종료될 때까지 실행됩니다.
- s**
모델 큐입니다. 선택적으로 사용할 모델 큐를 지정하십시오. (기본적으로 **amqsrua** 는 SYSTEM.DEFAULT.MODEL.QUEUE 큐.)
- h**
사용법

프로시저

1. 샘플 디렉토리에서 다음 명령을 발행하여 큐 관리자에 사용 가능한 데이터를 표시할 수 있습니다.

-  Linux의 경우 `MQ_INSTALLATION_PATH/samp/bin:`

```
./amqsrua -m QMgrName
```

- **Windows** Windows의 경우 `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64:`

```
amqsrua -m QMgrName
```

여기서 `QMgrName`은 조회할 큐 관리자의 이름을 지정합니다. 큐 관리자가 실행 중이어야 합니다. 큐 관리자 이름을 지정하지 않으면 기본 큐 관리자가 사용됩니다.

다음 옵션을 사용할 수 있습니다.

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. CLASS 옵션 목록에 STATMQI를 입력하십시오.

```
==> STATMQI
CONNDISC : MQCONN and MQDISC
OPENCLOSE : MQOPEN and MQCLOSE
INQSET : MQINQ and MQSET
PUT : MQPUT
GET : MQGET
SYNCPOINT : Commit and rollback
SUBSCRIBE : Subscribe
PUBLISH : Publish
Enter Type selection
==>
```

3. TYPE 옵션 목록에 PUT을 입력하십시오.

```
==>PUT
Publication received PutDate:20170329 PutTime:17045485 Interval:4 minutes,13.978 seconds
Interval total MQPUT/MQPUT1 count 22
Interval total MQPUT/MQPUT1 byte count 25284 100/sec
Non-persistent message MQPUT count 22
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 25284 100/sec
Put persistent messages - byte count 0
MQSTAT count 0

Publication received PutDate:20170329 PutTime:17050485 Interval:10.001 seconds
Interval total MQPUT/MQPUT1 count 1
Interval total MQPUT/MQPUT1 byte count 524 52/sec
Non-persistent message MQPUT count 1
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 524 52/sec
Put persistent messages - byte count 0
MQSTAT count 0
```

결과

`amqsrua` 샘플 애플리케이션을 대화식으로 사용하여 메타데이터 접두부 `$SYS/MQ/INFO/QMGR`에서 큐 관리자가 시스템 토픽에 대해 발행하는 통계를 검색했습니다.

참고: 큐 관리자에 사용 가능한 자원 발행물의 클래스와 유형은 구성, 버전 및 플랫폼에 따라 다를 수 있습니다. 특정한 큐 관리자에 사용 가능한 클래스, 유형 및 요소를 찾으려면 `amqsrua`를 대화식으로 사용하십시오.

다음에 수행할 작업

사용자 고유의 모니터링 애플리케이션을 개발하려면 273 페이지의 『[고유 자원 모니터링 애플리케이션 개발](#)』의 내용을 참조하십시오.

관련 태스크

[Windows에서 샘플 프로그램 준비 및 실행](#)

[UNIX 및 Linux에서 샘플 프로그램 준비 및 실행](#)

ALW 고유 자원 모니터링 애플리케이션 개발

시스템 자원을 모니터링하는 고유 애플리케이션을 개발할 수 있습니다.

각 큐 관리자는 자원 사용량을 토픽으로 발행합니다. 이 데이터는 토픽 등록자가 사용합니다. 큐 관리자가 시작되면 큐 관리자는 메타 토픽에 대한 메시지 세트를 발행합니다. 이 메시지에서는 큐 관리자에서 지원하는 자원 사용량 토픽 및 해당 토픽으로 발행된 메시지의 콘텐츠를 설명합니다. 관리 도구는 사용 가능한 자원 사용 정보와 해당 토픽을 검색하기 위해 메타데이터에 등록하고, 해당 토픽에서 공개된 토픽에 등록할 수 있습니다.

메타데이터에 대한 토픽 트리의 구조는 다음과 같습니다.

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

토픽 트리의 메타데이터에 대해 다음 구조를 사용하십시오.

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

구조를 사용하여 모니터링되고 있는 자원을 지정하고 있으며 애플리케이션 이름은 최대 28자임을 명료화하고 내부 메타데이터 유형 속성과의 혼동을 피합니다.

가능한 클래스의 목록은 270 페이지의 『amqsrua 명령을 사용하여 시스템 자원 사용 모니터링』의 내용을 참조하십시오.

amqsrua 애플리케이션의 소스 코드는 IBM MQ 샘플로 제공됩니다. 고유 모니터링 애플리케이션을 작성하기 위한 안내서로 이 샘플 애플리케이션을 사용할 수 있습니다. IBM MQ 클라이언트 설치에서 샘플에 대한 소스를 검색할 수 있습니다. 소스 파일의 이름은 amqsruaa.c이고 이 파일은 샘플 디렉토리에 있습니다.

- ▶ **Linux** ▶ **AIX** AIX and Linux 플랫폼의 경우 `MQ_INSTALLATION_PATH/samp/`입니다.
- ▶ **Windows** Windows 플랫폼의 경우 `MQ_INSTALLATION_PATH\tools\c\Samples\`입니다.

amqsrua 애플리케이션은 IBM MQ 자원 사용량 토픽을 구독하고 그 결과로 발행된 PCF 데이터를 형식화합니다. 애플리케이션 소스는 이 유형의 관리 데이터를 구독하고 사용하는 방법에 대한 기본 예를 제공합니다. amqsrua 애플리케이션은 다음 태스크를 완료합니다.

- 입력 매개변수에서 식별한 토픽에 대한 지속 불가능 등록을 작성합니다.
- 토픽에서 메시지를 가져오기 위해 반복적으로 MQGET을 호출하고 stdout에 씁니다.
- MQRC_NONE을 제외한 각 MQI 이유에 대해 메시지를 씁니다.
- MQCC_FAILED의 MQI 완료 코드가 있거나 요청된 수의 자원 사용량 발행이 사용된 경우 중지합니다.

Multi 시스템 주제에 게시된 메트릭

메트릭은 클래스로 분류된 후 하위의 유형으로 분류됩니다. 각각의 메트릭 클래스 및 유형으로 게시된 다양한 메트릭이 존재합니다.

색인

- 274 페이지의 『CPU(플랫폼 중앙 처리 장치)』
- 275 페이지의 『디스크(플랫폼 지속적 데이터 저장소)』
- 275 페이지의 『STATMQI(API 사용량 통계)』
- 278 페이지의 『STATQ(API 큐별 사용량 통계)』
- 279 페이지의 『STATAPP(애플리케이션별 사용량 통계)』
- 279 페이지의 『NHAREPLICA (인스턴스당 기본 HA 통계)』

NHAREPLICA를 제외하고 나열된 옵션에 대한 데이터를 수집하는 방법에 대한 정보는 270 페이지의 『amqsrua 명령을 사용하여 시스템 자원 사용 모니터링』의 내용을 참조하십시오.

ALTER QMGR 명령을 사용하여 큐 관리자 레벨에서 STATMQI 및 STATQ를 모니터링하거나 로컬 큐 속성 STATQ를 사용하여 개별 큐를 모니터링할 수도 있습니다. 이 옵션은 ALTER QUEUES를 참조하십시오.

CPU(플랫폼 중앙 처리 장치)

소개

여기서 통계는 현재 간격을 참조하며, 이는 MQIAMO64_MONITOR_INTERVAL 매개변수에 의해 발행된 메시지에 정의된 간격입니다.

통계는 일반적으로 하나 이상의 활성 구독자가 있는 한 10초마다 발행 간격으로 발행되지만 정확한 간격은 항상 메시지에서 가져와야 합니다.

중요사항: 달리 지정되지 않는 한, 메트릭은 캡처될 때 특정 시점에 대한 절대값입니다.

시스템 요약(CPU 성능 - 플랫폼 전체)

사용자 CPU 시간 백분율 X%

권한이 없는 코드에 있을 때 CPU에서 사용한 시간의 평균 백분율 (마지막 10초간격 동안 소요됨)입니다.

시스템 CPU 시간 백분율 X%

CPU가 권한 코드에 있을 때 사용한 시간의 평균 백분율 (마지막 10초간격 동안 소요됨)입니다.

CPU 로드 - 1분 평균 X

1분로드 평균입니다. "로드 평균"은 업계 전체 용어이지만 보고된 정확한 값은 플랫폼마다 다를 수 있습니다.

CPU 로드 - 5분 평균 X

5분로드 평균입니다. "로드 평균"은 업계 전체 용어이지만 보고된 정확한 값은 플랫폼마다 다를 수 있습니다.

CPU 로드 - 15분 평균 X

15분로드 평균입니다. "로드 평균"은 업계 전체 용어이지만 보고된 정확한 값은 플랫폼마다 다를 수 있습니다.

CPU 시스템 요약

RAM 사용 가능 백분율 X%

RAM 합계 바이트 XMB

RAM 사용 가능 백분율 X%

RAM 합계 바이트 XMB

QMGR 요약(CPU 성능 - 실행 중인 큐 관리자)

사용자 CPU 시간 - 큐 관리자에 대한 백분율 예상 X%

이 큐 관리자의 프로세스가 권한이 없는 코드에 있을 때 CPU에서 사용한 시간의 평균 백분율 (최근 10초간격 동안 소요됨)입니다.

시스템 CPU 시간 - 큐 관리자에 대한 백분율 예상 X%

이 큐 관리자의 프로세스가 권한 코드에 있을 때 CPU에서 사용한 시간의 평균 백분율 (최근 10초간격 동안 소요됨)입니다.

RAM 합계 바이트 - 큐 관리자에 대한 예상 XMB

이는 큐 관리자가 사용하는 메모리의 근사치입니다.

디스크(플랫폼 지속적 데이터 저장소)

SystemSummary 및 *QMGrSummary* 는 캡처 시점의 절대값입니다. 공개 간격에 대한 세부사항은 [소개](#) 를 참조하십시오.

시스템 요약(디스크 사용량 - 플랫폼 전체)

MQ 오류 파일 시스템 - 사용 중인 바이트 XMB

MQ 오류 파일 시스템 - 여유 공간 X%

MQ FDC 파일 개수 X

MQ 추적 파일 시스템 - 사용 중인 바이트 XMB

MQ 추적 파일 시스템 - 여유 공간 X%

QMGr 요약(디스크 사용량 - 실행 중인 큐 관리자)

큐 관리자 파일 시스템 - 사용 중인 바이트 XMB

큐 관리자 파일 시스템 - 여유 공간 X%

로그(디스크 사용량 - 큐 관리자 복구 로그)

로그 - 사용 중인 바이트 X

로그 - 최대 바이트 X

모든 1차 및 2차 익스텐트가 가득 찬 경우 로그에 기록할 수 있는 최대 바이트 수입니다. 로그 파일 시스템 크기보다 작습니다.

로그 파일 시스템 - 사용 중인 바이트 X

로그 파일 시스템 - 최대 바이트 X

로그-현재 간격 X동안 기록된 물리적 바이트입니다.

현재 간격의 정의는 [소개](#) 를 참조하십시오.

현재 간격 X에 대해 기록된 로그 논리 바이트

로그 - 쓰기 지연 Xu초

디스크에 대한 단일 쓰기에 걸리는 시간을 나타내는 롤링 평균입니다.

여기서 **LogWriteIntegrity=TripleWrite**인 경우, 디스크에 기록된 실제 바이트 수는 기록된 논리 바이트 수보다 큽니다.

로그 쓰기 크기 X, 롤링 평균.

로그-아카이브 대기 중인 익스텐트 X가 차지합니다.

logtype= linear 및 **LogManagement = archive**인 경우에만 발행됩니다. 자세한 정보는 [qm.ini 파일의 로그 스탠자](#) 를 참조하십시오.

매체 복원 X에 필요한 로그 공간 (MB) 입니다.

logtype= linear인 경우에만 공개됩니다.

재사용 가능한 익스텐트 X가 차지하는 로그 스페이스 (MB)

logtype= linear 및 **LogManagement = automatic**인 경우에만 발행됩니다. 자세한 정보는 [qm.ini 파일의 로그 스탠자](#) 를 참조하십시오.

로그-현재 사용 중인 1차 스페이스 X%.

1차 로그의 백분율로 사용 중인 로그 파일 공간입니다. 이 값은 100%보다 클 수 있습니다.

로그 워크로드 1차 스페이스 활용도 X%.

최근 히스토리에서 롤링 평균으로 사용 중인 로그 파일 공간 백분율입니다.

STATMQI(API 사용량 통계)

모든 API 사용 통계는 공개된 간격의 발생 및/또는 백분율을 반영합니다. 공개 간격의 정의는 [소개](#) 를 참조하십시오.

통계는 실패한 MQI 호출 수의 계수를 출력하지만 모든 실패한 MQI 호출이 이러한 통계에 표시되는 것은 아닙니다. 실제로 모든 MQI 호출의 실패에는 통계가 기록되지 않습니다. 이는 MQI 호출이 통계가 기록되는 큐 관리자의 내부에 도달하기 전에 MQI 호출이 실패하는 많은 이유가 진단되기 때문입니다.

이에 대한 예는 클라이언트 애플리케이션에 리턴된 MQRC_HCONN_ERROR입니다. 클라이언트 애플리케이션이 잘못된 hconn를 전달하는 경우 MQ 클라이언트는 해당 오류를 진단하고 MQI 호출을 큐 관리자에 전달하지 않고 MQRC_HCONN_ERROR를 리턴합니다. 따라서 실패한 MQI 호출은 큐 관리자가 기록한 통계에 표시되지 않습니다.

실패한 MQI 호출의 통계는 고객이 불필요하게 실패한 MQI 호출을 생성하여 성능에 영향을 주는 잘못 작성된 애플리케이션의 문제점을 해결할 수 있도록 하므로 유용합니다. 통계에 기록되는 다양한 MQI 호출에 대한 실패 이유의 일부 예는 다음과 같습니다.

- MQCONN/MQCONN/MQOPEN은 클라이언트가 아닌 큐 관리자가 진단할 때 2035 MQRC_NOT_AUTHORIZED를 리턴합니다. 예를 들어, **amqsput** 를 nobody로 실행합니다.
- MQPUT/MQPUT1 은 MAXDEPTH가 초과되었으므로 2053 MQRC_Q_FULL을 리턴합니다.
- MQGET은 비어 있는 큐에서 찾아보거나 파괴적으로 가져오는 경우 2033 MQRC_NO_MSG_AVAILABLE을 리턴합니다.
- 보유된 메시지가 없으므로 MQSUBRQ가 2437 MQRC_NO_RETAINED_MSG를 리턴합니다.

CONNDISC(MQCONN 및 MQDISC)

MQCONN/MQCONNX 개수 X

실패한 MQCONN/MQCONNX 개수 X

동시 연결 - 하이 워터 마크 X

MQDISC 개수 X

OPENCLOSE(MQOPEN 및 MQCLOSE)

MQOPEN 개수 X Y/초

실패한 MQOPEN 개수 X

MQCLOSE 개수 X Y/초

실패한 MQCLOSE 개수 X

INQSET(MQINQ 및 MQSET)

MQINQ 개수 X

실패한 MQINQ 개수 X

MQSET 개수 X

실패한 MQSET 개수 X

PUT(MQPUT)

간격 전체 MQPUT/MQPUT1 개수 X

간격 전체 MQPUT/MQPUT1 바이트 수 X Y/초

비지속 메시지 MQPUT 개수 X

지속 메시지 MQPUT 개수 X

실패한 MQPUT 개수 X

비지속 메시지 MQPUT1 개수 X

지속 메시지 MQPUT1 개수 X

실패한 MQPUT1 개수 X

비지속 메시지 넣기 - 바이트 수 X Y/초

지속 메시지 넣기 - 바이트 수 X

MQSTAT 개수 X

GET(MQGET)

간격 전체 파괴적 가져오기 - 개수 X

간격 전체 파괴적 가져오기 - 바이트 수 X Y/초

비지속 메시지 파괴적 가져오기 - 개수 X
지속 메시지 파괴적 가져오기 - 개수 X
실패한 MQGET - 개수 X
비지속 메시지 가져오기 - 바이트 수 X Y/초
지속 메시지 가져오기 - 바이트 수 X
비지속 메시지 찾아보기 - 개수 X
지속 메시지 찾아보기 - 개수 X
실패한 찾아보기 개수 X
비지속 메시지 찾아보기 - 바이트 수 X Y/초
지속 메시지 찾아보기 - 바이트 수 X
만료된 메시지 개수 X
삭제된 큐 개수 X
MQCB 개수 X
실패한 MQCB 개수 X
MQCTL 개수 X

SYNCPOINT(커미트 및 롤백)

커미트 개수 X
롤백 개수 X

SUBSCRIBE(구독)

지속 가능 구독 작성 개수 X
지속 가능 구독 변경 개수 X
지속 가능 구독 재개 개수 X
지속 불가능 구독 작성 개수 X
실패한 구독 작성/변경/재개 개수 X
지속 가능 구독 삭제 개수 X
지속 불가능 구독 삭제 개수 X
구독 삭제 실패 개수 X
MQSUBRQ 개수 X
실패한 MQSUBRQ 개수 X
지속 가능 구독자 - 하이 워터 마크 X
지속 가능 구독자 - 로우 워터 마크 X
지속 불가능 구독자 - 하이 워터 마크 X
지속 불가능 구독자 - 로우 워터 마크 X

PUBLISH(공개)

주제 MQPUT/MQPUT1 간격 합계 X
간격 전체 주제 바이트 넣기 X Y/초
구독자에게 공개 - 메시지 개수 X
구독자에게 공개 - 바이트 수 X
비지속적 - 주제 MQPUT/MQPUT1 개수 X
지속적 - 주제 MQPUT/MQPUT1 개수 X
실패한 주제 MQPUT/MQPUT1 개수 X

STATQ(API 큐별 사용량 통계)

GENERAL(일반)

- 메시지 만료 X (IBM MQ 9.3.0 이상 CD 버전의 경우 [GET](#) 에서 이동됨)
- 큐 제거 수 X (IBM MQ 9.3.0 이상 CD 버전의 경우 [GET](#) 에서 이동됨)
- 평균 큐 시간 X uSec (IBM MQ 9.3.0 이상 CD 버전의 경우 [GET](#) 에서 이동됨)
- 큐 깊이 X (IBM MQ 9.3.0 이상 CD 버전의 경우 [GET](#) 에서 이동됨)

OPENCLOSE(MQOPEN 및 MQCLOSE)

- MQOPEN 개수 X
- MQCLOSE 개수 X

INQSET(MQINQ 및 MQSET)

- MQINQ 개수 X
- MQSET 개수 X

PUT(MQPUT 및 MQPUT1)

- MQPUT/MQPUT1 개수 X
- MQPUT 바이트 수 X
- MQPUT 비지속 메시지 개수 X
- MQPUT 지속 메시지 개수 X
- 롤백된 MQPUT 개수 X
- MQPUT1 비지속 메시지 개수 X
- MQPUT1 지속 메시지 개수 X
- 비지속적 바이트 수 X
- 지속적 바이트 수 X
- 잠금 경합 $X\%$

다른 프로세스가 잠금을 먼저 해제할 때까지 대기하게 된 큐를 잠그려는 시도의 백분율입니다. 현재 잠겨 있지 않은 잠금을 취하는 것이 잠금이 해제되기를 기다리는 것보다 더 효율적이기 때문에 잠금 경합을 줄이면 시스템의 최대 처리량이 증가할 수 있습니다.

큐 회피 넣기 $X\%$

대기 중인 Getter가 있을 때 메시지를 큐에 넣으면 메시지를 즉시 Getter로 전달할 수 있으므로 메시지를 큐에 넣을 필요가 없습니다. 따라서 이 메시지는 큐를 사용하지 않았다고 하며, "큐 방지 넣기" 는 이러한 메시지의 수입니다. 큐 회피를 늘리면 메시지를 큐에 넣고 다시 가져오는 비용을 피하므로 시스템의 최대 처리량이 증가할 수 있습니다.

큐 회피 바이트 $X\%$

대기 중인 Getter가 있을 때 메시지를 큐에 넣으면 메시지를 즉시 Getter로 전달할 수 있으므로 메시지를 큐에 넣을 필요가 없습니다. 따라서 이 메시지는 큐를 사용하지 않았다고 하며 "큐 방지 바이트" 는 이러한 바이트의 수입니다. 큐 회피를 늘리면 메시지를 큐에 넣고 다시 가져오는 비용을 피하므로 시스템의 최대 처리량이 증가할 수 있습니다.

GET(MQGET)

- MQGET 개수 X
- MQGET 바이트 수 X
- 파괴적 MQGET 비지속 메시지 개수 X
- 파괴적 MQGET 지속 메시지 개수 X
- 롤백된 MQGET 개수 X
- 파괴적 MQGET 비지속적 바이트 수 X
- 파괴적 MQGET 지속적 바이트 수 X
- MQGET 찾아보기 비지속 메시지 개수 X
- MQGET 찾아보기 지속 메시지 개수 X

MQGET 찾아보기 비지속적 바이트 수 X
 MQGET 찾아보기 지속적 바이트 수 X
 만료된 메시지 X (IBM MQ 9.3에서 GENERAL 로 이동됨)
 큐 제거된 개수 X (IBM MQ 9.3에서 GENERAL 로 이동됨)
 평균 큐 시간 X uSec (IBM MQ 9.3에서 GENERAL 로 이동됨)
 큐 용량 X (IBM MQ 9.3에서 GENERAL 로 이동됨)
 파괴적 MQGET 실패 X
 파괴적 MQGET 실패(MQRC_NO_MSG_AVAILABLE) X
 파괴적 MQGET 실패(MQRC_TRUNCATED_MSG_FAILED) X
 MQGET 찾아보기 실패 X
 MQGET 찾아보기 실패(MQRC_NO_MSG_AVAILABLE) X
 MQGET 찾아보기 실패(MQRC_TRUNCATED_MSG_FAILED) X

STATAPP(애플리케이션별 사용량 통계)

INSTANCE(인스턴스 통계)

인스턴스 수 X 절대
 이동 가능 인스턴스 수 X 절대
 인스턴스 부족분 수 X 절대
 시작된 인스턴스 X 간격
 시작된 아웃바운드 인스턴스 이동 X 간격
 완료된 아웃바운드 인스턴스 이동 X 간격
 다시 연결 중에 종료된 인스턴스 X 간격
 종료된 인스턴스 X 간격

NHAREPLICA (인스턴스당 기본 HA 통계)

REPLICATION (복제 통계)

보낸 동기 로그 바이트 X
 보낸 캐치업 로그 바이트 X
 ▶ V 9.4.0 전송된 동기 압축 로그 바이트 X
 ▶ V 9.4.0 전송된 캐치업 압축 로그 바이트 X
 ▶ V 9.4.0 전송된 동기 압축되지 않은 로그 바이트 X
 ▶ V 9.4.0 전송된 압축되지 않은 로그 바이트 캐치업 X
 로그 쓰기 평균 수신확인 대기 시간 X uSec
 로그 쓰기 평균 수신확인 크기 X
 백로그 바이트 X
 백로그 평균 바이트 X

관련 정보

IBM MQ Operator 사용 시 공개된 매트릭

실시간 모니터링

실시간 모니터링은 큐 관리자 내 채널과 큐의 현재 상태를 판별할 수 있게 하는 기술입니다. 명령이 발행될 때 리턴되는 정보는 정확합니다.

큐와 채널에 대한 리턴 실시간 정보가 발행될 때 여러 명령이 사용 가능합니다. 하나 이상의 큐 또는 채널에 대한 정보가 리턴될 수 있으며 정보의 양은 차이가 있을 수 있습니다. 다음 태스크에서 실시간 모니터링을 사용할 수 있습니다.

- 시스템 관리자가 IBM MQ 시스템의 정상 상태를 이해하도록 지원. 시스템에서 문제가 발생할 경우 이는 문제점 진단에 유용합니다.
- 특정 이벤트나 문제점이 감지되지 않았어도 어느 시점에나 큐 관리자의 조건 판별.
- 시스템의 문제점 원인 판별 지원.

실시간 모니터링으로 큐나 채널에 대한 정보가 리턴될 수 있습니다. 리턴되는 실시간 정보의 양은 큐 관리자, 큐, 채널 속성으로 제어합니다.

- 명령을 발행하여 큐를 모니터해서 큐가 제대로 서비스하는지 확인합니다. 일부 큐 속성을 사용하려면 실시간 모니터링에 해당 속성을 사용 가능하게 해야 합니다.
- 명령을 발행하여 채널을 모니터해서 채널이 제대로 실행 중인지 확인합니다. 일부 채널 속성을 사용하려면 실시간 모니터링에 해당 속성을 사용 가능하게 해야 합니다.

큐 및 채널에 대한 실시간 모니터링은 성능 및 채널 이벤트 모니터링 외의 것으로 이와는 별도로입니다.

실시간 모니터링을 제어하는 속성

실시간 모니터링이 사용되면 일부 큐 및 채널 상태 속성은 모니터링 정보를 보유합니다. 실시간 모니터링이 사용되지 않는 경우에는 이 모니터링 속성에 모니터링 정보가 보유되지 않습니다. 아래의 예는 이러한 큐 및 채널 상태 속성의 사용 방법을 보여줍니다.

개별 큐 또는 채널이나 여러 큐 또는 채널에 대해서 실시간 모니터링을 사용 가능 또는 불가능하게 할 수 있습니다. 개별 큐나 채널을 제어하려면 큐 속성 MONQ 또는 채널 속성 MONCHL을 설정하여 실시간 모니터링을 사용 가능 또는 불가능하게 하십시오. 여러 큐나 채널을 함께 제어하려면 큐 관리자 속성 MONQ 및 MONCHL을 사용하여 큐 관리자 레벨에서 실시간 모니터링을 사용 가능 또는 불가능하게 하십시오. 기본값이 지정된 모니터링 속성이 있는 모든 큐와 채널 오브젝트의 경우 큐 관리자 레벨에서 실시간 모니터링이 제어됩니다.

자동으로 정의된 클러스터 송신자 채널은 IBM MQ 오브젝트가 아니므로 채널 오브젝트와 동일한 방식으로 속성을 보유하지 않습니다. 자동으로 정의된 클러스터 송신자 채널을 제어하려면 큐 관리자 속성, MONACLS를 사용하십시오. 이 속성은 큐 관리자 내의 자동으로 정의된 클러스터 송신자 채널이 채널 모니터링에 대해 사용 가능 또는 불가능한지 여부를 판별합니다.

채널 실시간 모니터링의 경우 MONCHL 속성을 세 개의 모니터링 레벨 낮음, 중간 또는 높음 중 하나로 설정할 수 있습니다. 오브젝트 레벨 또는 큐 관리자 레벨에서 모니터링 레벨을 설정할 수 있습니다. 레벨의 선택은 시스템에 따라 다릅니다. 모니터링 데이터를 수집하려면 시스템 시간 확보와 같이 계산 비용이 상대적으로 값비싼 몇 가지 명령어가 필요할 수 있습니다. 실시간 모니터링의 영향을 줄이기 위해 중간 및 낮음 모니터링 옵션은 데이터를 항상 수집하기 보다는 데이터 샘플을 일정한 간격으로 측정합니다. 280 페이지의 표 30에는 채널의 실시간 모니터링에 사용 가능한 모니터링 레벨이 요약되어 있습니다.

표 30. 모니터링 레벨		
레벨	설명	사용법
낮음	일정한 간격으로 작은 데이터 샘플을 측정합니다.	높은 볼륨의 메시지를 처리하는 오브젝트의 경우.
중간	일정한 간격으로 데이터 샘플을 측정합니다.	대부분의 오브젝트의 경우.
높음	일정한 간격으로 모든 데이터를 측정합니다.	초당 몇 개의 메시지만을 처리하는 오브젝트의 경우(최신 정보가 중요함).

큐 실시간 모니터링의 경우에는 MONQ 속성을 세 개의 모니터링 레벨 낮음, 중간 또는 높음 중 하나로 설정할 수 있습니다. 하지만 이들 값의 차이가 뚜렷하지 않습니다. 모든 값이 데이터 콜렉션을 사용하지만 샘플 크기에 영향을 미치지 않습니다.

예

다음 예는 모니터링 레벨을 제어하기 위해 필요한 큐, 채널, 큐 관리자 속성의 설정 방법을 보여줍니다. 모든 예에서, 모니터링이 사용될 때 큐와 채널 오브젝트의 모니터링 레벨은 중간입니다.

1. 큐 관리자 레벨에서 모든 큐와 채널에 대해 큐 및 채널 모니터링을 모두 사용하려면 다음 명령을 사용하십시오.

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

2. 로컬 큐, Q1 및 송신자 채널, QM1.TO.QM2를 제외하고 모든 큐와 채널에 모니터링을 사용하려면 다음 명령을 사용하십시오.

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

3. 로컬 큐, Q1 및 송신자 채널, QM1.TO.QM2를 제외하고 모든 큐와 채널에 대해 큐 및 채널 모니터링을 모두 사용하지 않으려면 다음 명령을 사용하십시오.

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

4. 개별 오브젝트 속성과 무관하게, 모든 큐와 채널에 대해 큐 및 채널 모니터링을 모두 사용하지 않으려면 다음 명령을 사용하십시오.

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

5. 자동으로 정의된 클러스터 송신자 채널의 모니터링 기능을 제어하려면 다음 명령을 사용하십시오.

```
ALTER QMGR MONACLS(MEDIUM)
```

6. 자동으로 정의된 클러스터 송신자 채널에 채널 모니터링을 위한 큐 관리자 설정을 사용하도록 지정하려면 다음 명령을 사용하십시오.

```
ALTER QMGR MONACLS(QMGR)
```

관련 개념

[279 페이지의 『실시간 모니터링』](#)

실시간 모니터링은 큐 관리자 내 채널과 큐의 현재 상태를 판별할 수 있게 하는 기술입니다. 명령이 발행될 때 리턴되는 정보는 정확합니다.

[298 페이지의 『Using IBM MQ online monitoring』](#)

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

관련 태스크

[282 페이지의 『큐와 채널 모니터링 데이터 표시』](#)

큐나 채널에 대한 실시간 모니터링 정보를 표시하려면 IBM MQ Explorer 또는 해당 MQSC 명령을 사용하십시오. 일부 모니터링 필드는 큐 관리자의 조작을 모니터링하는 데 유용한 심표로 구분된 지표 값 쌍을 표시합니다. 아래의 예는 모니터링 데이터의 표시 방법을 보여줍니다.

[큐 관리자 속성 표시 및 대체
모니터링\(MONCHL\)](#)

큐와 채널 모니터링 데이터 표시

큐나 채널에 대한 실시간 모니터링 정보를 표시하려면 IBM MQ Explorer 또는 해당 MQSC 명령을 사용하십시오. 일부 모니터링 필드는 큐 관리자의 조작을 모니터링하는 데 유용한 심표로 구분된 지표 값을 표시합니다. 아래의 예는 모니터링 데이터의 표시 방법을 보여줍니다.

이 태스크 정보

심표로 구분된 값 쌍을 표시하는 모니터링 필드는 오브젝트에 대한 모니터링이 사용된 이후에 또는 큐 관리자가 시작되었을 때부터 측정된 시간의 단기 및 장기 지표를 제공합니다.

- 단기 지표는 쌍의 첫 번째 값이며 최신 측정에 더 높은 가중치를 주어서 이 값에 대한 영향을 더 크게 하는 방식으로 계산됩니다. 수행한 측정의 최근 추세를 표시합니다.
- 쌍에서 두 번째 값의 장기 지표는 최신 측정에 더 높은 가중치를 주지 않는 방식으로 계산됩니다. 자원의 성능에 대한 보다 장기적 활동을 표시합니다.

이 지표 값은 큐 관리자의 조작 변경사항을 감지하는 데 있어서 가장 유용합니다. 시간의 증가를 감지하려면 정상 사용일 때 지표가 표시하는 시간을 알아야 합니다. 이 값을 정기적으로 수집하고 확인해서 큐 관리자의 작동 변동을 감지할 수 있습니다. 이는 성능의 변화를 나타내는 것일 수 있습니다.

다음과 같이 실시간 모니터링 정보를 확보하십시오.

프로시저

1. 큐에 대한 실시간 모니터링 정보를 표시하려면 IBM MQ Explorer 또는 선택적 매개변수 MONITOR를 지정하여 MQSC 명령 DISPLAY QSTATUS를 사용하십시오.
2. 채널에 대한 실시간 모니터링 정보를 표시하려면 IBM MQ Explorer 또는 선택적 매개변수 MONITOR를 지정하여 MQSC 명령 DISPLAY CHSTATUS를 사용하십시오.

예

큐 Q1에는 기본값 QMGR로 설정된 속성 MONQ가 있고, 해당 큐를 소유한 큐 관리자에는 MEDIUM으로 설정된 속성 MONQ가 있습니다. 이 큐에 대해 수집된 모니터링 필드를 표시하려면 다음 명령을 사용하십시오.

```
DISPLAY QSTATUS(Q1) MONITOR
```

큐 Q1의 모니터링 필드 및 모니터링 레벨은 다음과 같이 표시됩니다.

```
QSTATUS(Q1)
TYPE(Queue)
MONQ(MEDIUM)
QTIME(11892157,24052785)
MSGAGE(37)
LPUTDATE(2005-03-02)
LPUTTIME(09.52.13)
LGETDATE(2005-03-02)
LGETTIME(09.51.02)
```

송신자 채널 QM1.TO.QM2에는 기본값 QMGR로 설정된 속성 MONCHL이 있고, 해당 큐를 소유한 큐 관리자에는 MEDIUM으로 설정된 속성 MONCHL이 있습니다. 이 채널에 대해 수집된 모니터링 필드를 표시하려면 다음 명령을 사용하십시오.

```
DISPLAY CHSTATUS(QM1.TO.QM2) MONITOR
```

송신자 채널 QM1.TO.QM2의 모니터링 필드 및 모니터링 레벨은 다음과 같이 표시됩니다.

```
CHSTATUS(QM1.TO.QM2)
XMITQ(Q1)
CONNNAME(127.0.0.1)
CURRENT
CHLTYPE(SDR)
STATUS(RUNNING)
```

```
SUBSTATE (MQGET)
MONCHL (MEDIUM)
XQTIME (755394737,755199260)
NETTIME (13372,13372)
EXITTIME (0,0)
XBATCHSZ (50,50)
COMPTIME (0,0)
STOPREQ (NO)
RQMNAME (QM2)
```

관련 개념

279 페이지의 『실시간 모니터링』

실시간 모니터링은 큐 관리자 내 채널과 큐의 현재 상태를 판별할 수 있게 하는 기술입니다. 명령이 발행될 때 리턴되는 정보는 정확합니다.

관련 참조

[큐 상태 표시](#)

큐 모니터링

이 페이지는 큐와 이 큐를 서비스하는 애플리케이션에 대한 문제점을 해결하는 데 도움을 줍니다. 다양한 모니터링 옵션을 사용하여 문제점을 판별할 수 있습니다.

서비스하는 큐에 대한 문제점의 첫 번째 신호는 종종 큐의 메시지 수(CURDEPTH) 증가입니다. 하루의 특정 시점이나 특정 워크로드 하에서 증가를 예상하는 경우 메시지 수 증가는 문제를 나타내는 것이 아닐 수 있습니다. 하지만 메시지 수 증가를 설명할 수 없으면 원인을 조사해야 할 수 있습니다.

애플리케이션에 대한 문제가 있는 애플리케이션 큐나 채널에 대한 문제가 있는 전송 큐가 있을 수 있습니다. 큐를 서비스하는 애플리케이션이 채널일 때 추가 모니터링 옵션이 사용 가능합니다.

다음 예는 Q1이라는 특정 큐에 대한 문제점을 조사하고 다양한 명령 출력에 표시되는 필드를 설명합니다.

애플리케이션이 큐를 여는지 판별

큐에 대한 문제가 있는 경우 애플리케이션이 큐를 여는지 여부를 확인하십시오.

이 태스크 정보

다음 단계를 수행하여 애플리케이션이 큐를 여는지 판별하십시오.

프로시저

1. 큐에 대해 실행 중인 애플리케이션이 예상한 애플리케이션인지 확인하십시오. 문제의 큐에 대해 다음 명령을 실행하십시오.

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

출력에서 APPLTAG 필드를 보고 애플리케이션의 이름이 표시되는지 확인하십시오. 애플리케이션의 이름이 표시되지 않은 경우나 출력이 전혀 없는 경우 애플리케이션을 시작하십시오.

2. 큐가 전송 큐이면 CHANNEL 필드에서 출력을 보십시오.
채널 이름이 CHANNEL 필드에 표시되지 않은 경우에는 채널이 실행 중인지 판별하십시오.
3. 큐에 대해 실행 중인 애플리케이션이 입력을 위한 큐를 열었는지 확인하십시오. 다음 명령을 실행하십시오.

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

출력에서 IPPROCS 필드를 보고 애플리케이션이 입력을 위한 큐를 열었는지 확인하십시오. 값이 0이고 큐가 애플리케이션 큐이면 애플리케이션이 큐에서 메시지를 가져오기 위해 입력을 위한 큐를 여는지 확인하십시오.

큐의 메시지가 사용 가능한지 확인

큐에 수많은 메시지가 있고 애플리케이션이 이 메시지를 처리하지 않는 경우 큐의 메시지가 애플리케이션에 사용 가능한지 확인하십시오.

이 태스크 정보

다음 단계를 수행하여 애플리케이션이 큐의 메시지를 처리하지 않는 이유를 조사하십시오.

프로시저

1. 애플리케이션이 큐의 모든 메시지를 처리해야 할 때 특정 메시지 ID 또는 상관 ID를 요구하지 않는지 확인하십시오.
2. 큐의 현재 용량이 큐의 메시지 수가 증가하고 있음을 표시하지만, 큐의 일부 메시지는 커밋되지 않아서 애플리케이션이 가져오지 못할 수 있습니다(큐의 현재 용량은 큐에 대한 메시지의 커밋되지 않은 MQPUT 수를 포함함). 다음 명령을 실행하십시오.

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

출력에서 UNCOM 필드를 보고 큐에 커밋되지 않은 메시지가 있는지 확인하십시오.

3. 애플리케이션이 큐에서 메시지를 가져오려 시도 중인 경우에는 넣고 있는 애플리케이션이 메시지를 제대로 커밋하는지 확인하십시오. 다음 명령을 발행하여 이 큐에 메시지를 넣고 있는 애플리케이션의 이름을 찾으십시오.

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) OPENTYPE(OUTPUT)
```

4. 그리고 나서 이전 명령 출력의 APPLTAG 값을 *appltag*에 삽입하여 다음 명령을 실행하십시오.

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

그러면 작업 단위가 언제 시작되었는지 표시되어 애플리케이션이 장기 실행 작업 단위를 작성 중인지 여부를 파악하는 데 도움이 됩니다. 넣고 있는 애플리케이션이 채널이면 배치를 완료하는 데 시간이 오래 걸리는 이유를 조사할 수 있습니다.

애플리케이션이 큐에서 메시지를 가져오는지 확인

큐와 이 큐를 서비스하는 애플리케이션에 문제가 있는 경우 애플리케이션이 큐에서 메시지를 가져오는지 여부를 확인하십시오.

이 태스크 정보

애플리케이션이 큐에서 메시지를 가져오는지 확인하려면 다음 검사를 수행하십시오.

프로시저

1. 큐에 대해 실행 중인 애플리케이션이 큐에서 실제로 메시지를 처리 중인지 확인하십시오. 다음 명령을 실행하십시오.

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

출력에서, 큐로부터 마지막 가져오기를 수행할 때 표시되는 LGETDATE 및 LGETTIME 필드를 보십시오.

2. 이 큐에서의 마지막 가져오기가 예상보다 오래 전이면 애플리케이션이 메시지를 제대로 처리하고 있는지 확인하십시오.

애플리케이션이 채널이면 메시지가 이 채널을 통해 이동하는지 확인하십시오.

애플리케이션이 메시지를 충분히 빨리 처리할 수 있는지 판별

메시지가 큐에 빌드되고 있지만 다른 검사에서 처리 문제점이 발견되지 않은 경우에는 애플리케이션이 메시지를 충분히 빨리 처리할 수 있는지 확인하십시오. 애플리케이션이 채널이면 채널이 메시지를 충분히 빠르게 처리할 수 있는지 확인하십시오.

이 태스크 정보

애플리케이션이 메시지를 빠르게 처리 중인지 판별하려면 다음 테스트를 수행하십시오.

프로시저

1. 다음 명령을 주기적으로 발행하여 큐에 대한 성능 데이터를 수집하십시오.

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

QTIME 표시기의 값이 높거나 일정 기간에 증가하고 있으며 큐의 메시지가 사용 가능한지 확인해서 작업 단위의 장기 실행 가능성을 이미 배제했다면, 가져오기 애플리케이션이 넣기 애플리케이션을 따라잡지 못하는 것일 수 있습니다.

2. 가져오기 애플리케이션이 넣기 애플리케이션을 따라잡지 못하는 경우 큐를 처리할 다른 가져오기 애플리케이션을 추가할 것을 고려하십시오.

다른 가져오기 애플리케이션을 추가할 수 있는지 여부는 애플리케이션의 설계와 둘 이상의 애플리케이션이 큐를 공유할 수 있는지 여부에 따라 다릅니다. 두 개의 애플리케이션이 한 큐를 동시에 처리할 수 있게 하려면 상관 ID별 메시지 가져오기나 그룹화와 같은 기능이 유용할 수 있습니다.

현재 용량이 증가하지 않을 때 큐 확인

큐의 현재 용량이 증가하지 않아도 큐를 모니터링하여 애플리케이션이 메시지를 제대로 처리 중인지 확인하는 데 여전히 유용할 수 있습니다.

이 태스크 정보

큐에 대한 성능 데이터를 수집하려면 다음 명령을 주기적으로 발행하십시오.

프로시저

다음 명령을 주기적으로 발행하십시오.

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

출력에서, MSGAGE의 값이 일정 기간에 증가하며 애플리케이션이 모든 메시지를 처리하도록 설계된 경우 이는 일부 메시지가 전혀 처리되고 있지 않음을 나타내는 것일 수 있습니다.

채널 모니터링

이 페이지를 사용하여 전송 큐와 이 큐를 서비스하는 채널에 대한 문제점을 해결하는 데 도움이 되는 태스크를 봅니다. 다양한 채널 모니터링 옵션을 사용하여 문제점을 판별할 수 있습니다.

서비스하는 큐에 대한 문제점의 첫 번째 신호는 종종 큐의 메시지 수(CURDEPTH) 증가입니다. 하루의 특정 시점이나 특정 워크로드 하에서 증가를 예상하는 경우 메시지 수 증가는 문제를 나타내는 것이 아닐 수 있습니다. 하지만 메시지 수 증가를 설명할 수 없으면 원인을 조사해야 할 수 있습니다.

전송 큐를 서비스하는 채널에 문제가 있을 수 있습니다. 다양한 채널 모니터링 옵션을 사용하여 문제점을 판별할 수 있습니다.

다음 예는 전송 큐 QM2 및 채널 QM1.TO.QM2에 대한 문제점을 조사합니다. 이 채널은 큐 관리자 QM1에서 큐 관리자 QM2로 메시지를 송신하는 데 사용됩니다. 큐 관리자 QM1의 채널 정의는 송신자 또는 서버 채널이고 큐 관리자 QM2의 채널 정의는 수신자 또는 요청자 채널입니다.

채널이 실행 중인지 판별

전송 큐에 대한 문제가 있는 경우 채널이 실행 중인지 여부를 확인하십시오.

이 태스크 정보

다음 단계를 수행하여 전송 큐를 서비스하는 채널의 상태를 확인하십시오.

프로시저

1. 다음 명령을 발행하여 전송 큐 QM2를 처리할 것으로 예상하는 채널을 찾으십시오.

```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```

이 예에서, 이 명령의 출력은 전송 큐를 서비스하는 채널이 QM1.TO.QM2임을 보여줍니다.

- 다음 명령을 발행하여 채널 QM1.TO.QM2의 상태를 판별하십시오.

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

- CHSTATUS** 명령의 출력에서 STATUS 필드를 조사하십시오.

- STATUS 필드의 값이 RUNNING이면 채널이 메시지를 이동시키고 있는지 확인하십시오.
- 명령의 출력에 상태가 표시되지 않거나 STATUS 필드의 값이 STOPPED, RETRY, BINDING 또는 REQUESTING인 경우 다음과 같이 적절한 단계를 수행하십시오.

- 옵션: STATUS 필드의 값이 상태를 표시하지 않는 경우 채널이 비활성 상태이므로 다음 단계를 수행하십시오.


- 트리거를 통해 채널을 자동으로 시작했으면 전송 큐의 메시지가 사용 가능한지 확인하십시오. 전송 큐에 사용 가능한 메시지가 있으면 전송 큐에 대한 트리거 설정이 올바른지 확인하십시오.
- 다음 명령을 발행하여 채널을 다시 수동으로 시작하십시오.

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

- 옵션: STATUS 필드의 값이 STOPPED인 경우 다음 단계를 수행하십시오.

- 오류 로그를 확인하여 채널이 중지된 원인을 판별하십시오. 오류로 인해 채널이 중지된 경우 문제점을 정정하십시오. 채널에 재시도 속성 *SHORTRTY* 및 *LONGRTY*의 값이 지정되어 있는지도 확인하십시오. 네트워크 오류와 같은 일시적 장애의 경우에는 채널이 자동으로 재시작을 시도합니다.
- 다음 명령을 발행하여 채널을 다시 수동으로 시작하십시오.

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

 IBM MQ for z/OS에서는, 명령 이벤트 메시지를 사용하여 사용자가 채널을 중지하는 시기를 감지할 수 있습니다.

- 옵션: STATUS 필드의 값이 RETRY인 경우 다음 단계를 수행하십시오.

- 오류 로그를 확인하여 오류를 식별한 후 문제점을 정정하십시오.
- 다음 명령을 발행하여 채널을 다시 수동으로 시작하십시오.

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

또는 채널이 다음 번 재시도에 연결할 때까지 대기하십시오.


- 옵션: STATUS 필드의 값이 BINDING 또는 REQUESTING이면 채널이 아직 파트너에 연결되지 않은 것입니다. 다음 단계를 수행하십시오.

- 채널의 양쪽 끝에서 다음 명령을 발행하여 채널의 하위 상태를 판별하십시오.

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

참고:

- 일부 경우 채널의 한쪽 끝에만 하위 상태가 있을 수도 있습니다.
 - 많은 하위 상태는 일시적이므로 명령을 몇 번 발행해서 채널이 특정 하위 상태로 유지되는지 판별하십시오.
- 287 페이지의 표 31를 확인하여 수행할 조치를 판별하십시오.

표 31. 바인딩 또는 요청 중 상태로 표시된 하위 상태		
시작 MCA 하위 상태 ¹	응답 MCA 하위 상태 ²	참고
NAMESERVER		시작 MCA가 이름 서버 요청이 완료될 때까지 대기하고 있습니다. 채널 속성 CONNAME에 올바른 호스트 이름이 지정되었고 이름 서버가 제대로 설정되었는지 확인하십시오.
SCYEXIT	SCYEXIT	MCA가 현재 보안 엑시트를 통해 대화 중입니다. 자세한 정보는 289 페이지의 『채널이 메시지를 충분히 빨리 처리할 수 있는지 판별』의 내용을 참조하십시오.
	CHADEXIT	채널 자동 정의 엑시트가 현재 실행 중입니다. 자세한 정보는 289 페이지의 『채널이 메시지를 충분히 빨리 처리할 수 있는지 판별』의 내용을 참조하십시오.
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	MQXR_INIT에 대한 채널 시작 시 엑시트가 호출됩니다. 시간이 오래 걸리면 엑시트의 이 부분에서 처리를 검토하십시오. 자세한 정보는 289 페이지의 『채널이 메시지를 충분히 빨리 처리할 수 있는지 판별』의 내용을 참조하십시오.
SERIALIZE	SERIALIZE	이 하위 상태는 SHARED 속성 지정의 채널에만 적용됩니다.
NETCONNECT		올바르지 않은 네트워크 구성으로 인해 연결 지연이 있으면 이 하위 상태가 표시됩니다.
SSLHANDSHAKE	SSLHANDSHAKE	TLS 데이터 교환은 많은 송신 및 수신으로 이루어집니다. 네트워크 시간이 느리거나 검색 CRL 연결이 느린 경우 이는 데이터 교환을 수행하는 데 걸리는 시간에 영향을 미칩니다.  IBM MQ for z/OS에서 이 하위 상태는 충분한 SSLTASKS가 없음을 나타내는 것일 수도 있습니다.

참고:

- i) 시작 MCA는 대화를 시작한 채널의 끝입니다. 송신자, 클러스터 송신자, 완전한 서버, 요청자일 수 있습니다. 서버-요청자 쌍에서는 채널을 시작한 끝입니다.
- ii) 응답 MCA는 대화 시작 요청에 응답한 채널의 끝입니다. 수신자, 클러스터 수신자, 요청자(서버 또는 송신자가 시작될 때), 서버(요청자가 시작될 때), 송신자(채널의 요청자-송신자 콜백 쌍에서)일 수 있습니다.

채널이 메시지를 이동시키는지 확인

전송 큐에 대한 문제가 있는 경우 채널이 메시지를 이동시키는지 확인하십시오.

시작하기 전에

명령 DIS CHSTATUS(QM1.TO.QM2) ALL을 발행하십시오. STATUS 필드의 값이 RUNNING이면 채널이 파트너 시스템에 연결된 것입니다.

283 페이지의 『큐의 메시지가 사용 가능한지 확인』에 설명된 대로, 전송 큐에 커밋되지 않은 메시지가 없는지 확인하십시오.

이 태스크 정보

채널의 가져오기 및 송신에 사용 가능한 메시지가 있으면 다음 검사를 수행하십시오.

프로시저

1. 채널 상태 표시 명령 DIS CHSTATUS(QM1.TO.QM2) ALL의 출력에서 다음 필드를 확인하십시오.

MSGS

이 세션 동안(채널이 시작된 이후) 송신했거나 수신한 메시지의 수(또는 서버 연결 채널의 경우, 처리된 MQI 호출 수).

BUFSSENT

송신된 전송 버퍼의 수. 여기에는 제어 정보만을 송신하는 전송이 포함됩니다.

BYTSSENT

이 세션 동안 송신된 바이트 수(채널이 시작된 이후로). 여기에는 메시지 채널 에이전트가 송신한 제어 정보가 포함됩니다.

LSTMSGDA

마지막 메시지가 송신되었거나 MQI 호출이 처리된 날짜(LSTMSGTI 참조).

LSTMSGTI

마지막 메시지가 송신되었거나 MQI 호출이 처리된 시간. 송신자 또는 서버의 경우에는 마지막 메시지(메시지가 분할된 경우에는 메시지의 마지막 부분)가 송신된 시간입니다. 요청자 또는 수신자의 경우에는 마지막 메시지가 대상 큐에 넣어진 시간입니다. 서버 연결 채널의 경우에는 마지막 MQI 호출이 완료된 시간입니다.

CURMSGS

송신 채널의 경우, 현재 배치에서 송신된 메시지 수. 수신 채널의 경우, 이는 현재 배치에서 수신된 메시지 수입니다. 배치가 확약되면, 송신 및 수신 채널 둘 다에 대해 값이 0으로 재설정됩니다.

STATUS

채널의 상태이며, Starting, Binding, Initializing, Running, Stopping, Retrying, Paused, Stopped 또는 Requesting입니다.

SUBSTATE

채널이 현재 수행 중인 조치입니다.

INDOUBT

채널이 현재 인다우트(in-doubt) 상태에 있는지의 여부. 송신 메시지 채널 에이전트가 송신한 메시지의 배치가 성공적으로 수신되었음을 수신확인하기 위해 대기하는 동안에만 YES입니다. 메시지가 송신되고 있는 동안 긍정 응답을 요청하기 전까지의 기간을 포함하여 다른 모든 시점에서는 NO입니다. 수신 채널의 경우, 값은 항상 NO입니다.

2. 채널이 시작된 이후에 메시지를 송신했는지 판별하십시오. 송신한 경우 언제 마지막 메시지를 송신했는지 판별하십시오.
3. CURMSGS에 0이 아닌 값으로 표시된 대로 아직 완료되지 않은 배치를 채널이 시작했을 수도 있습니다. INDOUBT가 YES인 경우 채널의 다른 끝이 배치를 수신했다는 수신확인을 수신하기 위해 대기 중임을 나타냅니다. 출력의 SUBSTATE 필드를 보고 288 페이지의 표 32의 내용을 참조하십시오.

표 32. 송신자 및 수신자 MCA 하위 상태		
송신자 SUBSTATE	수신자 SUBSTATE	참고
MQGET	RECEIVE	휴면 중인 채널의 정상 상태.
SEND	RECEIVE	SEND는 일반적으로 일시적 상태입니다. SEND가 표시될 경우 이는 통신 프로토콜 버퍼가 채워졌음을 나타냅니다. 네트워크 문제 점을 나타내는 것일 수 있습니다.
RECEIVE		송신자가 일정 기간 동안 RECEIVE 하위 상태로 표시되는 경우 송신자가 배치 완료 또는 하트비트에 대한 응답을 대기 중인 것입니다. 배치 완료에 시간이 오래 걸리는 이유를 확인할 수 있습니다.

참고: 특히, 채널의 하위 상태가 엑시트 처리와 연관된 경우 채널이 메시지를 충분히 빠르게 처리할 수 있는지를 여부를 판별할 수도 있습니다.

배치 완료에 시간이 오래 걸리는 이유 확인

배치를 완료하는 데 시간이 오래 걸리는 이유로는 네트워크가 느리거나 채널이 메시지 재시도 처리를 사용하는 원인이 있습니다.

이 태스크 정보

송신자 채널은 메시지 배치를 송신한 후 채널이 파이프라인되었는지 여부와 무관하게, 수신자로부터 해당 배치의 확인을 대기합니다. 이 태스크에 설명된 요인은 송신자 채널의 대기 시간에 영향을 미칠 수 있습니다.

프로시저

- 네트워크가 느린지 확인하십시오.
NETTIME 값은 배치 끝 요청을 채널의 원격 끝에 송신하고 응답을 수신하는 데 걸리는 시간에서 배치 끝 요청을 처리한 시간을 뺀 시간입니다(마이크로초로 표시). 이 값은 다음 이유 중 하나로 커질 수 있습니다.
 - 네트워크가 느립니다. 느린 네트워크는 배치를 완료하는 데 걸리는 시간에 영향을 미칠 수 있습니다. NETTIME 필드의 지표를 생성하는 측정은 배치 끝에서 측정됩니다. 하지만 느린 네트워크의 영향을 받은 첫 번째 배치는 배치 끝에서 측정되었기 때문에 NETTIME 값의 변경이 표시되지 않습니다.
 - 요청이 원격 끝에서 큐 대기 상태로 있거나(예를 들어, 채널이 넣기를 재시도 중일 수 있음) 페이지 세트 I/O로 인해 넣기 요청이 느릴 수 있습니다. 큐에 대기된 요청이 완료되면 배치 끝 요청의 지속 기간이 측정됩니다. 따라서 NETTIME 값이 크면 원격 끝의 비정상 처리를 확인하십시오.
- 채널이 메시지 재시도를 사용 중인지 확인하십시오.
수신자 채널에서 대상 큐에 메시지를 넣는 데 실패하는 경우 해당 메시지를 즉시 데드레터 큐에 넣지 않고 메시지 재시도 처리를 사용할 수도 있습니다. 재시도 처리는 배치 속도를 느리게 할 수 있습니다. MQPUT 시도 사이에 채널의 상태가 메시지 재시도 간격이 패스되기를 대기 중임을 나타내는 STATUS(PAUSED)가 됩니다.

채널이 메시지를 충분히 빨리 처리할 수 있는지 판별

메시지가 전송 큐에 빌드되고 있지만 처리 문제점이 발견되지 않은 경우 채널이 메시지를 충분히 빨리 처리할 수 있는지 판별하십시오.

시작하기 전에

다음 명령을 일정 기간 반복적으로 발행하여 채널에 대한 성능 데이터를 수집하십시오.

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

이 태스크 정보

283 페이지의 『[큐의 메시지가 사용 가능한지 확인](#)』에 설명된 대로, 전송 큐에 커밋되지 않은 메시지가 없는지 확인한 후 채널 상태 표시 명령의 출력에서 XQTIME 필드를 확인하십시오. XQTIME 지표의 값이 일관되게 높거나 측정 기간 중에 증가하면 채널이 넣고 있는 애플리케이션을 따라잡지 못하는 것입니다.

다음 테스트를 수행하십시오.

프로시저

1. 엑시트가 처리 중인지 확인하십시오.
이 메시지를 전달 중인 채널에 엑시트가 사용되는 경우 엑시트가 메시지 처리에 소모되는 시간을 늘릴 수 있습니다. 이러한 경우인지 식별하려면 다음 검사를 수행하십시오.
 - a) 명령 DIS CHSTATUS(QM1.TO.QM2) ALL의 출력에서 EXITIME 필드를 확인하십시오.
엑시트에 소모된 시간이 예상보다 많으면 엑시트의 처리에서(특히 메시지, 송신, 수신 엑시트에서) 불필요한 루프나 여분의 처리를 검토하십시오. 이러한 처리는 채널에서 이동하는 모든 메시지에 영향을 줍니다.
 - b) 명령 DIS CHSTATUS(QM1.TO.QM2) ALL의 출력에서 SUBSTATE 필드를 확인하십시오.
채널이 상당한 시간 동안 다음 하위 상태 중 하나로 유지되면 엑시트의 처리를 검토하십시오.
 - SCYEXIT
 - RCVEXIT
 - SENDEXIT
 - MSGEXIT
 - MREXIT

채널 하위 상태에 대한 자세한 정보는 [288 페이지의 표 32](#) 테이블을 참조하십시오.

2. 네트워크가 느린지 확인하십시오.

메시지가 채널에서 충분히 빠르게 이동하지 않는 경우 이는 네트워크가 느리기 때문일 수 있습니다. 이러한 경우인지 식별하려면 다음 검사를 수행하십시오.

a) 명령 `DIS CHSTATUS(QM1.TO.QM2) ALL`의 출력에서 `NETTIME` 필드를 확인하십시오.

이 지표는 송신 채널이 파트너에 응답을 요청할 때 측정됩니다. 이는 각 배치의 끝에서 그리고 하트비트 중 에 채널이 유휴 상태일 때 발생합니다.

b) 이 지표가 라운드트립에 예상보다 시간이 오래 걸린다고 표시하는 경우에는 다른 네트워크 모니터링 도구를 사용하여 네트워크 성능을 조사하십시오.

3. 채널이 압축을 사용 중인지 확인하십시오.

채널이 압축을 사용하면 메시지 처리 시간이 증가합니다. 채널이 압축 알고리즘을 하나만 사용 중인 경우 다음 검사를 수행하십시오.

a) 명령 `DIS CHSTATUS(QM1.TO.QM2) ALL`의 출력에서 `COMPTIME` 필드를 확인하십시오.

이 지표는 압축 또는 압축 해제 중에 소모된 시간을 표시합니다.

b) 선택된 압축이 송신할 데이터 양을 예상하는 만큼 감소시키지 않으면 압축 알고리즘을 변경하십시오.

4. 채널이 여러 압축 알고리즘을 사용 중인 경우에는 다음 검사를 수행하십시오.

a) 명령 `DIS CHSTATUS(QM1.TO.QM2) ALL`의 출력에서 `COMPTIME`, `COMPHDR`, `COMPMSG` 필드를 확인하십시오.

b) 압축률 또는 알고리즘 선택이 필요한 압축 또는 성능을 제공하지 않는 경우에는, 채널 정의에 지정된 압축 알고리즘을 변경하거나 특정 메시지에 대한 채널의 압축 알고리즘 선택을 대체할 메시지 엑시트를 쓸 것을 고려하십시오.

클러스터 채널에 대한 문제점 해결

`SYSTEM.CLUSTER.TRANSMIT.QUEUE` 큐에 메시지를 빌드하는 경우 문제점 진단의 첫 단계는 메시지 전달 문제가 있는 단일 또는 다중 채널을 찾는 것입니다.

이 태스크 정보

`SYSTEM.CLUSTER.TRANSMIT.QUEUE`를 사용하여 메시지 전달 문제가 있는 단일 또는 다중 채널을 찾으려면 다음 검사를 수행하십시오.

프로시저

1. 다음 명령을 실행하십시오.

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

참고: 많은 메시지가 이동하는 사용 중인 클러스터가 있으면 전달 가능한 몇 개의 메시지만 있는 채널을 제거하기 위해 이 명령을 보다 높은 수로 발행할 것을 고려하십시오.

2. 출력에서 `XQMSGSA` 필드의 값이 큰 채널을 찾아보십시오. 채널이 메시지를 이동시키지 않거나 충분히 빠르게 이동시키지 않는 이유를 판별하십시오. [285 페이지의 『채널 모니터링』](#)에 개괄된 태스크를 사용하여 빌드를 유발하는 채널에 대한 문제점을 진단하십시오.

클러스터 모니터링

클러스터 안에서 애플리케이션 메시지, 제어 메시지 및 로그를 모니터링할 수 있습니다. 클러스터가 큐의 둘 이상의 인스턴스 사이에 로드 밸런싱할 때 특별한 모니터링 고려사항이 있습니다.

클러스터의 애플리케이션 메시지 모니터링

일반적으로 메시지를 전송하는 데 사용되는 클러스터 송신자 채널에 관계없이 큐 관리자에서 나가는 모든 클러스터 메시지가 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`(를) 통해 전달됩니다. 각 채널은 해당 채널을 대상으로 하는 메시지를 다른 모든 클러스터 송신자 채널과 병렬로 내보내고 있습니다. 이 큐에서 점점 쌓여가는 메시지는 하나 이상의 채널의 문제점을 표시할 수 있으며 조사해야 합니다.

- 큐의 용량이 클러스터 설계에 대해 적절하게 모니터링되어야 합니다.
- 다음 명령은 전송 큐에서 대기 중인 둘 이상의 메시지를 갖는 모든 채널을 리턴합니다.

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

모든 클러스터 메시지가 단일 큐에 있을 때, 항상 채널이 채우기 시작할 때 문제가 있는 채널을 확인하는 것이 쉽지는 않습니다. 이 명령의 사용은 책임있는 채널을 확인하는 쉬운 방법입니다.

복수 전송 큐를 갖도록 클러스터 큐 관리자를 구성할 수 있습니다. 큐 관리자 속성 DEFCLXQ를 CHANNEL(으)로 변경하면 모든 클러스터 송신자 채널이 다른 클러스터 전송 큐와 연관됩니다. 그렇지 않으면 별도의 전송 큐를 수동으로 구성할 수 있습니다. 클러스터 송신자 채널과 연관된 모든 클러스터 전송 큐를 표시하려면 다음 명령을 실행하십시오.

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

왼쪽에 큐 이름의 고정된 여백을 갖는 패턴을 따르도록 클러스터 전송 큐를 정의하십시오. 그러면 일반 큐 이름을 사용하여 **DISPLAY CLUSMGR** 명령에 의해 리턴되는 모든 클러스터 전송 큐의 용량을 조회할 수 있습니다.

```
DISPLAY QUEUE (qname *) CURDEPTH
```

클러스터의 제어 메시지 모니터링

SYSTEM.CLUSTER.COMMAND.QUEUE 큐는 로컬 큐 관리자가 생성하거나 클러스터의 다른 큐 관리자에서 이 큐 관리자로 전송되는 큐 관리자에 대한 모든 클러스터 제어 메시지를 처리하는 데 사용됩니다. 큐 관리자가 그의 클러스터 상태를 올바르게 유지보수 중일 때, 이 큐는 0으로 가는 경향이 있습니다. 그러나 이 큐의 메시지 용량이 일시적으로 커질 수 있는 상황이 있습니다.

- 큐에 많은 메시지를 갖는 것은 클러스터 상태의 뒤섞임을 표시합니다.
- 상당한 변경을 수행할 때, 큐가 해당 변경 사이에서 안정할 수 있게 하십시오. 예를 들어, 저장소를 이동할 때 두 번째 저장소를 이동하기 전에 큐가 0에 도달할 수 있도록 허용하십시오.

메시지의 백로그가 이 큐에 존재할 때, 클러스터 상태 업데이트나 클러스터 관련 명령이 처리되지 않습니다. 메시지가 오랜 시간 동안 이 큐에서 제거되지 않으면 처음에 이 상황을 유발하는 프로세스에 대해 설명할 수 있는 큐 관리자 오류 로그(또는 z/OS의 경우 CHINIT 로그)의 검사를 통해 추가 조사가 필요합니다.

SYSTEM.CLUSTER.REPOSITORY.QUEUE에서는 클러스터 저장소 캐시 정보를 여러 메시지로 보유합니다. 메시지가 항상 이 큐에 존재하고 더 큰 클러스터의 경우 더 많이 존재하는 것이 유용합니다. 그러므로 이 큐의 메시지 용량은 관심이 가는 문제가 아닙니다.

로그 모니터링

문제점이 원래 정보의 캐싱 및 클러스터링의 분산 본질로 인해 발생한 후 여러 날(심지어 여러 달) 동안 클러스터에서 발생하는 문제점이 애플리케이션에 외부 증상을 보이지 않을 수 있습니다. 그러나 원래 문제점이 종종 IBM MQ 오류 로그(및 z/OS의 경우 CHINIT 로그)에 기록됩니다. 이 때문에 클러스터링과 관련하여 작성되는 모든 메시지에 대해 이들 오류를 능동적으로 모니터링하는 것이 중요합니다. 이들 메시지를 읽고 이해하고, 필요한 경우 조치를 취해야 합니다.

예를 들어, 클러스터에 있는 큐 관리자와의 통신 중단은 클러스터가 정보를 재발행하여 클러스터 자원을 정기적으로 다시 유효성 검증하는 방식으로 인해 삭제되고 있는 특정 클러스터 자원을 알 수 있게 합니다. 잠재적으로 발생하는 이러한 이벤트의 경고는 [AMQ9465](#) 또는 [CSQX465I](#)(z/OS 시스템) 메시지로 보고합니다. 이 메시지는 문제점을 조사해야 함을 표시합니다.

로드 밸런싱을 위한 특별 고려사항

클러스터가 한 큐의 둘 이상의 인스턴스 사이에서 로드 밸런싱할 때, 이용하는 애플리케이션이 각 인스턴스의 메시지를 처리 중이어야 합니다. 이용 애플리케이션의 하나 이상이 종료하거나 메시지 처리를 중지하는 경우, 클러스터링이 큐의 해당 인스턴스로 메시지를 계속 송신할 수 있습니다. 이 상황에서는 애플리케이션이 다시 올바르게 기능 중일 때까지 해당 메시지가 처리되지 않습니다. 이 때문에 애플리케이션의 모니터링은 솔루션의 중요한

부분이며 해당 상황에서 메시지를 경로 재지정하는 조치를 취해야 합니다. 그런 모니터링을 자동화하는 메커니즘의 예를 샘플 클러스터 큐 모니터링 샘플 프로그램(AMQSCLM)에서 찾을 수 있습니다.

관련 개념

352 페이지의 『분산 발행/구독 네트워크 성능 조정』

IBM MQ 분산 발행/구독 클러스터 및 계층의 성능을 개선하려면 이 절의 성능 조정 팁을 사용하십시오.

357 페이지의 『발행/구독 네트워크에서 작성자와 이용자 밸런스 유지』

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

전송 큐 전환 모니터링

엔터프라이즈에 미치는 영향이 최소화되도록 전송 큐를 전환하는 클러스터-송신자 채널의 프로세스를 모니터링하는 것이 중요합니다. 예를 들어, 워크로드가 높거나 여러 채널을 동시에 전환하여 이 프로세스를 시도해서는 안 됩니다.

채널 전환 프로세스

채널을 전환하는 데 사용되는 프로세스는 다음과 같습니다.

1. 채널이 입력을 위해 새 전송 큐를 열고 이 큐에서 메시지 가져오기를 시작합니다 (상관 ID별 가져오기 사용).
2. 큐 관리자가 백그라운드 프로세스를 시작하여 채널에 대해 큐된 메시지를 이전 전송 큐에서 새 전송 큐로 이동합니다. 메시지가 이동되는 동안 채널에 대한 모든 새 메시지는 순서 지정을 보존하기 위해 이전 전송 큐에 대기됩니다. 이전 전송 큐의 채널에 대해 많은 수의 메시지가 있거나 새 메시지가 빠르게 도착하는 경우 이 프로세스를 완료하는 데 시간이 걸릴 수 있습니다.
3. 커밋되거나 커밋되지 않은 메시지가 이전 전송 큐의 채널에 대해 큐에 남아 있지 않으면 전환이 완료됩니다. 이제 새 메시지를 새 전송 큐에 직접 넣습니다.

여러 채널이 동시에 전환되는 상황을 방지하기 위해 IBM MQ 는 실행 중이지 않은 하나 이상의 채널의 전송 큐를 전환하는 기능을 제공합니다. 위치:

- IBM MQ for Multiplatforms 명령이라고 함 **runswch1**
- IBM MQ for z/OS CSQUTIL 유틸리티를 사용하여 대신 SWITCH CHANNEL 명령을 처리할 수 있습니다.

스위치 조작의 상태 모니터링

스위치 조작의 상태를 이해하기 위해 관리자는 다음 조치를 수행할 수 있습니다.

- 큐 관리자 오류 로그 (AMQERR01.LOG). 여기서 메시지는 조작 중에 다음 단계를 표시하기 위한 출력입니다.
 - 전환 조작이 시작되었습니다.
 - 메시지 이동이 시작되었습니다.
 - 이동할 남아 있는 메시지 수에 대한 주기적 업데이트 (전환 조작이 빠르게 완료되지 않는 경우)
 - 메시지 이동이 완료되었습니다.
 - 전환 조작이 완료되었습니다.

z/OS에서 이러한 메시지는 채널 시작기 작업 로그가 아닌 큐 관리자 작업 로그에 출력되지만 시작 시 전환을 시작하는 경우 채널 시작기 작업 로그에 대한 단일 메시지가 채널 시작기 작업 로그에 의해 출력됩니다.

- DISPLAY CLUSQMGR 명령을 사용하여 각 클러스터-송신자 채널이 현재 사용 중인 전송 큐를 조회하십시오.
- 퀴리 모드에서 **runswch1** 명령 (또는 z/OS의 경우 CSQUTIL) 을 실행하여 하나 이상의 채널의 전환 상태를 확인하십시오. 이 명령의 출력은 각 채널에 대해 다음을 식별합니다.
 - 채널에 보류 중인 전환 조작이 있는지 여부
 - 채널이 전환되는 전송 큐
 - 이전 전송 큐에 남아 있는 메시지 수

한 번의 호출로 모든 채널의 상태, 구성 변경의 영향 및 모든 스위치 조작이 완료되었는지 여부를 판별할 수 있기 때문에 각 명령은 매우 유용합니다.

발생할 수 있는 잠재적 문제

전송 큐를 전환할 때 발생할 수 있는 일부 문제, 해당 원인 및 가장 가능성이 높은 솔루션의 목록은 [전송 큐를 전환할 때 발생할 수 있는 문제](#) 를 참조하십시오.

관련 개념

352 페이지의 『[분산 발행/구독 네트워크 성능 조정](#)』

IBM MQ 분산 발행/구독 클러스터 및 계층의 성능을 개선하려면 이 절의 성능 조정 팁을 사용하십시오.

357 페이지의 『[발행/구독 네트워크에서 작성자와 사용자 밸런스 유지](#)』

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

Multi 애플리케이션 밸런싱 모니터링

DISPLAY APSTATUS 명령을 사용하여 균등 클러스터에서 애플리케이션 밸런싱 상태를 모니터링하고 예상치 못한 경우 애플리케이션의 밸런스가 맞지 않는 이유를 조사할 수 있습니다.

클러스터에 있는 큐 관리자에서 애플리케이션의 현재 상태 모니터링

균등 클러스터에 있는 큐 관리자에서 **DIS APSTATUS** 명령을 실행하여 한 클러스터의 모든 큐 관리자에서 애플리케이션의 현재 상태를 가져올 수 있습니다.

IBM MQ 9.2.0부터는 **TYPE** 필드도 출력에 표시됩니다.

예를 들어 한 큐 관리자를 방금 시작한 후에는 다음과 같은 출력이 표시될 수 있습니다.

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)
```

이는 균등 클러스터에 MYAPP이라는 애플리케이션이 있고, 현재 8개의 인스턴스가 있으며 이들 모두는 균등 클러스터에서 이동 가능한 것으로 간주됨을 표시합니다. 밸런스 값 UNKNOWN은 임시 값이며, 이는 큐 관리자가 필요한 경우 애플리케이션의 재밸런싱 시도를 아직 완전히 완료하지 않았음을 표시합니다.

잠시 후 다음과 같은 출력이 표시될 수 있습니다.

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

이 출력은 애플리케이션에 8개의 인스턴스가 있지만 균등 클러스터에서 아직 밸런싱되지 않았음을 표시합니다. 이 때 클러스터에서 애플리케이션의 분포를 살펴보는 것이 좋습니다.

이를 수행하려면 **DIS APSTATUS** 명령을 다시 실행하십시오. 균등 클러스터에 있는 임의의 큐 관리자에 대해 이 명령을 실행할 수 있습니다.

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                      MOVCCOUNT(6)
BALSTATE(HIGH)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)             QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
```

```

AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(1)                 MOVCCOUNT(1)
BALSTATE(LOW)             LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)       QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(1)                 MOVCCOUNT(1)
BALSTATE(LOW)             LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)       QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

여기에서는 큐 관리자 UNID001에 6개의 인스턴스가 있지만 큐 관리자 UNID0002 및 UNID0003에는 각각 하나만 있음을 알 수 있습니다. BALSTATE 출력은 큐 관리자가 이를 보고한 마지막 시간의 밸런스 상태를 표시합니다. 그러나 인스턴스 수는 BALSTATE 필드보다 더 최신일 수 있습니다.

이 출력은 또한 균등 클러스터가 이 애플리케이션에 대한 정보를 분배하고 있음을 나타냅니다. 애플리케이션이 균등 클러스터에 있는 모든 큐 관리자 인스턴스에게 알려지고 마지막 메시지 날짜 및 시간이 가장 최신입니다.

또한 ACTIVE 필드는 클러스터에 있는 모든 큐 관리자가 서로 통신하는 것으로 간주됨을 나타냅니다. 임의의 큐 관리자에서 ACTIVE가 NO로 설정되면 이는 정기 통신이 인터럽트됨을 나타냅니다.

자체 재밸런싱할 내용이 남아있으면 출력은 결국 다음과 유사하게 표시하며 끝납니다.

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(3)                 MOVCCOUNT(3)
BALSTATE(OK)              LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)       QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(3)                 MOVCCOUNT(3)
BALSTATE(OK)              LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)       QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(2)                 MOVCCOUNT(2)
BALSTATE(OK)              LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)       QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

큐 관리자의 BALSTATE는 이제 OK이며 이는 안정된 상태에 도달했음을 표시합니다.

별도 애플리케이션 인스턴스 모니터링

애플리케이션 인스턴스를 개별적으로 살펴볼 수 있지만 각 큐 관리자에 대해 다음을 수행할 수 있습니다. UNID001의 출력은 다음과 같습니다.

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)

```

```
TYPE (LOCAL)
One MQSC command read.
```

MOVABLE(YES)은 클러스터에 있는 다른 큐 관리자에 필요한 경우 이 인스턴스를 이동할 수 있음을 표시합니다. 다음 예제에서는 클라이언트로서 연결되지 않았으므로 애플리케이션 인스턴스를 이동할 수 없습니다.

```
3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)
```

CONNTAG 필드를 사용하여 해당 인스턴스로부터의 개별 큐 관리자 연결을 확인할 수 있습니다. 이는 애플리케이션 인스턴스가 IMMREASN(NOTRECONN)인 경우 유용할 수 있습니다. 다음 예제에서 클라이언트 애플리케이션은 이동할 수 없고 연결 옵션 조사에서 **CONNOPTS** 필드에 MQCNO_RECONNECT가 없음을 표시합니다.

```
1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE (CONN)
PID(14656) TID(20)
APPLDESC(IBM MQ Channel) APPLTAG(ClientApp)
APPLTYPE(USER) ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN) CLIENTID( )
CONNAME(127.0.0.1)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserid) UOWLOG( )
UOWSTDA( ) UOWSTTI( )
UOWLOGDA( ) UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0) UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
TYPE (CONN)
```

관련 개념

[자동 애플리케이션 밸런싱](#)

Monitoring performance and resource usage on z/OS

Use this topic to understand the facilities available to monitor the performance, and resource usage of your IBM MQ for z/OS subsystems.

Related tasks

[Configuring queue managers on z/OS](#)

[Administering IBM MQ for z/OS](#)

Introduction to monitoring IBM MQ for z/OS

Use this topic as an overview of the monitoring facilities available for IBM MQ for z/OS. For example, obtaining snapshots, using IBM MQ trace, online monitoring, and events.

This topic describes how to monitor the performance and resource usage of IBM MQ.

- It outlines some of the information that you can retrieve and briefly describes a general approach to investigating performance problems. See [“Investigating performance problems” on page 302](#) for more information.
- It describes how you can collect statistics about the performance of IBM MQ by using SMF records.
- It describes how to gather accounting data to enable you to charge your customers for their use of your IBM MQ systems.
- It describes how to use IBM MQ events (alerts) to monitor your systems.

Here are some of the tools you might use to monitor IBM MQ; they are described in the sections that follow:

- Tools provided by IBM MQ:
 - [Using DISPLAY commands](#)
 - [“Using CICS adapter statistics” on page 297](#)
 - [“Using IBM MQ events” on page 299](#)
- z/OS service aids:
 - [“Using System Management Facility” on page 299](#)
- Other IBM licensed programs:
 - [Using the Resource Measurement Facility](#)
 - [Using Tivoli Decision Support for z/OS](#)
 - [Using the CICS monitoring facility](#)

Information about interpreting the data gathered by the performance statistics trace is given in [“Interpreting IBM MQ for z/OS performance statistics” on page 303](#).

Information about interpreting the data gathered by the accounting trace is given in [“Interpreting IBM MQ for z/OS accounting data” on page 339](#).

Getting snapshots of IBM MQ using the DISPLAY commands

IBM MQ provides the MQSC facility which can give a snapshot of the performance, and resource usage using the DISPLAY commands.

You can get an idea of the current state of IBM MQ by using the DISPLAY commands and, for the CICS adapter, the CICS adapter panels.

Using DISPLAY commands

You can use the IBM MQ MQSC DISPLAY or PCF Inquire commands to obtain information about the current state of IBM MQ. They provide information about the status of the command server, process definitions, queues, the queue manager, and its associated components. These commands are:

MQSC command	PCF command
DISPLAY ARCHIVE	Inquire Archive
DISPLAY AUTHINFO	Inquire Authentication Information Object
DISPLAY CFSTATUS	Inquire CF Structure Status
DISPLAY CFSTRUCT	Inquire CF Structure
DISPLAY CHANNEL	Inquire Channel
DISPLAY CHINIT	Inquire Channel Initiator
DISPLAY CHSTATUS	Inquire Channel Status
DISPLAY CMDSERV	
DISPLAY CLUSQMGR	Inquire Cluster Queue Manager

MQSC command	PCF command
DISPLAY CONN	Inquire Connection
DISPLAY GROUP	Inquire Group
DISPLAY LOG	Inquire Log
DISPLAY PROCESS	Inquire Process
DISPLAY QMGR	Inquire Queue Manager
DISPLAY QSTATUS	Inquire Queue Status
DISPLAY QUEUE	Inquire Queue
DISPLAY SECURITY	Inquire Security
DISPLAY STGCLASS	Inquire Storage Class
DISPLAY SYSTEM	Inquire System
DISPLAY TRACE	
DISPLAY USAGE	Inquire Usage

For the detailed syntax of each command, see [MQSC commands](#) or [PCF commands](#). All of the functions of these commands (except DISPLAY CMDSERV and DISPLAY TRACE) are also available through the operations and control panels.

These commands provide a snapshot of the system only at the moment the command was processed. If you want to examine trends in the system, you must start an IBM MQ trace and analyze the results over a period of time.

Using CICS adapter statistics

If you are an authorized CICS user, you can use the CICS adapter control panels to display CICS adapter statistics dynamically.

These statistics provide a snapshot of information related to CICS thread usage and situations when all threads are busy. The display connection panel can be refreshed by pressing the Enter key.

For more information about configuring the IBM MQ CICS adapter, see the [Configuring connections to MQ](#) section of the CICS documentation.

Using IBM MQ trace

You can record performance statistics and accounting data for IBM MQ by using the IBM MQ trace facility. Use this topic to understand how to control IBM MQ trace.

The data generated by IBM MQ is sent to:

- The System Management Facility (SMF), specifically as SMF record type 115, subtypes 1 and 2 for the performance statistics trace
- The SMF, specifically as SMF record type 116, subtypes zero, 1, and 2 for the accounting trace.


If you prefer, the data generated by the IBM MQ accounting trace can also be sent to the generalized trace facility (GTF).

Starting IBM MQ trace

You can start the IBM MQ trace facility at any time by issuing the IBM MQ [START TRACE](#) command.


Accounting data can be lost if the accounting trace is started or stopped while applications are running. To collect accounting data successfully, the following conditions must apply:

- The accounting trace must be active when an application starts, and it must still be active when the application finishes.
- If the accounting trace is stopped, any accounting data collection that was active stops.

You can also start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro.  These parameters are described in [Using CSQ6SYSP](#).

Before starting an IBM MQ trace, read [“Using System Management Facility”](#) on page 299.

Controlling IBM MQ trace

To control the IBM MQ trace data collection at start-up, specify values for the parameters in the CSQ6SYSP macro when you customize IBM MQ.  See [Using CSQ6SYSP](#) for details.

You can control IBM MQ tracing when the queue manager is running with these commands:

- START TRACE
- ALTER TRACE
- STOP TRACE

You can choose the destination to which trace data is sent. Possible destinations are:

SMF

System Management Facility

GTF


Generalized Trace Facility (accounting trace only)

SRV

Serviceability routine for diagnostic use by IBM service personnel

For daily monitoring, information is sent to SMF (the default destination). SMF data sets typically contain information from other systems; this information is not available for reporting until the SMF data set is dumped.

You can also send accounting trace information to the GTF. This information has an event identifier of 5EE.

 The MQI call and user parameter, and [z/OS generalized trace facility \(GTF\)](#) describes how to deal with IBM MQ trace information sent to the GTF.

For information about IBM MQ commands, see [MQSC commands](#).

Effect of trace on IBM MQ performance

Using the IBM MQ trace facility can have a significant effect on IBM MQ and transaction performance. For example, if you start a global trace for class 1 or for all classes, it is likely to increase processor usage and transaction response times by approximately 50%. However, if you start a global trace for classes 2 - 4 alone, the increase in processor usage and transaction response times is likely to be less than 1% additional processor cost to the cost of IBM MQ calls. The same applies for a statistics or accounting trace.

Using IBM MQ online monitoring

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

[Table 33 on page 299](#) summarizes the commands to set these attributes at different levels and to display the monitoring information.

Table 33. Setting and displaying attributes to control online monitoring

Attribute	Applicable at this level	Set using command	Display monitoring information using command
MONQ	Queue	DEFINE QLOCAL DEFINE QMODEL ALTER QLOCAL ALTER QMODEL	DISPLAY QSTATUS
	Queue manager	ALTER QMGR	
MONCHL	Channel	DEFINE CHANNEL ALTER CHANNEL	DISPLAY CHSTATUS
	Queue manager	ALTER QMGR	
MONACLS	Queue manager	ALTER QMGR	

For full details of these commands, see [MQSC commands](#). For more information about online monitoring, see “[IBM MQ 네트워크 모니터링](#)” on page 5.

Using IBM MQ events

IBM MQ instrumentation events provide information about errors, warnings, and other significant occurrences in a queue manager. You can monitor the operation of all your queue managers by incorporating these events into your own system management application.

IBM MQ instrumentation events fall into the following categories:

Queue manager events

These events are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist.

Performance events

These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached, or the queue was not serviced within a predefined time limit.

Channel events

These events are reported by channels as a result of conditions detected during their operation. For example, a channel instance is stopped.

Configuration events

These events are notifications that an object has been created, changed, or deleted.

When an event occurs, the queue manager puts an *event message* on the appropriate *event queue*, if defined. The event message contains information about the event that can be retrieved by a suitable IBM MQ application.

IBM MQ events can be enabled using the IBM MQ commands or the operations and control panels.

See “[이벤트 유형](#)” on page 11 for information about the IBM MQ events that generate messages, and for information about the format of these messages. See [Event message reference](#) for information about enabling the events.

Using System Management Facility

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

System management facility (SMF) is a z/OS service aid used to collect information from various z/OS subsystems. This information is dumped and reported periodically, for example, hourly. You can use SMF with the IBM MQ trace facility to collect data from IBM MQ. In this way you can monitor *trends*, for example, in system utilization and performance, and collect accounting information about each user ID using IBM MQ.

To record performance statistics (record type 115) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(115))
```

To record accounting information (record type 116) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(116))
```

To use the z/OS command SETSMF, either PROMPT(ALL) or PROMPT(LIST) must be specified in the SMFPRM xx member. See [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) for more information.

You can start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro; this is described in [Using CSQ6SYSP](#).

Specifying YES on the SMFSTAT and SMFACCT parameters enables you to collect trace information as a queue manager starts.

You can also start collection of the data when the queue manager is running with the **START TRACE** command, specifying START TRACE (A) or START TRACE (S).

You can turn on or off the recording of accounting information at the queue or queue manager level using the ACCTQ parameter of the **DEFINE QLOCAL**, **DEFINE QMODEL**, **ALTER QLOCAL**, **ALTER QMODEL**, or **ALTER QMGR** commands. See [MQSC commands](#) for details of these commands.

You can control the collection of channel accounting data at the channel or queue manager level using the **STATCHL** parameter of the **DEFINE CHANNEL**, **ALTER CHANNEL** or **ALTER QMGR** commands.

You can specify the interval at which IBM MQ collects statistics and accounting data in one of these ways:

- You can collect statistics data and accounting data at different intervals, using STATIME (statistics data) and ACCTIME (accounting data) in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data at the same interval by specifying a value for STATIME in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data by specifying zero for STATIME.

SMF must be running before you can send data to it. For more information about SMF, see the [z/OS MVS System Management Facilities \(SMF\)](#) manual.

For the statistics and accounting data to be reset, at least one MQI call must be issued during the accounting interval.

Allocating additional SMF buffers

When you start a trace, you must ensure that you allocate adequate SMF buffers. Specify SMF buffering on the VSAM BUFSP parameter of the access method services DEFINE CLUSTER statement. Specify CISZ(4096) and BUFSP(81920) on the **DEFINE CLUSTER** statement for each SMF VSAM data set.

If an SMF buffer shortage occurs, SMF rejects any trace records sent to it. IBM MQ sends a CSQW133I message to the z/OS console when this occurs. IBM MQ treats the error as temporary and remains active

even though SMF data can be lost. When the shortage has been alleviated and trace recording has resumed, IBM MQ sends a CSQW123I message to the z/OS console.

Reporting data in SMF

You can use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed.

There are several ways to report on this data, for example:

- Write an application program to read and report information from the SMF data set. You can then tailor the report to fit your exact needs.
- Use Performance Reporter to process the records. For more information, see [“Using other products with IBM MQ”](#) on page 301.

No interval CLASS(03) SMF accounting records produced during long running processes

You are collecting CLASS(3) SMF116 accounting records for IBM MQ, but are getting no records produced while a long running process runs.

The CLASS(3) SMF116 accounting records normally are produced only when a process ends. For long running processes, for example CICS, this might not produce a sufficient number of records, as the process can run for a month or longer. However, you might want to gather SMF116 records at set time intervals while a process is running.

To gather CLASS(3) SMF116 accounting records you must set the following:

SMFACCT

=YES

SMFSTAT

=YES or NO, where

YES

Causes records to be produced if a collection broadcast is received.

No

Causes you to get a CLASS(3) SMF116 record produced only when a process ends

and issue the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(03)
```

If you have set SMFSTAT=YES and a collection broadcast occurs, an interval CLASS(3) SMF116 accounting record is produced for any process currently running that was also running at the time of the previous collection broadcast.

You can set the collection broadcast to occur on a regular time interval by setting STATIME in [CSQ6SYSP](#) as follows:

- If your STATIME has been set to a value greater than 0, that is your broadcast interval in minutes.
- If your STATIME = 0 the SMF broadcast of your system is used (SMF INTVAL)
- If your STATIME = 0 and your SMF INTVAL is not set, no broadcast occurs and no interval records are produced

Using other products with IBM MQ

You can use other products to help you to improve the presentation of, or to augment statistics related to, performance and accounting. For example, Resource Measurement Facility, Tivoli Decision Support, and CICS monitoring.

Using the Resource Measurement Facility

Resource Measurement Facility (RMF) is an IBM licensed program (program number 5685-029) that provides system-wide information about processor utilization, I/O activity, storage, and paging. You can use RMF to monitor the utilization of physical resources across the whole system dynamically. For more information, see the [z/OS Resource Measurement Facility User's Guide](#).

Using Tivoli Decision Support for z/OS

You can use Tivoli Decision Support for z/OS to interpret RMF and SMF records.

Tivoli Decision Support for z/OS is an IBM licensed program (program number 5698-B06) that enables you to manage the performance of your system by collecting performance data in a Db2® database and presenting the data in various formats for use in systems management. Tivoli Decision Support for can generate graphic and tabular reports using systems management data it stores in its Db2 database. It includes an administration dialog, a reporting dialog, and a log collector, all of which interact with a standard Db2 database.

This is described in the [IBM Tivoli Decision Support for z/OS: Administration Guide and Reference](#).

Using the CICS monitoring facility

The CICS monitoring facility provides performance information about each CICS transaction running. It can be used to investigate the resources used and the time spent processing transactions. For background information, see the [CICS Performance Guide](#) and [Developing CICS System Programs](#), together with the two companion reference manuals, formerly called the [CICS Customization Guide](#).

Investigating performance problems

Performance problems can arise from various factors. For example, incorrect resource allocation, poor application design, and I/O restraints. Use this topic to investigate some of the possible causes of performance problems.

Performance can be adversely affected by:

- Buffer pools that are an incorrect size
- Lack of real storage
- I/O contention for page sets or logs
- Log buffer thresholds that are set incorrectly
- Incorrect setting of the number of log buffers
- Large messages
- Units of recovery that last a long time, incorporating many messages for each sync point
- Messages that remain on a queue for a long time
- RACF® auditing
- Unnecessary security checks
- Inefficient program design

When you analyze performance data, always start by looking at the overall system before you decide that you have a specific IBM MQ problem. Remember that almost all symptoms of reduced performance are magnified when there is contention. For example, if there is contention for DASD, transaction response times can increase. Also, the more transactions there are in the system, the greater the processor usage and greater the demand for both virtual and real storage.

In such situations, the system shows heavy use of *all* its resources. However, the system is actually experiencing normal system stress, and this stress might be hiding the cause of a performance reduction. To find the cause of such a loss of performance, you must consider all items that might be affecting your active tasks.

Investigating the overall system

Within IBM MQ, the performance problem is either increased response time or an unexpected and unexplained heavy use of resources. First check factors such as total processor usage, DASD activity, and paging. An IBM tool for checking total processor usage is resource management facility (RMF). In general, you must look at the system in some detail to see why tasks are progressing slowly, or why a specific resource is being heavily used.


Start by looking at general task activity, then focus on particular activities, such as specific tasks or a specific time interval.

Another possibility is that the system has limited real storage; therefore, because of paging interrupts, the tasks progress more slowly than expected.

Investigating individual tasks

You can use the accounting trace to gather information about IBM MQ tasks. These trace records tell you a great deal about the activity that the task has performed, and about how much time the task spent suspended, waiting for latches. The trace record also includes information about how much Db2 and coupling facility activity were performed by the task.

Interpreting IBM MQ accounting data is described in [“Interpreting IBM MQ for z/OS accounting data”](#) on page 339.

Long running units of work can be identified by the presence of message CSQR026I in the job log. This message indicates that a task has existed for more than three queue manager checkpoints and its log records have been shunted.  For a description of log record shunting, see [The log files](#).

Interpreting IBM MQ for z/OS performance statistics

Use this topic as an index to the different SMF records created by IBM MQ for z/OS.

IBM MQ for z/OS performance statistics are written as SMF type 115 records. Statistics records are produced periodically at a time interval specified by the **STATIME** parameter of the CSQ6SYSP system parameter module, or at the SMF global recording interval if you specify zero for **STATIME**. The information provided in the SMF records comes from the following components of IBM MQ:

Buffer manager	Manages the buffer pools in virtual storage and the writing of pages to page sets as the buffer pools become full. Also manages the reading of pages from page sets.
Coupling facility manager	Manages the interface with the coupling facility.
Data manager	Manages the links between messages and queues. It calls the buffer manager to process the pages with messages on them.
Db2 manager	Manages the interface with the Db2 database that is used as the shared repository.
Lock manager	Manages locks
Log manager	Manages the writing of log records, which are essential for maintaining the integrity of the system if there is a back out request, or for recovery, if there is a system or media failure.
Message manager	Processes all IBM MQ API requests.
Storage manager	Manages storage, for example, storage pool allocation, expansion, and deallocation.
Topic manager	Manages the topic and subscription information

Coupling facility SMDS manager Manages the shared message data sets (SMDS) for large messages stored in the coupling facility.

IBM MQ statistics are written to SMF as SMF type 115 records. The following subtypes can be present:

1

System information, for example, related to the logs and storage.

2

Information about number of messages and paging information. Queue sharing group information related to the coupling facility and Db2.

5 and 6

Detailed information about internal storage usage in the queue manager address space. While you can view this information, some of it is intended only for IBM use.

7

Storage manager summary information. While you can view this information, some of it is intended only for IBM use.

201

Page set input/output information

215

Buffer pool information

216

Queue information

231

System information for the channel initiator address space.

Note that:

- Subtype 1, 2, 201, and 215 records are created with statistics trace class 1.
- Subtype 5, 6, and 7 records are created with statistics trace class 3.
- Subtype 231 records are created with statistics trace class 4.
- Subtype 216 records are created with statistics trace class 5.

The subtype is specified in the SM115STF field (shown in [Table 34 on page 307](#)).

SMF 유형 115 및 유형 116 레코드를 수집하는 데 필요한 명령

이 주제를 유형 115 및 유형 116 SMF 레코드를 수집하는 데 필요한 명령에 대한 참조로 사용하십시오.

START TRACE 명령 사용

다음 옵션과 함께 [START TRACE](#) 명령의 동적 버전을 사용하여 레코드를 수집하십시오.

- [START TRACE\(STAT\) DEST SMF CLASS\(*\)](#) 및 [START TRACE\(ACCTG\) DEST SMF CLASS\(*\)](#) 는 클래스 1-3에 대한 추적을 시작합니다.
- [START TRACE\(STAT\) DEST SMF CLASS\(4\)](#) 및 [START TRACE\(ACCTG\) DEST SMF CLASS\(4\)](#) 는 각각 채널 시작기 통계 및 채널 회계 데이터를 시작합니다.

클래스 4 채널 시작기 정보에 대한 자세한 정보는 [채널 시작기 SMF 데이터 계획](#) 을 참조하십시오.

- [START TRACE\(STAT\) DEST SMF CLASS\(5\)](#) 는 큐 통계를 시작합니다.



주의: 명령의 콘솔 버전을 사용 중인 경우, 실행하는 명령의 시작 부분에 `cpf` 를 추가하십시오.

CLASS (4) 채널 통계의 경우, CHANNEL 정의에서 STACHL 속성을 설정해야 합니다. 자세한 정보는 [132 페이지의 『채널 통계 콜렉션 제어』](#) 의 내용을 참조하십시오.

CLASS (5) 큐 통계의 경우, QUEUE 및/또는 QMGR 정의에서 STATQ 속성을 설정해야 합니다. 자세한 정보는 [131 페이지의 『큐 통계 콜렉션 제어』](#) 의 내용을 참조하십시오.

SMF 유형 115레코드- IBM MQ 통계

현재 통계 수집을 확인하려면 DISPLAY TRACE 명령 DISPLAY TRACE (STAT)을 실행하십시오. 다음과 같이 표시되어야 합니다.

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
02 STAT 01,02,03,04 SMF * *
END OF TRACE REPORT
```

참고: ZPARM SMFSTAT 또는 ZPARM SMFACCT 속성을 별표로 설정하면 이제 클래스 1, 2 및 3만 제어합니다. START TRACE 명령을 사용하여 클래스 4 및 5를 켜야 합니다.

SMF 유형 116클래스 3 및 4데이터- IBM MQ 태스크 및 채널 회계 레코드

현재 통계 수집을 확인하려면 DISPLAY TRACE 명령 DISPLAY TRACE (ACCT)을 실행하십시오. 다음과 같이 표시되어야 합니다.

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
03 ACCTG * SMF * *
END OF TRACE REPORT
CSQ9022I QML1 CSQWVC1 'DISPLAY TRACE' NORMAL COMPLETION
```

다음 명령을 사용하여 큐 관리자의 주기가 필요하지 않으므로 SMF116 데이터 콜렉션을 동적으로 켜고 끌 수 있습니다.

```
START TRACE (ACCTG) DEST (SMF) CLASS (3)
START TRACE (ACCTG) DEST (SMF) CLASS (4)

STOP TRACE (ACCTG) DEST (SMF) CLASS (3)
STOP TRACE (ACCTG) DEST (SMF) CLASS (4)
```

SMF를 생성하려면 채널 회계 데이터에 추가 설정이 필요합니다. 그 단계는 다음과 같습니다.

- 다음 명령을 실행하여 모든 송신자 수신자 및 클라이언트 연결에 대한 정보 수집을 켜십시오.

```
ALTER QMGR STATCHL(HIGH)
```

- 클러스터링이 사용 중인 경우 명령을 실행하여 자동 정의된 클러스터 채널에 대한 정보 수집을 켜십시오.

```
ALTER QMGR STATCLS(HIGH)
```

참고: 앞의 명령문에서 HIGH, MED 또는 LOW의 값은 동일한 효과를 갖습니다.

- 모든 채널에 QMGR로 설정된 STATCHL이 있는지 확인하십시오.

```
DISPLAY CHANNEL(*) STATCHL
```

STATIME 확인



주의: 기본값은 30분이지만 SET SYSTEM 명령을 사용하여 최신 도구를 준수하도록 간격을 5분 이하로 설정해야 합니다.

큐 공유 그룹 (QSG)을 평가하는 경우 STATIME이 QSG에서 일치하는지 확인하십시오.

- 다음 DISPLAY SYSTEM 명령을 사용하여 시스템 설정을 표시하십시오.

```
DISPLAY SYSTEM
```

- STATIME값이 0인 경우 (통계 간격이 기본 LPAR값으로 설정됨을 의미함) 이는 일반적으로 정상입니다.

SMF 데이터 프로덕션의 다양한 지속 기간을 작성하는 매우 높은 볼륨의 LPAR 인스턴스가 있습니다. 데이터의 평가에서 광범위하게 다양한 지속 기간이 있는 경우 모든 큐 관리자에 대해 STATIME을 0이 아닌 값으로 설정하십시오.

- 다음 예제는 STATIME 간격을 5분으로 설정합니다.

```
SET SYSTEM STATIME(05)
```

또는 IBM MQ 9.2.4 이상인 경우:

```
SET SYSTEM STATIME(05.00)
```

참고: 새 간격은 현재 간격이 경과할 때까지 적용되지 않으므로 데이터 콜렉션을 시작하기 전에 변경해야 합니다.

릴리스 레벨 IBM MQ 9.2.4 이상의 큐 관리자에 대한 ACCTIME값 확인:

- ACCTIME이 -1로 설정되면 STATIME 속성과 동일합니다.
- 이 평가가 QSG에 대한 것이면 ACCTIME이 QSG에서 일치하는지 확인하십시오. 그렇지 않으면 데이터 수집을 시작하기 전에 값을 동일한 간격으로 설정하십시오.
- ACCTIME이 30이상으로 설정된 경우 SET SYSTEM 명령을 사용하여 15이하로 변경하십시오. 다음 예제에서는 ACCTIME 간격을 5분으로 설정합니다.

```
SET SYSTEM ACCTIME(05.00)
```

참고: 새 간격은 현재 간격이 경과할 때까지 적용되지 않으므로 데이터 콜렉션을 시작하기 전에 변경해야 합니다.

큐 관리자에서 ACCTQ 설정 확인

- 다음 명령을 사용하여 조사 중인 큐 관리자에 대한 ACCTQ 설정을 표시하십시오.

```
DISPLAY QMGR ACCTQ
```

- 값이 ACCTQ (ON) 인 경우 추가 조치가 필요하지 않습니다. 그렇지 않으면 다음 명령을 실행하십시오.

```
ALTER QMGR ACCTQ(ON)
```

- 클러스터링이 사용 중인 경우 SYSTEM.CLUSTER.TRANSMIT.QUEUE 및 이름 지정된 다른 클러스터는 큐 관리자가 호스트하는 큐를 전송합니다. 클러스터 전송 큐에 대해 계정이 사용 가능한지 여부를 판별하려면 다음을 수행하십시오.
 - 다음 명령을 사용하여 하나 이상의 큐에 대한 ACCTQ 설정을 표시하십시오.

```
DISPLAY QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ
```

- 값이 ACCTQ (ON) 인 경우 추가 조치가 필요하지 않습니다. 그렇지 않으면 다음 명령을 실행하십시오.

```
ALTER QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ(ON)
```

- 기타 모든 고용량 큐의 경우, ACCTQ의값이 ON인지 확인하거나 ACCTG를 QMGR로 설정하십시오.

Layout of an SMF type 115 record

You can use this section as a reference for the format of an SMF type 115 record.

The standard layout for SMF records involves three parts:

SMF header

Provides format, identification, and time and date information about the record itself.

Self-defining section

Defines the location and size of the individual data records within the SMF record.

Data records

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

Related reference

[“The SMF header” on page 307](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 307](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 309](#)

Use this topic to understand some example SMF records.

The SMF header

Use this topic as a reference for the format of the SMF header.

Table 34 on page 307 shows the format of SMF record header (SM115).

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM115	SMF record header.	
0	0	Integer	2	SM115LEN	SMF record length.	14A0
2	2		2		Reserved.	
4	4	Integer	1	SM115FLG	System indicator.	5E
5	5	Integer	1	SM115RTY	Record type. The SMF record type, for IBM MQ statistics records this is always 115 (X'73').	73
6	6	Integer	4	SM115TME	Time when SMF moved record.	00355575
10	A	Integer	4	SM115DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM115SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM115SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM115STF	Record subtype.	0002
24	18	Character	3	SM115REL	IBM MQ version.	F6F0F0 (600)
27	1B		1		Reserved	
28	1C	Character	0	SM115END	End of SMF header and start of self-defining section.	

Self-defining sections

Use this topic as a reference for format of the self-defining sections of the SMF record.

A self-defining section of a type 115 SMF record tells you where to find a statistics record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at fixed offsets from the start of the SMF record. Each statistics record can be identified by an eye-catcher string.

The following types of self-defining section are available to users for type 115 records. Each self-defining section points to statistics data related to one of the IBM MQ components. Table 35 on page 308 summarizes the sources of the statistics, the eye-catcher strings, and the offsets of the self-defining sections from the start of the SMF record header.

Table 35. Offsets to self-defining sections

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Eye-catcher of data
		Dec	Hex	
Storage manager	1	100	X'64'	QSST
Log manager	1	116	X'74'	QJST
Message manager	2	36	X'24'	QMST
Data manager	2	44	X'2C'	QIST
No longer used. The self-defining section will be binary zeros.	2	52	X'34'	
Lock manager	2	60	X'3C'	QLST
Db2 manager	2	68	X'44'	Q5ST
Coupling Facility manager	2	76	X'4C'	QEST
Topic manager	2	84	X'54'	QTST
SMDS usage	2	92	X'5C'	QESD
Buffer manager - one for each buffer pool	215	36	X'24'	QPST
Channel initiator	231			QWSX
Data manager page set - one for each page set	201	36	X'24'	QIS1
Storage manager	5	36	X'24'	QSPH
Storage manager	6	36	X'24'	QSGM
Storage manager	7	36	X'24'	QSRS
Queues	216	36	X'24'	QQST

Note: Some of the storage manager information in subtype 5, 6 and 7 records is intended only for IBM use. Other self-defining sections that are not listed contain data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssllllnnnn
```

where:

- ssssssss is a fullword containing the offset from the start of the SMF record.
- llll is a halfword giving the length of this data record.
- nnnn is a halfword giving the number of data records in this SMF record.

For more information see, “Examples of SMF statistics records” on page 309.

Note: Always use offsets in the self-defining sections to locate the statistics records.

z/OS Examples of SMF statistics records

Use this topic to understand some example SMF records.

Figure 20 on page 309 shows an example of part of the SMF record for subtype 1. Subtype 1 includes the storage manager and log manager statistics records. The SMF record header is shown underlined.

The self-defining section at offset X'64' refers to storage manager statistics and the self-defining section at offset X'74' refers to log manager statistics, both shown in **bold**.

The storage manager statistics record is located at offset X'0000011C' from the start of the header and is X'48' bytes long. There is one set of storage manager statistics, identified by the eye-catcher string QSST. The start of this statistics record is also shown in the example.

The log manager statistics record is located at offset X'00000164' from the start of the header and is X'78' bytes long. There is one set of log manager statistics, identified by the eye-catcher string QJST.

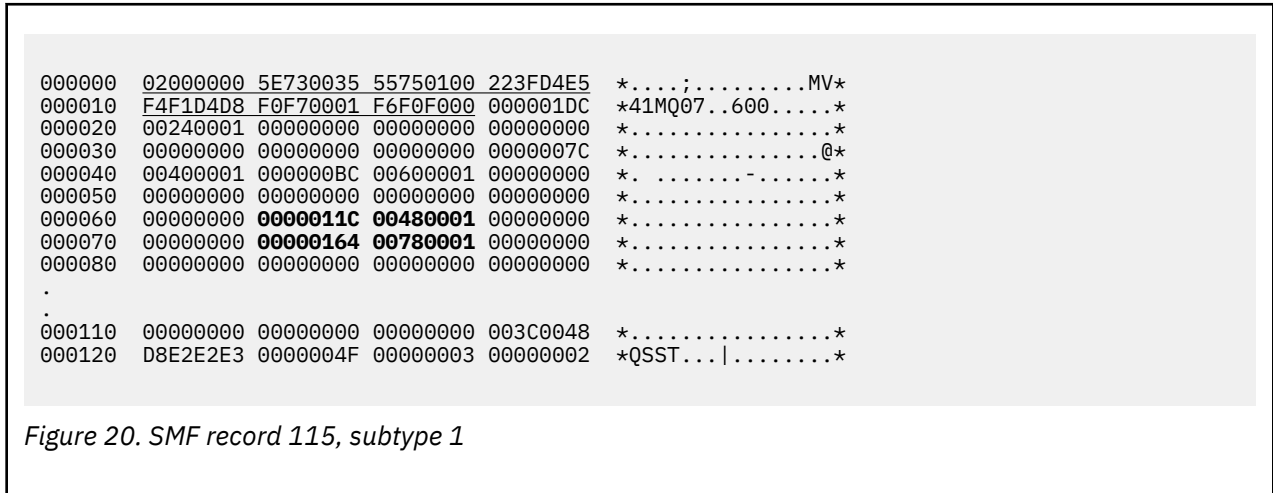


Figure 20. SMF record 115, subtype 1

Figure 21 on page 310 shows an example of part of the SMF record for subtype 2. Subtype 2 includes the statistics records for the message, data, lock, coupling facility, topic, and Db2 managers. The SMF record header is shown underlined; the self-defining sections are shown alternately **bold** and *italic*.

- The self-defining section at offset X'24' refers to message manager statistics. The message manager statistics record is located at offset X'00000064' from the start of the header and is X'48' bytes long. There is one set of these statistics, identified by the eye-catcher string QMST.
- The self-defining section at offset X'2C' refers to data manager statistics. The data manager statistics record is located at offset X'000000AC' from the start of the header and is X'50' bytes long. There is one set of these statistics, identified by the eye-catcher string QIST.
- In releases prior to IBM MQ 9.1.0, the self-defining section at offset X'34' referred to buffer manager statistics. As this SMF record was taken from an IBM MQ 9.1.0 queue manager, the buffer manager self-defining section is set to zeros to indicate that there are no buffer manager statistics. Instead, these statistics are in SMF 115 subtype 215 records.
- The self-defining section at offset X'3C' refers to lock manager statistics. The lock manager statistics record is located at offset X'000000FC' from the start of the header and is X'20' bytes long. There is one set of these statistics, identified by the eye-catcher string QLST.
- The self-defining section at offset X'44' refers to Db2 manager statistics. The Db2 manager statistics record is located at offset X'0000011C' from the start of the header and is X'2A0' bytes long. There is one set of these statistics, identified by the eye-catcher string Q5ST.
- The self-defining section at offset X'4C' refers to coupling facility manager statistics. The coupling facility manager statistics record is located at offset X'000003BC' from the start of the header and is X'1008' bytes long. There is one set of these statistics, identified by the eye-catcher string QEST.
- The self-defining section at offset X'54' refers to topic manager statistics. The topic manager statistics record is located at offset X'000013C4' from the start of the header and is X'64' bytes long. There is one set of these statistics, identified by the eye-catcher string QTST.

- The self-defining section at offset X'5C' is for SMDS statistics. This self defining section is set to zeros indicating that SMDS is not being used.

```

000000 09F40000 5E730033 4DBE0113 142FD4E5 *.4...;...(. ....MV*
000010 F4F1D4D8 F2F10002 F9F1F000 00001428 *41MQ21..910.....*
000020 00240001 00000064 00480001 000000AC *. ....*
000030 00500001 00000000 00000000 000000FC *. ....*
000040 00200001 0000011C 02A00001 000003BC *. ....*
000050 10080001 000013C4 00640001 00000000 *. ....D.....*
000060 00000000 D40F0048 D8D4E2E3 00000000 *. ...M...QMST....*
000080 00000000 00000000 00000000 00000000 *. ....*
000090 00000000 00000000 00000000 00000000 *. ....*
0000A0 00000000 00000000 00000000 C90F0050 *. ....I..&*
0000B0 D8C9E2E3 00000000 00000000 00000000 *QIST.....*
0000C0 00000000 00000000 00000000 00000000 *. ....*
0000D0 00000000 00000000 00000000 00000000 *. ....*
0000E0 00000000 00000000 00000000 00000000 *. ....*
0000F0 00000000 00000000 00000000 D30F0020 *. ....L...*
000100 D8D3E2E3 00000000 00000000 00000000 *QLST.....*
000110 00000000 00000000 00000000 F50F02A0 *. ....5...*
000120 D8F5E2E3 00000008 00000000 00000000 *Q5ST.....*
.
.

```

Figure 21. SMF record 115, subtype 2

z/OS Processing type 115 SMF records

Use this topic as a reference for processing type 115 SMF records.

You must process any data you collect from SMF to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the values of the following fields:

- SM115RTY, the SMF record number, must be X'73' (115)
- SM115STF, the record subtype, must be 0001, 0002, 0005, 0006, 0007, 0201, 0215, or 0231

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility” on page 299](#).

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

z/OS Storage manager data records

Use this topic as a reference for storage manager data records.

The format of the storage manager statistics record is described in assembler macro thlqual.SCSQMACS(CSQDQSST).

The data contains information about the number of fixed and variable storage pools that the queue manager has allocated, expanded, contracted, and deleted during the statistics interval, plus the number of GETMAIN, FREEMAIN, and STORAGE requests to z/OS, including a count of those requests that were unsuccessful. Additional information includes a count of the number of times the short-on-storage condition was detected and a count of the number of abends that occurred as a result of that condition.

Additional data about storage usage in the queue manager is produced by class 2 and class 3 statistics trace. While you can view this information, some of it is intended only for IBM use.

- The format of the storage manager pool header statistics record, which is present in subtype 5 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSPH).
- The format of the storage manager getmain statistics record, which is present in subtype 6 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSGM).
- The format of the storage manager region summary record, which is present in subtype 7 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSRs).

z/OS 로그 관리자 데이터 레코드

이 주제를 로그 관리자 데이터 레코드 형식에 대한 참조로 사용하십시오.

로그 관리자 통계 레코드의 형식은 어셈블러 매크로 thlqual.SCSQMACS(CSQDQJST)에 설명되어 있습니다.

통계에서 다음 수치가 중요합니다.

1. 로그 쓰기 요청의 총 수:

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. 로그 읽기 요청의 총 수:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

로그 관리자 통계를 사용하여 조사할 수 있는 문제점 증상은 다음 표에 설명되어 있습니다.

<p>증상 1 QJSTWTB가 0이 아닙니다.</p> <p>원인 인스토키지 버퍼가 활성 로그에 기록되는 동안 태스크가 일시중단됩니다. 활성 로그 쓰기에 문제가 있는 것일 수 있습니다. CSQ6LOGP 내의 OUTBUFF 매개변수가 너무 작습니다.</p> <p>Action 활성 로그 쓰기에 관한 문제점을 조사하십시오. CSQ6LOGP 내 OUTBUFF 매개변수의 값을 늘리십시오.</p>
<p>증상 2 QJSTWTL/N_{logread} 비율이 1%보다 큼니다.</p> <p>원인 활성 로그에서 읽어야 하는 로그 읽기가 시작되었지만 MAXRTU 데이터 세트가 이미 할당되었기 때문에 IBM MQ가 데이터 세트를 할당할 수 없습니다.</p> <p>Action MAXRTU을 증가시키십시오.</p>

증상 3

QJSTRARH/N logread 비율이 정상보다 큼니다.

원인

대부분의 로그 읽기 요청은 출력 버퍼나 활성 로그에서 나와야 합니다. 백아웃에 대한 요청을 충족시키기 위해 복구 단위 레코드는 인스토키지 버퍼, 활성 로그, 아카이브된 로그에서 읽습니다.

시간이 몇 분에 달하는 장기 실행 복구 단위에는 여러 다른 로그에 분산된 로그 레코드가 있을 수 있습니다. 이는 로그 레코드를 복구하기 위해 추가 작업을 수행해야 하므로 성능을 저하시킵니다.

Action

복구 단위의 길이를 줄이도록 애플리케이션을 변경하십시오. 또한 활성 로그의 크기를 늘려서 단일 복구 단위 가 둘 이상의 로그에 분산될 가능성을 줄일 것을 고려하십시오.

기타 포인터

N logread /N logwrite 비율은 백아웃해야 하는 작업의 양을 표시합니다.

증상 4

QJSTLLCP가 시간당 10을 초과합니다.

원인

사용 중인 시스템에서는 일반적으로 시간당 10개의 체크포인트가 있을 것으로 예상합니다. QJSTLLCP 값이 이보다 크면 이는 큐 관리자의 설정 문제점을 나타냅니다.

가장 가능한 원인은 CSQ6SYSP의 LOGLOAD 매개변수가 너무 작기 때문입니다. 체크포인트를 야기하는 기타 이벤트는 활성 로그가 가득 차서 다음 활성 로그 데이터 세트로 전환될 때입니다. 로그가 너무 작은 경우 이는 빈번한 체크포인트를 유발할 수 있습니다.

QJSTLLCP는 체크포인트의 총 수입입니다.

Action

LOGLOAD 매개변수를 증가시키거나 필요에 따라 로그 데이터 세트의 크기를 늘리십시오.

증상 5

QJSTCmpFail > 0 또는 QJSTCmpComp가 QJSTCmpUncmp보다 훨씬 작지 않음

원인

큐 관리자가 현저하게 로그 레코드를 압축할 수 없습니다.

QJSTCmpFail은 큐 관리자가 레코드 길이 축소를 수행할 수 없는 횟수입니다. 이 수치를 QJSTCmpReq(압축 요청 수)와 비교하여 실패 수가 유효한지 확인해야 합니다.

QJSTCmpComp는 로그에 쓴 압축된 바이트의 총계이고 QJSTCmpUncmp는 압축 전의 바이트 총계입니다. 두 총계 모두 압축하기에 적합한 로그 레코드에 대해 기록된 바이트는 포함하지 않습니다. 이 수치가 유사하면 압축의 이점이 거의 없는 것입니다.

Action

로그 압축을 끄십시오. SET LOG COMPLOG(NONE) 명령을 발행하십시오. 자세한 내용은 SET LOG 명령을 참조하십시오.

참고: 시스템 시동 후에 생성된 첫 번째 통계 세트에는 인플라이트 복구 단위의 해결로 인한 중요한 로그 활동이 있을 수 있습니다.

zHyper쓰기에 대한 변경사항

▶ V 9.4.0

IBM MQ 9.3.5 부터 zHyper쓰기 처리는 QJSTHWC 및 QJSTHWE 통계의 동작을 변경합니다.

IBM MQ 9.3.5 이전:

- QJSTHWC는 zHyper쓰기 가능 볼륨에 있는 SMF 간격에서 사용되는 로그 데이터 세트의 수입입니다. 볼륨의 zHyper쓰기 기능은 큐 관리자가 시작될 때 확보되며 시간이 지남에 따라 변경될 수 있으므로 이 정보는 시간이 경과될 수 있습니다.

- QJSTHWE는 zHyper쓰기를 사용하여 작성되는 SMF 간격에서 사용되는 로그 데이터 세트의 수입입니다. 이는 로그 데이터 세트가 zHyper쓰기 가능 볼륨에 있고 큐 관리자가 ZHYWRITE (YES) 를 설정하여 zHyperWrite에 대해 사용으로 설정된 경우에 발생합니다.

IBM MQ 9.3.5부터:

- QJSTHWC는 zHyper쓰기 가능 볼륨에 있는 SMF 간격에서 사용되는 로그 데이터 세트의 수입입니다. 볼륨의 zHyper쓰기 기능은 큐 관리자가 시작될 때 확보되며 시간이 지남에 따라 변경될 수 있으므로 이 정보는 시간이 경과될 수 있습니다.
- QJSTHWE는 zHyper쓰기를 사용하여 작성되는 SMF 간격에서 사용되는 로그 데이터 세트의 수입입니다. 이는 큐 관리자가 ZHYWRITE (YES) 를 설정하여 zHyperWrite에 대해 사용으로 설정된 경우에 발생합니다.



주의: 볼륨의 zHyper쓰기 기능은 시간이 지남에 따라 변경될 수 있습니다. IBM MQ 9.3.5 부터는 QJSTHWE가 QJSTHWC보다 큰 시나리오가 발생할 수 있습니다.

zHyper링크 통계

V 9.4.0

IBM MQ 9.4.0부터는 zHyperLink의 성능을 추적하기 위해 로그 관리자 통계가 추가되었습니다.

다음 통계가 QJST에 추가되었습니다.

이름	유형	길이 바이트	설명
QJSTHLSCIW	부호 없는 정수	4	zHyperLink를 요청한 단일 페이지 쓰기 수입입니다.
QJSTHLMCIW	부호 없는 정수	4	zHyper링크를 요청한 다중 페이지 쓰기 수입입니다.
QJSTHLCICNTW	부호 없는 정수	4	쓰기에서 zHyper링크를 요청한 위치에 쓴 페이지 수입입니다.
QJSTHLSCIS	부호 없는 정수	4	zHyperLink를 성공적으로 사용한 페이지 쓰기 수입입니다. 즉, 쓰기가 동기적으로 수행되었습니다.
QJSTHLMCIS	부호 없는 정수	4	zHyperLink를 사용한 다중 페이지 쓰기 수입입니다. 즉, 쓰기가 동기적으로 수행되었습니다.
QJSTHLCICNTS	부호 없는 정수	4	zHyperLink를 사용하여 성공적으로 쓴 페이지 수입입니다. 즉, 쓰기가 동기적으로 수행되었습니다.
QJSTHLSCIF	부호 없는 정수	4	zHyperLink를 사용하려고 시도했지만 동기적으로 쓰기를 수행할 수 없는 단일 페이지 쓰기 수입입니다. 대신 쓰기가 비동기식으로 성공적으로 수행되었습니다.
QJSTHLMCIF	부호 없는 정수	4	zHyperLink를 사용하려고 시도했지만 동기적으로 쓰기를 수행할 수 없는 다중 페이지 쓰기 수입입니다. 대신 쓰기가 비동기식으로 성공적으로 수행되었습니다.
QJSTHLCICNTF	부호 없는 정수	4	쓰기가 zHyperLink를 사용하려고 시도했지만 동기적으로 쓰기를 수행할 수 없는 기록된 페이지 수입입니다. 대신 쓰기가 비동기식으로 성공적으로 수행되었습니다.
QJSTHLSCICONT	부호 없는 정수	4	zHyper링크 쓰기 세션이 설정되기 전의 단일 페이지 쓰기 수입입니다. 314 페이지의 『1』의 내용을 참조하십시오.
QJSTHLMCICONT	부호 없는 정수	4	zHyper링크 쓰기 세션이 설정되기 전의 다중 페이지 쓰기 수입입니다. 314 페이지의 『1』의 내용을 참조하십시오.
QJSTHLCONCICNT	부호 없는 정수	4	zHyper링크 연결이 설정되기 전에 쓴 페이지 수입입니다. 314 페이지의 『1』의 내용을 참조하십시오.

이름	유형	길이 바이트	설명
최대 QJSTHLSCIWTM AX	부호 없는 정수	8	간격에서 가장 긴 zHyper링크 쓰기 시간, 단일 페이지입니다.
QJSTHLSCIWTM IN	부호 없는 정수	8	간격에서 가장 짧은 zHyper링크 쓰기 시간입니다 (단일 페이지).
QJSTHLSCIWTT OT	부호 없는 정수	8	간격 내의 총 zHyper링크 쓰기 시간, 단일 페이지입니다.
최대 QJSTHLMCIWT MAX	부호 없는 정수	8	간격에서 가장 긴 zHyper링크 쓰기 시간, 다중 페이지.
QJSTHLMCIWT MIN	부호 없는 정수	8	간격에서 가장 짧은 zHyper링크 쓰기 시간입니다 (다중 페이지).
QJSTHLMCIWTT OT	부호 없는 정수	8	간격 내의 총 zHyper링크 쓰기 시간 (다중 페이지)입니다.
QJSTHLIOSQU	부호 없는 정수	16	단일 페이지 쓰기에 대한 zHyper링크 쓰기 시간의 제곱합입니다.
QJSTHLC	부호 없는 정수	4	zHyper링크가 가능한 이 SMF 간격에서 사용되는 새 로그의 수입입니다.
큐	부호 없는 정수	4	이 SMF 간격에서 사용되는 zHyper링크가 사용 가능한 새 로그의 수입입니다.

참고:

1. 새 활성 로그 사본으로 전환할 때 DASD를 사용하여 zHyper링크 쓰기 세션이 설정되기 전에 일정 기간이 있을 수 있습니다.

Message manager data records

Use this topic as a reference for message manager data records.

The format of the message manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQMST)`.

The data gives you counts of different IBM MQ API requests.

Data manager data records

Use this topic as a reference for the format of the Data Manager data records.

The format of the data manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIST)`.

The data gives you counts of different object requests.

Data manager page set data records

Use this section as a reference for the format of the data manager page set data records

The format of the data manager page set statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIS1)`.

The page set usage information helps to facilitate better management of local queues within a queue manager by recording information such as page set input-output rates and highest usage.

The data provides the same basic page set information as output by the MQSC **DISPLAY USAGE TYPE (PAGESET)** command, or the PCF Inquire Usage (**MQCMD_INQUIRE_USAGE**) command.

For example:

- The total pages
- The current used pages
- Unused persistent and nonpersistent pages
- Expansion method
- Number of extends
- Number of stripes

The data also provides some performance indicators, together with other performance information. For example:

- How many times deferred write, immediate write, and read page I/O requests happened during the SMF interval,
- The number of pages moved, elapsed time and number of read and write operations.
- How many pages written in checkpoints.
- Has expansion occurred during the SMF interval?
- How many times the page set became full.
- An indication of where new space is being allocated within the page set.

From the information displayed, you should be able to understand the general status of each page set, and consider whether you need to retune the system.

Related reference

[“The SMF header” on page 307](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 307](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 309](#)

Use this topic to understand some example SMF records.

Buffer manager data records

Use this topic as a reference for the format of buffer manager data records.

The format of the buffer manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQPST)`.

Note: Buffer manager statistics records will only be created for buffer pools that are defined. If a buffer pool is defined but not used then no values will be set and its buffer manager statistics record will not contain any data.

For information about efficiently managing your buffer pools, see [“Managing your buffer pools” on page 317](#).

When interpreting the statistics, you are recommended to consider the following factors because the values of these fields can be used to improve the performance of your system:

1. If QPSTSOS, QPSTDMC, or QPSTIMW is greater than zero, you should either increase the size of the buffer pool or reallocate the page sets to different buffer pools.
 - QPSTSOS is the number of times that there were no buffers available for page get requests. If QPSTSOS ever becomes nonzero, it shows that IBM MQ is under severe stress. The buffer pool size should be increased. If increasing the buffer pool size does not make the value of QPSTSOS zero, there might be I/O contention on the DASD page sets.
 - QPSTDMC is the number of updates that were performed synchronously because there was either more than 95% of the pages in the buffer pool waiting for write I/O, or there was less than 5% of the buffer pool available for read requests. If this number is not zero, the buffer pool might be too small

and should be enlarged. If increasing the buffer pool size does not reduce QPSTDMC to zero, there might be I/O contention on the DASD page sets.

- QPSTIMW is a count of the number of times pages were written out synchronously. If QPSTDMC is zero, QPSTIMW is the number of times pages were found on the queue waiting for write I/O that had been there for at least two checkpoints.

2. For buffer pool zero and buffer pools that contain short-lived messages:

- QPSTDWT should be zero, and the percentage QPSTCBSL/QPSTNBUF should be greater than 15%.
QPSTDWT is the number of times the asynchronous write processor was started because there was either more than 85% of the pages in the buffer pool waiting for write I/O, or there was less than 15% of the buffer pool available for read requests. Increasing the buffer pool size should reduce this value. If it does not, the pattern of access is one of long delays between puts and gets.
- QPSTTPW might be greater than zero due to checkpointing activity.
- QPSTRIO should be zero unless messages are being read from a page set after the queue manager is restarted.

The ratio of QPSTRIO to QPSTGETP shows the efficiency of page retrieval within the buffer pool. Increasing the buffer pool size should decrease this ratio and, therefore, increase the page retrieval efficiency. If this does not happen, it indicates that pages are not being frequently reaccessed. This implies a transaction pattern where there is a long delay between messages being put and then later retrieved.

The ratio of QPSTGETN to QPSTGETP indicates the number of times an empty page, as opposed to a non-empty page, has been requested. This ratio is more an indication of transaction pattern, than a value that can be used to tune the system.

- If QPSTSTL has a value greater than zero, this indicates that pages that have not been used before are now being used. This might be caused by an increased message rate, messages not being processed as fast as they were previously (leading to a buildup of messages), or larger messages being used.

QPSTSTL is a count of the number of times a page access request did not find the page already in the buffer pool. Again, the lower the ratio of QPSTSTL to (QPSTGETP + QPSTGETN) is, the higher the page retrieval efficiency. Increasing the buffer pool size should decrease this ratio but, if it does not, it is an indication that there are long delays between puts and gets.

- You are recommended to have sufficient buffers to handle your peak message rate.

3. For buffer pools with long-lived messages, where there are more messages than can fit into the buffer pool:

- $(QPSTRIO + QPSTWIO) / \text{Statistics interval}$ is the I/O rate to page sets. If this value is high, you should consider using multiple page sets on different volumes to allow I/O to be carried out in parallel.
- Over the period of time that the messages are processed (for example, if messages are written to a queue during the day and processed overnight) the number of read I/Os (QPSTRIO) should be approximately the total number of pages written (QPSTTPW). This shows that one page is read for every page written.

If QPSTRIO is much larger than QPSTTPW, this shows that pages are being read in multiple times. This might be a result of the application using MQGET by *MsgId* or *CorrelId* when the queue is not indexed, or browsing messages on the queue using get next.

The following actions might relieve this problem:

- a. Increase the size of the buffer pool so that there are enough pages to hold the queue, in addition to any changed pages.
- b. Use the INDXTYPE queue attribute, which allows a queue to be indexed by *MsgId* or *CorrelId* and eliminates the need for a sequential scan of the queue.
- c. Change the design of the application to eliminate the use of MQGET with *MsgId* or *CorrelId*, or the get next with browse option.

Note: Applications using long-lived messages typically process the first available message and do not use MQGET with *MsgId* or *CorrelId*, and they might browse only the first available message.

- d. Move page sets to a different buffer pool to reduce contention between messages from different applications.

Managing your buffer pools

To manage your buffer pools efficiently, you must consider the factors that affect the buffer pool I/O operations and also the statistics associated with the buffer pools.

The following factors affect buffer pool I/O operations.

- If a page containing the required data is not found in the buffer pool, it is read in synchronously to an available buffer from its DASD page set.
- Whenever a page is updated, it is put on an internal queue of pages to be (potentially) written out to DASD. This means that the buffer used by that page is unavailable for use by any other page until the buffer has been written to DASD.
- If the number of pages queued to be written to DASD exceeds 85% of the total number of buffers in the pool, an asynchronous write processor is started to put the buffers to DASD.

Similarly, if the number of buffers available for page get requests become less than 15% of the total number of buffers in the pool, the asynchronous write processor is started to perform the write I/O operations.

The write processor stops when the number of pages queued to be written to DASD has fallen to 75% of the total number of buffers in the pool.

- If the number of pages queued for writing to DASD exceeds 95% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

Similarly, if the number of buffers available for page get requests becomes less than 5% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

- If the number of buffers available for page get requests ever reaches zero, a transaction that encounters this condition is suspended until the asynchronous write processor has finished.
- If a page is frequently updated, the page spends most of its time on the queue of pages waiting to be written to DASD. Because this queue is in least recently used order, it is possible that a frequently updated page placed on this least recently used queue is never written out to DASD. For this reason, at the time of update, if the page is found to have been waiting on the write operation to DASD queue for at least two checkpoints, it is synchronously written to DASD. Updating occurs at checkpoint time and is suspended until the asynchronous write processor has finished.

The aim of this algorithm is to maximize the time pages spend in buffer pool memory while allowing the system to function if the system load puts the buffer pool usage under stress.

Lock manager data records

Use this topic as a reference to the format of the lock manager data records.

The format of the lock manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQLST)`.

The records contain data about the following information:

- The number of lock get requests and lock release requests.
- The number of times a lock get request determined that the requested lock was already held.

Db2 manager data records

Use this topic as a reference to the format of the Db2 manager data records.

The format of the Db2 manager statistics record is described in the following table and in assembler macro `thlqual.SCSQMACS(CSQDQ5ST)` and C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `q5st`, `q5stid`.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

<i>Table 36. Db2 statistics record (Q5ST)</i>					
Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	668	Q5ST	Db2 manager statistics
0	0	Bitstring	2	Q5STID	Control block identifier
2	2	Integer	2	Q5STLL	Control block length
4	4	Character	4	Q5STEYEC	Control block eye catcher
8	8	Character	660	Q5STZERO	QMST part cleared on occasion
8	8	Integer	4	NUMTASK	Number of server tasks
12	C	Integer	4	ACTTASK	Number of active server tasks
16	10	Integer	4	CONNCNT	Number of connect requests
20	14	Integer	4	DISCCNT	Number of disconnect requests
24	18	Integer	4	DHIGMAX	Max. request queue depth
28	1C	Integer	4	ABNDCNT	Number of Db2SRV task abends
32	20	Integer	4	REQUCNT	Number of requests queued
36	24	Integer	4	DEADCNT	Number of deadlock timeouts
40	28	Integer	4	DELECNT	Number of delete requests
44	2C	Integer	4	LISTCNT	Number of list requests
48	30	Integer	4	READCNT	Number of read requests
52	34	Integer	4	UPDTCNT	Number of update requests
56	38	Integer	4	WRITCNT	Number of write requests
60	3C	Integer	4	SCSSEL	SCST (shared-channel-status) selects
64	40	Integer	4	SCSINS	SCST inserts
68	44	Integer	4	SCSUPD	SCST updates
72	48	Integer	4	SCSDEL	SCST deletes
76	4C	Integer	4	SSKSEL	SSKT (shared-sync-key) selects
80	50	Integer	4	SSKINS	SSKT inserts
84	54	Integer	4	SSKDEL	SSKT deletes
88	58	Integer	4	SCSBFTS	SCST number of times buffer too small
92	5C	Integer	4	SCSMAXR	SCST maximum rows on query
96	60	Integer	4	* (2)	Reserved
104	68	Character	8	DELETCUW	Cumulative STCK difference - Thread delete
112	70	Character	8	DELETMXW	Maximum STCK difference - Thread delete
120	78	Character	8	DELESCUW	Cumulative STCK difference - SQL delete
128	80	Character	8	DELESMXW	Maximum STCK difference - SQL delete

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
136	88	Character	8	LISTTCUW	Cumulative STCK difference - Thread list
144	90	Character	8	LISTTMXW	Maximum STCK difference - Thread list
152	98	Character	8	LISTSCUW	Cumulative STCK difference - SQL list
160	A0	Character	8	LISTSMXW	Maximum STCK difference - SQL list
168	A8	Character	8	READTCUW	Cumulative STCK difference - Thread read
176	B0	Character	8	READTMXW	Maximum STCK difference - Thread read
184	B8	Character	8	READSCUW	Cumulative STCK difference - SQL read
192	C0	Character	8	READSMXW	Maximum STCK difference - SQL read
200	C8	Character	8	UPDTTCUW	Cumulative STCK difference - Thread update
208	D0	Character	8	UPDTTMXW	Maximum STCK difference - Thread update
216	D8	Character	8	UPDTSCUW	Cumulative STCK difference - SQL update
224	E0	Character	8	UPDTSMXW	Maximum STCK difference - SQL update
232	E8	Character	8	WRITTCUW	Cumulative STCK difference - Thread write
240	F0	Character	8	WRITTMXW	Maximum STCK difference - Thread write
248	F8	Character	8	WRITSCUW	Cumulative STCK difference - SQL write
256	100	Character	8	WRITSMXW	Maximum STCK difference - SQL write
264	108	Character	8	SCSSTCUW	Cumulative STCK difference - Thread select
272	110	Character	8	SCSSTMXW	Maximum STCK difference - Thread select
280	118	Character	8	SCSSSCUW	Cumulative STCK difference - SQL select
288	120	Character	8	SCSSSMXW	Maximum STCK difference - SQL select
296	128	Character	8	SCSITCUW	Cumulative STCK difference - Thread insert
304	130	Character	8	SCSITMXW	Maximum STCK difference - Thread insert
312	138	Character	8	SCSISCUW	Cumulative STCK difference - SQL insert
320	140	Character	8	SCSISMXW	Maximum STCK difference - SQL insert
328	148	Character	8	SCSUTCUW	Cumulative STCK difference - Thread update
336	150	Character	8	SCSUTMXW	Maximum STCK difference - Thread update
344	158	Character	8	SCSUSCUW	Cumulative STCK difference - SQL update
352	160	Character	8	SCSUSMXW	Maximum STCK difference - SQL update
360	168	Character	8	SCSDTCUW	Cumulative STCK difference - Thread delete
368	170	Character	8	SCSDTMXW	Maximum STCK difference - Thread delete
376	178	Character	8	SCSDSCUW	Cumulative STCK difference - SQL delete
384	180	Character	8	SCSDSMXW	Maximum STCK difference - SQL delete
392	188	Character	8	SSKSTCUW	Cumulative STCK difference - Thread select
400	190	Character	8	SSKSTMXW	Maximum STCK difference - Thread select

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
408	198	Character	8	SSKSSCUW	Cumulative STCK difference - SQL select
416	1A0	Character	8	SSKSSMXW	Maximum STCK difference - SQL select
424	1A8	Character	8	SSKITCUW	Cumulative STCK difference - Thread insert
432	1B0	Character	8	SSKITMXW	Maximum STCK difference - Thread insert
440	1B8	Character	8	SSKISCUW	Cumulative STCK difference - SQL insert
448	1C0	Character	8	SSKISMXW	Maximum STCK difference - SQL insert
456	1C8	Character	8	SSKDTCUW	Cumulative STCK difference - Thread delete
464	1D0	Character	8	SSKDTMXW	Maximum STCK difference - Thread delete
472	1D8	Character	8	SSKDSCUW	Cumulative STCK difference - SQL delete
480	1E0	Character	8	SSKDSMXW	Maximum STCK difference - SQL delete
488	1E8	Integer	4	LMSSEL	Number of Db2 BLOB read requests
492	1EC	Integer	4	LMSINS	Number of Db2 BLOB insert requests
496	1F0	Integer	4	LMSUPD	Number of Db2 BLOB update requests
500	1F4	Integer	4	LMSDEL	Number of Db2 BLOB delete requests
504	1F8	Integer	4	LMSLIS	Number of Db2 BLOB list requests
508	1FC	64 bit integer	8	LMSSTCUW	Total elapsed time for all thread read BLOB requests
516	204	64 bit integer	8	LMSSTMXW	Maximum elapsed time for a thread read BLOB request
524	20C	64 bit integer	8	LMSSSCUW	Total elapsed time for all SQL read BLOB requests
532	214	64 bit integer	8	LMSSSMXW	Maximum elapsed time for an SQL read BLOB request
540	21C	64 bit integer	8	LMSITCUW	Total elapsed time for all thread insert BLOB requests
548	224	64 bit integer	8	LMSITMXW	Maximum elapsed time for a thread insert BLOB request
556	22C	64 bit integer	8	LMSISCUW	Total elapsed time for all SQL insert BLOB requests
564	234	64 bit integer	8	LMSISMXW	Maximum elapsed time for an SQL insert BLOB request
572	23C	64 bit integer	8	LMSUTCUW	Total elapsed time for all thread update BLOB requests
580	244	64 bit integer	8	LMSUTMXW	Maximum elapsed time for a thread update BLOB request
588	24C	64 bit integer	8	LMSUSCUW	Total elapsed time for all SQL update BLOB requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
596	254	64 bit integer	8	LMSUSMXW	Maximum elapsed time for an SQL update BLOB request
604	25C	64 bit integer	8	LMSDTCUW	Total elapsed time for all thread delete BLOB requests
612	264	64 bit integer	8	LMSDTMXW	Maximum elapsed time for a thread delete BLOB request
620	26C	64 bit integer	8	LMSDSCUW	Total elapsed time for all SQL delete BLOB requests
628	274	64 bit integer	8	LMSDSMXW	Maximum elapsed time for an SQL delete BLOB request
636	27C	64 bit integer	8	LMSLTCUW	Total elapsed time for all thread list BLOB requests
644	284	64 bit integer	8	LMSLTMXW	Maximum elapsed time for a thread list BLOB request
652	28C	64 bit integer	8	LMSLSCUW	Total elapsed time for all SQL list BLOB requests
660	294	64 bit integer	8	LMSLSMXW	Maximum elapsed time for an SQL list BLOB request

The data contains counts for each request type that the Db2 resource manager supports. For these request types, maximum and cumulative elapse times are kept for the following:

- The time spent in the Db2 resource manager as a whole (called the thread time).
- The time that was spent performing the RRSF and SQL parts of the request (a subset of the thread time called the SQL time).

Information is also provided for:

- The number of server tasks attached.
- The maximum overall request depth against any of the server tasks.
- The number of times any of the server task requests terminated abnormally.

If the abnormal termination count is not zero, a requeue count is provided indicating the number of queued requests that were requeued to other server tasks as a result of the abnormal termination.

If the average thread time is significantly greater than the average SQL time, this might indicate that thread requests are spending an excessive amount of time waiting for a server task to process the SQL part of the request. If this is the case, examine the DHIGMAX field and, if the value is greater than one, consider increasing the number of Db2 server tasks specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro.

Coupling facility manager data records

Use this topic as a reference to the format of the coupling facility manager data records.

The format of the coupling facility manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQEST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qest, qestid.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Table 37. Coupling facility statistics record (QEST)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	4104	QEST	CF manager statistics
0	0	Bitstring	2	QESTID	Control block identifier
2	2	Integer	2	QESTLL	Control block length
4	4	Character	4	QESTEYEC	Control block eye catcher
8	8	Character	4096	QESTZERO	QEST part cleared on occasion
8	8	Character	64	QESTSTUC (0:63)	Array (one entry per structure)
8	8	Character	12	QESTSTR	Structure name
20	14	Integer	4	QESTSTRN	Structure number
24	18	Integer	4	QESTCSEC	Number of IXLLSTE calls
28	1C	Integer	4	QESTCMEC	Number of IXLLSTM calls
32	20	Character	8	QESTSSTC	Time spent doing IXLLSTE calls
40	28	Character	8	QESTMSTC	Time spent doing IXLLSTM calls
48	30	Integer	4	QESTRSEC	Number of IXLLSTE redrives
52	34	Integer	4	QESTRMEC	Number of IXLLSTM redrives
56	38	Integer	4	QESTSFUL	Number of structure fulls
60	3C	Integer	4	QESTMNUS	Maximum number of entries in use
64	40	Integer	4	QESTMLUS	Maximum number of elements in use
68	44	Character	4	*	Reserved
4104	1008	Character	0	*	End of control block

The data contains information for each coupling facility list structure, including the CSQ_ADMIN structure, that the queue manager could connect to during the statistics interval. The information for each structure includes the following:

- The number of and cumulative elapsed times for IXLLSTE and IXLLSTM requests.
- The number of times a request had to be retried because of a timeout.
- The number of times a 'structure full' condition occurred.

Topic manager data records

Use this topic as a reference to the format of the topic manager data records.

The format of the topic manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQTST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qtst, qtstid.

Table 38. Topic manager statistics record (QTST)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	96	QTST	Topic manager statistics
0	0	Bitstring	2	QTSTID	Control block identifier
2	2	Integer	2	QTSTLL	Control block length
4	4	Character	4	TESTEYEC	Control block eye catcher
8	8	Character	88	QTSTZERO	QTST part cleared on occasion
8	8	Integer	4	QTSTSTOT	Total subscription requests
12	0C	Integer	4	QTSTSDUR	Durable subscription requests
16	10	Integer	4	QTSTSHIG (1:3)	Subscription high water mark array (API, ADMIN, PROXY)
28	1C	Integer	4	QTSTSLOW (1:3)	Subscription low water mark array (API, ADMIN, PROXY)
40	28	Integer	4	QTSTSEXP	Subscriptions expired
44	2C	Integer	4	QTSTMSG	Total messages put to Sub queue
48	30	Integer	4	QTSTSPHW	Single publish subscriber high water mark
52	34	Integer	4	QTSTPTOT (1:3)	Total Publication requests (API, ADMIN, PROXY)
64	40	Integer	4	QTSTPTHI	Total publish high water mark
68	44	Integer	4	QTSTPTLO	Total publish low water mark
72	48	Integer	4	QTSTPNOS	Count of publishes to no subscriber
76	4C	Integer	4	*	Reserved
80	50	Bitstring	8	QTSTETHW	Elapse time HW on publish
88	58	Bitstring	8	QTSTETTO	Elapse time total on publish

Coupling facility manager SMDS data records

Use this topic as a reference to the format of the coupling facility manager shared message data set (SMDS) data records.

The format of the coupling facility manager shared message data set (SMDS) statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQESD)` and in C header file `thlqual.SCSQC370(CSQDSMFC)`.

The statistics provide information about the utilization of the owned shared message data set, I/O activity for the group of shared message data sets, and SMDS buffer utilization.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Layout of channel initiator SMF type 115 records

The layout of channel initiator statistics data (SMF type 115, subtype 231) records is described in this topic.

Self-defining section

The self-defining section for the channel initiator statistics data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure qwsx in the C programming language header file th1qua1.SCSQC370(CSQDSMFC), and in assembler macro th1qua1.SCSQMACS(CSQDQWSX).

Table 39 on page 324 shows the format of the self-defining section.

<i>Table 39. Structure of the channel initiator statistics self-defining section</i>					
Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Integer	4	QWSX0PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWSX0PSL	Length of the QWHS
6	6	Integer	2	QWSX0PSN	Number of instances of QWHS
8	8	Integer	4	QWSX0R1O	Offset from the start of the SMF record to the first channel initiator control information block (QCCT)
12	C	Integer	2	QWSX0R1L	Length of the QCCT
14	E	Integer	2	QWSX0R1N	Number of instances of QCCT
16	10	Integer	4	QWSX0R2O	Offset from the start of the SMF record to the first dispatcher task block (QCT_DSP)
20	14	Integer	2	QWSX0R2L	Length of the QCT_DSP
22	16	Integer	2	QWSX0R2N	Number of instances of QCT_DSP
24	18	Integer	4	QWSX0R3O	Offset from the start of the SMF record to the first adapter task block (QCT_ADP)
28	1C	Integer	2	QWSX0R3L	Length of the QCT_ADP
30	1E	Integer	2	QWSX0R3N	Number of instances of QCT_ADP
32	20	Integer	4	QWSX0R4O	Offset from the start of the SMF record to the first SSL task block (QCT_SSL)
36	24	Integer	2	QWSX0R4L	Length of the QCT_SSL
38	26	Integer	2	QWSX0R4N	Number of instances of QCT_SSL
40	28	Integer	4	QWSX0R5O	Offset from the start of the SMF record to the first DNS task block (QCT_DNS)
44	2C	Integer	2	QWSX0R5L	Length of the QCT_DNS
46	2E	Integer	2	QWSX0R5N	Number of instances of QCT_DNS

Typically one record contains all the data. If there are a large number of dispatchers, adapters, or SSL tasks, the data is split over more than one record.

If this happens, the count of instances of some type of tasks can be zero, and information about a group of tasks can be spread across multiple records. The channel initiator control information block (QCCT) is only present in the first record. For example the data could be split between two SMF records like this:

Table 40. Example data

Count	First record	Last record
QWHS	1	1
QCCT	1	0
QCT_DSP	50	5
QCT_ADP	0	10
QCT_SSL	0	3
QCT_DNS	0	1

This example shows that there were 55 dispatcher TCBs running during the SMF interval.

Instrumentation standard header (QWHS)

The format of the QWHS is described in structure `qwhs` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 115 records:

Table 41. Key fields in the QWHS

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHS SMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record. The subsystem ID in QWHSSSID, and the SMF interval start time in QWHSTIME, can be used to group multiple records for the same interval.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHS DURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

Channel initiator statistics data records

Use this topic as a reference for channel initiator statistics data records.

The format of the channel initiator statistics data record contains two parts:

- The first part is the channel initiator control information block, described in assembler macro `thlqual.SCSQMACS(CSQDQCCT)`. For further information, see [“Channel initiator control information block”](#) on page 326.
- The second part is the channel initiator task block, described in assembler macro `thlqual.SCSQMACS(CSQDQCTA)`.

The channel initiator task block contains information about the four types of task within the CHINIT. For further information, see:

- [“Dispatcher tasks”](#) on page 327

- “Adapter tasks” on page 328
- “Domain Name Server (DNS) task” on page 329
- “SSL tasks” on page 330

Each task includes:

- The elapsed time that the task spent processing requests in the interval (*qcteltm*)
- The CPU time used by the task in the interval, which is made up of CPU used while processing requests and CPU used between requests (*qctcptm*)
- The total wait time of this task in the interval (*qctwttm*)
- The number of requests in the interval (*qctreqn*)

You can use this information to see how busy the task was, and determine whether you need to add more tasks based on the analysis.

For TLS and DNS tasks, the duration of the longest request (*qctlgdu*, *qctlsdu*) and the time of day when this occurred (*qctlgdm*, *qctlsdm*) are also included.

These can be useful to identify when channel requests took a long time. For example, a DNS lookup request going to a server outside of your enterprise taking seconds rather than milliseconds.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.




The example accounting data in the following tasks has been formatted using IBM MQ SupportPac MP1B.

Both of the parts are also described in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, *qcct*, *qct_adp*.

Channel initiator control information block

Use this topic as a reference for the channel initiator control information block.

The channel initiator control information block contains basic information for this CHINIT, including:

- CHINIT job name (*qcctjobn*)
- QSG name if it is in a queue sharing group (*qcctqsgn*)
- Peak number used of current channels (*qcctnocc*)
- Peak number used of active channels (*qcctnoac*)
- MAXCHL - maximum permitted current channels (*qcctmxcc*)
- ACTCHL - maximum permitted active channels (*qcctmxac*)
- TCPCHL - maximum permitted TCP/IP channels (*qcctmxtp*)
- LU62CHL - maximum permitted LU62 channels (*qcctmxlu*)
-  31-bit storage used by CHINIT in the extended private region (*qcctstus*). This information is also provided by the `CSQX004I` message in the CHINIT job log.
-  64-bit storage limit available to the CHINIT (*qcctslim*)
-  64-bit storage used by CHINIT (*qcctstab*). This information is also provided by the `CSQX004I` message in the CHINIT job log.

The format of the channel initiator control information block is described in structure *qcct* in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCCS)`.

You can use this information to see if the number of active channels is approaching the configured maximum value. Note that the number of current and active channels are the values when the record was created. So, between the two intervals there might have been more than this number of channels active.

Channel information from SMF data

Here is an example of channel information from SMF data:

```

V9.4.0
MV4A,MQ27,2023/10/02,11:53:02,VRM:934,
From 2023/10/02,11:52:52 to 2023/10/02,11:53:02, duration 10 seconds.
Peak number used of current channels..... 1
Peak number used of active channels ..... 1
MAXCHL. Max allowed current channels..... 9999
ACTCHL. Max allowed active channels..... 9999
TCPCHL. Max allowed TCP/IP channels..... 9999
LU62CHL. Max allowed LU62 channels..... 200
31-bit storage used..... 436 MB
64-bit storage limit.....16384 PB
64-bit storage used..... 187 MB
64-bit storage free.....16384 PB

```

You can monitor the storage usage and see whether the value is trending upwards. If the total used is approaching the total storage available, you might be running out of storage, and so might not be able to support many more channels.

If the numbers of active current channels are tending towards the maximum number of channels, you might need to increase the maximum number of channels.

z/OS Dispatcher tasks

This topic contains example data for the dispatcher tasks statistics, and information about how to interpret the data.

The format of the dispatcher task block is described in structure `qct_dsp` in the C programming language header file `thlqual.SCSQC370 (CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS (CSQDQCTA)`.

Example data

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	DISP	26587	0.4	0.592463	0.1	22	127
1	DISP	26963	0.3	0.588092	0.1	22	112
2	DISP	864329	2.7	2.545668	0.3	3	28
3	DISP	26875	0.4	0.590825	0.1	22	120
4	DISP	26874	0.4	0.603285	0.1	22	123
Summ	DISP	971628	0.8	4.920332	0.1	5	38

The example data shows that there were five dispatchers. A channel is associated with a dispatcher when it starts. The channel initiator tries to distribute work across all the dispatchers when allocating a channel to a dispatcher. This example shows that one dispatcher is processing more requests than other dispatchers. This is normal, as some channels might stop, so the dispatcher is processing fewer channels, and some channels can be busier than others.

- 4.9 seconds of CPU were used by the dispatchers.
- The average request used 5 microseconds of CPU and took 38 microseconds elapsed time.
- A dispatcher is used to send and receive data over a communications network, and this is not usually dependent on external events. The average elapsed time should, therefore, be close to the average CPU time used. The CPU time (`qctcptm`) value includes all CPU consumed by the task, both processing requests and between processing requests.

The elapsed time (`qcteltm`) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time. If the CHINIT is delayed due to lack of CPU, then the ratio of

average elapsed time to average CPU time is much larger, compared to when the CHINIT is not delayed for CPU.

- The average CPU used per request depends on the message traffic. For example, bigger messages use more CPU than smaller messages.

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests : `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn`
- Average ET: `qcteltm` and `qctreqn`

Usually, the number of dispatchers should be less than, or equal to, the number of processors in the LPAR. If you have more dispatchers than processors in the LPAR they might compete for CPU resources. For more information about tuning your system, see [SupportPac MP16](#).

Channels have an affinity to a dispatcher, so you might find that some dispatchers process many more requests than another dispatcher.

You can use the ALTER QMGR CHIDISPS() command to change the number of dispatchers used. Any change comes into effect the next time the channel initiator is started.

Adapter tasks

This topic contains example data for the adapter tasks statistics, and information about how to interpret the data.

The format of the adapter task block is described in structure `qct_adp` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

Example data

Task	Type	Requests	Busy %	CPU used Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	ADAP	470297	10.2	41.290670	4.6	88	194
1	ADAP	13907	0.6	1.589428	0.2	114	365
2	ADAP	2517	0.2	0.185325	0.0	74	746
3	ADAP	1095	0.1	0.085774	0.0	78	907
4	ADAP	535	0.1	0.040743	0.0	76	947
5	ADAP	220	0.0	0.016228	0.0	74	1175
6	ADAP	82	0.0	0.005521	0.0	67	1786
7	ADAP	80	0.0	0.004248	0.0	53	1160
Summ	ADAP	488733	1.4	43.217938	0.6	88	205

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests: `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn` average
- ET: `qcteltm` and `qctreqn`

This example shows that there were eight adapter tasks.

Adapter number 0

- Processed the majority of the requests (470297 out of 488733)
- Was busy 10.2% of the interval
- Used 41.3 seconds of CPU

Overall

The average CPU per request was 88 microseconds of CPU and took 205 microseconds

The adapters process IBM MQ requests. Some of these requests might wait, for example, for log I/O during a commit, so the average Elapsed Time per request has little meaning.

The CPU time (*qctcptom*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

When an IBM MQ request is made the first free adapter task is used.

- If there is at least one adapter that has been little used (less than 1%) busy, you have enough adapters.
- If at least one adapter was not used, you have enough adapters defined.
- If all the adapters were used, you might need to allocate more adapters.
- If all of the adapters were used, and they were all busy for most of the interval, you need to allocate more adapters.

You can use the ALTER QMGR CHIADAPS() command to change the number of adapters used. Any changes come into effect the next time the channel initiator is started.



Attention: If there are too many adapters acting on a small set of queues, you might get contention within the queue manager.

Related reference

[ALTER QMGR](#)



Domain Name Server (DNS) task

This topic contains example data for the DNS tasks statistics, and information about how to interpret the data.

The format of the DNS task block is described in structure *qct_dns* in the C programming language header file *th1qua1.SCSQC370(CSQDSMFC)*, and in assembler macro *th1qua1.SCSQMACS(CSQDQCTA)*.

Example data

```
Task, Type, Requests, Busy %, CPU used, CPU %, "avg CPU", "avg ET", longest,
date, time
0, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
Summ, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
```

The channel initiator uses a single DNS task. The example shows that the task processed 14002 requests and on average the request used 9 microseconds of CPU and took 11 microseconds of elapsed time.

The longest DNS request took 463 microseconds elapsed time, and this occurred at 12:56:33 local time.

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests : *qctreqn*
- Busy %: *qcteltm* and duration
- CPU used: *qctcptom*

- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltm* and *qctreqn*
- Longest: *qctlgdu*
- Longest at: *qctlgtm*

The DNS task can go out of your enterprise to look up the IP address associated with a name. If the average Elapsed time is significantly more than the average CPU time used, you might have some long requests.

If the value of the longest request time is unacceptable you should work with your network team to investigate why you are having long requests. It might be that you have an invalid name in your connections.

If the DNS task is busy for 25% of the duration, consider investigating the cause further.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Note: There are requests to the DNS task that are not DNS lookups, so you might have the number of requests being greater than zero - but no longest request information.

z/OS SSL tasks

This topic contains example data for the SSL tasks statistics, and information about how to interpret the data.

The format of the SSL task block is described in structure `qct_ssl` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

Example data

Task, date,	Type,	Requests, time	Busy %,	CPU used, Seconds,	CPU %,	"avg CPU", uSeconds,	"avg ET", uSeconds,	longest, uSeconds,	
0,	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,	2014/03/18,
12:46:40.237697									
1,	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,	2014/03/18,
12:46:18.938022									
2,	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,	2014/03/18,
12:46:35.358145									
3,	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,	2014/03/18,
12:46:44.514045									
4,	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,	2014/03/18,
12:46:22.134123									
Summ,	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,	2014/03/18,
12:46:22.134123									

This example data shows that the average request took 364 microseconds. The longest request was for SSL task 4, took 22,438 microseconds, and occurred at 12:46:22.134123 local time.

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests : *qctreqn*
- Busy %: *qcteltm* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltm* and *qctreqn*

- Longest: *qctlsdu* longest at: *qctlstm*

A running channel is associated with an SSL task, in a similar way that a channel is associated with a dispatcher. The SSL tasks can use the cryptographic coprocessors available to the LPAR. So, the elapsed time can include time spent on a coprocessor. You should monitor the average elapsed time throughout the day. If this time increases significantly during peak periods you should work with your z/OS systems programmers, as your coprocessors might be over-used.

If the SSL tasks are busy for a significant proportion of the interval, increasing the number of SSL tasks might help. If the SSL tasks are waiting for external resources such as a coprocessor, increasing the number of SSL tasks has little effect.

You can use the ALTER QMGR SSLTASKS() command to change the number of SSL tasks used. Any changes come into effect the next time the channel initiator is started.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qctelmtm*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Related reference

[ALTER QMGR](#)

Queue data records

Use this topic as a reference for queue (SMF type 115, subtype 216) data records. The statistics are designed to make it easier for you to monitor usage and performance of your queue over time, and give an insight into what happened with your queue during the last SMF interval. This includes all the DISPLAY QSTATUS information and information on message flow, expiry, high and low watermarks and more.

The format of the queue statistics data record is described in assembler macro `th1qual.SCSQMACS(CSQDQQST)`.

The queue statistics record contains information on the performance of selected queues and includes the following fields:

QQSTID – Control block identifier

The identifier for the queue statistics control block; is always `x' D80F '`.

QQSTLL – Length of control block

The length of a queue statistics record.

QQSTEYEC – Control block eyecatcher

The eyecatcher used to make identification of the control block easier; is always `' QQST '`

QQSTQNAM – Queue name

The name of the queue.

QQSTFLAG

An array of bits containing the following information about the queue:

QQSTDISP – Queue disposition

This bit identifies whether the queue is of private or shared disposition. If the bit is on, then it is a shared queue.

QQSTPART – Partial record identifier

This bit identifies whether the record is a full or partial record. If the bit is on, then it is a partial record. When this flag is set there was an issue accessing the information on the queue, for example if there is a CF structure failure.

In a shared queue partial record, the accuracy of **qqstdpth**, **qqstmage**, and **qqstuncm** cannot be guaranteed. Therefore, the fields **qqstdpth** and **qqstmage** are populated with `x' 00 '` and the **qqstuncm** flag is not set.

In a private queue partial record, the accuracy of **qqstmage** cannot be guaranteed, therefore the field is populated with `x' 00 '`.

QQSTUNCM – Uncommitted changes pending

This bit indicates whether there are any uncommitted changes (puts and gets) pending for the queue. If the bit is on, there are uncommitted changes.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would if run at the time the SMF record was generated.

If the queue is a shared queue and QQSTPART is set, this bit is always off as there might have been an issue obtaining the correct value.

QQSTPSID – Page set ID

The page set ID where the queue is located, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a page set assigned to it, this field is set to -1 (x 'FFFF').

This value is correct at the time the SMF record was generated. It is possible that the page set changed during the SMF interval, in which case, the value reflected in the next SMF record will be the new page set.

QQSTBPID – Buffer pool ID

The buffer pool ID used by the queue, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a buffer pool assigned to it, this field is set to -1 (x 'FFFF').

This value is correct at the time the SMF record was generated. It is possible that the buffer pool changed during the SMF interval. In this case, the value reflected in the SMF record is the new buffer pool.

QQSTQSGN – QSG name

The Queue Sharing Group name that the queue manager is a member of, if it is a shared queue. If the queue is a private queue this field is blank.

QQSTCFST – CF Structure name

The coupling facility (CF) structure name the queue uses if it is a shared queue. If the queue is a private queue this field is blank.

QQSTDPTH – Current depth

The depth of the queue at the time the SMF data was captured.

If the queue is a shared queue and QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

QQSTOPCT – Current open for output count

The number of handles that are currently open for output for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as OPPROCS from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTIPCT – Current open for input count

The number of handles that are currently open for input for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as IPPROCS from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTMAGE – Oldest message age

The age, in seconds, of the oldest message on the queue.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

If QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

QQSTQTST – Short term QTIME

The interval, in microseconds, between messages being put on the queue and then being destructively read. Value based on the last few messages processed. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the first value in QTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTQTLT – Long term QTIME

The interval, in microseconds, between messages being put on the queue and then being destructively read. The value is based on a larger sample of the recently processed messages. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the second value in QTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTLPUT – Last put date/time

The time, in store clock format, at which the last message was put to the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

This is the same as LPUTDATE and LPUTTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTLGET – Last get date/time

The time, in store clock format, at which the last message was retrieved from the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

A message being browsed does not count as a message being retrieved.

This is the same as LGETDATE and LGETTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTDPHI – Highest depth

The highest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPHI value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPHI is set to zero.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is higher than the current value of QQSTDPHI, then it is used as the new value of QQSTDPHI.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is higher than QQSTDPHI, if so the current queue depth is used as the new value of QQSTDPHI.

This approach means that the value of QQSTDPHI does not take into account messages put by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

QQSTDPLO – Lowest depth

The lowest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPLO value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPLO is set to a special value.
- The first time during the interval the queue manager obtains the depth of the queue, QQSTDPLO to that value.

- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is lower than the current value of QQSTDPLO, then it is used as the new value of QQSTDPLO.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is lower than QQSTDPLO, if so the current queue depth is used as the new value of QQSTDPLO.

This approach means that the value of QQSTDPLO does not take into account messages got by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

QQSTPUTS – MQPUT count

The number of messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPUT1 – MQPUT1 count

The number of messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPPT – Non-persistent MQPUT count

The number of non-persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPPT – Persistent MQPUT count

The number of persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPP1 – Non-persistent MQPUT1 count

The number of non-persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPP1 – Persistent MQPUT1 count

The number of persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPUTB – MQPUT bytes

The number of bytes of message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPT1B – MQPUT1 bytes

The number of bytes of message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPPB – Non-persistent MQPUT bytes

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPPB – Persistent MQPUT bytes

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNP1B – Non-persistent MQPUT1 bytes

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTP1B – Persistent MQPUT1 bytes

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTFLPT – Failed MQPUT count

The number of MQPUT calls targeting the queue, which failed with a completion code of MQCC_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

QQSTFLP1 – Failed MQPUT1 count

The number of MQPUT1 calls targeting the queue, which failed with a completion code of MQCC_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

QQSTFPTC – Fast puts to a waiting getter count

The number of MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval.

Note: *Put to a waiting getter* is a technique whereby a message might not actually be put onto a queue if there is an application already waiting to get the message. Certain conditions must be satisfied for this to occur, in particular the message must be non-persistent and the putting and getting application must be processing the message outside syncpoint control.

If these conditions are met, then the message is transferred from the putting application's buffer into the getting application's buffer without actually touching the IBM MQ queue. This removes a lot of processing involved in putting the message on the queue and therefore leads to increased throughput and lower CPU costs.

QQSTFPTB – Fast puts to a waiting getter bytes

The number of message and properties bytes from MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval. This does not include message headers in the calculation of the size.

QQSTSTRM – Streamed message count

The number of messages that were successfully streamed from the queue during the interval. This is always zero if STREAMQ is not set for the queue.

QQSTMSMI – Minimum message size put

The minimum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTMSMA – Maximum message size put

The maximum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTMSAV – Average message size put

The average message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTGETS – Destructive MQGET count

The number of messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTNPDG – Non-persistent destructive MQGET count

The number of non-persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTPDG – Persistent destructive MQGET count

The number of persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTGETB – Destructive MQGET byte count

The number of message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTNPDB – Non-persistent destructive MQGET bytes

The number of non-persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTPDB – Persistent destructive MQGET bytes

The number of persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTBRWS – Non-destructive MQGET count

The number of messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTNPBR – Non-persistent non-destructive MQGET count

The number of non-persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTPBR – Persistent non-destructive MQGET count

The number of persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTBRWB – Non-destructive MQGET bytes

The number of message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTNPBB – Non-persistent non-destructive MQGET bytes

The number of non-persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTPBB – Persistent non-destructive MQGET bytes

The number of persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTFLGT – Failed destructive MQGET count

The number of destructive MQGET calls targeting the queue, that failed with a completion code of MQCC_FAILED, during the SMF interval.

Also included in the count is MQCC_WARNING when accompanied by a return code of MQRC_TRUNCATED_MSG_FAILED. However, not included in this count is any MQGET with a wait that receives MQRC_NO_MSG_AVAILABLE.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

QQSTNMAG – Failed destructive MQGET with MQRC_NO_MSG_AVAILABLE count

The number of destructive MQGET calls, without wait, targeting the queue, that failed with both a completion code of MQCC_FAILED and a return code of MQRC_NO_MSG_AVAILABLE, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET, without wait, attempts through the queue manager that generated the SMF record.

QQSTTMFB – Failed destructive MQGET with MQRC_TRUNCATED_MSG_FAILED count

The number of destructive MQGET calls targeting the queue, that failed with both a completion code of MQCC_WARNING and a return code of MQRC_TRUNCATED_MSG_FAILED, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

QQSTFLGW – No message available for destructive MQGET with a wait count

The number of times that there is no message available for destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

QQSTRDGW – Re-driven destructive MQGET with a wait count

The number of times that destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to get the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to get the message, QQSTFLGW is incremented by one and the MQGET goes back into a waiting state.

QQSTFLBW – No message available for non-destructive MQGET with a wait count

The number of times that there is no message available for non-destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

QQSTRDBW – Re-driven non-destructive MQGET with a wait count

The number of times that non-destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to browse the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to browse the message, QQSTFLBW is incremented by one and the MQGET goes back into a waiting state.

QQSTSAGT – Destructive MQGET with MQRC_SIGNAL_REQUEST_ACCEPTED count

The number of destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC_WARNING and a return code of MQRC_SIGNAL_REQUEST_ACCEPTED, during the SMF interval.

QQSTSABR – Non-destructive MQGET with MQRC_SIGNAL_REQUEST_ACCEPTED count

The number of non-destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC_WARNING and a return code of MQRC_SIGNAL_REQUEST_ACCEPTED, during the SMF interval.

QQSTIPHI – High watermark for IPPROC

The highest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTIPL0 – Low watermark for IPPROC

The lowest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPHI – High watermark for OPPROC

The highest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPLO – Low watermark for OPPROC

The lowest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPEN – Successful MQOPEN count

The number of times the queue was successfully opened during the SMF interval. This does not include opens performed as part of an MQPUT1 call.

For shared queues, the count only includes when the queue is opened through the queue manager that generated the SMF record.

QQSTCLOS – MQCLOSE count

The number of times the queue was successfully closed using MQCLOSE, during the SMF interval.

For shared queues, the count only includes when the queue is closed through the queue manager that generated the SMF record.

QQSTINQR – MQINQ count

The number of MQINQ calls that completed with a completion code of MQCC_OK or MQCC_WARNING, during the SMF interval.

QQSTSET – MQSET count

The number of MQSET calls that completed with a completion code of MQCC_OK during the SMF interval.

QQSTEXPR – Expired messages count

The number of expired messages cleared from the queue during the SMF interval.

This includes messages expired by an application issuing an MQGET, by the expired message scanning task or by a REFRESH QMGR TYPE(EXPIRY) command.

QQSTRBPT – Rolled back MQPUT counts

The number of messages that were put to the queue, that have been rolled back off the queue, during the SMF interval.

QQSTRBGT – Rolled back MQGET counts

The number of messages destructively read from the queue, that have been rolled back onto the queue, during the SMF interval.

Interpreting IBM MQ for z/OS accounting data

IBM MQ for z/OS accounting data is written as SMF type 116 records. Use this topic as a reference to the different types of accounting data records.

IBM MQ accounting information can be collected for the following subtypes:

0

Message manager accounting records (how much processor time was spent processing IBM MQ API calls and the number of MQPUT and MQGET calls). This information is produced when a named task disconnects from IBM MQ, and so the information contained within the record might cover many hours.

1

Accounting data for each task, at thread and queue level.

2

Additional queue-level accounting data (if the task used more queues than could fit in the subtype 1 record).

10

Accounting data for channels.

Note: Accounting information for specific channels can be enabled or suppressed by the **STATCHL** channel attribute, and the **STATACLS** queue manager attribute.

Note that:

- Subtype 0 records are produced with accounting trace class 1.
- Subtype 1 and 2 records are produced with accounting trace class 3.
- Subtype 10 records are produced with accounting trace class 4.

Layout of an SMF type 116 record

Use this topic as a reference to the format of an SMF type record.

The standard layout for SMF records involves three parts:

SMF header

Provides format, identification, and time and date information about the record itself.

Self-defining section

Defines the location and size of the individual data records within the SMF record.

Data records

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

The SMF header

Table 42 on page 339 shows the format of SMF record header (SM116).

Offset : Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM116	SMF record header.	
0	0	Integer	2	SM116LEN	SMF record length.	01A4

Table 42. SMF record header description (continued)

Offset : Dec	Offset: Hex	Type	Len	Name	Description	Example
2	2		2		Reserved.	
4	4	Integer	1	SM116FLG	System indicator.	5E
5	5	Integer	1	SM116RTY	Record type. The SMF record type, for IBM MQ accounting records this is always 116 (X'74').	74
6	6	Integer	4	SM116TME	Time when SMF moved record.	00356124
10	A	Integer	4	SM116DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM116SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM116SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM116STF	Record subtype.	0000
24	18	Character	3	SM116REL	IBM MQ version.	F9F3F0 (930)
27	1B		1		Reserved.	
28	1C	Character	0	SM116END	End of SMF header and start of self-defining section.	

Note: The (hexadecimal) values in the right-hand column relate to [Figure 22 on page 342](#).

Self-defining sections

A self-defining section of an SMF record tells you where to find an accounting record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at a fixed offset from the start of the SMF record.

Each self-defining section points to accounting related data. [Table 43 on page 340](#) summarizes the offsets from the start of the SMF record header.

Table 43. Offsets to self-defining sections

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
All	Common header	28	X'1C'	“Common IBM MQ SMF header” on page 343
0	Message manager	44	X'2C'	“Message manager data records” on page 344
1	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 345

Table 43. Offsets to self-defining sections (continued)

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
1	Thread-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 345
1	Queue-level accounting	52	X'34'	“Thread-level and queue-level data records” on page 345. This section is present only if the WTASWQCT field in the task-related information (WTAS) structure is non-zero.
2	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 345
2	Queue-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 345
10	Channel accounting			“Channel accounting data records” on page 349

Note: Other self-defining sections refer to data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssllllnnnn
```

where:

sssssss

Fullword containing the offset from start of the SMF record.

llll

Halfword giving the length of this data record.

nnnn

Halfword giving the number of data records in this SMF record.

Figure 22 on page 342 shows an example of part of an SMF type 116 record. The numbers in the left-hand column represent the offset, in hexadecimal, from the start of the record. Each line corresponds to sixteen bytes of data, where each byte is two hexadecimal characters, for example 0C. The characters in the right-hand column represent the printable characters for each byte. Non-printable characters are shown by a period (.) character.

In this example, alternate fields in the SMF header are underlined to help you to see them; refer to [Table 42 on page 339](#) to identify them. The self defining section for one of the message manager accounting data records (at the offset given in [Table 43 on page 340](#)) is shown in **bold**.

```
000000 01A40000 5E740035 61240100 223FD4E5 *...;.../....MV*
000000 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000000 00700001 00000054 00B00001 00000104 *.....*
000000 00300001 00000000 00000000 00000000 *.....*
000000 00000000 00000000 00000000 00000000 *.....*
```

Figure 22. Part of an SMF record 116 showing the header and self-defining sections

The self-defining section for the type of message manager accounting data is located at offset X'2C' from the start of the SMF record and contains this information:

- The offset of the message manager accounting data is located X'00000104' bytes from the start of the SMF record.
- This message manager record is X'0030' bytes long.
- There is one record (X'0001').

Note: Always use offsets in the self-defining sections to locate the accounting records.

Processing type 116 SMF records

Use this topic as a reference to the format of the processing type accounting record.

Any accounting data you collect from SMF must be processed to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the value of the following fields:

- SM116RTY, the SMF record number = X'74' (116)
- SM116STF, the record subtype, must be 0000, 0001, 0002, or 0010

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMFDL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility” on page 299](#).

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

You need to update the SMFIN DD card with the name of the SMF data set. Use the z/OS command '/D SMF' to show the name of the data set, and you need to update the DUMPOUT DD card with the name for the output data set.

You also need to specify the START and END times that you require.

The following sample JCL extracts SMF records from SMF data sets and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=
//SMFIN DD DSN=xxxxxx.MANA,DISP=SHR
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
INDD(SMFIN,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

The following sample JCL extracts SMF records from the SMF log stream named in LSNNAME and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMF DL,REGION=0M
//SYSPRINT DD SYSOUT=*
//SMFOUT DD DSN=xxxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
LSNAME(IFASMF.MQ,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

Common IBM MQ SMF header

Use this topic as a reference to the common IBM MQ SMF header type accounting record.

The format of this record is described in [Table 44 on page 343](#) and in assembler macros `thlqual.SCSQMACS(CSQDQWHS)` and `thlqual.SCSQMACS(CSQDQWHC)`, and C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `qwhs`, `qwhsnsda`.

The QWHS data includes the subsystem name. For subtype 1 records, it also shows whether there are queue-level accounting records present. If the QWHSNSDA field is 3 or less, there are not, and the corresponding self-defining section (at offset X'34') is not set.

The QWHC data gives you information about the user (for example, the user ID (QWHCAID) and the type of application (QWHCATYP)). The QWHC section is completed only for subtype 0 records. The equivalent information is present in the thread identification record for subtype 1 and 2 records.

Table 44. Structure of the common IBM MQ SMF header record QWHS

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Structure	128	QWHS	
0	0		6		Reserved
6	6	Character	1	QWHSNSDA	Number of self defining sections in the SMF records
7	7		5		Reserved
12	C	Character	4	QWHSSSID	Subsystem name
16	10		24		Reserved
40	28	Character	8	QWHCAID	User ID associated with the z/OS job
48	30	Character	12	QWHCCV	Thread cross-reference
60	3C	Character	8	QWHCCN	Connection name
68	44		8		Reserved
76	4C	Character	8	QWHCOPID	User ID associated with the transaction
84	54	Integer	4	QWHCATYP	Type of connecting system (1=CICS, 2=Batch or TSO, 3=IMS control region, 4=IMS MPP or BMP, 5=Command server, 6=Channel initiator, 7=RRS Batch)
88	58	Character	22	QWHCTOKN	Accounting token set to the z/OS accounting information for the user
110	6E	Character	16	QWHCNID	Network identifier
126	7E		2		Reserved

z/OS Combining CICS and IBM MQ performance data

Use this topic as a reference to the combination of IBM MQ and CICS performance data.

The common IBM MQ SMF header type accounting record section, QWHCTOKN, is used to correlate CICS type 110 SMF records with IBM MQ type 116 SMF records.

CICS generates an LU6.2 unit-of-work token, for each CICS task. The token is used to generate an accounting token that is written to QWHCTOKN in the correlation header of subtype zero records.

Details are also written to the WTIDACCT section in subtype 1 and 2 records. The accounting token enables correlation between CICS and IBM MQ performance data for a transaction.

z/OS Thread cross-reference data

Use this topic as a reference to the format of the thread cross-reference type accounting record.

The interpretation of the data in the thread cross-reference (QWHCCV) field varies. This depends on what the data relates to:

- CICS connections (QWHCATYP=1) - see [Table 45 on page 344](#)
- IMS connections (QWHCATYP=3 or 4) - see [Table 46 on page 344](#)
- Batch connections (QWHCATYP=2 or 7) - this field consists of binary zeros
- Others - no meaningful data

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	CICS thread number.
52	34	Character	4	CICS transaction name.
56	38	Integer	4	CICS task number.

Some entries contain blank characters. These apply to the task, rather than to a specific transaction.

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	IMS partition specification table (PST) region identifier.
52	34	Character	8	IMS program specification block (PSB) name.

z/OS Message manager data records

Use this topic as a reference to the format of the message manager accounting records.

The message manager is the component of IBM MQ that processes all API requests. The format of the message manager accounting records is described in assembler macro thlqual.SCSQMACS(CSQDQMAC).

The QMAC data gives you information about the processor time spent processing IBM MQ calls, and counts of the number of MQPUT and MQGET requests for messages of different sizes.

Note: A single IMS application might write two SMF records. In this case, add the figures from both records to provide the correct totals for the IMS application.

Records containing zero processor time

Records are sometimes produced that contain zero processor time in the QMACPUT field. These records occur when long running tasks identified to IBM MQ either terminate or are prompted to output accounting records by accounting trace being stopped. Such tasks exist in the CICS adapter and in the channel initiator (for distributed queuing). The number of these tasks with zero processor time depends upon how much activity there has been in the system:

- For the CICS adapter, this can result in up to nine records with zero processor time.
- For the channel initiator, the number of records with zero processor time can be up to the sum of `Adapters + Dispatchers + 6`, as defined in the queue manager attributes.

These records reflect the amount of work done under the task, and can be ignored.

Sample subtype zero accounting record

Use this topic as a reference to the format of the subtype zero accounting records.

Figure 23 on page 345 shows a type 116, subtype zero SMF record. In this figure, the SMF record header and the QMAC accounting data record are underlined. The self-defining sections are in bold.

```
000000 01A40000 5E740035 61240100 223FD4E5 *...;.../....MV*
000010 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600.....*
000020 00700001 00000054 00B00001 00000104 *.....*
000030 00300001 00000000 00000000 00000000 *.....*
000040 00000000 00000000 00000000 00000000 *.....*
000050 00000000 B478AB43 9C6C2280 B478AB47 *.....%.....*
000060 9DB47E02 00000000 04C0F631 00000001 *..=.....%6.....*
000070 9880E72D 00000000 014D9540 00000000 *..X.....(. ....*
000080 08480C80 00000010 40404040 40404040 *.....*
000090 00000000 00000000 00000051 00000000 *.....*
0000A0 00000000 00000000 00000000 00000000 *.....*
0000B0 00000000 00000000 00000000 00000000 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 00000000 *.....*
000100 00000000 D4140030 D8D4C1C3 00000000 *...M...QMAC...*
000110 689C738D 00000050 00000000 00000050 *.....&.....&*
000120 0000000A 00000000 00000000 00000000 *.....*
000130 00000000 0024011A 00030710 02DAACF0 *......0*
```

Figure 23. Example SMF type 116, subtype zero record

Thread-level and queue-level data records

Use this topic as a reference to the format of the thread-level and queue-level accounting records.

Thread level accounting records are collected for each task using IBM MQ. In addition, queue-level accounting records are gathered about each queue that the task opens. A queue-level accounting record is written for each queue that the task has used since the thread-level accounting record was last written.

If the task uses a queue that is configured with a streaming queue, there is no queue-level accounting record for the streaming queue. Instead, the accounting record for the original queue accumulates data for the data points that would have been associated with the streaming queue.

The only exception to this is that the PUTN/PUT1N value shows the number of MQPUT/MQPUT1 requests made by the application, and excludes the extra MQPUT requests made to the streaming queue.

So, for example, if an application issues a single MQPUT request, the:

- PUTN value is 1
- Elapsed time (PUTET) and CPU time (PUTCT) for the MQPUT include the time taken to put to both the primary and streaming queue

- Number of page set requests (PUTPSN) includes those for both the primary and secondary queue, and so on

For each task, data is written to SMF when the task finishes.

From IBM MQ 9.3.0 onwards, for long running tasks, data is also written at the interval specified by either the ACCTIME, or STATIME, parameter of the CSQ6SYSP system parameter macro, or by the system SMF statistics broadcast, provided that the task was running the previous time data was gathered.

Thread-level and queue-level accounting records are produced if you specify class 3 when you start the accounting trace. For example, use the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
```

The thread level accounting information is written to an SMF type 116, subtype 1 record, and is followed by queue-level records. If the task opened many queues, further queue information is written to one or more SMF type 116 subtype 2 records. A thread identification control block is included in each subtype 1 and 2 record to enable you to relate each record to the correct task. Typically, the maximum number of queue-level records in each SMF record is about 45.

The format of the thread-level accounting record is described in assembler macro `th1qua1.SCSQMACS(CSQDWTAS)`. The format of the queue-level accounting record is described in assembler macro `th1qua1.SCSQMACS(CSQDWQ)`. The format of the thread identification record is described in assembler macro `th1qua1.SCSQMACS(CSQDWTID)`. All these records are also described in C header file `th1qua1.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `wtas`, `wtassex`.

Meaning of the channel names

Use this topic as a reference to the meaning of channel names.

The channel name in the WTID is constructed as shown in the following example. In this example a sender channel exists from queue manager QM1 to queue manager QM2.

The meaning of channel names are described in the following table.

<i>Table 47. Meaning of channel names</i>		
Field name	Meaning	Example
For queue manager QM1 the sender channel has the following fields set:		
WTIDCCN	The job name	QM1CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	This is defined in the CONNAME of the channel	WINMVS2B(2162)
For queue manager QM2 the receiver channel has the following fields set:		
WTIDCCN	The job name	QM2CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	Where the channel came from	9.20.101.14

Sample subtype 1 and subtype 2 records

Use this topic as a reference to the format of the subtype 1 and subtype 2 accounting records.

Figure 24 on page 347 and Figure 25 on page 347 show examples of SMF type 116, subtype 1 and subtype 2 records. These two accounting records were created for a batch job that opened 80 queues. Because many queues were opened, a subtype 2 record was required to contain all the information produced.

```

000000 703C0000 5E74002D 983B0100 229FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 00006FCC *41MQ07..600...?*
000020 00700001 0000003C 00D00001 0000010C *.....}.....*
000030 02C00001 000003CC 02400030 F70000D0 *.}.....7..}*
000040 E6E3C9C4 00000000 00000000 00000040 *WTID..... *
.
.
000100 00000000 00000000 7F4A4BB8 F70102C0 *....."...7..}*
000110 E6E3C1E2 B4802373 0BF07885 7F4AE718 *WTAS.....0.."X.*

```

Figure 24. Example SMF type 116, subtype 1 record

The first self-defining section starts at X'24' and is **bold** in the example; X'0000003C' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'0000010C' is the offset to the WTAS data record, X'02C0' is the length of the WTAS record, and X'0001' is the number of WTAS records.

The third self-defining section starts at X'34' and is **bold** in the example; X'000003CC' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0030' is the number of WQST records.

Figure 25 on page 347 shows an example of an SMF type 116, subtype 2 record.

```

000000 49740000 5E74002D 983B0100 229FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 00004904 *41MQ07..600.....*
000020 00700001 00000034 00D00001 00000104 *.....}.....*
000030 02400020 F70000D0 E6E3C9C4 00000002 *. ..7..}WTID....*
.
.
000100 7F4A4BB8 F7020240 E6D8E2E3 00000001 *"...7.. WQST....*

```

Figure 25. Example SMF type 116, subtype 2 record

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'00000104' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0020' is the number of WQST records.

Figure 26 on page 347 shows an example of an SMF type 116, subtype 1 record where no queues have been opened and there are consequently no self-defining sections for WQST records..

```

000000          5E740039 4E9B0104 344FD4E5 * .....|MV*
000010 F4F1D4D8 F0F70001 F6F0F000 000003DC *41MQ07..600.....*
000020 00800001 00000034 00D00001 00000104 *.....}.....*
000030 02D80001 F70000D0 E6E3C9C4 00000002 *.Q..7..WTID....*
000040 C1F8C5C1 C4C5D740 C1F8C5C1 C4C54040 *A8EADEP A8EADE *
000050 40404040 40404040 00000000 00000000 * .....*
000060 40404040 40404040 4040          *          *

```

Figure 26. Example SMF type 116, subtype 1 record with no WQST data records

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'0000010C' is the offset to the WTAS data record, X'02D8' is the length of the WTAS record, and X'0001' is the number of WTAS records.

There is no self-defining section describing a WQST data record, equivalent to the third self-defining section in [Figure 24 on page 347](#).

Layout of channel initiator SMF type 116 records

The layout of channel accounting data (SMF type 116, subtype 10) records is described in this topic.

Self-defining section

The self-defining section for the channel accounting data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure `qws5` in the C programming language header file `thlqua1.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqua1.SCSQMACS(CSQDQWS5)`.

[Table 48 on page 348](#) shows the format of the self-defining section.

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Integer	4	QWS50PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWS50PSL	Length of the QWHS
6	6	Integer	2	QWS50PSN	Number of instances of QWHS
8	8	Integer	4	QWS50R1O	Offset from the start of the SMF record to the first channel accounting data record (QCST)
12	C	Integer	2	QWS50R1L	Length of the QCST
14	E	Integer	2	QWS50R1N	Number of instances of QCST

Instrumentation standard header (QWHS)

The format of the QWHS is described in structure `qwhs` in the C programming language header file `thlqua1.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqua1.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 116 records:

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHSSMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHSDURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

Channel accounting data records

Use this topic as a reference for channel accounting data records.

The format of the channel accounting data record is described in assembler macro `th1qua1.SCSQMACS (CSQDQCST)`. The format is also described in the C programming language header file `th1qua1.SCSQC370 (CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, `qcst`.

The channel accounting data gives you information about the status and statistics of each channel instance, including:

- Average network time (`qcstntav`)
- Average time on exit (`qcstetav`)
- Channel batch data limit (`qcstcbdl`)
- Channel batch interval (`qcstcbit`)
- Channel batch size (`qcstcbisz`)
- Channel dispatcher number (`qcstdspn`)
- Channel disposition (`qcstchdp`)
- Channel name (`qcstchnm`)
- Channel state (`qcstchst`)
- Channel started time (`qcststrt`)
- Channel status collected time (`qcstcltm`)
- Channel stopped time (`qcstludt`)
- Channel type (`qcstchty`)
- Common name (CN) from SSLCERTI (`qcstslcn`)
- Compression rate (`qcstcpra`)
- Connection name (`qcstcnnm`)
- Current shared conversations (`qcstcscv`)
- DNS resolution time (`qcstdnrt`)
- Effective value of STATCHL parameter (`qcststcl`)
- Last message time (`qcstlmst`)
- Maximum network time (`qcstntmx`)
- Maximum time on exit (`qcstetmx`)
- Minimum network time (`qcstntmn`)
- Minimum time on exit (`qcstetmn`)
- Name of the remote queue manager or application (`qcstrqmn`)
- Number of batches (`qcstbatc`)
- Number of bytes for message data (`qcstnbyt`)
- Number of bytes for persistent message data (`qcstnpby`)
- Number of bytes received for both message data and control information (`qcstbyrc`)
- Number of bytes sent for both message data and control information (`qcstbyst`)
- Number of full batches (`qcstfuba`)
- Number of messages, or number of MQI calls (`qcstnmsg`)
- Number of persistent messages (`qcstnmsg`)
- Number of put retries (`qcstptry`)
- Number of transmission queue becoming empty (`qcstqetc`)
- Number of transmission buffers received (**qcstbfrc**)

- Number of transmission buffers sent (*qcstbfst*)
- Serial number from SSLPEER (*qcstslsn*)
- SSL CipherSpec (zero means TLS not used) (*qcstslcs*)
- The date and time of maximum network time (*qcstntdt*)
- The date and time of maximum time on exit (*qcstetdt*)

Note, that for the channel accounting field *qcstetmn* (Minimum time on exit) and *qcstntmn* (Minimum network time) these two fields will be initialized to the hexadecimal value of 8FFFFFFF when unused.

You can use this information to see the throughput of a channel, if the actual batches are approaching the limit, the latency of the network, information about the remote end, performance of user exit, and so on.

Here is an example of the channel accounting data which has been formatted with IBM MQ SupportPac MP1B.

The fields available are based on the display channel status command (DIS CHS) and channel statistics by IBM MQ on platforms except z/OS, with some additional fields.

```
The data and time of the start and end of the record in local time, and the duration
SMF interval start      2014/03/26,02:30:00
SMF interval end       2014/03/26,02:45:00
SMF interval duration   899.997759 seconds
```

Information about the channel

```
Connection name      9.20.4.159
Channel disp         PRIVATE
Channel type         RECEIVER
Channel status       CLOSING
Channel STATCHL     HIGH
```

```
Start date & time      2014/03/26,02:44:58
Channel status collect time 2014/03/26,02:45:00
Last status changed   1900/01/01,00:00:00
Last msg time         2014/03/26,02:44:59
```

```
Batch size            50
Messages/batch        3.3
Number of messages    1,102
Number of persistent messages 1,102
Number of batches     335
Number of full batches 0
Number of partial batches 335
Buffers sent          337
Buffers received      1,272
Message data          5,038,344  4 MB
Persistent message data 5,038,344  4 MB
Non persistent message data 0  0 B
Total bytes sent      9,852  9 KB
Total bytes received  5,043,520  4 MB
Bytes received/Batch  15,055  14 KB
Bytes sent/Batch      29  29 B
Batches/Second       1
Bytes received/message 4,576  4 KB
Bytes sent/message    8  8 B
Bytes received/second 28,019  27 KB/sec
Bytes sent/second     54  54 B/sec
Compression rate      0
```

The name of the queue manager at the remote end of the connection

```
Remote qmgr/app      MQPH
Put retry count      0
```

IBM MQ 네트워크 성능 조정

큐 관리자 네트워크의 성능을 개선하려면 이 절의 성능 조정 팁을 사용하십시오.

클라이언트 및 서버 연결 채널 조정

SHARECNV의 기본 설정은 10이며, 각 채널 인스턴스에 대해 최대 10개의 클라이언트 대화를 허용합니다. 그러나 다른 수의 공유 대화를 사용하면 성능을 향상시킬 수 있습니다. 공유 대화 필요하지 않거나 분산 서버를 사용하는 경우 **SHARECNV**를 1로 설정하십시오. **SHARECNV**를 1 이상으로 설정할 때 올바르게 실행되지 않는 기존 클라이언트 애플리케이션이 있는 경우 **SHARECNV**를 0로 설정하십시오.

이 태스크 정보

일부 구성의 경우 공유 대화 사용이 상당한 이점을 제공합니다. 그러나 분산 서버의 경우 10 공유 대화의 기본 구성을 사용하는 채널에서의 메시지 처리는 공유 대화를 사용하지 않는 채널에서보다 평균 15% 더 느립니다. 대화를 공유하는 MQI 채널 인스턴스에서 한 소켓의 모든 대화는 동일한 스레드로 수신됩니다. 소켓을 공유하는 대화가 모두 사용 중인 경우 대화 스레드는 수신하는 스레드를 사용하기 위해 서로 경합합니다. 경합은 지연을 유발하며, 이 상황에서는 더 작은 수의 공유 대화를 사용하는 것이 더 좋습니다.

SHARECNV 매개변수를 사용하여 특정 TCP/IP 클라이언트 채널 인스턴스를 통해 공유할 최대 대화 수를 지정합니다. 가능한 모든 값에 대한 세부사항은 [지원되는 IBM MQ 클라이언트: 클라이언트 연결 및 서버 연결 채널의 기본 동작](#)을 참조하십시오.

SHARECNV를 1 이상으로 설정하면 다음 성능 개선사항을 사용할 수 있습니다.

- 양방향 하트비트
- 관리자 중지-정지
- 미리 읽기
- 클라이언트 애플리케이션의 비동기 이용

공유 대화 필요하지 않은 경우 다음 두 설정이 최상의 성능을 제공합니다.

- **SHARECNV(1)**.
- **SHARECNV(0)**.

참고:

- 클라이언트 연결 **SHARECNV** 값이 서버 연결 **SHARECNV** 값과 일치하지 않으면 더 낮은 값이 사용됩니다.
- 애플리케이션이 비재진입 라이브러리에 대해 링크되거나 컴파일되는 경우, **CLNTCONN** 및 **SVRCONN**에서 더 높은 값이 설정되는 경우에도 **CURSHCNV(0)** 값이 조정됩니다.

지정된 채널 인스턴스에 대한 성능을 최적화하려면 다음 단계를 완료하십시오.

프로시저

- 기본 **SHARECNV** 값 10를 사용하는 채널을 모니터하십시오.

SHARECNV(10)의 기본 설정은 여러 시나리오에서 잘 작동하지만, 지정된 채널 인스턴스에 대한 최적 설정이 아닐 수 있습니다. 예를 들어, 분산 서버의 경우 이 설정을 사용하는 채널에서 메시지를 처리하는 것은 공유 대화를 사용하지 않는 채널에서보다 평균 15% 느립니다.

기본 설정이 지정된 채널 인스턴스에 적합한지 확인하려면 이 설정으로 채널이 수행되는 방법을 모니터하십시오.

- **SHARECNV** 값을 2 이상으로 설정하십시오.

SHARECNV(2)를 **SHARECNV(99999999)**로 설정할 수 있습니다. 선택한 설정이 지정된 채널 인스턴스에 적합한지 확인하려면 새 설정으로 채널이 수행되는 방법을 모니터하십시오.

- **SHARECNV** 값을 1로 설정하십시오.

공유 대화 필요하지 않은 경우 가능할 때마다 이 설정을 사용하십시오. 이는 수신 스레드를 사용하기 위한 경합을 제거하며 클라이언트 애플리케이션은 "이 태스크 정보" 절에 설명된 성능 향상을 이용할 수 있습니다.

이 설정을 사용하면 분산 서버 성능이 상당히 향상됩니다. 성능 향상은 미리 읽기가 아닌 동기 가져오기 대기 호출을 발행하는 클라이언트 애플리케이션에 적용됩니다 (예: C 클라이언트 **MQGET** 대기 호출). 이들 클라이언트 애플리케이션이 연결될 때, 분산 서버는 스레드와 메모리를 더 적게 사용하므로 처리량이 증가됩니다.

서버에 소켓을 통해 대화를 공유하는 클라이언트가 연결되어 있고 공유 대화 설정을 SHARECNV(10)에서 SHARECNV(1)(으)로 줄이면 다음과 같은 영향을 미칩니다.

- 서버에서 소켓 사용량을 증가시켰습니다.
- 서버의 채널 인스턴스를 증가시켰습니다.

이 경우 **MaxChannels** 및 **MaxActiveChannels**에 대한 설정을 늘리도록 선택할 수도 있습니다.

참고: 또한 MQCONNX 옵션 MQCNO_NO_CONV_SHARING을 설정하고 **SHARECNV**가 1보다 큰 값으로 설정된 채널에 애플리케이션을 연결할 수도 있습니다. 결과는 **SHARECNV**가 1로 설정된 채널에 애플리케이션을 연결하는 경우와 같습니다.

- SHARECNV 값을 0으로 설정하십시오.

채널 인스턴스는 IBM WebSphere MQ 6.0 서버 또는 클라이언트 연결 채널인 것처럼 정확하게 작동합니다. **SHARECNV**를 1 이상으로 설정할 때 사용할 수 있는 공유 대화 또는 성능 개선사항을 가져오지 않습니다.

SHARECNV를 1 이상으로 설정할 때 올바르게 실행하지 않는 기존 클라이언트 애플리케이션이 있는 경우에만 0의 값을 사용하십시오.

관련 개념

[지원되는 IBM MQ 클라이언트: 클라이언트 연결 및 서버 연결 채널의 기본 동작](#)

분산 발행/구독 네트워크 성능 조정

IBM MQ 분산 발행/구독 클러스터 및 계층의 성능을 개선하려면 이 절의 성능 조정 팁을 사용하십시오.

관련 개념

290 페이지의 『클러스터 모니터링』

클러스터 안에서 애플리케이션 메시지, 제어 메시지 및 로그를 모니터링할 수 있습니다. 클러스터가 큐의 둘 이상의 인스턴스 사이에 로드 밸런싱할 때 특별한 모니터링 고려사항이 있습니다.

직접 라우트된 발행/구독 클러스터 성능

직접 라우트된 발행/구독 클러스터에서, 모든 클러스터 큐 관리자가 발행/구독 메시징에 능동적으로 참여하고 있는지 여부와 상관없이 클러스터된 토픽 및 프록시 구독 같은 정보가 클러스터의 모든 멤버에 푸시됩니다. 이 프로세스는 시스템에 상당한 추가 로드를 생성합니다. 성능에 대한 클러스터 관리의 효과를 줄이기 위해 피크가 아닌 시간에 업데이트를 수행하고 발행/구독에 참여하는 큐 관리자의 훨씬 더 작은 서브세트를 정의하고 그것을 "중첩" 클러스터로 만들거나 토픽 호스트 라우팅 사용으로 전환할 수 있습니다.

발행/구독 클러스터의 큐 관리자에는 두 가지 워크로드 소스가 있습니다.

- 애플리케이션 프로그램에 대한 메시지를 직접 처리.
- 클러스터를 관리하는 데 필요한 메시지 및 채널 처리.

일반적인 포인트-투-포인트 클러스터에서, 클러스터 시스템 워크로드는 클러스터의 멤버에 의해 필요할 때 명시적으로 요청되는 정보로 크게 제한됩니다. 따라서 매우 큰 포인트-투-포인트 클러스터(예를 들어, 수천 개의 큐 관리자를 포함하는 클러스터) 이외의 모든 클러스터에서, 클러스터 관리의 성능 효과를 대체로 무시할 수 있습니다. 그러나 직접 라우트된 발행/구독 클러스터에서, 클러스터 토픽, 큐 관리자 멤버십 및 프록시 구독과 같은 정보는 모든 클러스터 큐 관리자가 발행/구독 메시징에 능동적으로 참여하고 있는지 여부와 상관없이 클러스터의 모든 멤버로 푸시됩니다. 이것은 시스템에 상당한 추가 로드를 생성합니다. 그러므로 타이밍과 크기 모두에서 큐 관리자 성능에 대한 클러스터 관리 효과를 고려해야 합니다.

직접 라우트된 클러스터의 성능 특성

핵심 관리 태스크의 측면에서 포인트-투-포인트 클러스터를 직접 라우트된 발행/구독 클러스터와 비교하십시오.

첫 번째, 포인트-투-포인트 클러스터:

1. 새 클러스터 큐가 정의될 때 목적지 정보가 전체 저장소 큐 관리자로 푸시되고 처음 클러스터 큐를 참조할 때 (예를 들어, 애플리케이션이 열려고 시도할 때) 다른 클러스터 멤버로만 송신됩니다. 이 정보는 큐가 액세스될 때마다 정보를 원격으로 검색할 필요성을 제거하기 위해 큐 관리자에 의해 로컬로 캐시됩니다.

- 클러스터에 큐 관리자 추가가 다른 큐 관리자의 로드와 직접적으로 영향을 주지 않습니다. 새 큐 관리자에 관한 정보가 전체 저장소로 푸시되지만, 클러스터의 다른 큐 관리자로부터 새 큐 관리자로의 채널은 트래픽이 새 큐 관리자로 또는 새 큐 관리자로부터 이동하기 시작할 때만 작성되고 시작됩니다.

요약하면, 포인트-투-포인트 클러스터에 있는 큐 관리자의 로드는 애플리케이션 프로그램에 대해 처리하는 메시지 트래픽에 관련되며 클러스터의 크기에는 직접 관련되지 않습니다.

두 번째, 직접 라우트된 발행/구독 클러스터:

- 새 클러스터 토픽이 정의될 때, 정보가 전체 저장소 큐 관리자로 푸시되고, 그곳에서 클러스터의 모든 멤버로 직접 푸시되어 아직 시작되지 않은 경우 전체 저장소에서 클러스터의 각 멤버로 채널이 시작되게 합니다. 이것이 첫 번째 직접 라우트된 토픽인 경우, 각 큐 관리자 멤버에게 클러스터의 다른 모든 큐 관리자 멤버에 관한 정보가 송신됩니다.
- 새 토픽 문자열의 클러스터 토픽에 구독이 작성될 때, 해당 정보가 즉시 해당 큐 관리자에서 클러스터의 다른 모든 멤버로 직접 푸시되어 아직 시작되지 않은 경우 해당 큐 관리자에서 클러스터의 각 멤버로 채널이 시작되게 합니다.
- 새 큐 관리자가 기존 클러스터에 조인할 때, 모든 클러스터된 토픽(및 직접 클러스터 토픽이 정의된 경우 모든 큐 관리자 멤버)에 관한 정보가 전체 저장소 큐 관리자에서 새 큐 관리자로 푸시됩니다. 그러면 새 큐 관리자가 클러스터에 있는 클러스터 토픽에 대한 모든 구독 지식을 클러스터의 모든 멤버와 동기화합니다.

요약하면, 직접 라우트된 발행/구독 클러스터의 모든 큐 관리자에서의 클러스터 관리 로드는 각 큐 관리자에서 해당 클러스터 토픽의 로컬 사용과 상관 없이 큐 관리자의 수, 클러스터된 토픽 및 클러스터 내의 다른 토픽 문자열에 대한 구독의 변화와 함께 커집니다.

대형 클러스터 또는 구독 변경 비율이 높은 클러스터에서, 이 레벨의 클러스터 관리는 모든 큐 관리자 사이에 상당한 오버헤드일 수 있습니다.

성능에 대한 직접 라우트된 발행/구독의 효과 줄이기

직접 라우트된 발행/구독 클러스터의 성능에 대한 클러스터 관리의 효과를 줄이려면 다음 옵션을 고려하십시오.

- 하루의 오프 피크 시간에 클러스터, 토픽 및 구독 업데이트를 수행하십시오.
- 발행/구독에 포함되는 큐 관리자의 훨씬 더 작업 서브세트를 정의하고 그것을 "중첩" 클러스터로 만드십시오. 이 클러스터가 클러스터 토픽이 정의되는 클러스터입니다. 일부 큐 관리자가 이제 두 클러스터에 있지만, 발행/구독의 전체 효과는 줄어듭니다.
 - 발행/구독 클러스터의 크기가 더 작습니다.
 - 발행/구독 클러스터에 있지 않은 큐 관리자는 클러스터 관리 트래픽에 의해 영향을 훨씬 덜 받습니다.

이전 옵션이 성능 문제를 충분히 해결하지 못하는 경우, 대신 토픽 호스트 라우트된 발행/구독 클러스터 사용을 고려하십시오. 발행/구독 클러스터에서 직접 라우팅 및 토픽 호스트 라우팅의 상세한 비교에 대해서는 [발행/구독 클러스터 설계를 참조하십시오](#).

관련 개념

[토픽 호스트 라우트된 발행/구독 클러스터 성능](#)

토픽 호스트 라우트된 발행/구독 클러스터는 각 토픽을 호스트하는 큐 관리자에 대한 정확한 제어를 제공합니다. 이런 토픽 호스트는 토픽 트리의 해당 분기에 대한 라우팅 큐 관리자가 됩니다. 게다가, 구독 또는 발행자가 없는 큐 관리자는 토픽 호스트에 연결할 필요가 없습니다. 이 구성은 클러스터의 큐 관리자 사이의 연결 수 및 큐 관리자 사이에서 전달되는 정보의 양을 상당히 줄일 수 있습니다.

[발행/구독 네트워크에서 작성자와 이용자 밸런스 유지](#)

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

[발행/구독 네트워크에서 구독 성능](#)

IBM MQ에서 분배된 발행/구독은 다른 토픽 문자열에 대한 구독이 큐 관리자 네트워크에서 작성된 위치의 지식을 전파하여 작업합니다. 이것은 메시지가 발행되는 큐 관리자가 자신의 구독과 일치하기 위해 발행된 메시지의 사본이 필요한 다른 큐 관리자를 식별할 수 있게 합니다.

토픽 호스트 라우트된 발행/구독 클러스터 성능

토픽 호스트 라우트된 발행/구독 클러스터는 각 토픽을 호스트하는 큐 관리자에 대한 정확한 제어를 제공합니다. 이런 토픽 호스트는 토픽 트리의 해당 분기에 대한 라우팅 큐 관리자가 됩니다. 게다가, 구독 또는 발행자가 없는 큐 관리자는 토픽 호스트에 연결할 필요가 없습니다. 이 구성은 클러스터의 큐 관리자 사이의 연결 수 및 큐 관리자 사이에서 전달되는 정보의 양을 상당히 줄일 수 있습니다.

토픽 호스트 라우트 발행/구독 클러스터는 다음과 같이 작동합니다.

- 토픽은 클러스터에 있는 개별 토픽 호스트 큐 관리자에서 수동으로 정의됩니다.
- 클러스터 큐 관리자에 대한 구독이 작성될 때, 프록시 구독이 토픽 호스트에만 작성됩니다.
- 애플리케이션이 토픽에 정보를 발행할 때, 수신 큐 관리자가 해당 토픽을 호스트하는 큐 관리자로 구독을 전달합니다. 그런 다음 토픽 호스트가 토픽에 대한 유효한 구독을 갖는 클러스터의 모든 큐 관리자로 구독을 송신합니다.

토픽 호스트 라우팅에 대한 더 상세한 소개에 대해서는 [클러스터의 토픽 호스트 라우팅](#)을 참조하십시오.

많은 구성의 경우 토픽 호스트 라우팅은 다음과 같은 이점을 제공하기 때문에 직접 라우팅보다 더 적절한 토폴로지입니다.

- 대형 클러스터의 확장성 개선. 토픽 호스트 큐 관리자만 클러스터의 다른 모든 큐 관리자에 연결할 수 있어야 합니다. 따라서 큐 관리자 간 채널이 감소하며 직접 라우팅에 비해 큐 관리자 간의 발행/구독 관리 트래픽이 줄어듭니다. 큐 관리자에서 구독이 변경되면, 토픽 호스트 큐 관리자에게만 알려야 합니다.
- 실제 구성에 대한 추가 제어. 직접 라우팅에서는 모든 큐 관리자가 모든 역할을 담당하는 것으로 간주하므로, 모두가 동등한 능력을 가지고 있어야 합니다. 토픽 호스트 라우팅에서는 토픽 호스트 큐 관리자를 명시적으로 선택합니다. 그러므로 해당 큐 관리자가 적합한 장치에서 실행 중인지 확인할 수 있고 다른 큐 관리자에 대해서는 기능이 적은 시스템을 사용할 수 있습니다.

그러나 토픽 호스트 라우팅을 사용하는 경우에도 특정 제한조건이 시스템에 적용됩니다.

- 직접 라우팅보다 시스템 구성과 유지보수에 더 많은 계획이 필요합니다. 토픽 트리에서 클러스터할 지점 및 클러스터의 토픽 정의 위치를 결정해야 합니다.
- 직접 라우트 토픽과 마찬가지로, 새 토픽 호스트 라우트 토픽을 정의할 경우 정보가 전체 저장소 큐 관리자로 푸시되어 거기에서 클러스터의 모든 멤버에게 전달됩니다. 이 이벤트는 채널이 아직 시작되지 않은 경우 전체 저장소에서 클러스터의 각 멤버로 시작되도록 합니다.
- 클러스터에 구독이 없는 경우에도 발행물이 항상 비호스트 큐 관리자에서 호스트 큐 관리자로 송신됩니다. 따라서 일반적으로 구독이 있을 것으로 예상되는 경우 또는 글로벌 연결 및 지식의 오버헤드가 추가 발행 트래픽의 위험보다 클 경우 라우트 토픽을 사용해야 합니다.
- 비호스트 큐 관리자에서 발행되는 메시지는 구독을 호스팅하는 큐 관리자로 전달되지 않습니다. 이러한 메시지는 항상 토픽 호스트 큐 관리자를 통해 라우트됩니다. 이 접근 방식을 사용할 경우 클러스터에 대한 총 오버헤드와 메시지 대기 시간이 증가하고 성능이 저하될 수 있습니다.

참고: 특정 구성의 경우 중앙 집중된 발행자 또는 구독자를 사용하는 토픽 호스트 라우팅에서 설명하는 대로 이 제한조건을 유용하게 제거할 수 있습니다.

- 하나의 토픽 호스트 큐 관리자를 사용할 경우 토픽에 발행되는 모든 메시지에 대해 단일 실패 지점이 제공됩니다. 이 단일 실패 지점은 토픽 호스트를 여러 개 정의하여 제거할 수 있습니다. 그러나 호스트가 여러 개 있으면 구독을 통해 수신되는 발행 메시지의 순서에 영향을 미칩니다.
- 토픽 호스트 큐 관리자가 여러 큐 관리자의 발행 트래픽을 처리해야 하기 때문에 토픽 호스트 큐 관리자에서 추가 메시지 로드가 발생합니다. 이 로드를 줄이려면 하나의 토픽에 대해 여러 개의 토픽 호스트를 사용하거나(메시지 순서가 유지되지 않는 경우), 토픽 트리의 여러 분기에 대한 라우트 토픽을 호스팅할 여러 큐 관리자를 사용하십시오.

중앙 집중된 발행자 또는 구독자를 갖는 토픽 호스트 라우팅

발행물이 항상 토픽 호스트 큐 관리자를 통해 구독으로 라우트될 때 발생하는 추가 "홉(Hop)"을 제거하려면, 토픽을 호스트하는 동일한 큐 관리자에서 발행자 또는 구독을 구성하십시오. 이 접근법은 다음 두 경우에 최대 성능 이점을 제공합니다.

- 다수의 발행자와 소수의 구독을 가진 토픽. 이 경우, 토픽 호스트 큐 관리자에 구독을 호스트하십시오.

- 소수의 발행자와 다수의 구독을 가진 토픽. 이 경우, 토픽 호스트 큐 관리자에 발행자를 호스트하십시오.

다음 그림은 구독도 호스트하는 토픽 호스트 큐 관리자를 표시합니다. 이 접근법은 발행자와 구독자 사이의 추가 "홉(Hop)"을 제거하며, 클러스터의 모든 멤버 사이에 구독 지식의 불필요한 공유를 줄입니다.

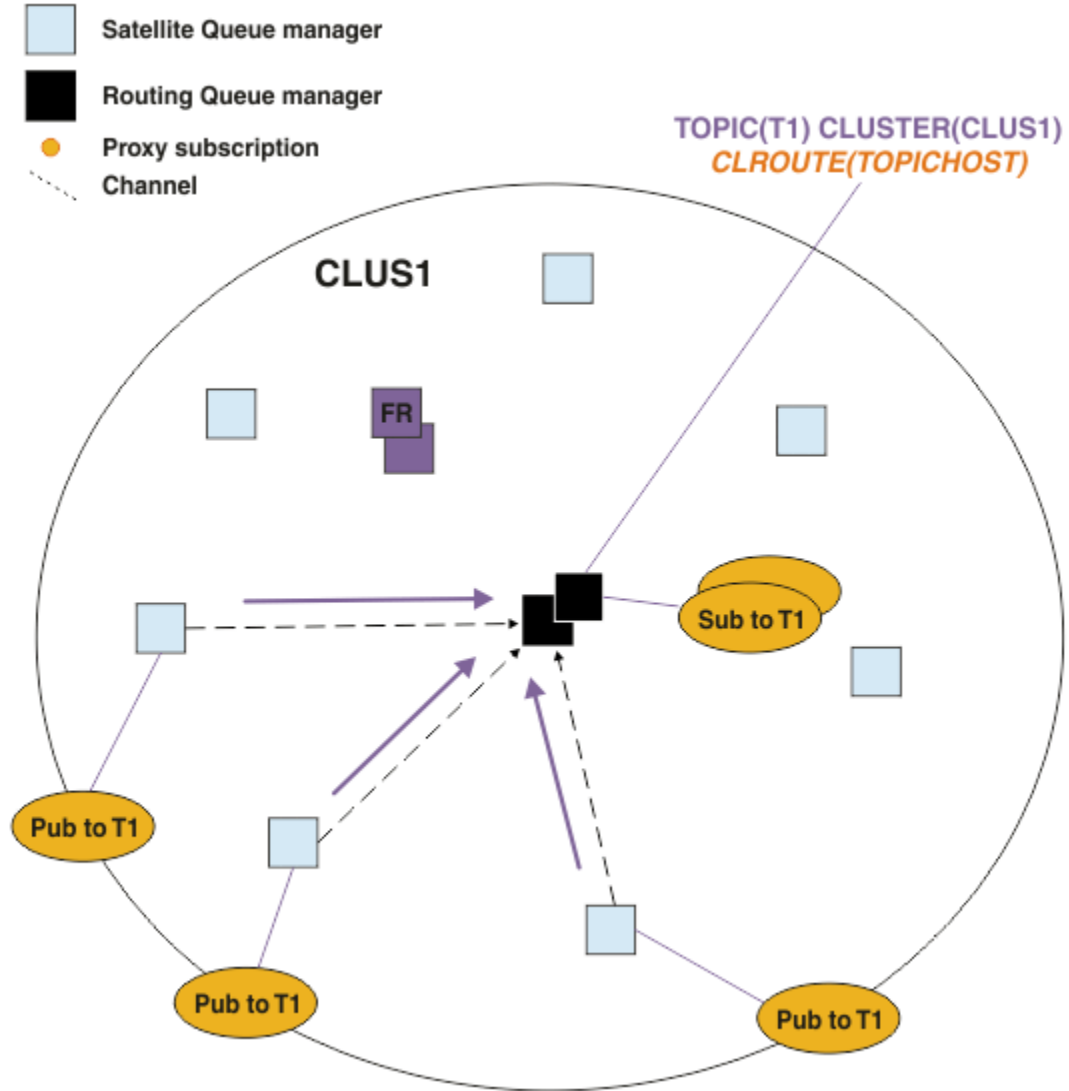


그림 27. 토픽 호스트 큐 관리자에서 구독 호스트

다음 그림은 발행자도 호스트하는 토픽 호스트 큐 관리자를 표시합니다. 이 접근법은 발행자와 구독자 사이의 추가 "홉(Hop)"을 제거하며, 클러스터의 모든 멤버 사이에 구독 지식의 불필요한 공유를 줄입니다.

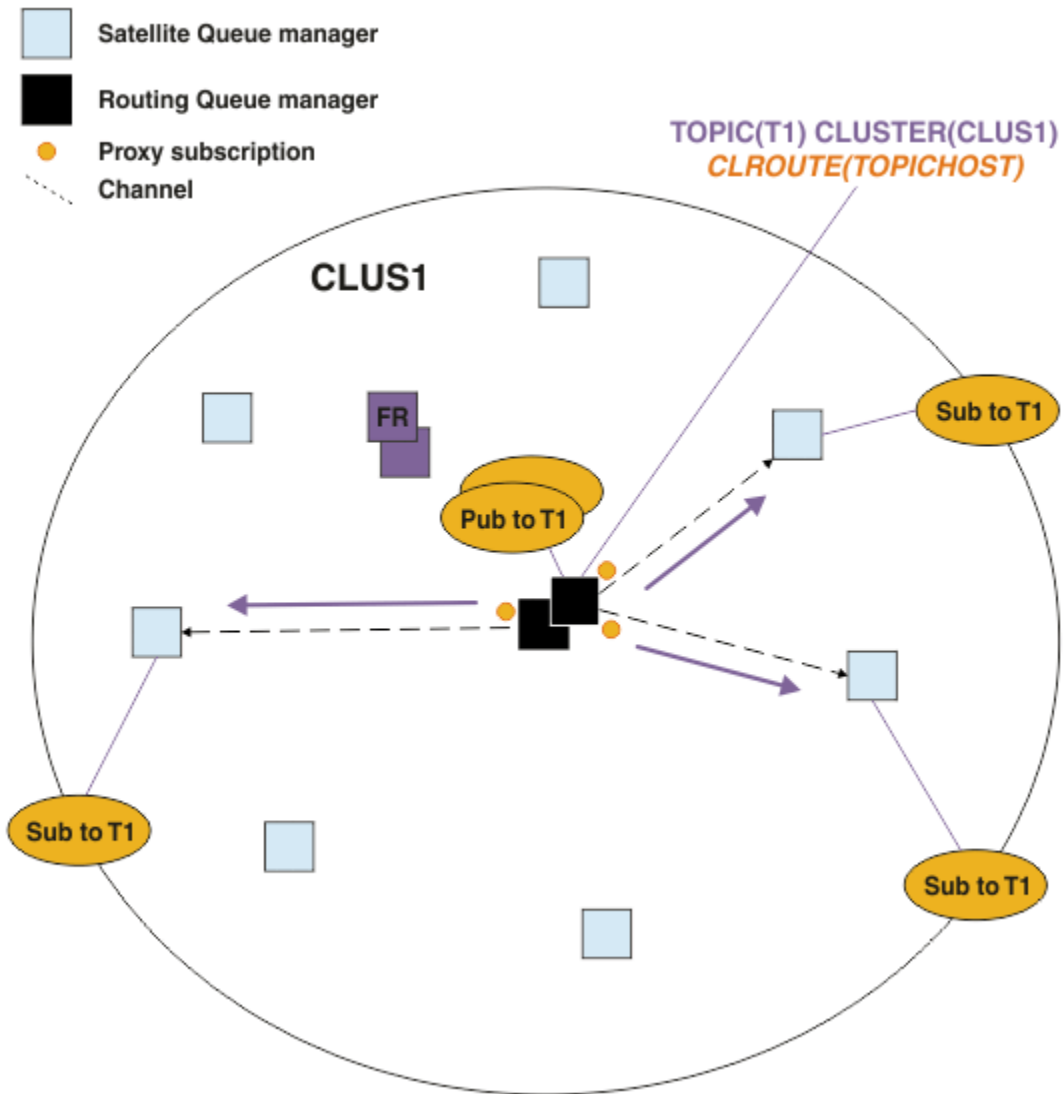


그림 28. 토픽 호스트 큐 관리자에서 발행물 호스트

관련 개념

직접 라우트된 발행/구독 클러스터 성능

직접 라우트된 발행/구독 클러스터에서, 모든 클러스터 큐 관리자가 발행/구독 메시징에 능동적으로 참여하고 있는지 여부와 상관없이 클러스터된 토픽 및 프록시 구독 같은 정보가 클러스터의 모든 멤버에 푸시됩니다. 이 프로세스는 시스템에 상당한 추가 로드를 생성합니다. 성능에 대한 클러스터 관리의 효과를 줄이기 위해 피크가 아닌 시간에 업데이트를 수행하고 발행/구독에 참여하는 큐 관리자의 훨씬 더 작은 서브세트를 정의하고 그것을 "중첩" 클러스터로 만들거나 토픽 호스트 라우팅 사용으로 전환할 수 있습니다.

발행/구독 네트워크에서 작성자와 사용자 밸런스 유지

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

발행/구독 네트워크에서 구독 성능

IBM MQ에서 분배된 발행/구독은 다른 토픽 문자열에 대한 구독이 큐 관리자 네트워크에서 작성된 위치의 지식을 전파하여 작업합니다. 이것은 메시지가 발행되는 큐 관리자가 자신의 구독과 일치하기 위해 발행된 메시지의 사본이 필요한 다른 큐 관리자를 식별할 수 있게 합니다.

발행/구독 네트워크에서 작성자와 이용자 밸런스 유지

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

포인트-투-포인트 메시징 토폴로지에서 메시지 이용자와 메시지 작성자 사이의 관계가 쉽게 이해됩니다. 큐별로, 채널별로 메시지 생성 및 이용의 추정값을 얻을 수 있습니다. 밸런스 부족이 있는 경우 병목현상을 쉽게 식별한 후 구제합니다.

발행자 및 구독자가 발행/구독 토폴로지에서 밸런스를 이루는지 여부를 밝히는 것은 더 어렵습니다. 각 구독에서 시작하고, 토픽에 발행자를 갖는 큐 관리자로 돌아가서 작업하십시오. 각 큐 관리자에서 각 구독자로 흐르는 발행물의 수를 계산하십시오.

리모트 큐 관리자의 구독과(프록시 구독을 바탕으로) 일치하는 각 발행이 전송 큐에 넣어집니다. 다중 리모트 큐 관리자가 해당 발행에 대한 프록시 구독을 갖는 경우 메시지의 복수 사본이 각각 서로 다른 송신자 채널을 대상으로 하는 전송 큐에 넣어집니다.

발행/구독 클러스터에서는 해당 발행물이 구독을 호스트하는 리모트 큐 관리자의 SYSTEM.INTER.QMGR.PUBS 큐를 대상으로 합니다. 계층에서는 각 발행물이 SYSTEM.BROKER.DEFAULT.STREAM 큐 또는 리모트 큐 관리자의 SYSTEM.QPUBSUB.QUEUE.NAMELIST에 나열되는 다른 모든 스트림 큐를 대상으로 합니다. 각 큐 관리자는 해당 큐에 도착하는 메시지를 처리하고 해당 큐 관리자의 올바른 구독으로 전달합니다.

이 때문에 병목현상이 발생할 수 있는 다음 지점의 로드를 모니터하십시오.

- 개별 구독 큐의 로드를 모니터하십시오.
 - 이 병목 현상은 구독 애플리케이션이 발행되자마자 가능한 빨리 발행물을 이용 중이 아님을 의미합니다.
- SYSTEM.INTER.QMGR.PUBS 큐 또는 스트림 큐의 로드를 모니터하십시오.
 - 이 병목현상은 큐 관리자가 로컬 구독으로 분배할 수 있는 것보다 더 빨리 하나 이상의 리모트 큐 관리자로부터 발행물을 수신 중임을 의미합니다.
 - 클러스터에서 토픽 호스트 라우팅을 사용할 때 토픽 호스트 큐 관리자에서 볼 때, 추가 큐 관리자 토픽 호스트를 작성하여 발행 워크로드가 그들 사이에서 밸런싱될 수 있도록 할 것을 고려하십시오. 그러나 이것은 발행물 사이의 메시지 순서 지정에 영향을 줍니다. 단일 토픽에 대한 다중 토픽 호스트를 사용하는 토픽 호스트 라우팅을 참조하십시오.
- 발행 큐 관리자와 구독 큐 관리자 사이의 채널에서 로드를 모니터하십시오. 이것은 구독 큐 관리자의 전송 큐에 의해 공급됩니다.
 - 이 병목 현상은 하나 이상의 채널이 실행 중이 아니거나, 메시지가 채널이 리모트 큐 관리자로 전달할 수 있는 것보다 더 빨리 로컬 큐 관리자로 발행되고 있음을 의미합니다.
 - 발행/구독 클러스터를 사용할 때, 대상 큐 관리자에 추가 클러스터 수신자 채널 정의를 고려하십시오. 이것은 발행 워크로드가 그들 사이에 밸런싱되도록 허용합니다. 그러나, 이것은 발행 사이의 메시지 순서 지정에 영향을 줍니다. 또한 특정 상황에서는 성능을 개선할 수 있기 때문에 다중 클러스터 전송 큐 구성으로 이동을 고려하십시오.
- 발행 애플리케이션이 큐 처리된 발행/구독 인터페이스를 사용 중인 경우, (a) SYSTEM.BROKER.DEFAULT.STREAM 큐 및 SYSTEM.QPUBSUB.QUEUE.NAMELIST에 나열된 다른 모든 스트림 큐 및 (b) SYSTEM.BROKER.DEFAULT.SUBPOINT 큐 및 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST에 나열된 다른 모든 서브포인트 큐에서의 로드를 모니터하십시오.
 - 이 병목현상은 메시지가 로컬 큐 관리자가 메시지를 처리할 수 있는 것보다 더 빨리 로컬 발행 애플리케이션에 의해 넣어지고 있음을 의미합니다.

관련 개념

직접 라우트된 발행/구독 클러스터 성능

직접 라우트된 발행/구독 클러스터에서, 모든 클러스터 큐 관리자가 발행/구독 메시징에 능동적으로 참여하고 있는지 여부와 상관없이 클러스터된 토픽 및 프록시 구독 같은 정보가 클러스터의 모든 멤버에 푸시됩니다. 이 프로세스는 시스템에 상당한 추가 로드를 생성합니다. 성능에 대한 클러스터 관리의 효과를 줄이기 위해 피크가 아닌 시간에 업데이트를 수행하고 발행/구독에 참여하는 큐 관리자의 훨씬 더 작은 서브세트를 정의하고 그것을 "중첩" 클러스터로 만들거나 토픽 호스트 라우팅 사용으로 전환할 수 있습니다.

토픽 호스트 라우트된 발행/구독 클러스터 성능

토픽 호스트 라우트된 발행/구독 클러스터는 각 토픽을 호스트하는 큐 관리자에 대한 정확한 제어를 제공합니다. 이런 토픽 호스트는 토픽 트리의 해당 분기에 대한 라우팅 큐 관리자가 됩니다. 게다가, 구독 또는 발행자가 없는 큐 관리자는 토픽 호스트에 연결할 필요가 없습니다. 이 구성은 클러스터의 큐 관리자 사이의 연결 수 및 큐 관리자 사이에서 전달되는 정보의 양을 상당히 줄일 수 있습니다.

발행/구독 네트워크에서 구독 성능

IBM MQ에서 분배된 발행/구독은 다른 토픽 문자열에 대한 구독이 큐 관리자 네트워크에서 작성된 위치의 지식을 전파하여 작업합니다. 이것은 메시지가 발행되는 큐 관리자가 자신의 구독과 일치하기 위해 발행된 메시지의 사본이 필요한 다른 큐 관리자를 식별할 수 있게 합니다.

290 페이지의 『클러스터 모니터링』

클러스터 안에서 애플리케이션 메시지, 제어 메시지 및 로그를 모니터링할 수 있습니다. 클러스터가 큐의 둘 이상의 인스턴스 사이에 로드 밸런싱할 때 특별한 모니터링 고려사항이 있습니다.

발행/구독 네트워크에서 구독 성능

IBM MQ에서 분배된 발행/구독은 다른 토픽 문자열에 대한 구독이 큐 관리자 네트워크에서 작성된 위치의 지식을 전파하여 작업합니다. 이것은 메시지가 발행되는 큐 관리자가 자신의 구독과 일치하기 위해 발행된 메시지의 사본이 필요한 다른 큐 관리자를 식별할 수 있게 합니다.

이 방식은 일치하는 구독이 없는 큐 관리자로 발행된 메시지의 송신을 최소화합니다. 그러나, 구독 지식의 전파는 구독될 토픽 문자열의 수가 크고 빈번한 구독 작성 및 삭제를 통해 끊임없이 변하고 있을 때 상당한 오버헤드가 될 수 있습니다.

발행 및 구독이 발행/구독 네트워크 주위에서 이동되는 방법을 조정하여 성능에 영향을 미칠 수 있습니다. 네트워크 트래픽이 소수의 발행물과 빠른 구독 작성, 삭제 또는 변경을 갖는 경우, 모든 큐 관리자로 이동되는 구독 정보를 중지하고 대신 모든 발행물을 네트워크의 모든 큐 관리자로 전달할 수 있습니다. 또한 연결된 큐 관리자 사이에서 주어진 토픽에 대한 프록시 구독 및 발행의 흐름을 제한하고, 와일드카드를 포함하는 프록시 구독의 흐름을 제한하고 토픽 문자열의 수와 과도적 특성을 줄일 수 있습니다.

개별 구독 전파 및 모든 위치에서 발행

모든 위치에서 발행은 개별 구독 전파에 대한 대안입니다. 개별 전파에서는 큐 관리자에서 일치하는 구독을 갖는 발행물만 해당 큐 관리자로 전달됩니다. 모든 위치에서 발행을 사용하면 모든 발행물이 네트워크의 모든 큐 관리자에게 전달됩니다. 수신 큐 관리자는 로컬 구독과 일치하는 발행물을 전달합니다.

개별 구독 전파

이 메커니즘은 큐 관리자의 구독과 일치하는 발행물만이 송신되기 때문에 큐 관리자 간 발행 트래픽이 최소화됩니다.

그러나,

- 구독되는 각 개별 토픽 문자열에 대해, 프록시 구독이 발행/구독 토폴로지의 다른 큐 관리자로 송신됩니다. 큐 관리자 세트는 분산 발행/구독 네트워크 계획에서 설명하는 대로 사용되는 라우팅 모델에 따라 다릅니다.
 - 작성 또는 삭제할 수천 개의 구독이 있는 경우(예를 들어, 큐 관리자의 재시작 후 지속 불가능한 모든 구독 재작성) 또는 구독 세트가 빠르게 변하고 있으며 각각이 서로 다른 토픽 문자열에 대한 것인 경우 이 메시징 오버헤드는 상당할 수 있습니다.
 - 프록시 구독이 전파되는 큐 관리자의 수도 오버헤드의 스케일에 영향을 줍니다.
- 프록시 구독은 비동기 메시지를 사용하여 다른 큐 관리자로 이동됩니다. 이것은 다음과 같은 효과를 갖습니다.
 - 구독의 작성과 다른 큐 관리자에 의한 프록시 구독의 작성, 전달 및 처리 사이에 지연이 있습니다.
 - 해당 간격으로 이 큐 관리자에서 발행되는 메시지는 원격 구독으로 전달되지 않습니다.

모든 위치에서 발행

이 메커니즘을 사용하면 시스템에 토픽 문자열당 프록시 구독 오버헤드가 없습니다. 이것은 빠른 구독 작성, 삭제 또는 변경이 네트워크 로드 및 처리의 증가로 이어지지 않음을 의미합니다.

또한 모든 발행물이 모든 큐 관리자로 이동되기 때문에 큐 관리자로 이동되는 구독 작성과 발행 사이에 지연이 없습니다. 따라서 발행물이 새로 작성되는 원격 구독으로 전달되지 않는 구간이 없습니다.

그러나,

- 모든 발행물을 발행/구독 토플로지의 모든 큐 관리자로 보내는 것은 발행물이 각 큐 관리자에서 일치하는 구독을 갖지 않는 과도한 네트워크 트래픽을 유발할 수 있습니다.
 - 토플로지의 큐 관리자 수가 클수록, 오버헤드는 더 커집니다.

발행물이 큐 관리자의 상당한 부분으로 구독될 것을 예상할 때 또는 구독 변경의 빈도 때문에 프록시 구독 오버헤드가 너무 큰 경우에 모든 위치에서 발행 메커니즘 사용을 고려해야 합니다. 일치하는 구독을 갖는 큐 관리자로 보낼 때보다 모든 큐 관리자로 발행물이 전송될 때 메시징 트래픽 증가를 경험하는 경우에는 개별 프록시 구독 전달을 사용해야 합니다.

토플 트리 내의 임의의 레벨에서 모든 위치에서 발행 작동을 설정할 수 있습니다. 모든 위치에 발행을 사용으로 설정하려면 상위 레벨 토플 오브젝트에 대해 **PROXYSUB** 매개변수를 **FORCE**로 설정합니다. 그러면 토플 트리에서 이 토플 오브젝트 아래 모든 토플과 일치하는 단일 와일드카드 프록시 구독이 존재합니다. 클러스터링된 토플 오브젝트에서 설정하면 **PROXYSUB(FORCE)** 속성이 토플이 정의된 큐 관리자만이 아니라 네트워크의 모든 큐 관리자로 전파됩니다.

참고: 계층 구조에서 사용되면 각 큐 관리자에서 **PROXYSUB(FORCE)** 을(를) 개별적으로 설정하므로 토플로지 메커니즘에서 기본적으로 채널 수를 제한합니다. 그러나, 클러스터에서 사용될 때는 많은 추가 채널이 시작될 수 있습니다.

- 토플 호스트 라우트된 클러스터에서, 채널이 각 큐 관리자에서 각 토플 호스트 큐 관리자로 시작됩니다.
- 직접 라우트된 클러스터에서는 채널이 모든 큐 관리자에서 다른 모든 큐 관리자로 시작됩니다.

많은 채널을 시작하는 오버헤드는 직접 라우트된 클러스터에서 가장 많이 표명되고 성능 문제를 유발할 수 있습니다. [352 페이지의 『직접 라우트된 발행/구독 클러스터 성능』](#)의 내용을 참조하십시오.

연결된 큐 관리자 사이의 프록시 구독 및 발행의 흐름을 제한하는 다른 방법

토플 문자열 통합

많은 특징적인 임시 토플 문자열의 사용은 발행자 또는 구독이 첨부되는 시스템의 각 큐 관리자에서 일정 수준의 관리 오버헤드를 가져옵니다. 주기적으로 토플 문자열의 사용을 평가하여 토플 문자열이 통합될 수 있는지 확인해야 합니다. 토플 문자열의 수와 임시 네이처 및 그로 인해 발행자 및 구독을 줄이면 시스템에 대한 영향을 줄여줍니다.

발행 및 구독 범위 제한

지정된 토플에 대해 발행 범위 및 구독 범위 설정을 사용하여 발행물 및 구독을 이들이 정의된 큐 관리자에 대해 로컬로 유지할 수 있습니다.

와일드카드 토플에 작성되는 구독 차단

Topic 속성 **WILDCARD**를 **BLOCK**(으)로 설정하여 와일드카드를 포함하는 프록시 구독 플로우를 제한할 수 있습니다. 프록시 구독의 와일드카드를 참조하십시오.

[357 페이지의 『발행/구독 네트워크에서 작성자와 사용자 밸런스 유지』](#)의 내용도 참조하십시오.

클러스터에서 프록시 구독 트래픽 모니터링

프록시 구독 트래픽으로부터 시스템에 대한 로드를 고려할 때, [357 페이지의 『발행/구독 네트워크에서 작성자와 사용자 밸런스 유지』](#)에서 나열되는 큐 모니터링 외에 다음 클러스터 큐를 모니터링하십시오.

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

이들 큐에 있는 모든 상당한 메시지 백로그는 구독 변경 비율이 시스템에 대해 너무 크거나 큐 관리자가 클러스터에서 올바르게 기능 중이 아님을 암시합니다. 문제점이 특정 큐 관리자에 있다고 의심하는 경우, 해당 큐 관리자에 대해 발행/구독 지원이 사용 불가능하지 않은지 확인하십시오. **ALTER QMGR**의 **PSMODE**을(를) 참조하십시오.

관련 개념

직접 라우트된 발행/구독 클러스터 성능

직접 라우트된 발행/구독 클러스터에서, 모든 클러스터 큐 관리자가 발행/구독 메시징에 능동적으로 참여하고 있는지 여부와 상관없이 클러스터된 토플 및 프록시 구독 같은 정보가 클러스터의 모든 멤버에 푸시됩니다. 이 프로세스는 시스템에 상당한 추가 로드를 생성합니다. 성능에 대한 클러스터 관리의 효과를 줄이기 위해 피크가 아닌

시간에 업데이트를 수행하고 발행/구독에 참여하는 큐 관리자의 훨씬 더 작은 서브세트를 정의하고 그것을 "중첩" 클러스터로 만들거나 토픽 호스트 라우팅 사용으로 전환할 수 있습니다.

토픽 호스트 라우트된 발행/구독 클러스터 성능

토픽 호스트 라우트된 발행/구독 클러스터는 각 토픽을 호스트하는 큐 관리자에 대한 정확한 제어를 제공합니다. 이런 토픽 호스트는 토픽 트리의 해당 분기에 대한 라우팅 큐 관리자가 됩니다. 게다가, 구독 또는 발행자가 없는 큐 관리자는 토픽 호스트에 연결할 필요가 없습니다. 이 구성은 클러스터의 큐 관리자 사이의 연결 수 및 큐 관리자 사이에서 전달되는 정보의 양을 상당히 줄일 수 있습니다.

발행/구독 네트워크에서 작성자와 이용자 밸런스 유지

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

발행/구독 네트워크에서의 프록시 구독

토픽 트리에서 불필요한 토픽 수 감소

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

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

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

큐 관리자는 30분 동안 사용되지 않은 토픽을 원하지 않는 토픽으로 인식합니다. 큐 관리자는 토픽 트리에서 원하지 않는 토픽을 제거합니다. 큐 관리자 속성 **TREELIFE**(를) 변경하여 30분 지속 기간을 변경할 수 있습니다. 토픽이 큐 관리자에 사용되지 않는 것으로 나타나는지 확인하여 큐 관리자가 원하지 않는 토픽을 제거하도록 도울 수 있습니다. [360 페이지의 『사용되지 않는 토픽의 개념』](#) 절에서 사용되지 않는 토픽의 정의를 설명합니다.

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

사용되지 않는 토픽의 개념

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

토픽 문자열(예: USA/Alabama/Auburn)이 토픽으로 변환될 때 토픽이 토픽 트리에 추가됩니다. 추가 토픽 노드 및 해당 토픽이 필요한 경우 트리에 작성됩니다. 토픽 문자열 USA/Alabama/Auburn이 세 개의 토픽을 포함하는 트리로 변환됩니다.

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

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

토픽 트리의 미사용 토픽은 다음 특성을 갖습니다.

토픽 오브젝트와 연관되지 않음

관리 토픽 오브젝트는 토픽과 연관시키는 토픽 문자열을 갖습니다. 토픽 오브젝트 Alabama를 정의할 때 연관될 토픽 USA/Alabama가 존재하지 않는 경우 해당 토픽이 토픽 문자열에서 작성됩니다. 토픽이 존재하는 경우 토픽 오브젝트 및 토픽이 토픽 문자열을 사용하여 함께 연관됩니다.

보유된 발행을 갖지 않음

MQPMO_RETAIN 옵션이 있는 토픽에 메시지를 넣는 공개자의 보유된 발행 결과가 있는 토픽입니다.

USA/Alabama가 보유된 발행을 갖는지 확인하려면 **runmqsc** 명령 DISPLAY TPSTATUS('USA/Alabama') RETAINED를 사용하십시오. 응답은 YES 또는 NO입니다.

USA/Alabama에서 보유된 발행을 제거하려면 **runmqsc** 명령 CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)를 사용하십시오.

하위 토픽이 없음

USA/Alabama/Auburn은(는) 하위 토픽이 없는 토픽입니다. USA/Alabama/Auburn은(는) USA/Alabama의 직접 하위 토픽입니다.

runmqsc 명령 DISPLAY TPSTATUS('USA/Alabama+')로 USA/Alabama의 직접 하위를 표시하십시오.

노드에 대한 활성 발행자가 없음

노드에 대한 활성 발행자는 출력을 위해 열린 토픽을 갖는 애플리케이션입니다.

예를 들어, 애플리케이션은 열려 있는 옵션 MQ00_OUTPUT을(를) 사용하여 **Alabama**(이)라는 토픽 오브젝트를 엽니다.

USA/Alabama 및 그의 모든 하위에 대한 활성 발행자를 표시하려면 **runmqsc** 명령 DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN을 사용하십시오.

노드에 대한 활성 구독자가 없음

활성 구독자는 지속 가능 구독이거나 MQSUB을 갖는 토픽에 구독을 등록했고 닫지 않은 애플리케이션 중 하나일 수 있습니다.

USA/Alabama에 대한 활성 구독을 표시하려면 **runmqsc** 명령 DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN을 사용하십시오.

USA/Alabama 및 그의 모든 하위에 대한 활성 구독을 표시하려면 **runmqsc** 명령 DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN을 사용하십시오.

토픽 트리에서 토픽 수 관리

요약하자면, 토픽 트리의 토픽 수를 관리하는 많은 방법이 있습니다.

TPCOUNT 표시

TPCOUNT 특성을 표시하려면 주기적으로 **runmqsc** 명령 DISPLAY PUBSUB ALL을 사용하십시오. 이것은 토픽 트리에 있는 토픽 노드의 수입입니다. 숫자가 커지고 있는 경우 더 짧은 TREELIFE가 필요하거나 토픽 자체의 재설계가 필요함을 표시할 수 있습니다.

TREELIFE 수정

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

예를 들어, **runmqsc** 명령 ALTER QMGR TREELIFE(900)은 미사용 토픽의 수명을 30분에서 15분으로 줄입니다.

예외적으로 큐 관리자 재시작

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

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

관련 개념

[토픽 트리](#)

Windows Linux MQ Adv. MQ Adv. VUE Aspera gateway는 대기 시간이 긴 네트워크에서 성능을 향상시킬 수 있음

IBM Aspera® faspio Gateway는 IBM MQ의 네트워크 처리량을 상당히 늘릴 수 있는 빠른 TCP/IP 터널을 제공합니다.

Aspera gateway를 사용하여 큐 관리자 채널의 성능을 향상시킬 수 있습니다. 특히 네트워크의 대기 시간이 긴 경우에 효과적이며 일반적으로 서로 다른 데이터 센터에 있는 큐 관리자 간의 연결 속도를 높이는 데 사용됩니다.

그러나 패킷을 손실하지 않는 빠른 네트워크의 경우 Aspera gateway 사용 시 성능이 감소되기 때문에 Aspera gateway 연결 정의 전후에 네트워크 성능을 검사하는 것이 중요합니다.

권한이 있는 플랫폼에서 실행 중인 큐 관리자는 Aspera gateway를 통해 연결할 수 있습니다. 게이트웨이 자체는 Red Hat® 또는 Ubuntu Linux 또는 Windows에 배치됩니다.

자세한 정보는 [Linux 또는 Windows에서 Aspera gateway 연결 정의를 참조하십시오](#).

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 디버그를 돕기 위해 제공된 것입니다.

중요사항: 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

상표

IBM, IBM 로고, ibm.com[®]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: