9.4

# *IBM MQ* 보안

**참고**

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 647 페이지의 『주의사항』 에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ 의 버전 9릴리스 4및 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

# IBM MQ 보호

보안은 IBM MQ 애플리케이션 개발자와 IBM MQ 시스템 관리자 모두에게 중요한 고려사항입니다. 절대적으로 최소한 보안 구역 및 운영자 워크스테이션 내의 모든 하드웨어 및 소프트웨어가 지원 수명 주기 내에 있고 필수 소프트웨어 업데이트가 있는 최신 상태이며 보안 업데이트가 즉시 적용되도록 해야 합니다.

**관련 참조**

IBM 보안 취약성 관리

z/OS IBM Z및 LinuxOne Security Portal

# 보안 개요

이 토픽 콜렉션은 IBM MQ 보안 개념을 소개합니다.

보안 개념 및 메커니즘은 모든 컴퓨터 시스템에 적용되기 때문에 이는 처음에 표시되고 그 뒤에 해당 보안 메커니즘에 대한 설명이 오며 이는 IBM MQ에서 구현됩니다.

공통적으로 허용되는 보안 관점은 다음과 같습니다.

- 7 페이지의 『식별 및 인증』
- 8 페이지의 『권한 부여』
- 9 페이지의 『감사』
- 9 페이지의 『기밀성』
- 10 페이지의 『데이터 무결성』

보안 메커니즘은 보안 서비스를 구현하는 데 사용되는 기술적인 도구와 기술들입니다. 메커니즘은 특정 서비스를 제공하기 위해 독자적으로 또는 다른 메커니즘으로 작동됩니다. 공통 보안 메커니즘의 예제는 다음과 같습니다.

- 10 페이지의 『암호화』
- 12 페이지의 『메시지 요약 및 디지털 서명』
- 12 페이지의 『디지털 인증서』
- 16 페이지의 『공개 키 기반 구조(PKI)』

IBM MQ 구현을 계획 중인 경우 사용자에게 중요한 보안 관점을 구현하기 위해 필요한 보안 메커니즘을 고려하십시오. 이러한 토픽을 읽은 후에 고려해야 할 사항에 대한 자세한 정보는 80 페이지의 『보안 요구사항 계획』의 내용을 참조하십시오.

## 식별 및 인증

식별은 시스템 사용자 또는 시스템에서 실행 중인 애플리케이션을 고유하게 식별하는 기능입니다. 인증은 사용자 또는 애플리케이션이 청구되는 진짜 사용자나 애플리케이션임을 증명해주는 기능입니다.

예를 들어, 사용자 ID와 비밀번호를 입력하여 시스템에 로그온하는 사용자를 고려해 보십시오. 시스템은 사용자 식별에 사용자 ID를 사용합니다. 시스템은 로그온 시에 입력한 비밀번호가 올바른지 검사하여 사용자를 인증합니다.

### IBM MQ에서의 식별 및 인증

애플리케이션이 IBM MQ에 연결할 때 사용자 ID는 항상 연결과 연관됩니다. 사용자 ID는 처음에 애플리케이션 프로세스와 연관된 운영 체제 사용자 ID입니다. 이 ID는 종종 큐 관리자와 동일한 시스템에서 호스트되는 로컬로 바인드된 애플리케이션에 충분합니다. 그러나 큐 관리자는 여러 가지 방법으로 연결과 연관된 ID를 인증하고 수정할 수도 있습니다. 연결과 연관된 ID 인증은 신뢰할 수 없는 클라이언트 애플리케이션이 네트워크를 통해 큐 관리자에 연결할 때 중요합니다.

IBM MQ 큐 관리자에 대한 애플리케이션 연결과 연관된 ID는 다음 메커니즘 중 하나를 사용하여 설정할 수 있습니다.

© Copyright IBM Corp. 2007년, 2024

- 애플리케이션이 큐 관리자에 연결할 때 사용자 ID및 비밀번호를 제공할 수 있습니다. 큐 관리자는 해당 구성에 따라 신임 정보의 유효성을 검증합니다. 예를 들어, 사용자 ID및 비밀번호를 큐 관리자의 운영 체제 또는 LDAP 서버에 전달하여 인증할 수 있습니다.
- V 9.4.0 IBM MQ 9.3.4부터 애플리케이션은 외부 인증 서버에서 얻은 인증 토큰을 제공할 수도 있습니다. 인증 토큰에 대한 자세한 정보는 305 페이지의 『인증 토큰에 대한 작업』의 내용을 참조하십시오.
- 유효한 디지털 인증서로 구성된 경우 TLS 상호 인증을 사용하도록 클라이언트 채널을 구성할 수 있습니다. TLS 인증을 채널 인증 (CHLAUTH) 규칙과 결합하여 적절한 사용자 ID를 연결과 연관시킬 수 있습니다. 자세한 정보는 18 페이지의 『TLS가 식별, 인증, 기밀성, 무결성을 제공하는 방법』의 내용을 참조하십시오.
- 채널 인증 (CHLAUTH) 규칙은 연결에 대한 정보를 기반으로 ID를 대체할 수 있습니다. 예를 들어, 채널 인증 규칙은 클라이언트의 IP 주소를 기반으로 연결과 연관된 사용자 ID를 설정할 수 있습니다.
- 사용자 정의 종료 코드는 선택한 기준에 따라 ID를 설정할 수 있습니다.

ID및 인증은 두 큐 관리자 간의 채널에도 적용 가능합니다. 이러한 채널을 메시지 채널이라고 합니다. 메시지 채널이 시작되면 채널의 각 끝에 있는 메시지 채널 에이전트 (MCA) 가 해당 파트너를 인증할 수 있습니다. 이 기술을 상호 인증이라고 합니다. MCA 전송의 경우, 메시지를 전송하는 파트너가 진짜 파트너이도록 보증합니다. 마찬가지로, 수신 MCA는 실제 파트너로부터 메시지를 수신하려고 합니다.

ID가 설정되고 필요한 경우 인증되면 다음과 같은 여러 가지 방법으로 IBM MQ 에서 사용됩니다.

- 중요한 점은 기본적으로 이 ID를 사용하여 후속 8 페이지의 『권한 부여』 검사가 수행된다는 점입니다. 예를 들어, 애플리케이션이 큐에 메시지를 넣으려고 시도하면 큐 관리자는 애플리케이션과 연관된 ID에 큐 오브젝트에 대한 '넣기' 권한이 있는지 확인합니다.
- 또한 모든 메시지는 메시지 컨텍스트 정보를 포함할 수 있습니다. 이 정보는 메시지 디스크립터 (MQMD) 에 보유됩니다. 애플리케이션이 큐에 메시지를 넣을 때 큐 관리자가 자동으로 메시지 컨텍스트를 생성할 수 있습니다. 또는 애플리케이션과 연관된 사용자 ID에 권한이 부여된 경우 애플리케이션이 메시지 컨텍스트를 제공할 수 있습니다. 메시지의 이 컨텍스트 정보는 메시지의 진원지에 대한 메시지 정보를 수신하는 애플리케이션을 제공합니다. 예를 들어, 메시지를 넣는 애플리케이션의 이름과 애플리케이션에 연관된 사용자 ID가 포함됩니다.

## 부인 방지

부인 방지 서비스는 결국 특정 메시지가 특정 개인과 연관되는지를 증명하기 위해서 사용됩니다.

부인 방지 서비스는 식별 및 인증 서비스의 확장으로 볼 수 있습니다. 일반적으로 부인 방지는 데이터가 전자적으로 전송될 때 적용됩니다. 예를 들어, 주식을 사거나 팔기 위한 주식 중개인의 주문이나 한 계좌에서 다른 계좌로 자금을 이체하기 위한 은행의 주문 등입니다.

부인 방지 서비스는 둘 이상의 컴포넌트를 포함할 수 있고 이들 각각의 컴포넌트는 다른 기능을 제공합니다. 메시지의 송신자가 송신했다는 것을 거부하면, 원본 증명이 있는 부인 방지 서비스가 수신자에게 그 특정 개인이 메시지를 송신했다는 거부할 수 없는 증거를 제공할 수 있습니다. 메시지의 수신자가 그것을 수신하는 것을 거부하면, 전달의 증명이 있는 부인 방지 서비스가 송신자에게 그 특정 개인이 메시지를 수신했다는 거부할 수 없는 증거를 제공할 수 있습니다.

실제로, 사실상 100% 확신을 가지는 증명, 또는 거부할 수 없는 증거는 어려운 목표입니다. 실제 상황에서는 어떤 것도 완전하게 보안되지는 않습니다. 보안 관리는 비즈니스가 받아들일 수 있는 레벨로 위험을 관리한다는 의미로 받아들여집니다. 이런 환경에서 부인 방지 서비스의 보다 현실적인 기대치는, 용인될 만한 증거를 제공하고 법원에서 사건을 지원할 수 있는 정도입니다.

부인 방지는 IBM MQ가 데이터를 전자적으로 전송하는 수단이기 때문에 IBM MQ 환경에서 적합한 보안 서비스입니다. 예를 들면, 특정 메시지가 송신되었거나 특정 개인과 연관된 애플리케이션에 의해 수신되었다는 동시 증거가 필요할 수 있습니다.

IBM MQ with Advanced Message Security 는 기본 기능의 일부로 부인 방지 서비스를 제공하지 않습니다. 그렇지만 이 제품 문서는 IBM MQ 환경 내에서 자체 엑시트 프로그램을 작성하여 자체적으로 부인 방지 서비스를 제공할 수 있는 방법에 대한 제한을 포함하고 있습니다.

## 권한 부여

권한 부여는 권한 있는 사용자 및 해당 애플리케이션으로만 액세스를 제한하여 시스템에서 중요한 자원을 보호합니다. 자원을 권한 없이 사용하는 것이나 권한이 부여되지 않은 방식으로 사용하는 것을 막습니다.

### IBM MQ에서 권한 부여

특정 개인이나 애플리케이션이 IBM MQ 환경에서 수행할 수 있는 작업을 제한하기 위해 권한 부여할 수 있습니다.

다음은 IBM MQ 환경에서의 권한 부여 예제입니다.

- 인증된 관리자만 IBM MQ 자원 관리 명령 발행 허용.
- 애플리케이션에 연관된 사용자 ID가 수행할 권한이 있는 경우에만 애플리케이션이 큐 관리자에 연결되도록 허용.
- 애플리케이션이 자체 기능에 필요한 해당 큐만 열도록 허용.
- 애플리케이션이 자체 함수에 필요한 해당 토픽에만 구독하도록 허용.
- 애플리케이션이 자체 함수에 필요한 큐의 해당 조작만 수행하도록 허용. 예를 들어, 애플리케이션이 특정 큐에 있는 메시지를 찾아 보기만 하고, 메시지를 넣거나 가져올 수는 없을 수 있습니다.

권한을 설정하는 방법에 대한 자세한 정보는 83 페이지의 『권한 부여 계획』 및 연관된 하위 주제를 참조하십시오.

## 감사

감사는 예상치 못한 또는 권한이 없는 활동이 발생했는지 또는 이런 활동 수행 시도가 있었는지를 감지하기 위해 이벤트를 기록 및 검사하는 프로세스입니다.

### IBM MQ의 감사

IBM MQ는 일반적이지 않은 활동이 발생하는 레코드에 이벤트 메시지를 발행할 수 있습니다.

다음은 IBM MQ 환경에서의 감사 예제입니다.

- 애플리케이션이 열기 권한이 없는 큐 열기를 시도합니다. 도구 이벤트 메시지가 발행됩니다. 이벤트 메시지를 검사하여 이 시도가 발행하는지 검색하고 필요한 조치를 결정할 수 있습니다.
- 애플리케이션이 채널을 열려고 시도하지만 TLS 연결이 허용되지 않기 때문에 시도가 실패합니다. 도구 이벤트 메시지가 발행됩니다. 이벤트 메시지를 검사하여 이 시도가 발행하는지 검색하고 필요한 조치를 결정할 수 있습니다.

## 기밀성

기밀성 서비스는 기밀 정보가 권한 없이 노출되는 것을 막습니다.

민감한 데이터가 로컬에 저장되어 있는 경우, 데이터에 액세스할 수 없으면 읽을 수 없다는 가정 하에 액세스 제어 메커니즘으로 충분히 데이터를 보호할 수 있습니다. 상위 레벨의 보안이 필요한 경우, 데이터를 암호화할 수 있습니다.

민감한 데이터가 통신 네트워크를 통해, 특히, 인터넷과 같이 안전하지 않은 네트워크를 통해 전송될 때 이를 암호화하십시오. 네트워킹 환경에서 회선 도청 등의 데이터를 가로채려는 시도에 대해 액세스 제어 메커니즘은 효율적이지 않습니다.

### IBM MQ의 기밀성

메시지를 암호화하여 IBM MQ에서 기밀성을 구현할 수 있습니다.

다음과 같이 IBM MQ 환경에서 기밀성이 보장됩니다.

- 송신 MCA가 전송 큐에서 메시지를 가져온 후 IBM MQ가 TLS를 사용하여 네트워크를 통해 수신 MCA로 송신하기 전에 메시지를 암호화합니다. 채널의 다른 쪽 끝에서는 수신 MCA가 그 목적지 큐에 메시지를 넣기 전에 메시지가 복호화됩니다.
- 메시지가 로컬 큐에 저장되면 IBM MQ에서 제공되는 액세스 제어 메커니즘은 권한 없는 노출에 대해 해당 컨텐츠를 보호하기에 충분한 것으로 간주될 수 있습니다. 그렇지만 최고 레벨을 보안을 위해 Advanced Message Security를 사용하여 큐에 저장된 메시지를 암호화할 수 있습니다.

- z/OS 로컬 큐에 저장된 메시지는 저장 중에 z/OS® 데이터 세트 암호화를 사용하여 암호화할 수 있습니다.

  데이터 세트 암호화를 사용하는 IBM MQ for z/OS의 저장 데이터에 대한 기밀성 섹션을 참조하십시오. 참조하십시오.

## 데이터 무결성

데이터 무결성 서비스는 데이터가 권한 없이 수정되었는지를 감지합니다.

데이터 변경을 가져올 수 있는 두 가지 원인이 있습니다. 하드웨어 및 전송 오류이거나 의도적인 침입을 통해서입니다. 다수의 하드웨어 제품 및 전송 프로토콜에는 하드웨어 및 전송 오류를 감지하고 해결하는 메커니즘이 있습니다. 데이터 무결성 서비스의 목적은 의도적인 침입을 감지하는 것입니다.

데이터 무결성 서비스는 데이터가 수정되었는지 여부를 감지하는 것만을 목적으로 합니다. 데이터가 수정된 경우에 데이터를 원래 상태로 복원하는 것은 목적으로 하지 않습니다.

액세스 제어 메커니즘은 액세스가 거부되면 데이터를 수정할 수 없다는 점에 있어서 데이터 무결성에 기여할 수 있습니다. 그러나, 기밀성과 마찬가지로 네트워크 환경에서는 액세스 제어 메커니즘은 효율적이지 않습니다.

### IBM MQ에서의 데이터 무결성

다음과 같이 IBM MQ 환경에서 데이터 무결성이 보장됩니다.

- TLS를 사용하여 메시지 컨텐츠가 네트워크를 통해 전송 중에 의도적으로 수정되었는지 감지할 수 있습니다. TLS에서 메시지 요약 알고리즘은 전송 중인 수정된 메시지 감지를 제공합니다.

  모든 IBM MQ CipherSpec은 메시지 요약 알고리즘을 제공하며 메시지 데이터 무결성을 제공하지 않는 TLS_RSA_WITH_NULL_NULL은 예외입니다.

  IBM MQ 는 메시지를 수신할 때 수정된 메시지를 발견합니다. 수정된 메시지를 수신할 때 IBM MQ AMQ9661 오류 메시지가 오류 로그에 기록되고 채널이 중지됩니다.
- 메시지가 로컬 큐에 저장되는 경우 IBM MQ에서 제공되는 메시지 제어 메커니즘은 메시지 컨텐츠의 고의적인 수정을 보호하기에 충분합니다.

  그렇지만 최고 레벨을 보안을 위해 Advanced Message Security를 사용하여 큐에 메시지를 넣을 때부터 큐에서 검색되는 사이의 시간 동안 메시지 컨텐츠가 고의적으로 수정되었는지를 감지할 수 있습니다.

  수정된 메시지가 감지되면 메시지를 수신하려고 시도하는 애플리케이션이 MQRC_SECURITY_ERROR (2063) 리턴 코드를 수신합니다. 애플리케이션이 MQGET 호출을 사용 중인 경우 메시지도 SYSTEM.PROTECTION.ERROR.QUEUE 큐를 큐에 넣습니다.

## 암호화 개념

이 주제 모음에서는 IBM MQ에 적용할 수 있는 암호화 개념에 대해 설명합니다.

용어 엔티티는 큐 관리자, IBM MQ MQI client, 개별 사용자 또는 메시지를 교환할 수 있는 기타 다른 시스템을 나타내기 위하여 사용됩니다.

### 암호화

암호화(cryptography)는 읽기 가능한 텍스트인 일반 텍스트와 읽을 수 없는 양식인 암호문 사이에서의 변환 프로세스입니다.

이는 다음과 같이 발생합니다.

1. 송신자가 일반 텍스트 메시지를 암호문으로 변환합니다. 이 프로세스 파트를 암호화*(encryption)*(때로 암호화*(encipherment)*)라고 합니다.
2. 암호문은 수신자에게 전송됩니다.
3. 수신자는 암호문 메시지를 다시 일반 텍스트 양식으로 변환합니다. 이 프로세스 파트를 복호화*(decryption)*(때로 판독*(decipherment)*)이라고 합니다.

변환은 전송하는 동안에 메시지의 모양을 변경하나 컨텐츠에 영향을 미치지 않는 일련의 산술적인 연산으로 진행됩니다. 암호화된 메시지는 이해할 수 없기 때문에 암호화 기술을 통해 기밀성이 유지되고 권한 없는 보기(도청)로부터 메시지를 보호합니다. 메시지 무결성을 보장하는 디지털 서명은 암호화 기술을 사용합니다. 자세한 정보는 21 페이지의 『SSL/TLS의 디지털 서명』의 내용을 참조하십시오.

암호화 기술은 키의 사용에 의해 특정하게 만들어진 일반적인 알고리즘과 관계가 있습니다. 알고리즘에는 두 가지로 분류됩니다.

- 두 당사자가 모두 같은 보안 키를 사용하도록 하는 알고리즘. 공유 키를 사용하는 알고리즘은 대칭 알고리즘이라고 합니다. 11 페이지의 그림 1에서는 대칭 키 암호화에 대해 설명합니다.
- 암호화에 임의 키를 사용하고 복호화에 다른 키를 사용하는 알고리즘. 이들 중 하나는 비밀로 유지되어야 하지만 다른 하나는 공개될 수 있습니다. 공개 및 개인 키 쌍을 사용하는 알고리즘을 비대칭 알고리즘이라고 합니다. 11 페이지의 그림 2에서는 공개 키 암호화라고도 하는 비대칭 키 암호화에 대해 설명합니다.

사용되는 암호화 및 복호화 알고리즘은 공개될 수 있지만 공유 보안 키와 개인 키는 비밀로 유지해야 합니다.

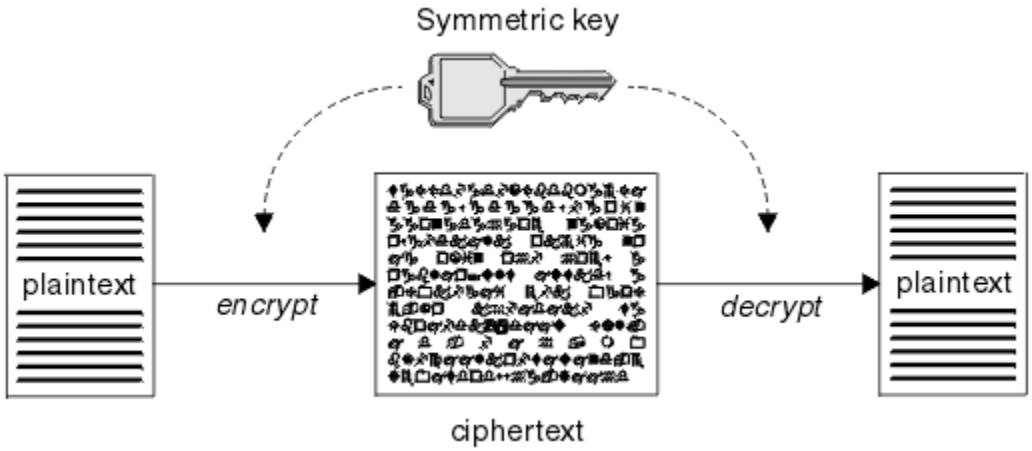



그림 *1.* 대칭 키 암호화

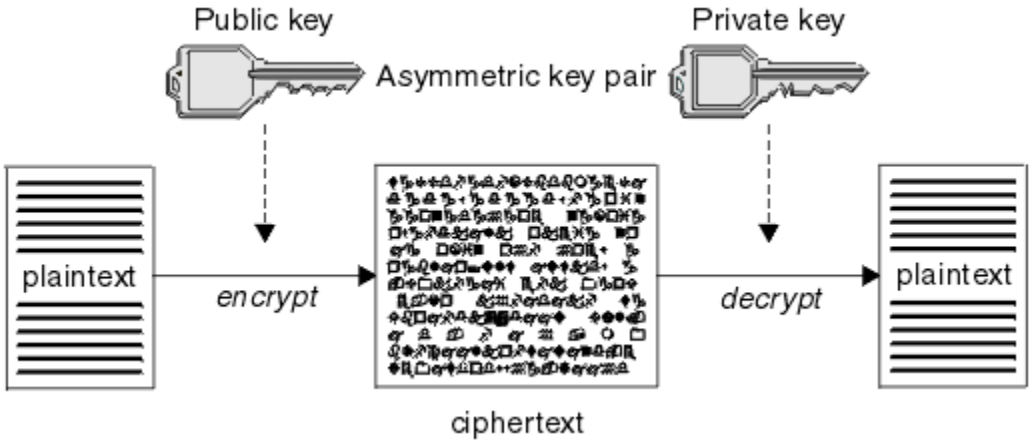



그림 *2.* 비대칭 키 암호화

11 페이지의 그림 2에서는 수신자의 공개 키로 암호화되고 수신자의 개인 키로 복호화되는 일반 텍스트를 보여줍니다. 의도한 수신자만 암호문 복호화를 위한 개인 키를 보유합니다. 메시지가 송신자로부터 나온다는 보장과 함께, 송신자가 개인 키로 메시지를 암호화할 수 있으며, 이렇게 하면 송신자의 공개 키를 가지고 있는 누구나 메시지를 복호화할 수 있다는 점에 주의하십시오.

비대칭 알고리즘에서 메시지는 공개 또는 개인 키로 암호화되지만 다른 키로만 복호화될 수 있습니다. 개인 키만 비밀이고, 공개 키는 누구나 알 수 있습니다. 대칭 알고리즘에서는 공유 키는 두 당사자에게만 알려져 있어야 합니다. 이를 키 분배 문제점이라고 합니다. 비대칭 알고리즘이 더 느리지만, 키 분배 문제점이 없다는 장점이 있습니다.

암호화에 연관되는 기타 용어는 다음과 같습니다.

**강도**

암호화 강도는 키 크기로 판별됩니다. 비대칭 알고리즘의 경우 큰 키가 필요합니다. 예를 들어,

| | |
|---|---|
| 1024비트 | 낮은 강도의 비대칭 키 |
| 2048비트 | 중간 강도의 비대칭 키 |
| 4096비트 | 높은 강도의 비대칭 키 |

대칭 키는 더 작습니다. 256비트 키가 강력한 암호화를 제공합니다.

**블록 암호 알고리즘**

이 알고리즘은 데이터를 블록으로 암호화합니다. 예를 들어, RSA Data Security Inc.의 RC2 알고리즘은 블록 8바이트 길이를 사용합니다. 블록 알고리즘은 일반적으로 스트림 알고리즘보다 느립니다.

**스트림 암호 알고리즘**

이 알고리즘은 각 데이터 바이트에서 조작됩니다. 스트림 알고리즘은 일반적으로 블록 알고리즘보다 빠릅니다.

## 메시지 요약 및 디지털 서명

메시지 요약은 메시지 컨텐츠에 대한 고정된 크기의 숫자 표시입니다. 메시지 요약은 해시 함수로 처리되어 디지털 서명을 양식화해서 암호화 가능합니다.

메시지 요약 처리에 사용되는 해시 함수는 다음 두 기준을 충족해야 합니다.

- 단방향이어야 합니다. 가능한 모든 메시지를 테스트하지 않고 특정 메시지 요약에 해당하는 메시지를 찾기 위해 함수를 되돌리는 것이 가능하면 안됩니다.
- 같은 요약으로 해시되는 두 개의 메시지를 찾기 위한 처리는 불가능해야 합니다.

메시지 요약은 메시지 자체와 같이 송신됩니다. 수신자는 메시지에 대한 요약을 생성하고 이를 송신자의 요약과 비교할 수 있습니다. 메시지 무결성은 두 메시지 요약이 동일한 경우에 확인됩니다. 전송하는 동안에 메시지를 도용하면 거의 분명하게 다른 메시지 요약이 생깁니다.

비밀 대칭 키를 사용하여 작성되는 메시지 요약은 메시지 인증 코드(MAC)라고 하는데 이를 통해 메시지가 수정되지 않았음을 확인할 수 있기 때문입니다.

송신자도 메시지 요약을 생성하고 디지털 서명을 양식화하는 비대칭 키 쌍의 개인 키를 사용하여 요약을 암호화할 수 있습니다. 그런 다음 서명은 로컬에서 생성된 요약과 비교하기 전에 수신자에서 복호화되어야 합니다.

**관련 개념**

21 페이지의 『SSL/TLS의 디지털 서명』

디지털 서명은 메시지의 표현을 암호화하여 형성됩니다. 암호화는 서명인의 개인 키를 사용하고, 효율성을 위해 보통 메시지 자체가 아니라 메시지 요약에 대해 작동합니다.

## 디지털 인증서

디지털 인증서는 위장으로부터 보호하고 공개 키가 지정된 엔티티에 속하는지 확인합니다. 이는 인증 기관에서 발행됩니다.

디지털 인증서는 소유자가 개인, 큐 관리자 또는 다른 어떤 엔티티인지에 상관없이 공개 키를 해당 소유자로 바인딩하기 때문에 위장에 대한 보호를 제공합니다. 비대칭 키 설계를 사용할 때 공개 키의 소유권에 대해 보장해주기 때문에, 디지털 인증서를 공개 키 인증서라고도 합니다. 디지털 인증서에는 엔티티에 대한 공개 키가 있고 공개 키가 그 엔티티에 속한다는 언급이 있습니다.

- 인증서가 개별 엔티티에 대한 것인 경우, 인증서는 개인 인증서 또는 사용자 인증서라고 합니다.
- 인증서가 인증 기관에 대한 것인 경우, 인증서는 *CA* 인증서 또는 서명자 인증서라고 합니다.

해당 소유자가 공개 키를 다른 엔티티로 직접 송신하는 경우, 메시지가 인터셉트되고 공개 키는 다른 키로 대체될 위험이 있습니다. 이는 중간자 공격이라고 합니다. 이 문제에 대한 솔루션은 신뢰하는 써드파티를 통해 공개 키를 교환하는 것으로 이를 통해 공개 키가 실제 통신 중인 엔티티에 속하는 것임을 강력하게 보장할 수 있습니다. 공개 키를 직접 송신하는 대신, 신뢰하는 써드파티에게 이를 디지털 인증서로 통합하도록 요청합니다. 디지털 인증서를 발행하는 신뢰하는 써드파티는 13 페이지의 『인증 기관』에서 설명하는 것처럼 인증 기관(CA)이라고 합니다.

## 디지털 인증서의 내용

디지털 인증서에는 X.509 표준으로 판별되는 정보의 특정 부분이 포함됩니다.

IBM MQ에서 사용하는 디지털 인증서는 X.509 표준을 준수하며 이는 필요한 정보 및 송신 형식을 지정합니다. X.509는 표준의 X.500 시리즈의 인증 프레임워크 부분입니다.

디지털 인증서에는 적어도 인증되고 있는 엔티티에 대한 다음 정보가 있습니다.

- 소유자 공개 키
- 소유자 식별 이름
- 인증서를 발행한 CA의 식별 이름
- 인증서 유효 시작 날짜
- 인증서 만기 날짜
- X.509에 정의된 것과 같은 인증서 데이터 형식의 버전 번호. 현재 X.509 표준 버전은 버전 3이며 대부분의 인증서가 해당 버전을 준수합니다.
- 일련 번호. 이는 인증서를 발행한 CA에서 지정하는 고유 ID입니다. 일련 번호는 인증서를 발행한 CA 내에서 고유합니다. 동일한 CA 인증서로 서명된 두 개의 인증서가 동일한 일련 번호를 가질 수는 없습니다.

X.509 버전 2 인증서도 발행인 ID 및 해당 ID를 포함하며 X.509 버전 3 인증서는 몇 개의 확장자를 포함할 수 있습니다. 기본 제한조건 확장자와 같은 일부 인증서 확장자는 표준이지만 다른 확장자는 구현 특정적입니다. 확장자는 중요할 수 있으며 이 경우 시스템은 필드를 인식할 수 있어야 하고 필드를 인식하지 않으면 인증서를 거부해야 합니다. 확장자가 중요하지 않은 경우에는 시스템이 이를 인식하지 못하는 경우 무시할 수 있습니다.

개인 인증서의 디지털 서명은 해당 인증서를 서명한 CA의 개인 키를 사용하여 생성됩니다. 개인 인증서를 확인해야 하는 모든 사용자는 CA 공개 키를 사용하여 확인 가능합니다. CA 인증서는 해당 공개 키를 포함합니다.

디지털 인증서는 개인 키를 포함하지 않습니다. 개인 키는 안전하게 보관해야 합니다.

## 개인 인증서에 대한 요구사항

IBM MQ에서는 X.509 표준을 준수하는 디지털 인증서를 지원합니다. 이 경우 클라이언트 인증 옵션이 필요합니다.

IBM MQ가 피어 투 피어 시스템이기 때문에 SSL/TLS 용어에서 클라이언트 인증으로 표시됩니다. 따라서 클라이언트 인증의 키 사용을 허용하려면 SSL/TLS 인증에 사용되는 개인 인증서가 필요합니다. 모든 인증서에서 이 옵션이 사용되지는 않기 때문에 인증서 제공자는 보안 인증서에 대한 루트 CA에서 클라이언트 인증이 사용되도록 해야 합니다.

디지털 인증서의 데이터 형식을 지정하는 표준뿐만 아니라 인증서의 유효성을 판별하는 표준도 있습니다. 이러한 표준은 특정 유형의 보안 위반을 방지하기 위해 시간에 걸쳐 업데이트되었습니다. 예를 들어, 이전의 X.509 버전 1 및 2 인증서는 인증서가 다른 인증서 서명에 적법하게 사용될 수 있는지를 표시하지 않았습니다. 따라서, 악의적인 사용자가 합법적인 소스에서 개인 인증서를 확보하여 다른 사용자를 모방하도록 설계된 새 인증서를 생성할 수도 있었습니다.

X.509 버전 3 인증서를 사용하는 경우 BasicConstraints 및 KeyUsage 인증서 확장자가 다른 인증서를 적법하게 서명할 수 있는 인증서를 지정하는 데 사용됩니다. IETF RFC 5280 표준은 준수 애플리케이션 소프트웨어가 위장 공격을 방지하기 위해 구현해야 하는 일련의 인증서 유효성 검증 규칙을 지정합니다. 인증서 규칙 세트는 인증서 유효성 검증 정책이라고도 합니다.

IBM MQ에서의 인증서 유효성 검증 정책에 대한 자세한 정보는 42 페이지의 『IBM MQ의 인증서 유효성 검증 정책』의 내용을 참조하십시오.

## 인증 기관

인증 기관(CA)은 엔티티의 공개 키가 진짜로 해당 엔티티에 속하는지 확인할 수 있도록 디지털 인증서를 발행하는 신뢰 가능한 써드파티입니다.

CA의 역할은 다음과 같습니다.

- 디지털 인증서에 대한 요청을 수신하면, 개인 인증서를 빌드, 서명, 리턴하기 전에 요청자의 ID를 확인
- 해당 CA 인증서에서 CA의 자체 공개 키를 제공
- 인증서 폐기 목록(CRL)에서 더 이상 신뢰되지 않는 인증서 목록을 공개. 자세한 정보는 318 페이지의 『폐기된 인증서에 대한 작업』의 내용을 참조하십시오.

- OCSP 응답자 서버를 운영하여 인증서 폐기 상태에 대한 액세스 제공

### 식별 이름

식별 이름(DN)은 X.509 인증서 내의 엔티티를 고유하게 식별합니다.

**주의:** 다음 표의 속성만 SSLPEER 필터에서 사용될 수 있습니다. 인증서 DN은 그 외의 속성을 포함할 수 있지만, 이러한 속성에 대해서는 필터링이 허용되지 않습니다.

표 *1. SSLPEER* 필터에서 사용할 수 있는 *DN*의 속성 유형

| 속성 유형 | 설명 |
|---|---|
| SERIALNUMBER | 인증서 일련 번호 |
| 메일 | 이메일 주소 |
| Deprecated E | 이메일 주소(MAIL보다 우선적으로는 더 이상 사용되지 않음) |
| UID 또는 USERID | 사용자 ID |
| CN | 공용 이름 |
| T | 제목 |
| OU | 조직 단위 이름 |
| DC | 도메인 컴포넌트 |
| O | 조직 이름 |
| STREET | 상세 주소/주소 두 번째 줄 |
| L | 지역 이름 |
| ST(또는 SP 또는 S) | 시 또는 도 이름 |
| PC | 우편번호 |
| C | 국가 |
| UNSTRUCTUREDNAME | 호스트 이름 |
| UNSTRUCTUREDADDRESS | IP 주소 |
| DNQ | 식별 이름 규정자 |

X.509 표준은 보통 DN의 부분을 이루지 않는 다른 속성을 정의하나, 선택적인 확장을 디지털 인증서에게 제공할 수 있습니다.

X.509 표준은 DN이 문자열 형식으로 지정되도록 제공합니다. 예를 들면, 다음과 같습니다.

```
CN=John Smith, OU=Test, O=IBM, C=GB
```

공용 이름(CN)은 개인 사용자나, 예를 들어 웹 서버와 같은 다른 모든 엔티티를 설명할 수 있습니다.

DN에 여러 개의 OU 및 DC 속성이 포함될 수 있습니다. 기타 각 속성의 경우 하나의 인스턴스만 허용됩니다. OU 입력 항목의 순서는 중요합니다. 순서가 조직 단위(OU) 이름의 계층을 최상위 레벨 단위부터 먼저 지정합니다. DC 입력 항목의 순서도 중요합니다.

IBM MQ에서는 생성 결함이 있는 특정 DN가 허용됩니다. 자세한 정보는 SSLPEER 값에 대한 IBM MQ 규칙을 참조하십시오.

**관련 개념**

13 페이지의 『디지털 인증서의 내용』
디지털 인증서에는 X.509 표준으로 판별되는 정보의 특정 부분이 포함됩니다.

## 인증 기관에서 개인 인증서 확보

신뢰할 수 있는 외부 인증 기관(CA)에서 인증서를 확보할 수 있습니다.

디지털 인증서는 인증서 요청 양식으로 정보를 CA에 보내서 확보할 수 있습니다. X.509 표준이 이 정보에 대한 형식을 정의하지만 일부 CA는 자체적으로 고유 형식이 있습니다. 인증서 요청은 일반적으로 시스템에서 사용하는 인증서 관리 도구로 생성됩니다. 예:

- ALW AIX, Linux, and Windows에서 **runmqakm** 및 V 9.4.0 V 9.4.0 **runmqktool** 명령.
- z/OS z/OS의 RACF®.

정보에는 식별 이름 및 공개 키가 포함됩니다. 인증서 관리 도구가 인증서 요청을 생성하면 개인 키도 생성하며 이는 안전하게 보관해야 합니다. 개인 키는 분배하지 마십시오.

CA가 요청을 수신하면, 기관이 인증서를 빌드하고 이를 개인 인증서로 사용자에게 리턴하기 전에 ID를 확인합니다.

15 페이지의 그림 3는 CA로부터 디지털 인증서를 확보하는 프로세스를 보여줍니다.

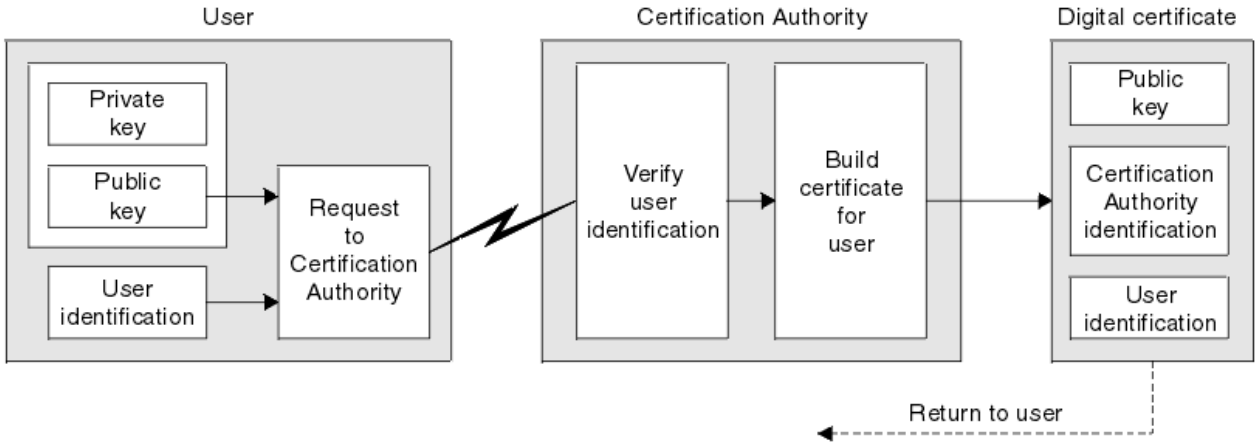



그림 *3.* 디지털 인증서 확보

다이어그램에서:

- 사용자 ID에는 제목 식별 이름이 포함됩니다.
- 인증 기관 ID에는 인증서를 발행하는 인증 기관의 식별 이름이 포함됩니다.

디지털 인증서에는 다이어그램에 표시되는 필드 이외에도 추가 필드가 있습니다. 디지털 인증서의 다른 필드에 대한 자세한 정보는 13 페이지의 『디지털 인증서의 내용』의 내용을 참조하십시오.

## 인증서 체인 작동 방법

다른 엔티티를 위해 인증서를 수신하면, 루트 *CA* 인증서를 얻기 위해 인증서 체인을 사용해야 할 수 있습니다.

인증 경로라고도 하는 인증서 체인은 엔티티를 인증하는 데 사용되는 인증서 목록입니다. 체인, 또는 경로는 그 엔티티의 인증서로 시작하고, 체인에 있는 각 인증서는 체인에 있는 다음 인증서에서 식별하는 엔티티에 의해 서명됩니다. 체인은 루트 CA 인증서로 종료됩니다. 루트 CA 인증서는 항상 인증 기관(CA) 자체로 서명됩니다. 체인의 모든 인증서 서명은 루트 CA 인증서에 도달할 때까지 확인해야 합니다.

16 페이지의 그림 4에서는 인증서 소유자로부터 신뢰 체인이 시작되는 루트 CA까지의 인증 경로를 보여줍니다.

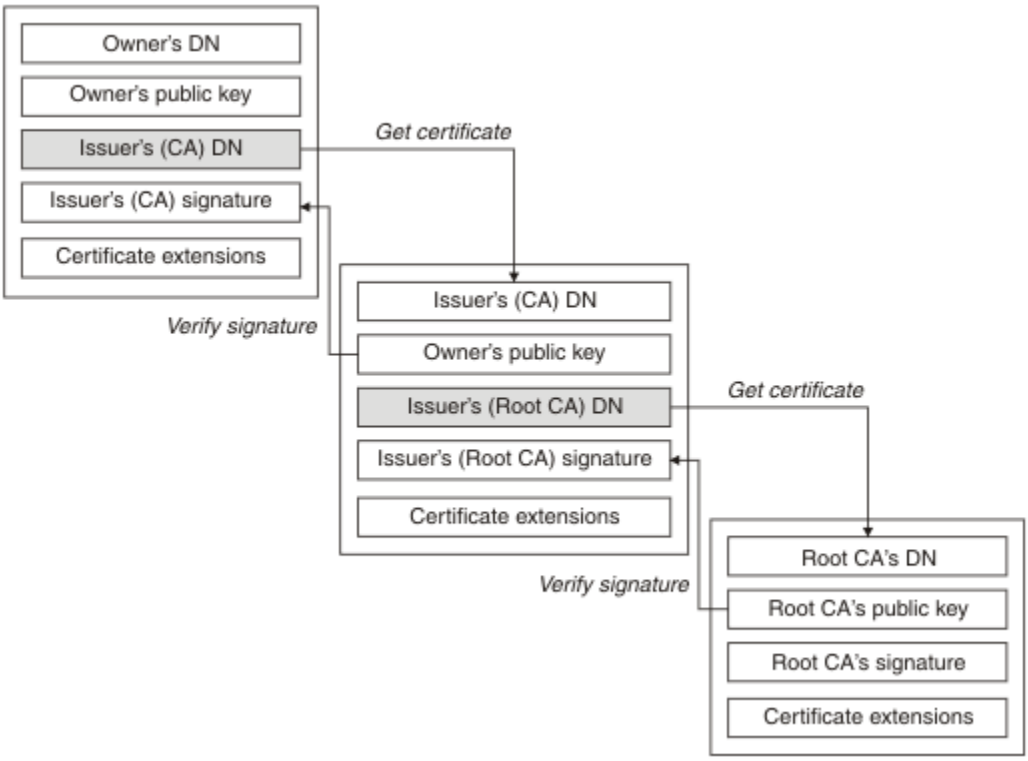



그림 *4.* 신뢰 체인

각 인증서는 하나 이상의 확장을 포함할 수 있습니다. CA에 속하는 인증서는 다른 인증서의 서명을 허용하도록 표시하기 위해 일반적으로 isCA 플래그가 설정되는 BasicConstraints 확장을 포함합니다.

### 인증서가 더 이상 유효하지 않은 경우

디지털 인증서는 만기되거나 폐기될 수 있습니다.

디지털 인증서는 고정된 기간에 대해 발행되고 그 만기 날짜 이후에는 유효하지 않습니다.

인증서는 다음을 비롯하여 다양한 이유로 폐기될 수 있습니다.

- 소유자가 다른 조직으로 이동했습니다.
- 개인 키가 더 이상 비밀이 아닙니다.

IBM MQ는 OCSP(Online Certificate Status Protocol) 응답자(AIX, Linux, and Windows전용)에 요청을 송신하여 인증서 폐기 여부를 검사할 수 있습니다. 또는, LDAP 서버의 인증서 폐기 목록(CRL)에 액세스할 수 있습니다. OCSP 폐기 및 CRL 정보는 인증 기관이 발행합니다. 자세한 정보는 318 페이지의 『폐기된 인증서에 대한 작업』의 내용을 참조하십시오.

## 공개 키 기반 구조(PKI)

PKI(Public Key Infrastructure)는 트랜잭션에 관련된 당사자들을 인증하기 위해 공개 키 암호화의 사용을 지원하는 기능, 정책, 서비스 시스템입니다.

PKI(Public Key Infrastructure)의 컴포넌트를 정의하는 단일 표준은 없지만 PKI는 일반적으로 인증 기관(CA) 및 등록대행 기관(RA)로 구성됩니다. CA는 다음 서비스를 제공합니다.

- 디지털 인증서 발행
- 디지털 인증서 유효성 검증
- 디지털 인증서 폐기
- 공개 키 분배

X.509 표준은 업계 표준의 PKI(Public Key Infrastructure)에 대한 기반을 제공합니다.

디지털 인증서 및 인증 기관(CA)에 대한 자세한 정보는 12 페이지의 『디지털 인증서』의 내용을 참조하십시오. RA는 디지털 인증서가 요청되면 제공된 정보를 확인합니다. RA가 해당 정보를 확인하면, CA가 디지털 인증서를 요청자에게 발행할 수 있습니다.

PKI는 디지털 인증서와 공개 키를 관리하기 위한 도구도 제공할 수 있습니다. PKI는 디지털 인증서를 관리하기 위한 신뢰 계층으로 설명되기도 하지만 대부분의 정의에는 추가 서비스가 있습니다. 일부 정의는 암호화 및 디지털 서명 서비스를 포함하지만 해당 서비스는 PKI 조작에 꼭 필요하지는 않습니다.

# 암호화 보안 프로토콜: TLS

암호화 프로토콜은 두 당사자가 개인정보 보호 및 데이터 무결성으로 통신하도록 하여 안전한 연결을 제공합니다. TLS(Transport Layer Security) 프로토콜은 SSL(Secure Socket Layer) 프로토콜에서 발전한 것입니다. IBM MQ는 TLS를 지원합니다.

두 프로토콜의 기본 목표는 기밀성(개인정보 보호라고도 함), 데이터 무결성, 인증, 디지털 인증서를 사용하는 인증을 제공하는 것입니다.

두 프로토콜이 유사하긴 하지만 SSL 3.0과 TLS의 다양한 버전이 상호 운용되지 않는다는 점은 상당한 차이점입니다.

**관련 개념**

22 페이지의 『IBM MQ의 TLS 보안 프로토콜』
IBM MQ는 메시지 채널 및 MQI 채널에 대한 링크 레벨 보안을 제공하기 위해 TLS(Transport Layer Security) 프로토콜을 지원합니다.

## TLS(Transport Layer Security) 개념

TLS 프로토콜을 사용하면 두 당사자가 기밀성과 데이터 무결성이 보장된 상태로 서로 식별 및 인증하고 통신할 수 있습니다. TLS 프로토콜은 Netscape SSL 3.0 프로토콜에서 발전되었으나 TLS 및 SSL이 상호 운영되지는 않습니다.

TLS 프로토콜은 인터넷에서 통신 보안을 제공하며, 클라이언트/서버 애플리케이션이 비밀리에 안정적인 방식으로 통신할 수 있도록 허용합니다. 프로토콜에는 레코드 프로토콜과 데이터 교환 프로토콜의 두 개의 계층이 있으며 이들은 TCP/IP 등과 같은 전송 프로토콜 위에 계층을 이루고 있습니다. 이들은 비대칭 및 대칭 암호화 기술을 모두 사용합니다.

TLS 연결은 애플리케이션에 의해 시작되고 이는 TLS 클라이언트가 됩니다. 연결을 수신하는 애플리케이션은 TLS 서버가 됩니다. 모든 새 세션은 TLS 프로토콜에 의해 정의된 대로 데이터 교환으로 시작됩니다.

IBM MQ가 지원하는 모든 CipherSpec 목록은 392 페이지의 『CipherSpec 사용 가능』에 있습니다.

SSL 프로토콜에 대한 자세한 정보는 https://developer.mozilla.org/docs/Mozilla/Projects/NSS에서 제공하는 정보를 참조하십시오. TLS 프로토콜에 대한 자세한 정보는 IETF(Internet Engineering Task Force) 웹 사이트(https://www.ietf.org)에서 TLS Working Group이 제공하는 정보를 참조하십시오.

## SSL/TLS 데이터 교환의 개요

SSL/TLS 데이터 교환을 사용하면 통신하는 TLS 클라이언트와 서버가 보안 키를 설정할 수 있습니다.

이 절에서는 TLS 클라이언트 및 서버가 서로 통신할 수 있도록 하는 단계의 요약을 제공합니다.

- 사용하는 프로토콜 버전 동의.
- 암호화 알고리즘 선택.
- 디지털 인증서를 교환 및 유효성 검증하여 서로 인증.
- 공유 보안 키 생성을 위한 키 분배 문제점을 막아주는 비대칭 암호화 기술 사용. 그러면 TLS는 메시지에 대해 비대칭 암호화보다 빠른 대칭 암호화의 공유 키를 사용합니다.

암호화 알고리즘 및 디지털 인증서에 대한 자세한 정보는 관련 정보를 참조하십시오.

대략적으로, TLS 데이터 교환에 연관된 단계는 다음과 같습니다.

1. TLS 클라이언트가 TLS 버전과 클라이언트가 지원하는 CipherSuite(클라이언트 환경 설정 순서대로) 등의 암호화 정보를 나열하는 "client hello" 메시지를 송신합니다. 메시지에는 나중의 처리에서 사용되는 무작위 바

이트 문자열도 포함됩니다. 프로토콜은 "client hello"가 클라이언트에서 지원되는 데이터 압축 메소드도 포함하도록 허용합니다.

2. TLS 서버는 클라이언트가 제공하는 목록에서 서버가 선택한 CipherSuite, 세션 ID, 다른 무작위 바이트 문자열을 포함하는 "server hello" 메시지로 응답합니다. 서버는 해당 디지털 인증서도 송신합니다. 서버가 클라이언트 인증을 위해 디지털 인증서를 필요로 하면, 서버가 지원되는 인증서 유형 목록과 허용 가능한 인증 기관(CA)의 식별 이름을 포함하는 "클라이언트 인증서 요청"을 송신합니다.
3. TLS 클라이언트는 서버의 디지털 인증서를 확인합니다. 자세한 정보는 18 페이지의 『TLS가 식별, 인증, 기밀성, 무결성을 제공하는 방법』의 내용을 참조하십시오.
4. TLS 클라이언트는 클라이언트와 서버 모두가 후속 메시지 데이터 암호화에 사용되는 보안 키를 처리할 수 있도록 하는 무작위 바이트 문자열을 송신합니다. 무작위 바이트 문자열 자체는 서버의 공개 키로 암호화됩니다.
5. TLS 서버가 "클라이언트 인증서 요청"을 송신하면 클라이언트는 클라이언트의 개인 키와 클라이언트의 디지털 인증서 또는 "디지털 인증서 경보 없음"을 같이 암호화한 무작위 바이트 문자열을 송신합니다. 이 경보는 단지 경고일 뿐이지만 일부 구현에서 클라이언트 인증이 필수인 경우에 데이터 교환은 실패합니다.
6. TLS 서버는 클라이언트 인증서를 확인합니다. 자세한 정보는 18 페이지의 『TLS가 식별, 인증, 기밀성, 무결성을 제공하는 방법』의 내용을 참조하십시오.
7. TLS 클라이언트는 데이터 교환의 클라이언트 측이 완료되었음을 표시하는 보안 키와 같이 암호화되는 "완료됨" 메시지를 서버에 송신합니다.
8. TLS 서버는 데이터 교환의 서버 측이 완료되었음을 표시하는 보안 키와 같이 암호화되는 "완료됨" 메시지를 클라이언트에 송신합니다.
9. 이제 TLS 세션 동안 서버와 클라이언트는 공유 보안 키로 대칭적으로 암호화되는 메시지를 교환할 수 있습니다.

18 페이지의 그림 5은 TLS 데이터 교환을 설명합니다.

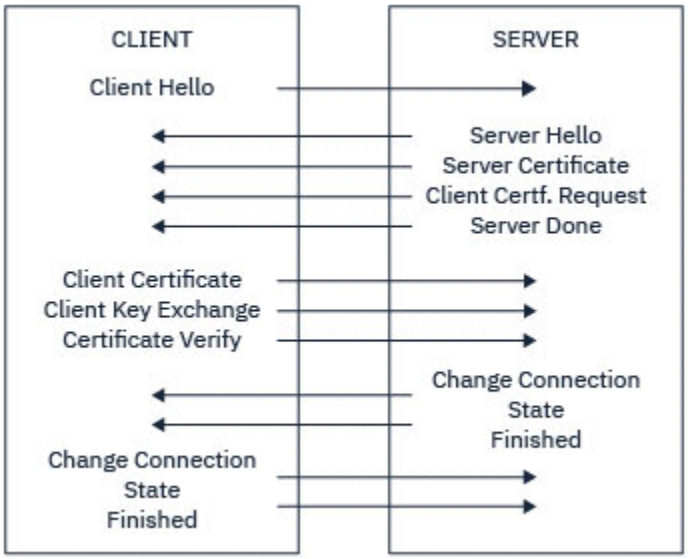



그림 *5. TLS* 데이터 교환의 개요

## TLS가 식별, 인증, 기밀성, 무결성을 제공하는 방법

클라이언트와 서버 둘 다의 인증 동안에 비대칭 키 쌍 중 하나로 데이터가 암호화되고 해당 쌍의 다른 키로 비밀번호가 복호화되어야 하는 단계가 있습니다. 메시지 요약은 무결성을 제공하는 데 사용됩니다.

TLS 핸드쉐이크와 관련된 단계에 대한 개요는 17 페이지의 『SSL/TLS 데이터 교환의 개요』의 내용을 참조하십시오.

## TLS가 인증을 제공하는 방법

서버 인증에서는 클라이언트가 보안 키를 처리하는 데 사용되는 데이터를 암호화하기 위해 서버의 공개 키를 사용합니다. 서버가 정확한 개인 키로 데이터의 비밀번호를 복호화하는 경우에만 보안 키를 생성할 수 있습니다. 무작위 바이트 문자열 자체는 서버의 공개 키로 암호화됩니다(개요에 있는 18 페이지의 『4』 단계).

클라이언트 인증에서는 서버가 데이터 교환의 18 페이지의 『5』 단계 동안에 클라이언트가 송신하는 데이터의 비밀번호를 해독하기 위해 클라이언트 인증서에 있는 공개 키를 사용합니다. 보안 키로 암호화된 완료 메시지의 교환(개요에 있는 단계 18 페이지의 『7』 및 18 페이지의 『8』)으로 인증이 완료되었음을 확인합니다.

임의 인증 단계가 실패하면, 데이터 교환이 실패하고 세션이 종료됩니다.

TLS 데이터 교환 중의 디지털 인증서 교환은 인증 프로세스의 부분입니다. 인증서가 위장에 대해 보호를 제공하는 방법에 대한 자세한 정보는 관련 정보를 참조하십시오. 필수 인증서는 다음과 같습니다. 여기서 CA X는 TLS 클라이언트에게 인증서를 발행하고, CA Y는 TLS 서버에게 인증서를 발행합니다.

서버 인증의 경우에 한해 TLS 서버는 다음을 필요로 합니다.

- CA Y에 의해 서버에게 발행된 개인 인증서
- 서버 개인 키

또한 TLS 클라이언트는 다음을 필요로 합니다.

- CA Y에 대한 CA 인증서

TLS 서버에 클라이언트 인증이 필요한 경우 서버는 클라이언트에 개인 인증서를 발행한 CA (이 경우 CA X) 의 공개 키를 사용하여 클라이언트의 디지털 인증서를 확인하여 클라이언트의 ID를 확인합니다. 서버 및 클라이언트 인증을 위해 서버는 다음을 필요로 합니다.

- CA Y에 의해 서버에게 발행된 개인 인증서
- 서버 개인 키
- CA X에 대한 CA 인증서

또한 클라이언트는 다음을 필요로 합니다.

- CA X에 의해 클라이언트에 발행된 개인 인증서
- 클라이언트의 개인 키
- CA Y에 대한 CA 인증서

TLS 서버와 클라이언트 둘 다 루트 CA 인증서로 인증서 체인을 형성하기 위해 다른 CA 인증서를 필요로 할 수 있습니다. 인증서 체인에 대한 자세한 정보는 관련 정보를 참조하십시오.

## 인증서 확인 중 발생하는 사항

개요의 18 페이지의 『3』 및 18 페이지의 『6』 단계에 기술된 대로 TLS 클라이언트는 서버의 인증서를 확인하고 TLS 서버는 클라이언트의 인증서를 확인합니다. 이 확인에 대한 네 가지 관점이 있습니다.

1. 디지털 서명이 검사됩니다(21 페이지의 『SSL/TLS의 디지털 서명』 참조).
2. 인증서 체인이 검사됩니다. CA 인증서를 중개해야 합니다(15 페이지의 『인증서 체인 작동 방법』 참조).
3. 만기 및 활성화 날짜와 검증 기간을 검사합니다.
4. 인증서의 폐기 상태를 검사합니다(318 페이지의 『폐기된 인증서에 대한 작업』 참조).

## 보안 키 재설정

TLS 데이터 교환 중에 TLS 클라이언트와 서버 간의 데이터를 암호화하기 위해 보안 키가 생성됩니다. 보안 키는 일반 텍스트를 읽을 수 없는 암호문으로 변환하고 암호문을 일반 텍스트로 변환할 수 있도록 데이터에 적용되는 수학 공식에서 사용됩니다.

보안 키는 데이터 교환 중에 송신되는 무작위 텍스트에서 생성되면 일반 텍스트를 암호문으로 암호화하는 데 사용됩니다. 또한 보안 키는 메시지의 대체 여부를 판별하는 데 사용되는 메시지 인증 코드(MAC) 알고리즘에 사용됩니다. 자세한 정보는 12 페이지의 『메시지 요약 및 디지털 서명』의 내용을 참조하십시오.

보안 키가 발견된 경우 메시지의 일반 텍스트를 암호문에서 복호화하거나 메시지 요약을 계산하여 감지하지 않고 메시지를 대체할 수 있습니다. 복잡한 알고리즘의 경우에도 암호문에 가능한 모든 수학적 변환을 적용하여 일반 텍스트를 발견할 수 있습니다. 보안 키가 절단된 경우 복호화하거나 대체할 수 있는 데이터의 양을 최소화하기 위해 보안 키를 주기적으로 재협상할 수 있습니다. 보안 키가 재협상되면 이전에 사용하던 보안 키는 새 보안 키로 암호화된 데이터를 복호화하는 데 더 이상 사용할 수 없습니다.

### TLS가 기밀성을 제공하는 방법

TLS는 메시지 개인정보 보호를 보장하기 위해 대칭과 비대칭 암호화의 결합을 사용합니다. TLS 데이터 교환 중에 TLS 클라이언트와 서버는 하나의 세션에만 사용될 암호화 알고리즘과 공유 보안 키에 동의합니다. TLS 클라이언트와 서버 사이에 전송되는 모든 메시지는 해당 알고리즘과 키를 사용하여 암호화되며, 메시지가 인터셉트되어도 반드시 개인용으로 남아있게 됩니다. TLS가 공유 보안 키를 전송할 때는 비대칭 암호화를 사용하기 때문에, 키 분배 문제점이 없습니다. 암호화 기술에 대한 자세한 정보는 10 페이지의 『암호화』의 내용을 참조하십시오.

### TLS가 무결성을 제공하는 방법

TLS는 메시지 축약을 계산하여 데이터 무결성을 제공합니다. 자세한 정보는 444 페이지의 『메시지의 데이터 무결성』의 내용을 참조하십시오.

채널 정의의 CipherSpec이 392 페이지의 『CipherSpec 사용 가능』의 표에 설명된 해시 알고리즘을 사용하는 경우 TLS를 사용하면 데이터 무결성을 확보할 수 있습니다.

특히 데이터 무결성이 중요한 경우 해시 알고리즘이 "없음"으로 표시된 CipherSpec을 선택하지 마십시오. MD5는 매우 오래되었으며 실제로도 사용 시 더 이상 안전하지 않기 때문에 사용하지 않도록 권장됩니다.

## CipherSpecs 및 CipherSuites

암호화 보안 프로토콜은 안전 연결에서 사용되는 알고리즘에 동의해야 합니다. CipherSpec 및 CipherSuite는 특정 알고리즘 결합을 정의합니다.

CipherSpec은 암호화 알고리즘과 메시지 인증 코드(MAC) 알고리즘의 결합을 식별합니다. TLS 연결의 양 측이 서로 통신하려면 동일한 CipherSpec에 동의해야 합니다.

IBM MQ는 TLS 1.3 및 TLS 1.2 프로토콜과 CipherSpec을 지원합니다. 하지만 이를 수행해야 하는 경우 더 이상 사용되지 않는 CipherSpec을 사용으로 설정할 수 있습니다.

다음에 대한 정보는 392 페이지의 『CipherSpec 사용 가능』의 내용을 참조하십시오.

- IBM MQ에서 지원되는 CipherSpec
- 더 이상 사용되지 않는 SSL 3.0 및 TLS 1.0 CipherSpec을 사용으로 설정하는 방법

**중요사항:** IBM MQ 채널 처리 시 CipherSpec을 사용합니다. Java 채널, JMS 채널 또는 MQTT 채널 처리 시 CipherSuite를 지정합니다.

CipherSpec에 대한 자세한 정보는 392 페이지의 『CipherSpec 사용 가능』의 내용을 참조하십시오.

CipherSuite는 TLS 연결에 사용되는 암호화 알고리즘 스위트입니다. 스위트는 세 개의 개별 알고리즘으로 구성됩니다.

- 데이터 교환 시에 사용되는 키 교환 및 인증 알고리즘
- 데이터 암호화 시에 사용되는 암호화 알고리즘
- 메시지 요약 생성 시에 사용되는 MAC(메시지 인증 코드) 알고리즘

스위트의 각 컴포넌트에 대해서는 몇 개의 옵션이 있지만 TLS 연결에 대해 지정되는 경우 특정 결합만 유효합니다. 유효한 CipherSuite의 이름은 사용되는 알고리즘의 결합을 정의합니다. 예를 들어, CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA는 다음을 지정합니다.

- RSA 키 교환과 인증 알고리즘
- 128비트 키 및 CBC(Cipher Block Chaining) 모드를 사용하는 AES 암호화 알고리즘
- SHA-1 메시지 인증 코드(MAC)

## SSL/TLS의 디지털 서명

디지털 서명은 메시지의 표현을 암호화하여 형성됩니다. 암호화는 서명인의 개인 키를 사용하고, 효율성을 위해 보통 메시지 자체가 아니라 메시지 요약에 대해 작동합니다.

디지털 서명은 손으로 쓴 서명과는 달리, 서명된 데이터에 따라 달라지며 이는 서명된 문서의 컨텐츠에 따라 다르지 않습니다. 두 개의 다른 메시지가 같은 엔티티에 의해 디지털로 서명되면 두 서명은 서로 다르지만 둘 다 같은 공개 키, 즉, 이 메시지에 서명한 엔티티의 공개 키를 사용하여 확인됩니다.

디지털 서명 프로세스의 단계는 다음과 같습니다.

1. 송신자가 메시지 요약을 처리한 후, 송신자의 개인 키를 사용하여 요약을 암호화하며, 이것이 디지털 서명을 구성합니다.
2. 송신자가 메시지와 같이 디지털 서명을 전송합니다.
3. 수신자가 송신자의 공개 키를 사용하여 디지털 서명을 복호화하며 이는 송신자의 메시지 요약을 다시 생성합니다.
4. 수신자가 수신된 메시지 데이터로부터 메시지 요약을 처리하고 두 개의 요약이 같은 것인지를 확인합니다.

21 페이지의 그림 6에서 이 프로세스를 설명합니다.

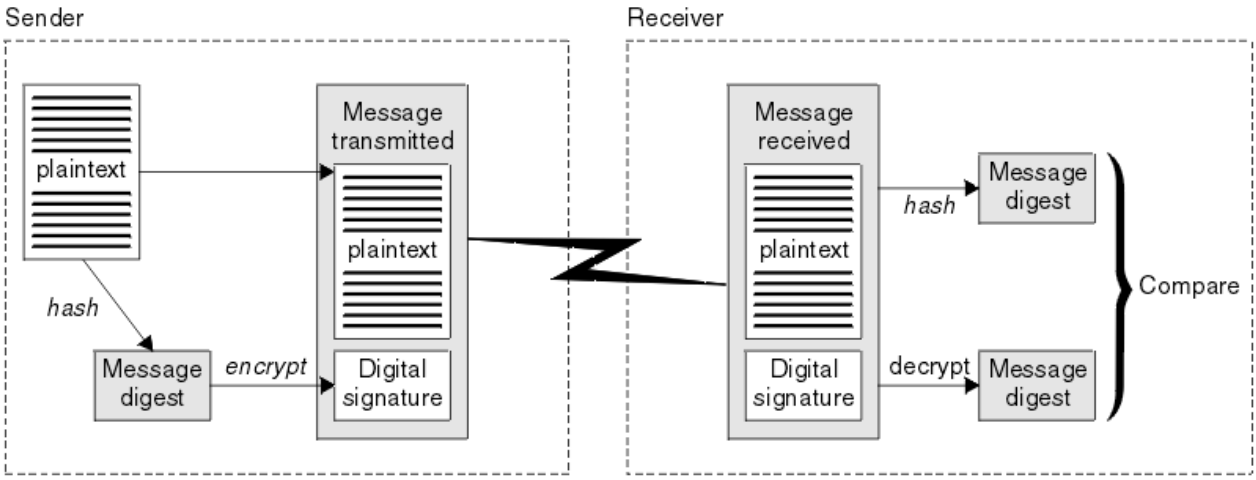



그림 *6.* 디지털 서명 프로세스

디지털 서명이 확인되면, 수신자가 다음을 알게 됩니다.

- 메시지가 전송 중에 수정되지 않았습니다.
- 메시지를 송신하도록 요청한 엔티티에 의해 메시지가 송신되었습니다.

디지털 서명은 무결성 및 인증 서비스의 일부입니다. 디지털 서명도 원본 증명을 제공합니다. 송신자만 개인 키를 알고 있으며 이는 송신자가 메시지의 진원지임을 증명하는 강력한 증거를 제공합니다.

**참고:** 메시지 자체를 암호화할 수도 있으며, 이는 메시지 안의 정보의 기밀성을 보호합니다.

## FIPS(Federal Information Processing Standard)

미국 정부에서는 데이터 암호화 같은 IT 시스템과 보안에 대한 기술적 자문을 만들고 있습니다. NIST(National Institute for Standards and Technology)는 IT 시스템 및 보안과 관련된 중요한 조직입니다. NIST는 FIPS(Federal Information Processing Standard)를 포함하는 표준 및 권장사항을 생성합니다.

이들 중 중요한 표준은 FIPS 140-2이며 강력한 암호화 알고리즘을 사용해야 합니다. FIPS 140-2는 전송 중에 수정되지 못하도록 패킷을 보호하는 데 사용될 해싱 알고리즘에 대한 요구사항도 지정합니다.

**참고:** AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

IBM MQ는 이를 수행하도록 구성된 경우 FIPS 140-2 지원을 제공합니다.

시간이 경과함에 따라 분석가들이 기존 암호화 및 해싱 알고리즘에 대한 공격을 개발했습니다. 이러한 공격에 저항하기 위해 새 알고리즘이 채택되었습니다. FIPS 140-2는 이러한 변경사항을 감안하도록 주기적으로 업데이트됩니다.

**관련 개념**

22 페이지의 『NSA(National Security Agency) 스위트 B 암호화』
미 정부는 데이터 암호화를 포함한 IT 시스템 및 보안에 대한 기술 자문을 생성합니다. US NSA(National Security Agency)에서는 해당 스위트 B 표준에서 상호 운용 가능한 암호화 알고리즘 세트를 권장합니다.

## NSA(National Security Agency) 스위트 B 암호화

미 정부는 데이터 암호화를 포함한 IT 시스템 및 보안에 대한 기술 자문을 생성합니다. US NSA(National Security Agency)에서는 해당 스위트 B 표준에서 상호 운용 가능한 암호화 알고리즘 세트를 권장합니다.

스위트 B 표준은 특정 안전 암호화 알고리즘 세트만 사용되도록 조작 모드를 지정합니다. 스위트 B 표준은 다음을 지정합니다.

- 암호화 알고리즘(AES)
- 키 교환 알고리즘(ECDH라고도 하는 Elliptic Curve Diffie-Hellman)
- 디지털 서명 알고리즘(ECDSA라고도 하는 Elliptic Curve Digital Signature Algorithm)
- 해시 알고리즘(SHA-256 또는 SHA-384)

또한, IETF RFC 6460 표준은 스위트 B 표준 준수를 위해 필요한 자세한 애플리케이션 구성 및 동작을 정의하는 스위트 B 준수 프로파일을 지정합니다. 이는 다음 두 프로파일을 정의합니다.

1. TLS 1.2에 사용할 스위트 B 준수 프로파일. 스위트 B 준수 조작을 위해 구성하는 경우 나열된 제한된 암호화 알고리즘 세트만 사용됩니다.
2. TLS 1.0 또는 TLS 1.1에 사용할 전이 프로파일. 이 프로파일을 사용하면 비스위트 B 준수 서버와의 상호 운용이 가능합니다. 스위트 B 전이 조작을 위해 구성하는 경우 추가 암호화 및 해싱 알고리즘이 사용될 수도 있습니다.

스위트 B 표준은 보증된 보안 레벨을 제공하기 위해서 사용 가능한 암호화 알고리즘 세트를 제한하기 때문에 개념상 FIPS 140-2와 비슷합니다.

AIX, Linux, and Windows 시스템에서 IBM MQ은(는) Suite B 호환 TLS 1.2 프로파일을 준수하도록 구성할 수 있지만, Suite B 전환 프로파일은 지원하지 않습니다. 추가 정보에 대해서는 39 페이지의 『IBM MQ에서의 NSA 스위트 B Cryptography』의 내용을 참조하십시오.

**관련 참조**

21 페이지의 『FIPS(Federal Information Processing Standard)』
미국 정부에서는 데이터 암호화 같은 IT 시스템과 보안에 대한 기술적 자문을 만들고 있습니다. NIST(National Institute for Standards and Technology)는 IT 시스템 및 보안과 관련된 중요한 조직입니다. NIST는 FIPS(Federal Information Processing Standard)를 포함하는 표준 및 권장사항을 생성합니다.

# IBM MQ 보안 메커니즘

이 주제 콜렉션에서는 다양한 보안 개념을 구현하는 IBM MQ 의 특정 메커니즘에 대해 설명합니다.

## IBM MQ의 TLS 보안 프로토콜

IBM MQ는 메시지 채널 및 MQI 채널에 대한 링크 레벨 보안을 제공하기 위해 TLS(Transport Layer Security) 프로토콜을 지원합니다.

메시지 채널 및 MQI 채널은 TLS 프로토콜을 사용하여 링크 레벨 보안을 제공합니다. 호출자 MCA는 TLS 클라이언트이며 응답자 MCA는 TLS 서버입니다.

IBM MQ에서는 TLS 프로토콜의 버전 1.2 및 1.3을 지원합니다. 이전 버전의 TLS 및 SSL은 기본적으로 사용되지 않지만 필요한 경우에는 사용 가능합니다. 채널 정의 중에 CipherSpec를 제공하여 TLS 프로토콜에서 사용되는 암호화 알고리즘을 지정할 수 있습니다.

IBM MQ 및 406 페이지의 『더 이상 사용되지 않는 CipherSpec』 에서 지원하는 CipherSpecs 목록은 392 페이지의 『CipherSpec 사용 가능』 의 내용을 참조하십시오.

SECPROT 및 SSLCIPH 매개변수를 사용하여 채널에서 사용 중인 보안 프로토콜 및 CipherSpec을 표시할 수 있습니다.

메시지 채널의 양 끝과 MQI 채널의 서버 쪽에서, MCA는 연결되는 큐 관리자 대신 작동합니다. TLS 데이터 교환 중에 MCA는 큐 관리자의 디지털 인증서를 채널의 다른 끝에 있는 해당 파트너 MCA로 송신합니다. MQI 채널의 클라이언트 끝에 있는 IBM MQ 코드는 IBM MQ 클라이언트 애플리케이션의 사용자 대신 작동합니다. TLS 데이터 교환 중에 IBM MQ 코드는 사용자의 디지털 인증서를 MQI 채널의 서버 측에 있는 MCA로 송신합니다.

큐 관리자 및 IBM MQ 클라이언트 사용자는 SSLCAUTH(REQUIRED)가 채널의 서버 측에 지정된 경우를 제외하고는 TLS 클라이언트로 작동 중인 경우 연관된 개인용 디지털 인증서가 없어도 됩니다.

디지털 인증서는 키 저장소에 저장됩니다. 큐 관리자 속성 **SSLKeyRepository**는 큐 관리자 디지털 인증서를 보유하는 키 저장소 위치를 지정합니다. IBM MQ 클라이언트 시스템에서 MQSSLKEYR 환경 변수는 사용자의 디지털 인증서를 보유하는 키 저장소 위치를 지정합니다. 또는 IBM MQ 클라이언트 애플리케이션은 MQCONNX 호출에서 TLS 구성 옵션 구조 MQSCO의 **KeyRepository** 필드에 해당 위치를 지정할 수 있습니다. 키 저장소 및 해당 위치 지정 방법에 대한 자세한 정보는 관련 주제를 참조하십시오.

## TLS에 대한 지원

IBM MQ에서는 모든 플랫폼에서 TLS 1.2 및 TLS 1.3에 대한 지원을 제공합니다. TLS 프로토콜에 대한 자세한 정보는 하위 주제의 정보를 참조하십시오.

**Java 및 JMS 클라이언트**
이 클라이언트는 JVM을 사용하여 TLS 지원을 제공합니다.

**AIX, Linux, and Windows**
TLS 지원은 IBM MQ와 같이 설치됩니다.

**IBM i**
TLS 지원은 IBM i 운영 체제에 통합되어 있습니다.

**z/OS**
TLS 지원은 z/OS 운영 체제에 통합되어 있습니다. z/OS의 TLS 지원은 시스템 *SSL*이라고 합니다.

IBM MQ TLS 지원에 대한 모든 필수조건 정보는 IBM MQ 의 시스템 요구사항의 내용을 참조하십시오.

**관련 개념**
17 페이지의 『암호화 보안 프로토콜: TLS』
암호화 프로토콜은 두 당사자가 개인정보 보호 및 데이터 무결성으로 통신하도록 하여 안전한 연결을 제공합니다. TLS(Transport Layer Security) 프로토콜은 SSL(Secure Socket Layer) 프로토콜에서 발전한 것입니다. IBM MQ는 TLS를 지원합니다.

## *SSL/TLS* 키 저장소

상호 인증된 TLS 연결에는 각 연결 측에 키 저장소가 필요합니다. 키 저장소는 디지털 인증서와 개인 키를 포함합니다.

이 정보는 일반 용어 키 저장소를 사용하여 디지털 인증서 및 해당 관련 개인 키에 대한 저장소를 설명합니다. 키 저장소는 TLS를 지원하는 다른 플랫폼 및 환경에서 다른 이름으로 참조됩니다.

- IBM i: 인증서 저장소
- Java 및 JMS: 키 저장소 및 신뢰 저장소
- ALW AIX, Linux, and Windows: 키 데이터베이스 파일
- z/OS: 키 링

자세한 정보는 12 페이지의 『디지털 인증서』 및 17 페이지의 『TLS(Transport Layer Security) 개념』의 내용을 참조하십시오.

상호 인증된 TLS 연결에는 각 연결 측에 키 저장소가 필요합니다. 키 저장소는 다음 인증서와 요청을 포함할 수 있습니다.

- 큐 관리자 또는 클라이언트가 연결의 원격 끝에 있는 해당 파트너에서 수신한 인증서의 확인을 허용하는 다양한 인증 기관에서 발행된 몇 개의 CA 인증서. 개별 인증서가 인증서 체인에 있을 수 있습니다.
- 인증 기관에서 수신된 하나 이상의 개인 인증서. 별도의 개인 인증서를 각 큐 관리자 또는 IBM MQ MQI client에 연관시킵니다. 개인 인증서는 상호 인증이 필요한 경우 TLS 클라이언트에서 필수입니다. 상호 인증이 필요하지 않은 경우, 개인 인증서가 클라이언트에 필요하지 않습니다. 키 저장소는 각 개인 인증서에 해당하는 개인 키를 포함할 수도 있습니다.
- 신뢰 가능한 CA 인증서의 서명을 대기 중인 인증서 요청.

키 저장소 보호에 대한 자세한 정보는 24 페이지의 『IBM MQ 키 저장소 보호』의 내용을 참조하십시오.

키 저장소의 위치는 사용하고 있는 플랫폼에 따라 다릅니다.

IBM i **IBM i**

키 저장소는 인증서 저장소입니다. 기본 시스템 인증서 저장소는 통합 파일 시스템(IFS)의 `/QIBM/UserData/ICSS/Cert/Server/Default`에 있습니다. IBM MQ는 인증서 저장소의 비밀번호는 비밀번호 스태쉬 파일에 저장합니다. 예를 들어, 큐 관리자 `QM1` 의 스태쉬 파일은 `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`입니다.

또는, IBM i 시스템 인증서 저장소가 대신 사용되도록 지정할 수도 있습니다. 이를 수행하려면 큐 관리자 **SSLKEYR** 속성 값을 `*SYSTEM`으로 변경하십시오. 이 값은 큐 관리자가 시스템 인증서 저장소를 사용하고 큐 관리자가 디지털 인증 관리자(DCM)를 애플리케이션으로 사용하도록 등록되는 것을 나타냅니다.

인증서 저장소는 큐 관리자에 대한 개인 키도 포함합니다.

ALW **AIX, Linux, and Windows 시스템**

키 저장소는 키 데이터베이스 파일입니다. 예를 들어, AIX and Linux에서 큐 관리자 `QM1`의 기본 키 데이터베이스 파일은 `/var/mqm/qmgrs/QM1/ssl/key.kdb`입니다. IBM MQ가 기본 위치에 설치되었다면 Windows에서 해당 경로는 `씨:\ProgramData \IBM \MQ\Qmgrs\QM1\ssl\key.kdb`입니다.

키 데이터베이스 파일 IBM MQ 에 액세스하려면 키 데이터베이스의 비밀번호를 제공해야 합니다. 이는 직접 또는 비밀번호 스태쉬 파일을 통해 수행할 수 있습니다. 비밀번호 스태쉬 파일이 사용되는 경우, 이 파일은 키 데이터베이스와 동일한 디렉토리에 있어야 하고 파일 스템이 동일해야 하며 접미부 `.sth`로 끝나야 합니다 (예: `/var/mqm/qmgrs/QM1/ssl/key.sth`).

**참고:** PKCS #11 암호화 하드웨어 카드는 인증서 및 키를 포함해야 하며 그렇지 않으면 키 데이터베이스 파일에 보유됩니다. 인증서와 키가 PKCS #11 카드에 보유되는 경우, IBM MQ는 여전히 키 데이터베이스 파일 및 비밀번호 스태쉬 파일 모두에 액세스해야 합니다.

AIX, Linux, and Windows 시스템에서 키 데이터베이스에는 큐 관리자나 IBM MQ MQI client과(와) 연관된 개인 인증서의 개인 키도 포함합니다.

z/OS **z/OS**

인증서는 z/OS의 키링에 보유됩니다.

기타 외부 보안 관리자(ESM)도 인증서 저장에 키링을 사용합니다.

개인 키는 RACF로 관리됩니다.

*IBM MQ* 키 저장소 보호

IBM MQ에 대한 키 저장소는 파일입니다. 의도된 사용자만 키 저장소 파일에 액세스할 수 있어야 합니다. 이렇게 하면 침입자나 다른 권한이 없는 사용자가 키 저장소 파일을 다른 시스템에 복사한 후, 그 시스템에서 동일한 사용자 ID를 설정하여 의도된 사용자로 위장하지 못하게 합니다.

파일의 권한은 사용자의 umask 및 사용하는 도구에 따라 다릅니다. Windows에서 IBM MQ 계정에는 `BypassTraverseChecking` 권한이 필요하며 이는 파일 경로의 폴더 권한이 아무런 영향을 주지 않음을 나타냅니다.

키 저장소 파일의 파일 권한을 확인하고 파일 및 해당 폴더에 대해 모두가 읽을 수 없도록 해야 하며 가능하면 그룹에서도 읽을 수 없어야 합니다.

사용 중인 시스템에서 키 저장소는 읽기 전용으로 하는 것이 바람직하며 관리자만 유지보수를 수행하기 위해 쓰기 조작을 사용하도록 권한 부여되는 것이 좋습니다.

실제로 모든 위치에서 비밀번호로 보호되는지 여부에 상관없이 모든 키 저장소(keystore)를 보호해야 하며 키 저장소(key repository)도 보호해야 합니다.

디지털 인증서 레이블, 요구사항 이해
디지털 인증서를 사용하기 위해 TLS를 설정할 때 사용된 플랫폼과 연결에 사용한 방법에 따라 반드시 따라야 하는 특정 레이블 요구사항이 있을 수 있습니다.

## 인증서 레이블의 개념

인증서 레이블은 키 저장소에 저장된 디지털 인증서를 나타내는 고유 ID이고, 키 관리 기능을 수행할 때 특정 인증서를 가리키는 데 사용되는 편리한 사람이 읽을 수 있는 이름을 제공합니다. 인증서를 키 저장소에 처음으로 추가할 때 인증서 레이블을 지정합니다.

인증서 레이블은 인증서의 **Subject Distinguished Name** 또는 **Subject Common Name** 필드와는 별개입니다. **Subject Distinguished Name** 및 **Subject Common Name**이 인증서 자체 내의 필드임을 유의하십시오. 이들은 인증서가 작성될 때 정의되고 변경될 수 없습니다. 그러나 필요한 경우 디지털 인증서와 연관된 레이블을 변경할 수 있습니다.

## 인증서 레이블 구문

인증서 레이블은 다음 조건과 함께 문자, 숫자 및 구두점을 포함할 수 있습니다.

- Multi 인증서 레이블은 최대 64자까지 포함할 수 있습니다.
- z/OS 인증서 레이블은 최대 32자까지 포함할 수 있습니다.
- 인증서 레이블은 공백을 포함할 수 있습니다.
- 레이블은 대소문자를 구분합니다.
- EBCDIC 가타카나를 사용하는 시스템에서는 소문자를 사용할 수 없습니다.

인증서 레이블 값의 추가 요구사항은 다음 섹션에 지정됩니다.

## 인증서 레이블 사용 방법

IBM MQ는 TLS 데이터 교환 동안에 전송되는 개인 인증서를 찾기 위해 인증서 레이블을 사용합니다. 이는 키 저장소에 있는 둘 이상의 개인 인증서가 있을 때 모호성을 없애줍니다.

인증서 레이블을 선택한 값으로 설정할 수 있습니다. 값을 설정하지 않으면 사용 중인 플랫폼에 따라 이름 지정 규칙을 따르는 기본 레이블이 사용됩니다. 자세한 내용은 특정 플랫폼에 대해 다음에 나오는 절을 참조하십시오.

**참고:**

1. Java 또는 JMS 시스템에서 인증서 레이블을 직접 설정할 수 없습니다.
2. 채널 자동 정의(CHAD) 엑시트에 의해 작성된 자동 정의된 채널은 인증서 레이블을 설정할 수 없습니다. TLS 데이터 교환은 채널이 작성되는 시간에 발생하기 때문입니다. 인바운드 채널의 경우 CHAD 엑시트에 인증서 레이블을 설정하는 것은 아무런 효과가 없습니다.

이 컨텍스트에서 TLS 클라이언트는 데이터 교환을 시작하는 연결 파트너를 가리키며 이는 IBM MQ 클라이언트 또는 또 다른 큐 관리자일 수 있습니다.

TLS 데이터 교환 중에 TLS 클라이언트는 항상 서버로부터 디지털 인증서를 확보 및 유효성을 검증합니다. IBM MQ 구현을 사용하여 TLS 서버는 항상 클라이언트로부터 인증서를 요청하고 클라이언트는 항상 서버에 인증서를 제공합니다(찾은 경우). 클라이언트가 개인 인증서를 찾을 수 없으면, 클라이언트는 서버에 no certificate 응답을 보냅니다.

TLS 서버는 항상 클라이언트 인증서가 송신되면 클라이언트 인증서의 유효성을 검증합니다. 클라이언트가 인증서를 송신하지 않은 경우에는 TLS 서버의 역할을 하고 있는 채널의 끝이 *REQUIRED*로 설정된 **SSLCAUTH** 매개변수 또는 **SSLPEER** 매개변수 값 세트로 정의된 경우 인증에 실패합니다.

인바운드 채널(수신자, 요청자, 클러스터 수신자, 자격이 없는 서버, 서버 연결 채널)은 리모트 피어의 IBM MQ 버전에서 인증서 레이블 구성을 완전하게 지원하고 채널에서 TLS CipherSpec을 사용 중인 경우에만 구성된 인증서를 송신합니다.

규정되지 않은 서버 채널은 CONNAME 필드가 설정되지 않은 채널입니다.

다른 모든 경우, 큐 관리자 **CERTLABL** 매개변수는 전송된 인증서를 판별합니다. 특히 다음은 채널 특정 레이블 설정에 관계없이 큐 관리자의 **CERTLABL** 매개변수에 의해 구성된 인증서만 수신합니다.

- SNI(Server Name Indication), 즉 채널별 인증서를 지원하는 Java 및 JMS 클라이언트
- IBM MQ prior to IBM MQ 8.0의 버전.
- 관리 .NET 클라이언트

또한 채널이 사용하는 인증서는 채널 CipherSpec에 적합해야 합니다. 자세한 정보는 43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』의 내용을 참조하십시오.

IBM MQ 8.0 이상에서는 채널 정의에서 **CERTLABL** 속성으로 지정된 채널별 인증서 레이블을 사용하여 동일한 큐 관리자에서 다중 인증서의 사용을 지원합니다. 큐 관리자의 인바운드 채널(예: 서버 연결 또는 수신자)은 큐 관리자로부터 올바른 인증서를 제공하기 위해 TLS 서버 이름 표시(SNI)를 사용하여 채널 이름 감지에 의존합니다. 큐 관리자에서 다중 인증서를 사용하는 방법에 대한 자세한 정보는 27 페이지의 『IBM MQ에서 다중 인증서 기능을 제공하는 방법』의 내용을 참조하십시오.

채널이 IBM MQ Internet Pass-Thru (MQIPT) 를 통해 목적지 큐 관리자에 연결하고 MQIPT 라우트에 **SSLServer** 및 **SSLClient** 가 모두 설정된 경우, 엔드포인트 간에 두 개의 개별 TLS 세션이 있습니다. MQIPT는 SNI를 채널 이름으로 설정하거나 인바운드 연결에서 수신된 SNI를 통해 라우트로 전달하여 목적지 큐 관리자가 다중 인증서를 사용할 수 있도록 구성할 수 있습니다. 다중 인증서 지원 및 MQIPT에 대한 자세한 정보는 MQIPT를 사용한IBM MQ 다중 인증서 지원을 참조하십시오.

단방향 인증을 사용하여 큐 관리자에 연결하는 방법 즉, TLS 클라이언트가 인증서를 송신하지 않을 경우에 대한 자세한 정보는 단방향 인증을 사용하여 두 큐 관리자 연결을 참조하십시오.

## 멀티플랫폼 시스템

Multi

멀티플랫폼에서 TLS 서버는 인증서를 클라이언트에 보냅니다.

큐 관리자와 클라이언트는 각각 다음 소스에서 비어 있지 않은 값을 순서대로 검색합니다. 첫 번째 비어 있지 않은 값은 인증서 레이블을 판별합니다. 인증서 레이블은 키 저장소에 존재해야 합니다. 레이블의 대소문자와 형식이 정확히 일치하는 인증서가 발견되지 않으면 오류가 발생하고 TLS 데이터 교환이 실패합니다.

**큐 관리자**

1. 채널 인증서 레이블 속성 **CERTLABL**.
2. 큐 관리자 인증서 레이블 속성 **CERTLABL**.
3. 기본적으로 이는 큐 관리자의 이름이 추가된 `ibmwebspheremq` 형식(모두 소문자)입니다. 예를 들어, `QM1` 이름이 지정된 큐 관리자의 경우 기본 인증서 레이블은 `ibmwebspheremqqm1`입니다.

**IBM MQ 클라이언트**

1. CLNTCONN 채널 정의에 있는 인증서 레이블 속성 **CERTLABL**.
2. MQSCO 구조 **CertificateLabel** 속성.
3. 환경 변수 **MQCERTLABL**.
4. 클라이언트 `.ini` 파일(해당 SSL 절에서) **CertificateLabel** 속성
5. 클라이언트 애플리케이션이 실행 중인 사용자 ID가 추가된 `ibmwebspheremq` 형식의 기본값(모두 소문자)입니다. 예를 들어, USER1의 사용자 ID의 경우 기본 인증서 레이블은 `ibmwebspheremquser1`입니다.

## z/OS 시스템

z/OS

IBM MQ 클라이언트는 z/OS에서 지원되지 않습니다. 그러나 z/OS 큐 관리자는 연결을 시작할 때 TLS 클라이언트 또는 연결 요청을 승인할 때 TLS 서버의 역할을 할 수 있습니다. z/OS 큐 관리자의 인증서 레이블 요구사항은 이러한 역할 둘 모두에 적용되고 멀티플랫폼의 요구사항과는 다릅니다.

큐 관리자와 클라이언트는 각각 다음 소스에서 비어 있지 않은 값을 순서대로 검색합니다. 첫 번째 비어 있지 않은 값은 인증서 레이블을 판별합니다. 인증서 레이블은 키 저장소에 존재해야 합니다. 레이블의 대소문자와 형식이 정확히 일치하는 인증서가 발견되지 않으면 오류가 발생하고 TLS 데이터 교환이 실패합니다.

1. 채널 인증서 레이블 속성, **CERTLABL**.
2. 공유된 경우, 큐 공유 그룹 인증서 레이블 속성, **CERTQSGL**.

   공유되지 않은 경우, 큐 관리자 인증서 레이블 속성, **CERTLABL**.
3. 큐 관리자 또는 큐 공유 그룹의 이름이 추가된 `ibmWebSphereMQ` 형식의 기본값입니다. 이 문자열이 대소문자를 구분하고 표시된 대로 기록되어야 함을 유의하십시오. 예를 들어, `QM1` 이름이 지정된 큐 관리자의 경우 기본 인증서 레이블은 `ibmWebSphereMQQM1`입니다.
4. 옵션 27 페이지의 『3』에 있는 형식의 인증서가 없는 경우 IBM MQ는 키 링에서 기본값으로 표시된 인증서를 사용하려고 시도합니다.

키 저장소를 표시하는 방법에 대한 정보는 290 페이지의 『Locating the key repository for a queue manager on z/OS』의 내용을 참조하십시오.

## IBM MQ Java 및 IBM MQ JMS 클라이언트

IBM MQ Java 및 IBM MQ JMS 클라이언트는 TLS 데이터 교환 동안에 개인 인증서를 선택하기 위해 JSSE(Java Secure Socket Extension) 제공자의 기능을 사용하므로 인증서 레이블 요구사항을 따르지 않습니다.

기본 작동은 JSSE 클라이언트가 키 저장소의 인증서를 통해 반복하여 첫 번째 발견된 승인 가능한 개인 인증서를 선택하는 것입니다. 그러나 이 작동은 단지 기본값이며 JSSE 제공자의 구현에 의존합니다.

또한 JSSE 인터페이스는 애플리케이션에 의한 런타임 시에 구성과 직접 액세스를 통해 고도로 사용자 정의 가능합니다. 특정 세부사항은 JSSE 제공자가 제공한 문서를 참조하십시오.

문제점 해결을 위해서 또는 특정 JSSE 제공자와 결합하여 IBM MQ Java 클라이언트 애플리케이션이 수행하는 데이터 교환을 보다 잘 이해하기 위해서 JVM 환경에서 `javax.net.debug=ssl`을 설정하여 디버깅을 사용 가능하게 할 수 있습니다.

구성을 통해서 또는 명령행에 `-Djavax.net.debug=ssl`을 입력하여 애플리케이션 내에 변수를 설정할 수 있습니다.

Linux *IBM MQ*에서 다중 인증서 기능을 제공하는 방법

서버 이름 표시(SNI)는 클라이언트가 필요로 하는 서비스를 표시할 수 있게 해 주는 TLS 프로토콜의 확장입니다. IBM MQ 용어에서 이는 채널과 동등합니다.

SNI 확장은 채널 정의에서 CERTLABL 매개변수를 사용하여 여러 채널에서 여러 인증서를 지정할 수 있도록 IBM MQ 에서 사용됩니다.

IBM MQ에서 사용하는 SNI 주소는 요청되는 채널 이름을 기반으로 하며, 그 뒤에는 접미부 `.chl.mq.ibm.com`이 옵니다.

IBM MQ 채널 이름은 올바른 SNI 이름이 되기 위해 다음과 같이 맵핑됩니다.

- 대문자 A - Z는 소문자로 폴드됨
- 숫자 0 - 9는 변경되지 않음
- 소문자 a - z를 포함한 다른 모든 문자는 뒤에 하이픈이 오는 두 자리 16진수 ASCII 문자 코드(소문자)로 변환됩니다.
  - 소문자 a - z는 각각 16진 코드 61- - 7a-에 맵핑됨
  - 퍼센트(%)는 16진 코드 25-에 맵핑됨

- 하이픈(-)은 16진 코드 2d-에 맵핑됨
- 점(.)은 16진 코드 2e-에 맵핑됨
- 슬래시(/)는 16진 코드 2f-에 맵핑됨
- 밑줄(_)은 16진 코드 5f-에 맵핑됨

EBCDIC 플랫폼에서, 채널 이름은 이 맵핑이 적용되기 전에 ASCII로 변환됩니다.

예를 들면, 채널 이름 TO.QMGR1은 SNI 주소 to2e-qmgr1.chl.mq.ibm.com에 맵핑됩니다.

반면 소문자 채널 이름 to.qmgr1은 SNI 주소 74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com에 맵핑됩니다.

**참고:** 생성된 SNI URL이 URL 형식 스펙을 준수해야 하는 환경에서(예를 들어, 클라이언트가 Red Hat® OpenShift® Route를 통해 Red Hat OpenShift에서 실행 중인 큐 관리자에 연결 중인 경우) 채널 이름은 소문자로 끝나지 않아야 합니다.

SSL 스탠자의 **OutboundSNI** 특성을 사용하여 TLS 연결을 시작할 때 또는 호스트 이름에 SNI를 리모트 시스템에 대상 IBM MQ 채널 이름으로 설정해야 하는지 여부를 선택할 수 있습니다. **OutboundSNI** 특성에 대한 자세한 정보는 qm.ini 파일의 SSL 스탠자 및 클라이언트 구성 파일의 SSL 스탠자의 내용을 참조하십시오.

여러 인증서를 사용하려면 SNI가 IBM MQ 채널 이름으로 설정되어 있어야 합니다. 인증서 레이블이 구성된 IBM MQ 채널에 연결하는 데 호스트 이름, 사용자 정의 또는 SNI가 사용되지 않는 경우 연결 애플리케이션은 MQRC_SSL_INITIALIZATION_ERROR와 함께 거부되고 AMQ9673 메시지가 리모트 큐 관리자 오류 로그에 인쇄됩니다.

채널이 IBM MQ Internet Pass-Thru(MQIPT)를 통해 목적지 큐 관리자에 연결하는 경우 다중 인증서가 목적지 큐 관리자에서 사용되도록 하려면 SNI를 채널 이름으로 설정하거나 라우트에 대한 인바운드 연결에서 수신된 SNI를 통과하도록 MQIPT를 구성해야 합니다. 다중 인증서 지원 및 MQIPT에 대한 자세한 정보는 MQIPT를 통한 IBM MQ 다중 인증서 지원을 참조하십시오.

이 특성이 사용되는 방법에 대한 자세한 정보는 Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결을 참조하십시오.

큐 관리자의 키 저장소 새로 고치기
키 저장소의 컨텐츠를 변경할 때 기존 큐 관리자 프로세스는 REFRESH SECURITY TYPE (SSL) 명령이 실행되거나 큐 관리자가 재시작될 때까지 새 컨텐츠를 선택하지 않습니다.

REFRESH SECURITY TYPE(SSL) 명령에 대한 자세한 정보는 REFRESH SECURITY를 참조하십시오.

큐 관리자가 키 저장소의 컨텐츠를 변경한 후 amqrmppa 또는 **runmqchl**를 사용하여 새 채널 프로세스를 작성하는 경우, 기존 프로세스가 키 저장소의 캐시된 사본을 계속 사용하는 동안 새 프로세스가 새 인증서를 즉시 사용하기 시작합니다. 자세한 내용은 287 페이지의 『AIX, Linux, and Windows에서 인증서 또는 키 저장소의 변경사항이 적용되는 시기』의 내용을 참조하십시오.

REFRESH SECURITY TYPE (SSL) 명령을 실행할 때까지 여러 실행 채널이 다른 버전의 키 저장소를 사용할 수 있습니다.

PCF 명령 또는 IBM MQ Explorer를 사용하여 키 저장소를 새로 고칠 수 있습니다. 자세한 정보는 MQCMD_REFRESH_SECURITY 명령 및 이 제품 문서의 IBM MQ Explorer 절에 있는 *TLS* 보안 새로 고치기 주제를 참조하십시오.

**관련 개념**

28 페이지의 『SSL/TLS 키 저장소 컨텐츠 및 SSl/TLS 설정의 클라이언트 보기 새로 고침』
키 저장소에 대해 새로 고친 컨텐츠가 포함된 클라이언트 애플리케이션을 업데이트하려면 클라이언트 애플리케이션을 중지한 후에 다시 시작해야 합니다.

*SSL/TLS* 키 저장소 컨텐츠 및 *SSl/TLS* 설정의 클라이언트 보기 새로 고침
키 저장소에 대해 새로 고친 컨텐츠가 포함된 클라이언트 애플리케이션을 업데이트하려면 클라이언트 애플리케이션을 중지한 후에 다시 시작해야 합니다.

IBM MQ 클라이언트에서 보안을 새로 고칠 수 없습니다. 클라이언트에 대해 REFRESH SECURITY TYPE(SSL) 명령에 해당하는 명령이 없습니다(자세한 정보는 REFRESH SECURITY 참조).

키 저장소의 새로 고쳐진 컨텐츠로 클라이언트 애플리케이션을 업데이트하려면 보안 인증서를 변경할 때마다 애플리케이션을 중지한 후 재시작해야 합니다.

채널을 재시작하면 구성이 새로 고쳐지며 애플리케이션에 다시 연결 논리가 있는 경우 STOP CHL STATUS(INACTIVE) 명령을 발행하여 클라이언트에서 보안을 새로 고칠 수 있습니다.

**관련 개념**

28 페이지의 『큐 관리자의 키 저장소 새로 고치기』

키 저장소의 컨텐츠를 변경할 때 기존 큐 관리자 프로세스는 REFRESH SECURITY TYPE (SSL) 명령이 실행되거나 큐 관리자가 재시작될 때까지 새 컨텐츠를 선택하지 않습니다.

## *MQCSP* 비밀번호 보호

MQCSP 구조에 지정된 인증 신임 정보는 IBM MQ MQCSP 비밀번호 보호 기능을 사용하여 보호하거나 TLS 암호화를 사용하여 암호화할 수 있습니다.

IBM MQ client 애플리케이션은 큐 관리자에 연결할 때 사용자 ID및 비밀번호를 제공할 수 있습니다.

V 9.4.0 IBM MQ 9.4.0부터 애플리케이션은 인증의 대체 방법으로 인증 토큰을 제공할 수도 있습니다. 이러한 신임 정보는 MQCSP 구조로 큐 관리자에 송신됩니다.

채널이 TLS 암호화를 사용 중인 경우 MQCSP의 신임 정보는 TLS 암호 스펙에 따라 암호화됩니다. 채널이 TLS 암호화를 사용하지 않는 경우 IBM MQ 는 네트워크를 통해 전송되기 전에 이러한 신임 정보를 보호하여 일반 텍스트로 네트워크를 통해 신임 정보를 전송하지 않도록 할 수 있습니다. 이러한 신임 정보를 보호하는 IBM MQ 기능을 MQCSP 비밀번호 보호라고 합니다.

MQCSP 비밀번호 보호가 사용되는 경우 MQCSP 구조의 다음 데이터가 보호됩니다.

- `MQCSP.AuthenticationType` 필드가 `MQCSP_AUTH_USER_ID_AND_PW`로 설정된 경우 비밀번호입니다.
- V 9.4.0 `MQCSP.AuthenticationType` 필드가 `MQCSP_AUTH_ID_TOKEN`으로 설정된 경우 인증 토큰입니다.

**중요사항:** MQCSP 비밀번호 보호 사용은 TLS 암호화 설정에 비해 단순하기 때문에 MQCSP 비밀번호 보호는 테스트 및 개발 용도로 유용하지만 TLS 암호화에 비해 안전하지 않습니다. 프로덕션 용도로는 IBM MQ 비밀번호 보호보다 우선하여 TLS 암호화를 사용하십시오. 특히 클라이언트와 큐 관리자 간의 네트워크를 신뢰할 수 없는 경우에는 TLS 암호화가 더 안전하기 때문에 TLS 암호화를 사용하십시오.

사용 중인 암호화 및 제공되는 보호 수준에 대해 걱정하는 경우 전체 TLS 암호화를 사용해야 합니다. TLS를 사용하면 알고리즘이 공개적으로 알려지며 **SSLCIPH** 채널 속성을 사용하여 엔터프라이즈에 적합한 알고리즘을 선택할 수 있습니다.

MQCSP 구조에 대한 자세한 정보는 MQCSP 구조를 참조하십시오.

MQCSP 구조의 신임 정보는 다음 조건이 모두 충족되는 경우 IBM MQ 비밀번호 보호를 사용하여 보호됩니다.

- 연결의 양쪽 끝 모두 IBM MQ 8.0 이상을 사용 중입니다.
- 채널은 TLS 암호화를 사용 중이 아닙니다. 채널에 공백 **SSLCIPH** 속성이 있거나 **SSLCIPH** 속성이 암호화를 제공하지 않는 암호 스펙으로 설정된 경우 채널이 TLS 암호화를 사용하지 않습니다. 예를 들어, NULL_SHA와 같은 널 암호는 암호화를 제공하지 않습니다.
- `MQCSP.AuthenticationType` 필드는 `MQCSP_AUTH_USER_ID_AND_PWD` 또는 `MQCSP_AUTH_ID_TOKEN`으로 설정됩니다. `MQCSP.AuthenticationType` 필드에 대한 자세한 정보는 **AuthenticationType**의 내용을 참조하십시오.
- 클라이언트가 IBM MQ Explorer 이고 사용자 ID 호환 모드가 사용으로 설정되지 않은 경우. 이 모드는 IBM MQ Explorer 에서 사용자 ID및 비밀번호를 전송하는 데 사용되는 기본 모드가 아닙니다. 이 조건은 IBM MQ Explorer에만 적용 가능합니다.

이러한 조건이 충족되지 않으면 신임 정보가 MQCSP 비밀번호 보호로 보호되지 않습니다.
**PasswordProtection** 속성의 값이 신임 정보가 평문으로 전송되는 것을 금지하고 채널이 TLS 암호화를 사용하지 않는 경우 연결이 실패하고 MQRC_PASSWORD_PROTECTION_ERROR (2594) 이유 코드가 리턴됩니다.

## PasswordProtection 구성 설정

클라이언트 및 큐 관리자 구성 파일의 **Channels** 스탠자에 있는 **PasswordProtection** 속성은 신임 정보가 일반 텍스트로 전송되는 것을 방지할 수 있습니다.

**참고:** 이 속성은 TLS 암호화를 사용하지 않는 연결에만 관련됩니다. 연결에서 TLS 암호화를 사용하는 경우 신임 정보는 MQCSP 비밀번호 보호로 보호되는 대신 TLS를 사용하여 암호화됩니다.

속성은 다음 값 중 하나로 설정할 수 있습니다. 기본값은 compatible입니다.

**호환 가능**
큐 관리자 또는 클라이언트가 IBM MQ 8.0이전 버전의 IBM MQ 를 실행 중인 경우 신임 정보가 일반 텍스트로 전송됩니다. 즉, MQCSP 비밀번호 보호를 지원하지 않는 IBM MQ 버전과의 호환성을 위해 일반 텍스트로 네트워크를 통해 신임 정보를 전송할 수 있습니다.

큐 관리자와 클라이언트 둘 다 IBM MQ 8.0 이상에서 IBM MQ 버전을 실행 중인 경우 신임 정보는 MQCSP 비밀번호 보호에 의해 보호됩니다.

큐 관리자와 클라이언트가 모두 IBM MQ 8.0 이상에서 IBM MQ 버전을 실행 중이고 MQCSP.AuthenticationType 필드가 MQCSP_AUTH_USER_ID_AND_PW 또는 MQCSP_AUTH_ID_TOKEN으로 설정되지 않은 경우 신임 정보가 송신되기 전에 연결이 실패합니다.

**always**
신임 정보는 보호되지 않은 네트워크를 통해 전송되지 않아야 합니다.

큐 관리자와 클라이언트 둘 다 IBM MQ 8.0 이상에서 IBM MQ 버전을 실행 중인 경우 신임 정보는 MQCSP 비밀번호 보호에 의해 보호됩니다.

다음과 같은 경우에 신임 정보가 전송되기 전에 연결이 실패합니다.

- MQCSP.AuthenticationType 필드가 MQCSP_AUTH_USER_ID_AND_PW 또는 MQCSP_AUTH_ID_TOKEN으로 설정되지 않습니다.
- 큐 관리자 또는 클라이언트가 IBM MQ 8.0이전의 IBM MQ 버전을 실행 중입니다.

**선택사항**
큐 관리자와 클라이언트 둘 다 IBM MQ 8.0 이상에서 IBM MQ 버전을 실행 중이고 MQCSP.AuthenticationType 필드가 MQCSP_AUTH_USER_ID_AND_PW 또는 MQCSP_AUTH_ID_TOKEN으로 설정된 경우 신임 정보가 MQCSP 비밀번호 보호에 의해 보호됩니다. 그렇지 않으면 신임 정보가 일반 텍스트로 전송됩니다.

**warn**
모든 클라이언트가 일반 텍스트 신임 정보를 전송할 수 있습니다. 평문 신임 정보가 수신되면 경고 메시지 AMQ9297W 가 큐 관리자 오류 로그에 기록됩니다.

이 옵션은 큐 관리자 구성 파일에서만 지정할 수 있습니다.

Java 및 JMS 클라이언트의 경우, **PasswordProtection** 속성의 동작은 클라이언트가 호환 모드를 사용하는지 또는 MQCSP 모드를 사용하는지에 따라 변경됩니다.

- Java 및 JMS 클라이언트가 호환 모드에서 작동 중인 경우 클라이언트가 연결할 때 MQCSP 구조를 사용하여 사용자 ID및 비밀번호를 전송하지 않습니다. 따라서 **PasswordProtection** 속성의 동작은 IBM MQ 8.0이전의 IBM MQ 버전을 실행 중인 클라이언트에 대해 설명된 동작과 동일합니다.
- Java 및 JMS 클라이언트가 MQCSP 모드에서 작동하는 경우 **PasswordProtection** 속성의 작동은 설명된 대로 작동합니다.

Java 및 JMS 클라이언트와의 연결 인증에 대한 자세한 정보는 77 페이지의 『Java 클라이언트와의 연결 인증』 의 내용을 참조하십시오.

## MQCSP 비밀번호 보호 및 MQIPT

V 9.4.0

클라이언트가 IBM MQ Internet Pass-Thru (MQIPT) 를 통해 큐 관리자에 연결하는 경우 MQIPT 라우트는 TLS 암호화를 추가하거나 제거하도록 구성될 수 있습니다. 즉, MQIPT 라우트는 SSLServer=true 및 SSLClient=false또는 SSLServer=true 및 SSLClient=false로 구성될 수 있습니다. 이 상황에서 채널

의 한쪽 끝은 TLS 암호화를 사용하고 다른 끝은 사용하지 않으므로 클라이언트 및 큐 관리자가 비밀번호 보호 알고리즘에 동의하는 데 실패할 수 있습니다. 이로 인해 연결이 이유 코드 MQRC_PASSWORD_PROTECTION_ERROR (2594) 로 실패합니다.

IBM MQ 9.4.0부터 MQIPT 는 TLS 암호화를 추가하거나 제거하는 MQIPT 라우트에 대한 클라이언트와 큐 관리자 간의 호환성을 유지하기 위해 MQCSP 구조에서 신임 정보에 대한 보호를 추가하거나 제거할 수 있습니다. MQIPT 의 MQCSP 비밀번호 보호는 **PasswordProtection** 라우트 특성을 사용하여 구성됩니다.

**PasswordProtection** 특성의 기본값은 required입니다. 이 값은 MQIPT 가 MQCSP 비밀번호 보호를 추가할 수 있지만 제거할 수는 없음을 의미합니다. TLS 암호화를 추가하는 MQIPT 라우트에 대한 연결은 이 값이 **PasswordProtection**인 이유 코드 MQRC_PASSWORD_PROTECTION_ERROR (2594) 로 실패할 수 있습니다. 이 문제를 해결하려면 MQIPT 라우트 구성에서 **PasswordProtection** 특성의 값을 호환  가능 으로 설정하십시오.

MQIPT의 **PasswordProtection** 특성에 대한 자세한 정보는 PasswordProtection의 내용을 참조하십시오.

## 디지털 인증 관리자*(DCM, Digital Certificate Manager)*

DCM을 사용하여 IBM i에서 디지털 인증서 및 개인 키를 관리합니다.

디지털 인증 관리자(DCM)를 사용하여 디지털 인증서를 관리하고 이를 IBM i 서버의 안전한 애플리케이션에서 사용할 수 있습니다. 디지털 인증 관리자를 사용하여 인증 기관(CA) 또는 다른 써드파티에서 디지털 인증서를 요청하고 처리할 수 있습니다. 또한, 로컬 인증 기관으로 작동하여 사용자를 위해 디지털 인증서를 작성하고 관리할 수도 있습니다.

DCM은 더 강력한 인증서 및 애플리케이션 유효성 검증 프로세스를 제공하기 위해 인증서 폐기 목록(CRL) 사용을 지원합니다. DCM을 사용하여 LDAP 서버에서 특정 인증 기관 CRL이 있는 위치를 정의하여 IBM MQ가 특정 인증서가 폐기되지 않았음을 확인할 수 있도록 합니다.

DCM은 다양한 형식의 인증서를 지원하며 이를 자동으로 감지할 수 있습니다. DCM이 PKCS #12 인코드 인증서 또는 암호화된 데이터를 포함하는 PKCS #7 인증서를 감지하면, 자동으로 사용자에게 이 인증서를 암호화할 때 사용한 비밀번호 입력을 프롬프트합니다. DCM은 암호화된 데이터를 포함하지 않는 PKCS #7 인증서에 대해서는 프롬프트를 표시하지 않습니다.

DCM은 애플리케이션 및 사용자에 대한 디지털 인증서를 관리하기 사용할 수 있는 브라우저 기반의 사용자 인터페이스를 제공합니다. 사용자 인터페이스는 탐색 프레임 및 태스크 프레임의 두 기본 프레임으로 나뉩니다.

탐색 프레임을 사용하여 인증서 또는 이를 사용하는 애플리케이션을 관리하는 태스크를 선택합니다. 일부 개별 태스크는 기본 탐색 프레임에 직접 표시되기도 하지만 탐색 프레임에서 대부분의 태스크는 범주로 구성됩니다. 예를 들어, 인증서 관리는 다양한 개별 안내 태스크(예: 인증서 보기, 인증서 갱신, 인증서 가져오기)가 포함되는 태스크 범주입니다. 탐색 프레임의 항목이 둘 이상의 태스크를 포함하는 범주인 경우에는 왼쪽에 화살표가 표시됩니다. 화살표는 범주 링크를 선택하면 펼쳐진 태스크 목록이 표시되어 수행할 태스크를 선택할 수 있음을 나타냅니다.

DCM에 대한 중요한 정보는 다음 IBM Redbooks® 발행물을 참조하십시오.

- *IBM i* 유선 네트워크 보안*: OS/400 V5R1 DCM and Cryptographic Enhancements*, SG24-6168. 특히, IBM i 시스템을 로컬 CA로 설정하는 데 필요한 필수 정보는 부록을 참조하십시오.
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*, SG24-5659. 특히, AS/400 DCM에 대해 설명하는 5장. *Digital Certificate Manager for AS/400*을 참조하십시오.

## *FIPS(Federal Information Processing Standard)*

이 주제에서는 미국 국립 표준 기술국의 FIPS(Federal Information Processing Standard) Cryptomodule Validation Program 및 TLS 채널에서 사용할 수 있는 암호화 기술을 소개합니다.

**참고:** AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

이 정보는 다음 플랫폼에 적용됩니다.

- ALW AIX, Linux, and Windows
- z/OS z/OS

ALW AIX, Linux, and Windows에서 IBM MQ TLS 연결의 FIPS 140-2준수에 대한 자세한 정보는 32 페이지의 『AIX, Linux, and Windows용 FIPS(Federal Information Processing Standard)』의 내용을 참조하십시오.

z/OS z/OS에서 IBM MQ TLS 연결의 FIPS 140-2준수에 대한 자세한 정보는 35 페이지의 『Federal Information Processing Standards (FIPS) for z/OS』의 내용을 참조하십시오.

암호화 하드웨어가 있는 경우 IBM MQ에서 사용되는 암호화 모듈은 하드웨어 생산자가 제공하는 모듈로 구성 가능합니다. 모듈 구성을 완료하면 구성이 해당 암호화 모듈이 FIPS에서 인증된 경우에만 FIPS를 준수합니다.

시간이 흐름에 따라 FIPS(Federal Information Processing Standards)는 암호화 알고리즘 및 프로토콜에 대한 새 공격을 반영할 수 있도록 업데이트되었습니다. 예를 들어, 일부 CipherSpec은 FIPS 인증 CipherSpec이 되기 위해 정지될 수 있습니다. 이런 경우 IBM MQ도 최신 표준 구현으로 업데이트됩니다. 그 결과 유지보수를 적용한 후에는 동작이 변경됨을 알 수 있습니다.

**관련 개념**

254 페이지의 『MQI 클라이언트에서 런타임 시 FIPS 인증 CipherSpec만 사용하도록 지정』
FIPS 준수 소프트웨어를 사용하여 키 저장소를 작성한 다음 채널이 FIPS-인증 CipherSpec을 사용해야 함을 지정하십시오.

**관련 태스크**

IBM MQ classes for Java에서 TLS 사용

IBM MQ classes for JMS에 TLS(Transport Layer Security) 사용

**관련 참조**

JMS 오브젝트의 TLS 특성

505 페이지의 『AIX, Linux, and Windows 의 runmqakm 및 runmqktool 명령』
AIX, Linux, and Windows 시스템에서 **runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 및 인증서를 관리하십시오.

21 페이지의 『FIPS(Federal Information Processing Standard)』
미국 정부에서는 데이터 암호화 같은 IT 시스템과 보안에 대한 기술적 자문을 만들고 있습니다. NIST(National Institute for Standards and Technology)는 IT 시스템 및 보안과 관련된 중요한 조직입니다. NIST는 FIPS(Federal Information Processing Standard)를 포함하는 표준 및 권장사항을 생성합니다.

ALW *AIX, Linux, and Windows용 FIPS(Federal Information Processing Standard)*

AIX, Linux, and Windows 시스템의 SSL/TLS 채널에서 암호화가 필요한 경우 IBM MQ 는 IBM Crypto for C (ICC)라는 암호화 패키지를 사용합니다. AIX, Linux, and Windows 플랫폼에서 ICC 소프트웨어는 미국 국립 표준 기술 연구소 (US National Institute of Standards and Technology) 의 FIPS (Federal Information Processing Standards) Cryptomodule Validation Program 레벨 140-2를 통과했습니다.

**참고:** AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

AIX, Linux, and Windows 시스템에서 IBM MQ TLS 연결의 FIPS 140-2준수는 다음과 같습니다.

- 다음과 같은 조건이 충족되면 모든 IBM MQ 메시지 채널(CLNTCONN 채널 유형 제외)에 대해 연결은 FIPS를 준수합니다.
  - 설치된 IBM Global Security Kit (GSKit) ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2를 준수하는 것으로 인증되었습니다.

  - 큐 관리자의 SSLFIPS 속성이 YES로 설정되었습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
  - 모든 키 저장소에 대한 액세스는 큐 관리자의 **KEYRPWD** 속성이 아닌 스태쉬 파일을 사용하여 제공됩니다.
- 모든 IBM MQ MQI client 애플리케이션에 대해 연결은 GSKit 를 사용하며 다음 조건이 충족되는 경우 FIPS를 준수합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2를 준수하는 것으로 인증되었습니다.
  - MQI 클라이언트에 대한 관련 주제에 설명된 대로 FIPS 인증 암호만 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
  - 모든 키 저장소에 대한 액세스는 키 저장소 비밀번호 메커니즘이 아닌 스태쉬 파일을 사용하여 제공됩니다.
- 다음 조건이 충족되는 경우, 클라이언트 모드를 사용하는 IBM MQ classes for Java 애플리케이션에 대해 연결은 JRE의 TLS 구현을 사용하며 FIPS를 준수합니다.
  - 애플리케이션 실행에 사용되는 Java Runtime Environment는 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS를 준수합니다.
  - Java 클라이언트에 대한 관련 주제에 설명된 대로 FIPS 인증 암호만 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
- 다음 조건이 충족되는 경우, 클라이언트 모드를 사용하는 IBM MQ classes for JMS 애플리케이션에 대해 연결은 JRE의 TLS 구현을 사용하며 FIPS를 준수합니다.
  - 애플리케이션 실행에 사용되는 Java Runtime Environment는 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS를 준수합니다.
  - JMS 클라이언트에 대한 관련 주제에 설명된 대로 FIPS 인증 암호만 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
- 비관리 .NET 클라이언트 애플리케이션의 경우, 연결은 GSKit 를 사용하며 다음 조건이 충족되면 FIPS를 준수합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2를 준수하는 것으로 인증되었습니다.
  - .NET 클라이언트에 대한 관련 주제에 설명된 대로 FIPS 인증 암호만 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
  - 모든 키 저장소에 대한 액세스는 키 저장소 비밀번호 메커니즘이 아닌 스태쉬 파일을 사용하여 제공됩니다.
- 비관리 XMS .NET 클라이언트 애플리케이션의 경우, 연결은 GSKit 를 사용하며 다음 조건이 충족되는 경우 FIPS를 준수합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2를 준수하는 것으로 인증되었습니다.
  - XMS .NET 문서에 설명된 대로 FIPS 인증 암호화만 사용하도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
  - 모든 키 저장소에 대한 액세스는 키 저장소 비밀번호 메커니즘이 아닌 스태쉬 파일을 사용하여 제공됩니다.

각 수정팩 또는 Refresh Pack에 포함되는 Readme 파일에서 언급하는 경우를 제외하고는 모든 지원 플랫폼은 FIPS 140-2 인증되었습니다.

GSKit를 사용하는 TLS 연결의 경우 FIPS 140-2인증 컴포넌트의 이름은 *ICC*입니다. 지정된 플랫폼에서 GSKit FIPS 준수를 판별하는 것은 이 구성요소의 버전입니다. 현재 설치된 ICC 버전을 판별하려면 **dspmqver -p 64 -v** 명령을 실행하십시오.

다음은 ICC와 관련된 **dspmqver -p 64 -v** 출력의 예제 추출입니다.

```
ICC
============
@(#)CompanyName:   IBM Corporation
@(#)LegalTrademarks: IBM
@(#)FileDescription: IBM Crypto for C-language
@(#)FileVersion:   8.0.0.0
@(#)LegalCopyright:   Licensed Materials - Property of IBM
@(#)          ICC
@ (#) (C) Copyright IBM Corp. 2002, 2024.
@ (#) All Rights Reserved. US Government Users
@(#)          Restricted Rights - Use, duplication or disclosure
@(#)          restricted by GSA ADP Schedule Contract with IBM Corp.
@(#)ProductName:   icc_8.0 (GoldCoast Build) 100415
@(#)ProductVersion:  8.0.0.0
@(#)ProductInfo:   10/04/15.03:32:19.10/04/15.18:41:51
@(#)CMVCInfo:
```

GSKit ICC 8 ( GSKit 8에 포함됨) 에 대한 NIST 인증 명령문은 Cryptographic Module Validation Program에서 찾을 수 있습니다.

암호화 하드웨어가 있는 경우 IBM MQ에서 사용되는 암호화 모듈은 하드웨어 생산자가 제공하는 모듈로 구성 가능합니다. 모듈 구성을 완료하면 구성이 해당 암호화 모듈이 FIPS에서 인증된 경우에만 FIPS를 준수합니다.

## FIPS 140-2 준수로 사용되는 경우 3중 DES(3DES) 제한이 강제로 적용됩니다.

IBM MQ가 FIPS 140-2 준수로 작동되도록 구성되면 추가 제한은 3중 DES(3DES) CipherSpec에 관련되도록 강제 적용됩니다. 이런 제한을 사용하여 US NIST SP800-67 권장사항이 준수됩니다.

1. 3중 DES 키의 모든 파트는 고유해야 합니다.
2. NIST SP800-67 정의에 따라 3중 DES 키의 어떤 파트도 Weak, Semi-Weak 또는 Possibly-Weak 키일 수 없습니다.
3. 32GB 이상의 데이터를 보안 키 재설정 수행 전에 연결로 전송할 수 없습니다. 기본적으로 IBM MQ는 비밀 세션 키를 재설정하지 않기 때문에 이 재설정은 구성해야 합니다. 3중 DES CipherSpec 및 FIPS 140-2 준수 사용 시 보안 키 재설정을 수행하지 않으면 최대 바이트 수가 초과된 후 연결은 AMQ9288 오류로 종료됩니다. 보안 키 재설정 구성 방법에 대한 정보는 435 페이지의 『SSL 및 TLS 비밀 키 재설정』의 내용을 참조하십시오.

IBM MQ는 규칙 1 및 2를 이미 준수하는 3중 DES 세션 키를 생성합니다. 그러나 세 번째 제한을 충족하려면 FIPS 140-2 구성에서 3중 DES CipherSpec을 사용할 때 보안 키 재설정을 사용으로 설정해야 합니다. 또는 3중 DES를 사용하지 않을 수 있습니다.

**관련 개념**

254 페이지의 『MQI 클라이언트에서 런타임 시 FIPS 인증 CipherSpec만 사용하도록 지정』
FIPS 준수 소프트웨어를 사용하여 키 저장소를 작성한 다음 채널이 FIPS-인증 CipherSpec을 사용해야 함을 지정하십시오.

**관련 태스크**

IBM MQ classes for Java에서 TLS 사용

IBM MQ classes for JMS에 TLS(Transport Layer Security) 사용

**관련 참조**

JMS 오브젝트의 TLS 특성

505 페이지의 『AIX, Linux, and Windows 의 runmqakm 및 runmqktool 명령』
AIX, Linux, and Windows 시스템에서 **runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 및 인증서를 관리하십시오.

21 페이지의 『FIPS(Federal Information Processing Standard)』
미국 정부에서는 데이터 암호화 같은 IT 시스템과 보안에 대한 기술적 자문을 만들고 있습니다. NIST(National Institute for Standards and Technology)는 IT 시스템 및 보안과 관련된 중요한 조직입니다. NIST는 FIPS(Federal Information Processing Standard)를 포함하는 표준 및 권장사항을 생성합니다.

z/OS *Federal Information Processing Standards (FIPS) for z/OS*

When cryptography is required on an SSL/TLS channel on z/OS , IBM MQ uses a service called System SSL. The objective of System SSL is to provide the capability to execute securely in a mode designed to adhere to the Federal Information Processing Standards (FIPS) Cryptomodule Validation Program of the US National Institute of Standards and Technology, at level 140-2.

When implementing FIPS 140-2 compliant connections with IBM MQ TLS connections there are a number of points to consider:

- To enable IBM MQ message channels for FIPS-compliance, ensure the following conditions are met:
  - System SSL Security Level 3 FMID is installed and configured (see Planning to install IBM MQ ).
  - System SSL modules are validated.
  - The queue manager's SSLFIPS attribute has been set to **YES**.

When executing in FIPS mode, System SSL exploits CP Assist for Cryptographic Function (CPACF) when available. Cryptographic functions performed by ICSF-supported hardware when running in non-FIPS mode continue to be exploited when executing in FIPS mode, with the exception of RSA signature generation which must be performed in software.

*Table 2. Differences between FIPS mode and non-FIPS mode algorithm support.*

| | **Non-FIPS** | | **FIPS** | |
|---|---|---|---|---|
| **Algorithm** | **Key sizes** | **Hardware** | **Key sizes** | **Hardware** |
| RC2 | 40 and 128 | | | |
| RC4 | 40 and 128 | | | |
| DES | 56 | x | | |
| TDES | 168 | x | 168 | x |
| AES | 128 and 256 | x | 128 and 256 | x |
| MD5 | 48 | | | |
| SHA-1 | 160 | x | 160 | x |
| SHA-2 | 224, 256, 384 and 512 | x | 224, 256, 384 and 512 | x |
| RSA | 512-4096 | x | 1024-4096 | x |
| DSA | 512-1024 | | 1024 | |
| DH | 512-2048 | | 2048 | |

In FIPS mode, System SSL can only use certificates that use the algorithms and key sizes shown in Table 1. During X.509 certificate validation if an algorithm that is incompatible with FIPS mode is encountered, then the certificate cannot be used and is treated as not valid.

For IBM MQ classes applications using client mode within WebSphere® Application Server , refer to Federal Information Processing Standard support.

For information on System SSL module configuration, see System SSL Module Verification Setup.

**Related reference**

"FIPS(Federal Information Processing Standard)" on page 21

미국 정부에서는 데이터 암호화 같은 IT 시스템과 보안에 대한 기술적 자문을 만들고 있습니다. NIST(National Institute for Standards and Technology)는 IT 시스템 및 보안과 관련된 중요한 조직입니다. NIST는 FIPS(Federal Information Processing Standard)를 포함하는 표준 및 권장사항을 생성합니다.

## Multi *mqcertck*를 사용한 큐 관리자의 *TLS* 구성 확인

**MQCERTCK** 명령은 큐 관리자의 TLS 구성에서 일반적인 실수를 찾고, 문제점을 해결하기 위한 제안사항을 제공합니다.

### 소개

**mqcertck** 명령은 다음 항목을 확인합니다.

- 큐 관리자 **SSLKEYR** 속성에서 참조되는 큐 관리자의 키 저장소의 존재 및 권한.
- 큐 관리자 **CERTLABL** 속성에서 참조되는 큐 관리자 인증서에 대한 인증서의 존재 및 유효성.
- TLS 사용 채널의 **CERTLABL** 속성에서 참조되는 모든 인증서의 존재 및 유효성.
- 클라이언트 애플리케이션의 키 저장소 및 인증서(인증서가 큐 관리자에 대해 권한 부여되었는지에 대한 확인 포함)

**참고: mqcertck** 명령은 z/OS 또는 IBM i에서 사용할 수 없습니다.

### 사용법

**mqcertck** 명령을 사용하려면 명령행에서 명령 mqcertck를 필수 매개변수, 그리고 자신이 필요로 하는 선택적 매개변수와 함께 실행하십시오.

명령 및 명령이 취하는 매개변수에 대한 설명은 mqcertck를 참조하십시오.

### 예

큐 관리자의 SVRCONN 채널에 연결하는 클라이언트에서 TLS를 연결할 수 있도록 큐 관리자 QM1 설정을 완료했습니다.

다중 인증 기능을 사용 중이므로 큐 관리자 및 채널 모두의 **CERTLABL** 속성에 인증서 레이블이 지정되어 있습니다. 채널을 작성하는 동안 채널의 **CERTLABL** 속성에 오류가 발생하므로 클라이언트가 연결을 시도할 때 큐 관리자가 MQRC_SSL_INITIALIZATION_ERROR의 2393 리턴 코드를 리턴합니다.

사용자가 큐 관리자를 활성화하기 전에 **mqcertck** 명령을 사용하여 큐 관리자의 TLS 구성을 확인합니다.

mqcertck QM1 명령을 실행하고 다음 출력을 수신합니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
+-----------------------------------------------------------
| IBM MQ TLS Configuration Test tool
+-----------------------------------------------------------
| Problem identified:
|  No certificate could be found for the channel
|  MQCERTCK.CHANNEL
|  This tool looked in the Queue Manager's key repository
|  located at: 'C:\MQ Data\qmgrs\QM1\ssl\key.kdb'
|  for a certificate with label 'chacert',
|  which is the certificate specified in the channel's
|  CERTLABL attribute, but was unable to find one.
|
| Possible resolution:
|  A valid certificate with the label chacert
|  needs to be added to the key repository.
|
|  Alternatively, alter the channel definition to remove
|  the CERTLABL value. This can be done by executing the
|  following command in runmqsc:
|      ALTER CHANNEL(<Name>) CHLTYPE(<TYPE>) CERTLABL(' ')
+-----------------------------------------------------------
| mqcertck has ended. See above for any problems found.
| If there are problems then resolve these and run this
| tool again.
|
+-----------------------------------------------------------
```

이 출력은 사용자가 서버 연결 채널 MQCERTCK.CHANNEL의 채널 정의를 확인하도록 지시합니다. 여기서, 작성한 오류가 표시되며 `mqcertck` 명령을 다시 실행하기 전에 오류를 정정하여 문제점을 해결했는지 확인할 수 있습니다.

## 클라이언트 연결 확인

**mqcertck** 명령에는 큐 관리자의 TLS 구성 외에 클라이언트 키 저장소를 확인하는 기능도 있습니다. 이를 수행하려면 **mqcertck**가 큐 관리자를 실행 중인 시스템에서 클라이언트의 키 저장소에 액세스할 수 있어야 합니다.

**mqcertck** 명령을 실행할 때, **-clientkeyr** 매개변수를 클라이언트 키 저장소의 위치(확장자는 제외)와 함께 제공하면 **mqcertck**가 이 키 저장소를 큐 관리자에 대해 확인합니다.

클라이언트가 큐 관리자에 연결하기 위해 사용할 채널을 알고 있으면 **-clientchannel** 플래그를 사용하여 지정할 수 있습니다.

클라이언트가 큐 관리자에 연결하는 데 상호 인증을 사용하고 있는 경우에는 **-clientusername** 또는 **-clientlabel** 매개변수를 사용하여 **mqcertck** 명령이 클라이언트 키 저장소에서 사용할 인증서를 지정할 수 있습니다.

기본 인증서를 사용하고 클라이언트 애플리케이션에 인증서 레이블을 제공하지 않는 경우, **-clientusername** 및 이 애플리케이션을 실행하는 **username** 매개변수를 사용할 수 있습니다.

**mqcertck** 명령의 조작 중에, 이 명령은 인증서 레이블 `ibmwebspheremqXXXX`를 생성하며 여기서 XXXX는 **-clientusername** 매개변수에서 전달된 값입니다.

클라이언트 키 저장소를 완전히 확인하기 위해 **mqcertck** 명령은 IBM Global Security Kit (GSKit)를 사용하여 더미 연결을 작성합니다. 이를 수행하려면 이 명령이 클라이언트 테스트 중에 바인드될 수 있는, 사용 가능한 포트가 있어야 합니다. 사용되는 기본 포트는 5857이지만, 이 포트가 이미 사용 중인 경우에는 클라이언트 테스트 중에 다른 포트를 사용하도록 지정할 수 있습니다.

**참고: mqcertck** 명령은 포트에 바인드되지만 **mqcertck**는 외부 통신을 사용하지 않으며, 모든 테스트는 로컬로 수행됩니다.

### *IBM MQ MQI client*의 *SSL/TLS*

IBM MQ는 클라이언트에서 TLS를 지원합니다. 다양한 방식으로 TLS 사용을 조정할 수 있습니다.

IBM MQ 는 AIX, Linux, and Windows 시스템의 IBM MQ MQI clients 에 대한 TLS 지원을 제공합니다. IBM MQ classes for Java를 사용 중인 경우 IBM MQ classes for Java 사용을 참조하고 IBM MQ classes for JMS를 사용 중인 경우 IBM MQ classes for JMS 사용을 참조하십시오. 이 절의 나머지 부분은 Java 또는 JMS 환경에는 적용되지 않습니다.

IBM MQ 클라이언트 구성 파일의 MQSSLKEYR값을 사용하거나 애플리케이션이 MQCONNX 호출을 작성할 때 IBM MQ MQI client 에 대한 키 저장소를 지정할 수 있습니다. 채널이 TLS를 사용하도록 지정하기 위한 세 가지 옵션이 있습니다.

- 채널 정의 테이블 사용
- MQCONNX 호출에서 SSL 구성 옵션 구조인 MQSCO 사용
- Active Directory 사용(Windows 시스템)

MQSERVER 환경 변수를 사용해서는 채널이 TLS를 사용하도록 지정할 수 없습니다.

TLS가 채널의 다른 쪽에 지정되지 않은 경우 TLS 없이 계속 기존 IBM MQ MQI client 애플리케이션을 실행할 수 있습니다.

클라이언트 시스템에서 TLS 키 저장소의 컨텐츠, TLS 키 저장소의 위치, 인증 정보 또는 암호화 하드웨어 매개변수를 변경하면, 애플리케이션이 큐 관리자에 연결하기 위해 사용 중인 클라이언트 연결 채널에서 이러한 변경사항을 반영하기 위해 모든 TLS 연결을 종료해야 합니다. 모든 연결이 종료되면 TLS 채널을 재시작하십시오. 모든 새 TLS 설정이 사용됩니다. 이 설정은 큐 관리자 시스템에서 REFRESH SECURITY TYPE(SSL) 명령으로 새로 고치기되는 설정과 유사합니다.

IBM MQ MQI client이(가) 암호화 하드웨어가 있는 AIX, Linux, and Windows 시스템에서 실행되면 MQSSLCRYP 환경 변수를 사용하여 해당 하드웨어를 구성합니다. 이 변수는 ALTER QMGR MQSC 명령에서의 SSLCRYP 매개변수와 동일합니다. ALTER QMGR MQSC 명령의 SSLCRYP 매개변수에 대한 설명은 ALTER

QMGR을 참조하십시오. SSLCRYP 매개변수의 GSK_PCS11 버전을 사용하는 경우에는 PKCS #11 토큰 레이블은 전적으로 소문자로 지정되어야 합니다.

TLS 보안 키 재설정 및 FIPS는 IBM MQ MQI clients에서 지원됩니다. 자세한 정보는 435 페이지의 『SSL 및 TLS 비밀 키 재설정』 및 32 페이지의 『AIX, Linux, and Windows용 FIPS(Federal Information Processing Standard)』의 내용을 참조하십시오.

IBM MQ MQI clients의 TLS 지원에 대한 자세한 정보는 253 페이지의 『IBM MQ MQI client 보안 설정』 의 내용을 참조하십시오.

**관련 태스크**
IBM MQ MQI client 구성 파일, `mqclient.ini`

*MQI* 채널이 *SSL/TLS*를 사용하도록 지정
MQI 채널이 TLS를 사용하도록 클라이언트 연결 채널의 *SSLCipherSpec* 속성 값은 클라이언트 플랫폼의 IBM MQ에서 지원되는 CipherSpec 이름이어야 합니다.

다음과 같은 방법으로 이 속성 값을 가진 클라이언트 연결 채널을 정의할 수 있습니다. 이는 우선순위 감소 순으로 나열됩니다.

1. PreConnect 엑시트가 사용할 채널 정의 구조를 제공하는 경우.

   PreConnect 엑시트는 채널 정의 구조 MQCD의 *SSLCipherSpec* 필드에서 CipherSpec 이름을 제공할 수 있습니다. 이 구조는 PreConnect 엑시트에서 사용하는 MQNXP 엑시트 매개변수 구조의 **ppMQCDArrayPtr** 필드에서 리턴됩니다.
2. IBM MQ MQI client 애플리케이션이 MQCONNX 호출을 발행하는 경우.

   애플리케이션은 채널 정의 구조 MQCD의 *SSLCipherSpec* 필드에 CipherSpec 이름을 지정할 수 있습니다.이 구조는 MQCONNX 호출의 매개변수인 연결 옵션 구조 MQCNO에서 참조됩니다.
3. 클라이언트 채널 정의 테이블(CCDT) 사용.

   클라이언트 채널 정의 테이블에 있는 하나 이상의 입력 항목은 CipherSpec의 이름을 지정할 수 있습니다.예를 들어, DEFINE CHANNEL MQSC 명령을 사용하여 항목을 작성한 경우 명령에 대해 SSLCIPH 매개변수를 사용하여 CipherSpec의 이름을 지정할 수 있습니다.
4. Windows에서 Active Directory 사용.

   Windows 시스템에서 **setmqscp** 제어 명령을 사용하여 Active Directory에 클라이언트 연결 채널 정의를 발행할 수 있습니다. 이 정의 중 하나 이상은 CipherSpec 이름을 지정할 수 있습니다.

예를 들어, 클라이언트 애플리케이션이 MQCONNX 호출에서 MQCD 구조에 클라이언트 연결 채널 정의를 제공하는 경우, 이 정의는 IBM MQ 클라이언트가 액세스할 수 있는 클라이언트 채널 정의 테이블의 임의 항목에 대한 환경 설정에 사용됩니다.

TLS를 사용하는 MQI 채널의 클라이언트 종료 시 채널 정의를 제공하는 데 MQSERVER 환경 변수를 사용할 수 없습니다.

클라이언트 인증서가 플로우되었는지 여부를 확인하려면 피어 이름 매개변수 값의 존재에 대한 채널의 서버 측에서 채널 상태를 표시하십시오.

**관련 개념**
414 페이지의 『IBM MQ MQI client를 위해 CipherSpec 지정』
IBM MQ MQI client를 위해 CipherSpec을 지정하기 위한 세 가지 옵션이 있습니다.

## *IBM MQ*의 *CipherSpec* 및 *CipherSuite*
IBM MQ는 TLS 1.3 및 TLS 1.2 CipherSpec, 그리고 RSA 및 Diffie-Hellman 알고리즘을 지원합니다. 하지만 이를 수행해야 하는 경우 더 이상 사용되지 않는 CipherSpec을 사용으로 설정할 수 있습니다.

다음에 대한 정보는 392 페이지의 『CipherSpec 사용 가능』의 내용을 참조하십시오.

- IBM MQ에서 지원되는 CipherSpec.
- 더 이상 사용되지 않는 SSL 3.0 및 TLS 1.0 CipherSpec을 사용으로 설정하는 방법

IBM MQ는 RAS 및 Diffie-Hellman 키 교환 및 인증 알고리즘을 지원합니다. TLS 데이터 교환 중 사용되는 키의 크기는 사용자가 사용하는 디지털 인증서에 따라 결정될 수 있지만 일부 CiperSpec은 데이터 교환 키 크기의 스

펙을 포함합니다. 데이터 교환 키 크기가 클수록 보다 강력한 인증이 제공됩니다. 키 크기가 작아지면 데이터 교환이 보다 빨라집니다.

**관련 개념**

20 페이지의 『CipherSpecs 및 CipherSuites』
암호화 보안 프로토콜은 안전 연결에서 사용되는 알고리즘에 동의해야 합니다. CipherSpec 및 CipherSuite는 특정 알고리즘 결합을 정의합니다.

## *IBM MQ*에서의 *NSA* 스위트 *B Cryptography*

이 주제에서는 Suite B 호환 TLS 1.2 프로파일을 준수하도록 IBM MQ for AIX, Linux, and Windows을(를) 구성하는 방법에 관한 정보를 제공합니다.

시간이 흐름에 따라 NSA Cryptography 스위트 B 표준은 암호화 알고리즘 프로토콜에 대한 새 공격을 반영할 수 있도록 업데이트되었습니다. 예를 들어, 일부 CipherSpec은 스위트 B 인증 CipherSpec이 되기 위해 정지될 수 있습니다. 이런 경우 IBM MQ도 최신 표준 구현으로 업데이트됩니다. 그 결과 유지보수를 적용한 후에는 동작이 변경됨을 알 수 있습니다. IBM MQ Readme 파일은 각 제품 유지보수 레벨에서 적용된 스위트 B의 버전을 나열합니다. IBM MQ가 스위트 B 준수를 강제 적용하도록 구성하려면 유지보수 적용 계획 시에 Readme 파일을 항상 참조하십시오. IBM MQ, WebSphere MQ및 MQSeries® 제품 Readme의 내용을 참조하십시오.

AIX, Linux, and Windows 시스템에서 IBM MQ이(가) 표 1에 표시된 보안 레벨에서 Suite B 호환 TLS 1.2 프로파일을 준수하도록 구성할 수 있습니다.

표 *3.* 허용된 *CipherSpec* 및 디지털 서명 알고리즘이 사용되는 스위트 *B* 보안 레벨

| **보안 레벨** | **허용되는 CipherSpec** | **허용되는 디지털 서명 알고리즘** |
|---|---|---|
| 128비트 | ECDHE_ECDSA_AES_128_GCM_SHA256<br>ECDHE_ECDSA_AES_256_GCM_SHA384 | SHA-256을 사용하는 ECDSA<br>SHA-384를 사용한 ECDSA |
| 192비트 | ECDHE_ECDSA_AES_256_GCM_SHA384 | SHA-384를 사용한 ECDSA |
| 모두 [1] | ECDHE_ECDSA_AES_128_GCM_SHA256<br>ECDHE_ECDSA_AES_256_GCM_SHA384 | SHA-256을 사용하는 ECDSA<br>SHA-384를 사용한 ECDSA |

1. 128비트 및 192비트 보안 레벨을 동시에 구성하는 것도 가능합니다. 스위트 B 구성은 최소 허용 가능 암호화 알고리즘을 판별하기 때문에 두 보안 레벨 구성은 128비트 보안 레벨 구성과 동일합니다. 192비트 보안 레벨의 암호화 알고리즘은 128비트 보안 레벨에 필요한 최소값보다 강력하기 때문에 192비트 보안 레벨이 사용 가능하지 않더라도 128비트 보안 레벨에 대해 허용됩니다.

**참고:** 보안 레벨에 사용되는 이름 지정 규칙은 Elliptic Curve 크기 또는 AES 암호화 알고리즘 키 크기를 표시할 필요는 없습니다.

## 스위트 B에 대한 CipherSpec 준수

IBM MQ 의 기본 동작은 스위트 B 표준을 준수하지 않는 것이지만, IBM MQ 는 AIX, Linux, and Windows 시스템에서 보안 레벨 중 하나 또는 둘 다를 준수하도록 구성될 수 있습니다. 스위트 B를 사용하도록 IBM MQ를 올바르게 구성한 후에 스위트 B를 준수하지 않는 CipherSpec를 사용하는 아웃바운드 채널 시작 시도는 오류 AMQ9282를 초래합니다. 이 활동은 MQI 클라이언트도 이유 코드 MQRC_CIPHER_SPEC_NOT_SUITE_B를 리턴하도록 합니다. 이와 유사하게 스위트 B 구성을 준수하지 않는 CipherSpec을 사용하는 인바운드 채널 시작 시도도 오류 AMQ9616을 초래합니다.

IBM MQ CipherSpec에 대한 자세한 정보는 392 페이지의 『CipherSpec 사용 가능』의 내용을 참조하십시오.

## 스위트 B 및 디지털 인증서

스위트 B는 디지털 인증서 서명에 사용할 수 있는 디지털 서명 알고리즘을 제한합니다. 스위트 B는 인증서가 포함할 수 있는 공개 키 유형도 제한합니다. 따라서, IBM MQ는 디지털 서명 알고리즘 및 공개 키 유형이 원격 파트너의 구성된 스위트 B 보안 레벨에서 허용되는 인증서를 사용하도록 구성해야 합니다. 보안 레벨 요구사항을 준수하지 않는 디지털 인증서는 거부되고 연결은 AMQ9633 또는 AMQ9285 오류로 실패합니다.

128비트 스위트 B 보안 레벨에 대해 인증서 주제의 공개 키는 NIST P-256 Elliptic Curve 또는 NIST P-384 Elliptic Curve를 사용하고 NIST P-256 Elliptic Curve 또는 NIST P-384 Elliptic Curve로 서명되어야 합니다. 192비트 스위트 B 보안 레벨에서 인증서 주제의 공개 키는 NIST P-384 Elliptic Curve를 사용하고 NIST P-384 Elliptic Curve로 서명되어야 합니다.

스위트 B 준수 조작에 적합한 인증서를 확보하려면 **runmqakm** 명령을 사용하여 **-sig_alg** 매개변수를 지정하여 적합한 디지털 서명 알고리즘을 요청하십시오. `EC_ecdsa_with_SHA256` 및 `EC_ecdsa_with_SHA384` **-sig_alg** 매개변수 값은 허용되는 스위트 B 디지털 서명 알고리즘으로 서명되는 Elliptic Curve 키에 대응합니다.

**runmqakm** 명령에 대한 자세한 정보는 504 페이지의 『AIX, Linux, and Windows에서 키 및 인증서 관리』의 내용을 참조하십시오.

## 디지털 인증서 작성 및 요청

스위트 B 테스트를 위해 자체 서명 디지털 인증서를 작성하려는 경우에는 506 페이지의 『AIX, Linux, and Windows에서 자체 서명된 개인 인증서 작성』의 내용을 참조하십시오.

스위트 B 프로덕션 사용을 위해 CA 서명 디지털 인증서를 요청하려는 경우에는 508 페이지의 『AIX, Linux, and Windows에서 개인 인증서 요청』의 내용을 참조하십시오.

**참고:** 사용 중인 인증 기관은 IETF RFC 6460에서 설명하는 요구사항을 충족시키는 디지털 인증서를 생성해야 합니다.

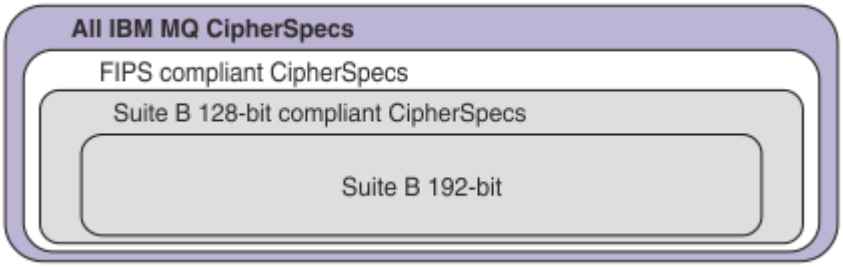
## FIPS 140-2 및 스위트 B

**참고:** AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

스위트 B 표준은 보증된 보안 레벨을 제공하기 위해서 사용 가능한 암호화 알고리즘 세트를 제한하기 때문에 개념상 FIPS 140-2와 비슷합니다. 현재 지원되는 스위트 B CipherSpec은 IBM MQ가 FIPS 140-2 준수 조작용으로 구성된 경우에 사용할 수 있습니다. 따라서, FIPS 및 스위트 B 준수 모두에 대해 동시에 IBM MQ를 구성할 수 있으며 이 경우 두 제한사항 세트가 모두 적용됩니다.

다음 다이어그램은 이러한 서브세트 간의 관계를 보여줍니다.




## 스위트 B 준수 조작을 위해 IBM MQ 구성

스위트 B 준수 조작을 위해 AIX, Linux, and Windows 에서 IBM MQ 를 구성하는 방법에 대한 정보는 41 페이지의 『스위트 B에 대해 IBM MQ 구성』의 내용을 참조하십시오.

IBM MQ 는 다음 플랫폼 및 클라이언트에서 스위트 B 준수 조작을 지원하지 않습니다.

- IBM i 플랫폼
- z/OS 플랫폼
- Java 클라이언트

- JMS 클라이언트

**관련 개념**

254 페이지의 『MQI 클라이언트에서 런타임 시 FIPS 인증 CipherSpec만 사용하도록 지정』
FIPS 준수 소프트웨어를 사용하여 키 저장소를 작성한 다음 채널이 FIPS-인증 CipherSpec을 사용해야 함을 지정하십시오.

## ALW 스위트 *B*에 대해 *IBM MQ* 구성

IBM MQ은(는) AIX, Linux, and Windows 플랫폼에서 NSA Suite B 표준을 준수하여 작동하도록 구성할 수 있습니다.

스위트 B는 보증된 보안 레벨을 제공하기 위해서 사용 가능한 암호화 알고리즘 세트를 제한합니다. IBM MQ는 보안의 확장 레벨을 제공하기 위해 스위트 B를 준수하여 작동하도록 구성 가능합니다. 스위트 B에 대한 추가 정보는 22 페이지의 『NSA(National Security Agency) 스위트 B 암호화』의 내용을 참조하십시오. 스위트 B 구성 및 TLS 채널의 효과에 대한 자세한 정보는 39 페이지의 『IBM MQ에서의 NSA 스위트 B Cryptography』의 내용을 참조하십시오.

### 큐 관리자

큐 관리자에 대해 **SUITEB** 매개변수를 지정한 **ALTER QMGR** 명령을 사용하여 필수 보안 레벨에 적합한 값을 설정하십시오. 자세한 정보는 ALTER QMGR을 참조하십시오.

**MQIA_SUITE_B_STRENGTH** 매개변수를 지정하여 PCF **MQCMD_CHANGE_Q_MGR** 명령을 사용하여 스위트 B 준수 조작을 위한 큐 관리자를 구성할 수도 있습니다.

**참고:** 큐 관리자 스위트 B 설정을 대체하는 경우, MQXR 서비스를 해당 설정이 적용되도록 재시작해야 합니다.

### MQI 클라이언트

기본적으로 MQI 클라이언트는 스위트 B 준수를 강제 적용하지 않습니다. 다음 옵션 중 하나를 실행하여 스위트 B 준수를 위해 MQI 클라이언트를 사용할 수 있습니다.

1. MQCONNX 호출에서 MQSCO 구조에 EncryptionPolicySuiteB 필드를 다음 값 중 하나 이상으로 설정.
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

   MQ_SUITE_B_NONE 를 다른 값과 함께 사용하는 것은 올바르지 않습니다.

   MQSCO 구조에 대한 자세한 정보는 MQSCO-SSL 구성 옵션을 참조하십시오.
2. **MQSUITEB** 환경 변수를 다음 값 중 하나 이상으로 설정합니다.
   - NONE
   - 128_BIT
   - 192_BIT

   쉼표로 구분된 목록을 사용하여 여러 개의 값을 지정할 수 있습니다. 다른 값과 함께 NONE 값을 사용하는 것은 올바르지 않습니다.
3. 클라이언트 구성 파일의 SSL 스탠자 에 있는 **EncryptionPolicySuiteB** 속성을 다음 값 중 하나 이상으로 설정합니다.
   - NONE
   - 128_BIT
   - 192_BIT

   쉼표로 구분된 목록을 사용하여 여러 개의 값을 지정할 수 있습니다. 다른 값과 함께 NONE 을 사용하는 것은 올바르지 않습니다.

**참고:** MQI 클라이언트 설정은 우선순위대로 나열됩니다. MQCONNX 호출의 MSCO 구조는 SSL 스탠자의 속성을 대체하는 **MQSUITEB** 환경 변수의 설정을 대체합니다.

### .NET

.NET 비관리 클라이언트에 대해 **MQC.ENCRYPTION_POLICY_SUITE_B** 특성은 필요한 스위트 B 보안 유형을 나타냅니다.

IBM MQ classes for .NET에서 스위트 B 사용에 대한 자세한 정보는 MQEnvironment .NET 클래스를 참조하십시오.

### AMQP

큐 관리자에 대한 스위트 B 속성 설정은 해당 큐 관리자의 AMQP 채널에 적용됩니다. 큐 관리자 스위트 B 설정을 수정하는 경우 채널의 AMQP 서비스를 재시작하여 변경사항을 적용해야 합니다.

## *IBM MQ*의 인증서 유효성 검증 정책

인증서 유효성 검증 정책에서는 인증서 체인 유효성 검증이 업계 보안 표준을 엄격하게 준수하는 방법을 판별합니다.

인증서 유효성 검증 정책은 다음과 같이 플랫폼 및 환경에 따라 다릅니다.

- 모든 플랫폼의 Java 및 JMS 애플리케이션의 경우, 인증서 유효성 검증 정책은 JRE(Java Runtime Environment)의 JSSE 컴포넌트에 따라 다릅니다. 인증서 유효성 검증 정책에 대한 자세한 정보는 JRE에 대한 문서를 참조하십시오.
- ALW AIX, Linux, and Windows 시스템의 경우 인증서 유효성 검증 정책은 IBM Global Security Kit (GSKit) 에서 제공하며 구성할 수 있습니다. V 9.4.0 V 9.4.0 세 개의 다른 인증서 유효성 검증 정책이 지원됩니다.
  - 현재 IETF 인증서 유효성 검증 표준을 준수하지 않는 이전 디지털 인증서와의 최대 역호환성 및 상호 운용성에 사용되는 레거시 인증서 유효성 검증 정책. 이 정책은 기본 정책이라고도 합니다.
  - RFC 5280 표준을 시행하는 엄격한 표준 준수 인증서 유효성 검증 정책. 이 정책은 표준 정책이라고도 합니다.
  - V 9.4.0 V 9.4.0 클라이언트 애플리케이션에만 사용 가능한 TLS 서버 인증서를 인증하지 않는 인증서 유효성 검증 정책입니다.
- IBM i IBM i 시스템의 경우 인증서 유효성 검증 정책은 운영 체제에서 제공하는 보안 소켓 라이브러리에 따라 다릅니다. 인증서 유효성 검증 정책에 대한 자세한 정보는 운영 체제에 대한 문서를 참조하십시오.
- z/OS z/OS 시스템의 경우 인증서 유효성 검증 정책은 운영 체제에서 제공하는 시스템 SSL 컴포넌트에 따라 다릅니다. 인증서 유효성 검증 정책에 대한 자세한 정보는 운영 체제에 대한 문서를 참조하십시오.

인증서 유효성 검증 정책을 구성하는 방법에 대한 정보는 42 페이지의 『IBM MQ에서 인증서 유효성 검증 정책 구성』의 내용을 참조하십시오. 기본 및 표준 인증서 유효성 검증 정책 사이의 차이점에 대한 자세한 정보는 AIX, Linux, and Windows에서 인증서 유효성 검증 및 신뢰 정책 디자인을 참조하십시오.

## *IBM MQ*에서 인증서 유효성 검증 정책 구성

원격 파트너 시스템에서 수신된 디지털 인증서의 유효성을 검증하는 데 사용되는 TLS 인증서 유효성 검증 정책을 지정할 수 있는 여러 가지 방법이 있습니다.

### 이 태스크 정보

인증서 유효성 검증 정책에서는 인증서 체인 유효성 검증이 업계 보안 표준을 엄격하게 준수하는 방법을 판별합니다. 인증서 유효성 검증 정책은 플랫폼 및 환경에 따라 다릅니다. 인증서 유효성 검증 정책에 대한 자세한 정보는 42 페이지의 『IBM MQ의 인증서 유효성 검증 정책』의 내용을 참조하십시오.

### 프로시저

- 큐 관리자에서 인증서 유효성 검증 정책을 설정하려면 큐 관리자 속성 **CERTVPOL**를 사용하십시오.
  이 속성 설정에 대한 자세한 정보는 ALTER QMGR (큐 관리자 설정 변경)을 참조하십시오.
- 클라이언트에서 인증서 유효성 검증 정책을 설정하려면 다음 메소드를 사용하십시오.
  정책을 설정하는 데 둘 이상의 방법이 사용되는 경우 클라이언트는 다음 우선순위에 따라 설정을 사용합니다.

  1. 클라이언트 MQSCO 구조에서 `CertificateValPolicy` 필드를 사용하십시오. 필드를 다음 값 중 하나로 설정하십시오.

     **MQ_CERT_VAL_POLICY_ANY**
     소스 소켓 라이브러리에서 지원하는 각 인증서 유효성 검증 정책을 적용합니다. 정책이 인증서 체인을 유효한 것으로 간주하는 경우 인증서 체인을 승인합니다.

     **MQ_CERT_VAL_POLICY_RFC5280**
     RFC5280 준수 인증서 유효성 검증 정책만 적용합니다. 이 설정은 임의(ANY) 설정보다 엄격한 유효성 검증을 제공하지만 일부 오래된 디지털 인증서는 거부합니다.

     V 9.4.0 V 9.4.0 **MQ_CERT_VAL_POLICY_NONE**
     인증서 유효성 검증 정책을 적용하지 않습니다. 이 설정은 클라이언트 애플리케이션 전용이며 신뢰 체인의 유효성을 검증하지 않고 TLS 서버 인증서를 승인합니다.

     이 필드를 사용하는 데 대한 자세한 정보는 MQSCO - SSL 구성 옵션을 참조하십시오.

  2. 클라이언트 환경 변수 **MQCERTVPOL**를 사용하십시오. 이 환경 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

     - Linux AIX AIX and Linux 시스템의 경우

       ```
       export MQCERTVPOL= value
       ```

     - Windows Windows 시스템의 경우

       ```
       SET MQCERTVPOL= value
       ```

     - IBM i IBM i 시스템의 경우

       ```
       ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
       ```

  3. 클라이언트 구성 파일에서 SSL 스탠자의 **CertificateValPolicy** 속성을 사용하십시오. 이 속성을 다음 값 중 하나로 설정하십시오.

     **ANY**
     기본 보안 소켓 라이브러리가 지원하는 모든 인증서 유효성 검증 정책을 사용합니다. 이 설정이 기본 설정입니다.

     **RFC5280**
     RFC 5280 표준을 준수하는 인증서 유효성 검증만 사용합니다.

     V 9.4.0 V 9.4.0 **NONE**
     인증서 유효성 검증 정책을 적용하지 않습니다. 이 설정은 신뢰 체인의 유효성을 검증하지 않고 TLS 서버 인증서를 승인합니다.

     이 속성 사용에 대한 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오.

## *IBM MQ*에서 디지털 인증서와 *CipherSpec*의 호환성

이 토픽은 IBM MQ에서 CipherSpec과 디지털 인증서 사이의 관계를 소개하여 보안 정책에 대해 적합한 CipherSpec 및 디지털 인증서를 선택하는 방법에 대한 정보를 제공합니다.

지원되는 CipherSpec 서브세트만 디지털 인증서의 지원되는 모든 유형에 사용할 수 있습니다. 따라서, 디지털 인증서에 대해 적합한 CipherSpec를 선택해야 합니다. 이와 유사하게 조직의 보안 정책에서 특정 CipherSpec을 사용하도록 요구하면 해당 CipherSpec에 대해 적합한 디지털 인증서를 확보해야 합니다.

## MD5 디지털 서명 알고리즘 및 TLS 1.2

MD5 알고리즘을 사용하여 서명된 디지털 인증서는 TLS 1.2 프로토콜이 사용되는 경우 거부됩니다. 이제는 많은 암호 분석가가 MD5 알고리즘을 취약하다고 생각하고 사용을 권장하지 않기 때문입니다. TLS 1.2 프로토콜을 기반으로 하는 최신 CipherSpec을 사용하려면 디지털 인증서가 해당 디지털 서명에 MD5 알고리즘을 사용하지 않도록 해야 합니다. TLS 1.0 프로토콜을 사용하는 이전 CipherSpec은 이 제한사항으로 제한되지 않으며 MD5 디지털 서명의 인증서를 계속 사용할 수 있습니다.

특정 인증서에 대한 디지털 서명 알고리즘을 보기 위해서 **runmqakm** 명령을 사용할 수 있습니다.

```
runmqakm -cert -details -db key.kdb -pw password -label cert_label
```

여기서, *cert_label*은 표시하려는 디지털 서명 알고리즘의 인증서 레이블입니다. 자세한 내용은 디지털 인증서 레이블을 참조하십시오.

**runmqakm** 명령 실행은 지정한 서명 알고리즘 사용을 표시하는 출력을 작성합니다.

```
Label : ibmmqexample
Key Size : 1024
Version : X509 V3
Serial : 4e4e93f1
Issuer : CN=Old Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : August 19, 2011 5:48:49 PM GMT+01:00
Not After : August 18, 2012 5:48:49 PM GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 98 5A 7A
    F0 18 21 EE E4 8A 6E DE C8 01 4B 3A 1E 41 90 3D
    CE 01 3F E6 32 30 6C 23 59 F0 FE 78 6D C2 80 EF
    BC 83 54 7A EB 60 80 62 6B F1 52 FE 51 9D C1 61
    80 A5 1C D4 F0 76 C7 15 6D 1F 0D 4D 31 3E DC C6
    A9 20 84 6E 14 A1 46 7D 4C F5 79 4D 37 54 0A 3B
    A9 74 ED E7 8B 0F 80 31 63 1A 0B 20 A5 99 EE 0A
    30 A6 B6 8F 03 97 F6 99 DB 6A 58 89 7F 27 34 DE
    55 08 29 D8 A9 6B 46 E6 02 17 C3 13 D3 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    09 4E 4F F2 1B CB C1 F4 4F 15 C9 2A F7 32 0A 82
    DA 45 92 9F
Fingerprint : MD5 :
    44 54 81 7C 58 68 08 3A 5D 75 96 40 D5 8C 7A CB
Fingerprint : SHA256 :
    3B 47 C6 E7 7B B0 FF 85 34 E7 48 BE 11 F2 D4 35
    B7 9A 79 53 2B 07 F5 E7 65 E8 F7 84 E0 2E 82 55
Signature Algorithm : MD5WithRSASignature (1.2.840.113549.1.1.4)
Value
    3B B9 56 E6 F2 77 94 69 5B 3F 17 EA 7B 19 D0 A2
    D7 10 38 F1 88 A4 44 1B 92 35 6F 3B ED 99 9B 3A
    A5 A4 FC 72 25 5A A9 E3 B1 96 88 FC 1E 9F 9B F1
    C5 E8 8E CF C4 8F 48 7B 0E A6 BB 13 AE 2B BD D8
    63 2C 03 38 EF DC 01 E1 1F 7A 6F FB 2F 65 74 D0
    FD 99 94 BA B2 3A D5 B4 89 6C C1 2B 43 6D E2 39
    66 6A 65 CB C3 C4 E2 CC F5 49 39 A3 8B 93 5A DD
    B0 21 0B A8 B2 59 5B 24 59 50 44 89 DC 78 19 51
Trust Status : Enabled
```

`Signature Algorithm` 행은 `MD5WithRSASignature` 알고리즘이 사용되는 것을 보여줍니다. 이 알고리즘은 MD5를 기반으로 하기 때문에 이 디지털 인증서는 TLS 1.2 CipherSpec과 같이 사용할 수 없습니다.

## Elliptic Curve 및 RSA CipherSpec의 상호 운용성

모든 CipherSpec을 모든 디지털 인증서와 같이 사용할 수 있는 것은 아닙니다. CipherSpecs 은 CipherSpec 이름 접두부로 표시됩니다. CipherSpec 의 각 유형은 사용할 수 있는 디지털 인증서의 유형에 대해 서로 다른 제한사항을 부과합니다. 이 제한은 모든 IBM MQ TLS 연결에 적용되지만 특히 Elliptic Curve 암호 사용자에게 연관됩니다.

다음 표는 CipherSpec과 디지털 인증서 사이의 관계를 요약합니다.

표 *4. CipherSpec*과 디지털 인증서 사이의 관계

| 유형 | CipherSpec 이름 접두부 | 설명 | 필수 공개 키 유형 | 디지털 서명 암호화 알고리즘 | 보안 키 설정 메소드 |
|---|---|---|---|---|---|
| 1 | ECDHE_ECDSA_ | Elliptic Curve 공개 키, Elliptic Curve 보안 키, Elliptic Curve 디지털 서명 알고리즘을 사용하는 CipherSpec. | Elliptic Curve | ECDSA | ECDHE |
| 2 | ECDHE_RSA_ | RSA 공개 키, Elliptic Curve 보안 키 및 RSA 디지털 서명 알고리즘을 사용하는 CipherSpec. | RSA | RSA | ECDHE |
| 3 | (모든 TLS 1.3 CipherSpecs) | CipherSpecs -Elliptic Curve 또는 RSA 공개 키, Elliptic Curve 비밀 키 및 Elliptic Curve 또는 RSA 디지털 서명 알고리즘을 사용합니다. | 타원 곡선 또는 RSA | ECDSA 또는 RSA | ECDHE 또는 RSA |
| 4 | (나머지 모두) | RSA 공개 키 및 RSA 디지털 서명 알고리즘을 사용하는 CipherSpec. | RSA | RSA | RSA |

**참고:** 유형 1 및 2 CipherSpec은 IBM MQ 큐 관리자 및 IBM i 플랫폼의 MQI 클라이언트에서는 지원되지 않습니다.

필수 공개 키 유형 열은 개인 인증서가 CipherSpec 각 유형 사용 시에 가져야 하는 공개 키 유형을 보여줍니다. 개인 인증서는 해당 원격 파트너에 대해 큐 관리자 또는 클라이언트를 식별하는 일반 엔티티 인증서입니다.

인증서 레벨에 이름 지정된 인증서가 채널 CipherSpec에 적합해야 합니다. 즉, EC(Elliptic Curve) 인증서가 필요한 CipherSpec에서 채널을 구성하는 경우 인증서 레이블에서 RSA 인증서 이름을 지정할 수 없습니다. RSA 인증서가 필요한 CipherSpec에서 채널을 구성하는 경우 인증서 레이블에서 EC 인증서 이름을 지정할 수 없습니다.

올바르게 IBM MQ를 구성한 경우 다음을 수행할 수 있습니다.

- RSA 및 EC 인증서가 혼합된 단일 큐 관리자.
- RSA 또는 EC 인증서를 사용하여 동일한 큐 관리자에서 다른 채널.

디지털 서명 암호화 알고리즘은 피어 유효성 검증에 사용되는 암호화 알고리즘을 나타냅니다. 암호화 알고리즘은 디지털 서명을 처리하기 위해 MD5, SHA-1 또는 SHA-256과 같은 해시 알고리즘을 같이 사용합니다. 사용할 수 있는 디지털 서명 알고리즘은 다양합니다(예: MD5가 포함된 RSA 또는 SHA-256이 포함된 ECDSA). 이 표에서 ECDSA는 ECDSA를 사용하는 디지털 서명 알고리즘 세트를 나타내며 RSA는 RSA를 사용하는 디지털 서명 알고리즘의 세트를 나타냅니다. 세트에서 지원되는 모든 디지털 서명 알고리즘은 언급된 암호화 알고리즘을 기반으로 하는 경우 사용 가능합니다.

유형 1 CipherSpec의 경우 개인 인증서에 Elliptic Curve 공개 키가 필요합니다. 이 CipherSpec이 사용되는 경우 Elliptic Curve Diffie Hellman Ephemeral 키 계약이 연결에 대한 보안 키 설정에 사용됩니다.

유형 2 CipherSpec의 경우 개인 인증서가 RSA 공개 키를 포함해야 합니다. 이 CipherSpec이 사용되는 경우 Elliptic Curve Diffie Hellman Ephemeral 키 계약이 연결에 대한 보안 키 설정에 사용됩니다.

유형 3 CipherSpec의 경우 개인 인증서는 RSA 공개 키를 포함해야 합니다. 이 CipherSpec이 사용되는 경우 RSA 키 교환이 연결에 대한 보안 키 설정에 사용됩니다.

이 제한 목록은 완벽하지 않으며 구성에 따라 상호 운영하는 기능에 추가 영향을 줄 수 있는 추가 제한이 있을 수도 있습니다. 예를 들어, IBM MQ가 FIPS 140-2 또는 NSA 스위트 B 표준을 준수하도록 구성되는 경우 이는 허용 가능한 구성 범위도 제한합니다. 자세한 정보는 다음 절을 참조하십시오.

동일한 큐 관리자 또는 클라이언트 애플리케이션에서 CipherSpec의 다른 유형을 사용해야 하는 경우, 적절한 인증서 레이블 및 CipherSpec 결합을 클라이언트 정의에서 구성하십시오.

세 가지 유형의 CipherSpec은 직접 상호 운용되지 않으며 이는 현재 TLS 표준의 제한사항입니다. 예를 들어, 이름이 QM1인 큐 관리자에서 이름이 TO.QM1 인 수신자 채널에 대해 ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec 를 사용하도록 선택했다고 가정하면 수신자는 Elliptic Curve키 및 ECDSA 기반 디지털 서명이 있는

개인 인증서를 가지고 있어야 합니다. 수신자 채널이 이러한 요구사항을 충족하지 않으면 채널이 시작되지 않습니다.

큐 관리자 QM1에 연결 중인 다른 채널은 다른 CipherSpec을 사용할 수 있으며 이는 해당 채널의 CipherSpec에 대해 올바른 유형의 인증서를 각 채널이 사용하는 경우에 해당합니다. 예를 들어, QM1이 TO.QM2 송신자 채널을 사용하여 메시지를 QM2의 다른 큐 관리자로 송신한다고 가정해 보십시오. 채널의 양쪽에서 RSA 공개 키를 포함하는 인증서를 사용하는 경우 채널 TO.QM2는 유형 3 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256을 사용할 수 있습니다. 인증서 레이블 채널 속성은 각 채널에 대해 다른 인증서를 구성하는 데 사용할 수 있습니다.

IBM MQ 네트워크를 계획할 때는 TLS가 필요한 채널을 신중하게 고려하고 각 채널에 사용되는 인증서의 유형이 해당 채널에서 CipherSpec과 함께 사용하기에 적합한지 확인하십시오.

디지털 서명에 대한 디지털 서명 알고리즘 및 공개 키 유형을 확인하기 위해 **runmqakm** 명령을 사용할 수 있습니다.

```
runmqakm -cert -details -db key.kdb -pw password -label cert_label
```

여기서, *cert_label*은 표시해야 하는 디지털 서명 알고리즘의 인증서 레이블입니다. 자세한 내용은 디지털 인증서 레이블을 참조하십시오.

**runmqakm** 명령 실행은 공개 키 유형을 표시하는 출력을 작성합니다.

```
Label : ibmmqexample
Key Size : 384
Version : X509 V3
Serial : 9ad5eeef5d756f41
Issuer : CN=Example Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : 21 August 2011 13:10:24 GMT+01:00
Not After : 21 August 2012 13:10:24 GMT+01:00
Public Key
    30 76 30 10 06 07 2A 86 48 CE 3D 02 01 06 05 2B
    81 04 00 22 03 62 00 04 3E 6F A9 06 B6 C3 A0 11
    F8 D6 22 78 FE EF 0A FE 34 52 C0 8E AB 5E 81 73
    D0 97 3B AB D6 80 08 E7 31 E9 18 3F 6B DE 06 A7
    15 D6 9D 5B 6F 56 3B 7F 72 BB 6F 1E C9 45 1C 46
    60 BE F2 DC 1B AD AC EC 64 4C 0E 06 65 6E ED 93
    B8 F5 95 E0 F9 2A 05 D6 21 02 BD FB 06 63 A1 CC
    66 C6 8A 0A 5C 3F F7 D3
Public Key Type : EC_ecPublicKey (1.2.840.10045.2.1)
Fingerprint : SHA1 :
    3C 34 58 04 5B 63 5F 5C C9 7A E7 67 08 2B 84 43
    3D 43 7A 79
Fingerprint : MD5 :
    49 13 13 E1 B2 AC 18 9A 31 41 DC 8C B4 D6 06 68
Fingerprint : SHA256 :
    6F 76 78 68 F3 70 F1 53 CE 39 31 D9 05 C5 C5 9F
    F2 B8 EE 21 49 16 1D 90 64 6D AC EB 0C A7 74 17
Signature Algorithm : EC_ecdsa_with_SHA384 (1.2.840.10045.4.3.3)
Value
    30 65 02 30 0A B0 2F 72 39 9E 24 5A 22 FE AC 95
    0D 0C 6D 6C 2F B3 E7 81 F6 C1 36 1B 9A B0 6F 07
    59 2A A1 4C 02 13 7E DD 06 D6 FE 4B E4 03 BC B1
    AC 49 54 1E 02 31 00 90 0E 46 2B 04 37 EE 2C 5F
    1B 9C 69 E5 99 60 84 84 10 71 1A DA 63 88 33 E2
    22 CC E6 1A 4E F4 61 CC 51 F9 EE A0 8E F4 DC B5
    0B B9 72 58 C3 C7 A4
Trust Status : Enabled
```

이 경우 공개 키 유형 행은 인증서에 Elliptic Curve 공개 키가 있음을 보여줍니다. 이 경우 서명 알고리즘 행은 EC_ecdsa_with_SHA384 알고리즘이 사용 중임을 보여주며 이는 ECDSA 알고리즘을 기반으로 합니다. 따라서 이 인증서는 유형 1 CipherSpec 사용에만 적합합니다.

## TLS 1.3 CipherSpecs

TLS 1.3 CipherSpecs 는 ECDSA및 RSA 인증서를 모두 지원합니다.

## Elliptic Curve CipherSpec 및 NSA 스위트 B

IBM MQ가 스위트 B 준수 TLS 1.2 프로파일을 준수하도록 구성되면 허용되는 CipherSpec 및 디지털 서명 알고리즘은 39 페이지의 『IBM MQ에서의 NSA 스위트 B Cryptography』에서 설명하는 대로 제한됩니다. 또한, 허용 가능한 Elliptic Curve 키 범위는 구성된 보안 레벨에 따라 줄어듭니다.

128비트 스위트 B 보안 레벨에서 인증서 제목 공개 키는 NIST P-256 또는 NIST P-384 Elliptic Curve를 사용하거나 NIST P-256 Elliptic Curve 또는 NIST P-384 Elliptic Curve로 서명되어야 합니다. **runmqakm** 명령은 EC_ecdsa_with_SHA256 또는 EC_ecdsa_with_SHA384의 -sig_alg 매개변수를 사용하여 이 보안 레벨에 대한 디지털 인증서를 요청하는 데 사용할 수 있습니다.

192비트 스위트 B 보안 레벨에서 인증서 주제 공개 키는 NIST P-384 Elliptic Curve를 사용하고 NIST P-384 Elliptic Curve로 서명되어야 합니다. **runmqakm** 명령은 EC_ecdsa_with_SHA384의 -sig_alg 매개변수를 사용하여 이 보안 레벨에 대한 디지털 인증서를 요청하는 데 사용할 수 있습니다.

지원되는 NIST Elliptic Curve는 다음과 같습니다.

표 *5.* 지원되는 *NIST Elliptic Curve*

| **NIST FIPS 186-3 커브 이름** | **RFC 4492 커브 이름** | **Elliptic Curve 키 크기(비트)** |
|---|---|---|
| P-256 | secp256r1 | 256 |
| P-384 | secp384r1 | 384 |
| P-521 | secp521r1 | 521 |

**참고:** NIST P-521 Elliptic Curve는 스위트 B 준수 조작에 사용할 수 없습니다.

**관련 개념**

392 페이지의 『CipherSpec 사용 가능』
**DEFINE CHANNEL** 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec을 사용 가능으로 설정합니다.

254 페이지의 『MQI 클라이언트에서 런타임 시 FIPS 인증 CipherSpec만 사용하도록 지정』
FIPS 준수 소프트웨어를 사용하여 키 저장소를 작성한 다음 채널이 FIPS-인증 CipherSpec을 사용해야 함을 지정하십시오.

39 페이지의 『IBM MQ에서의 NSA 스위트 B Cryptography』
이 주제에서는 Suite B 호환 TLS 1.2 프로파일을 준수하도록 IBM MQ for AIX, Linux, and Windows을(를) 구성하는 방법에 관한 정보를 제공합니다.

22 페이지의 『NSA(National Security Agency) 스위트 B 암호화』
미 정부는 데이터 암호화를 포함한 IT 시스템 및 보안에 대한 기술 자문을 생성합니다. US NSA(National Security Agency)에서는 해당 스위트 B 표준에서 상호 운용 가능한 암호화 알고리즘 세트를 권장합니다.

## 채널 인증 레코드

채널 레벨에서 연결 시스템에 부여된 액세스 권한에 대한 보다 정밀한 제어를 실행하려면 채널 인증 레코드를 사용하십시오.

클라이언트가 바람직하지 않은 조치를 수행하도록 허용하는 공백 사용자 ID 또는 상위 레벨 사용자 ID를 사용해 사용자의 큐 관리자에 연결을 시도하는 것을 발견할 수 있습니다. 채널 인증 레코드를 사용해 이 클라이언트에 대한 액세스를 차단할 수 있습니다. 아니면, 클라이언트가 클라이언트 플랫폼에서는 올바르지만 서버 플랫폼에서는 알 수 없거나 올바르지 않은 형식인 사용자 ID를 사용할 수 있습니다. 채널 인증 레코드를 사용해 사용한 사용자 ID를 올바른 사용자 ID에 맵핑할 수 있습니다.

사용자의 큐 관리자에 연결해 오류를 일으키는 클라이언트 애플리케이션이 발견될 수 있습니다. 이 애플리케이션이 일으키는 문제로부터 서버를 보호하려면 방화벽 규칙이 업데이트되거나 클라이언트 애플리케이션이 수정될 때까지 클라이언트 애플리케이션이 실행되고 있는 IP 주소를 사용해 일시적으로 차단해야 합니다. 채널 인증 레코드를 사용해 클라이언트 애플리케이션이 연결하는 IP 주소를 차단할 수 있습니다.

IBM MQ Explorer와 같은 관리 도구와 해당 용도의 채널을 설정했을 경우 특정 클라이언트 컴퓨터만 사용할 수 있도록 하고 싶을 수 있습니다. 채널 인증 레코드를 사용해 특정 IP 주소에서만 채널을 사용할 수 있게 만들 수 있습니다.

클라이언트로 실행 중인 일부 샘플 애플리케이션을 시작하는 경우 채널 인증 레코드를 사용하여 큐 관리자를 안전하게 설정하는 예로 샘플 프로그램 준비 및 실행을 참조하십시오.

채널 인증 레코드를 가져와서 인바운드 채널을 제어하려면 MQSC 명령 **ALTER QMGR CHLAUTH(ENABLED)**를 사용하십시오.

**CHLAUTH** 규칙은 새 인바운드 연결에 대한 응답으로 작성되는 채널 MCA에 적용됩니다. 로컬에서 시작되는 채널에 대한 응답으로 작성되는 채널 MCA의 경우 **CHLAUTH** 규칙이 적용되지 않습니다.

표 *6.* 서로 다른 채널 쌍에 대해 *CHLAUTH* 규칙이 적용되는 경우

| **채널 유형** | **CHLAUTH 규칙이 적용되는 MCA** |
|---|---|
| SDR-RCVR | RCVR |
| RQSTR-SVR(SVR에서 시작됨) | RQSTR |
| RQSTR-SVR(RQSTR에서 시작됨) | SVR |
| RQSTR-SDR(SDR에서 시작됨) | RQSTR |
| RQSTR-SDR(RQSTR에서 시작됨) | 초기 연결을 위한 SDR. 콜백 연결의 경우 RQSTR. |

다음 기능을 수행하도록 채널 인증 레코드를 작성할 수 있습니다.

- 특정 IP 주소로부터의 연결을 차단합니다.
- 특정 사용자 ID로부터의 연결을 차단합니다.
- 특정 IP 주소로부터 연결하는 모든 채널에 대해 사용하도록 MCAUSER 값을 설정합니다.
- 특정 사용자 ID를 이용하는 모든 채널에 대해 사용하도록 MCAUSER 값을 설정합니다.
- 특정 SSL 또는 TLS 식별 이름(DN)을 갖는 모든 채널에 대해 MCAUSER 값을 사용하도록 설정합니다.
- 특정 큐 관리자로부터 연결하는 모든 채널에 대해 사용하도록 MCAUSER 값을 설정합니다.
- 특정 IP 주소로부터의 연결이 아닌 경우 특정 큐 관리자로부터의 연결을 차단합니다.
- 특정 IP 주소로부터의 연결이 아닌 경우 특정 SSL 또는 TLS 인증서를 제시하는 연결을 차단합니다.

사용에 대해서는 다음 절에서 좀 더 자세히 설명합니다.

MQSC 명령 **SET CHLAUTH** 또는 PCF 명령 **Set Channel Authentication Record**를 사용하여 채널 인증 레코드를 작성, 수정 또는 제거합니다.

**참고:** 채널 인증 레코드 수가 많으면 큐 관리자의 성능에 부정적인 영향을 미칠 수 있습니다.

## IP 주소 차단

특정 IP 주소로부터의 액세스를 차단하는 것은 일반적으로 방화벽의 역할입니다. 하지만 IBM MQ 시스템에 대한 액세스 권한이 없는 IP 주소로부터 연결 시도가 있어 방화벽을 업데이트하기 전에 일시적으로 주소를 차단해야 할 경우가 있습니다. 이러한 연결은 IBM MQ 채널에서 시도되는 것이 아니라 IBM MQ 리스너를 대상으로 하도록 잘못 구성된 다른 소켓 애플리케이션에서 시도되는 것입니다. 유형 BLOCKADDR의 채널 인증 레코드를 설정하여 IP 주소를 차단하십시오. 한 개 이상의 주소, 일정 범위의 주소 또는 와일드카드를 포함한 패턴을 지정할 수 있습니다.

이 방식으로 IP 주소가 차단되어 인바운드 연결이 거부될 때마다 채널 이벤트가 사용 가능하며 큐 관리자가 실행 중인 경우, 이유 규정자가 MQRQ_CHANNEL_BLOCKED_ADDRESS인 이벤트 메시지 MQRC_CHANNEL_BLOCKED가 발행됩니다. 또한 차단된 연결 시도가 반복되어 리스너에 해당 시도가 쇄도하지 않도록 오류를 리턴하기 전에 연결을 30초간 연 상태로 유지합니다.

특정 채널에서만 IP 주소를 차단하거나 오류를 리턴하기 전에 지연되지 않도록 하려면 USERSRC(NOACCESS) 매개변수를 사용하여 ADDRESSMAP 유형의 채널 인증 레코드를 설정하십시오.

이러한 이유로 인해 인바운드 연결이 거부되면 채널 이벤트가 사용 가능하며 큐 관리자가 실행 중인 경우 이유 규정자가 MQRQ_CHANNEL_BLOCKED_NOACCESS인 이벤트 메시지 MQRC_CHANNEL_BLOCKED가 발행됩니다.

359 페이지의 『특정 IP 주소 차단』의 예를 참조하십시오.

### 사용자 ID 차단

특정 사용자 ID가 클라이언트 채널을 통해 연결하지 못하게 막으려면 BLOCKUSER 유형의 채널 인증 레코드를 설정하십시오. 이런 유형의 채널 인증 레코드는 클라이언트 채널에만 적용되며 메시지 채널에는 적용되지 않습니다. 한 개 이상의 개별 사용자 ID를 지정해 차단할 수 있지만 와일드카드는 사용할 수 없습니다.

이런 이유로 인바운드 연결이 거부되면 언제나 이벤트 메시지인 MQRC_CHANNEL_BLOCKED가 이유 규정자인 MQRQ_CHANNEL_BLOCKED_USERID와 함께 발행됩니다. 단, 채널 이벤트가 사용 가능해야 합니다.

360 페이지의 『특정 사용자 ID 차단』의 예를 참조하십시오.

USERSRC(NOACCESS) 매개변수와 함께 USERMAP 유형의 채널 인증 레코드를 설정해 특정 채널에서 지정된 사용자 ID로부터의 액세스를 차단할 수도 있습니다.

이러한 이유로 인해 인바운드 연결이 거부되면 채널 이벤트가 사용 가능하며 큐 관리자가 실행 중인 경우 이유 규정자가 MQRQ_CHANNEL_BLOCKED_NOACCESS인 이벤트 메시지 MQRC_CHANNEL_BLOCKED가 발행됩니다.

363 페이지의 『클라이언트 사용자 ID에 대한 액세스 차단』의 예를 참조하십시오.

### 큐 관리자 이름 차단

지정된 큐 관리자로부터 연결되는 임의의 채널이 액세스 권한을 갖지 않도록 지정하려면 채널 인증 레코드를 USERSRC(NOACCESS) 매개변수와 함께 QMGRMAP 유형으로 설정하십시오. 와일드카드가 포함된 패턴 또는 단일 큐 관리자 이름을 지정할 수 있습니다. 큐 관리자로부터의 액세스를 차단하는 BLOCKUSER 함수에 상응하는 것은 없습니다.

이러한 이유로 인해 인바운드 연결이 거부되면 채널 이벤트가 사용 가능하며 큐 관리자가 실행 중인 경우 이유 규정자가 MQRQ_CHANNEL_BLOCKED_NOACCESS인 이벤트 메시지 MQRC_CHANNEL_BLOCKED가 발행됩니다.

363 페이지의 『리모트 큐 관리자로부터의 액세스 차단』의 예를 참조하십시오.

### SSL 또는 TLS DN 차단

지정된 DN이 포함되어 있는 SSL 또는 TLS 개인 인증서를 제시하는 사용자가 액세스 권한을 갖지 않도록 지정하려면 USERSRC(NOACCESS) 매개변수와 함께 SSLPEERMAP 유형의 채널 인증 레코드를 설정하십시오. 와일드카드가 포함된 패턴 또는 단일 식별 이름을 지정할 수 있습니다. DN용 액세스를 차단하는 BLOCKUSER 함수에 상응하는 것은 없습니다.

이러한 이유로 인해 인바운드 연결이 거부되면 채널 이벤트가 사용 가능하며 큐 관리자가 실행 중인 경우 이유 규정자가 MQRQ_CHANNEL_BLOCKED_NOACCESS인 이벤트 메시지 MQRC_CHANNEL_BLOCKED가 발행됩니다.

364 페이지의 『SSL 또는 TLS 식별 이름의 액세스 차단』의 예를 참조하십시오.

### IP 주소를 사용할 사용자 ID에 맵핑

지정된 IP 주소로부터 연결되는 임의의 채널이 특정 MCAUSER를 사용하도록 지정하려면 채널 인증 레코드를 ADDRESSMAP 유형으로 설정하십시오. 와일드카드가 포함된 패턴 또는 단일 주소, 주소 범위를 지정할 수 있습니다.

포트 포워딩 장치, DMZ 세션 브레이크 또는 큐 관리자에 제시된 IP 주소를 변경하는 다른 설정을 사용한다면 IP 주소 맵핑이 꼭 사용하기에 적합하지는 않습니다.

364 페이지의 『MCAUSER 사용자 ID에 IP 주소 맵핑』의 예를 참조하십시오.

### 사용할 사용자 ID에 큐 관리자 이름 맵핑

지정된 큐 관리자로부터 연결되는 임의의 채널이 특정 MCAUSER를 사용하도록 지정하려면 채널 인증 레코드를 QMGRMAP 유형으로 설정하십시오. 와일드카드가 포함된 패턴 또는 단일 큐 관리자 이름을 지정할 수 있습니다.

361 페이지의 『MCAUSER 사용자 ID에 리모트 큐 관리자 맵핑』의 예를 참조하십시오.

### 클라이언트가 사용한 사용자 ID를 사용할 사용자 ID에 맵핑

IBM MQ MQI 클라이언트의 연결에 특정 사용자 ID가 사용될 경우 다른 지정된 MCAUSER를 사용하도록 지정하려면 USERMAP 유형의 채널 인증 레코드를 설정하십시오. 사용자 ID 맵핑에는 와일드카드를 사용하지 않습니다.

362 페이지의 『MCAUSER 사용자 ID에 클라이언트 사용자 ID 맵핑』의 예를 참조하십시오.

### 사용할 사용자 ID에 SSL 또는 TLS DN 맵핑

지정된 DN이 포함된 SSL/TLS 개인 인증서를 제시하는 임의의 사용자가 특정 MCAUSER를 사용하도록 지정하려면 채널 인증 레코드를 SSLPEERMAP 유형으로 설정하십시오. 와일드카드가 포함된 패턴 또는 단일 식별 이름을 지정할 수 있습니다.

362 페이지의 『MCAUSER 사용자 ID에 SSL 또는 TLS 식별 이름 맵핑』의 예를 참조하십시오.

### IP 주소에 따라 큐 관리자, 클라이언트 또는 SSL/TLS DN 맵핑

경우에 따라서는 써드파티가 큐 관리자 이름을 속일 수도 있습니다. SSL이나 TLS 인증서 또는 핵심 데이터베이스 파일이 도난되거나 재사용될 수도 있습니다. 이러한 위험으로부터 보호하기 위해 특정 큐 관리자나 클라이언트로부터 연결하거나 특정 DN을 사용할 경우 지정된 IP 주소로부터 연결하도록 지정할 수 있습니다. USERMAP, QMGRMAP 또는 SSLPEERMAP 유형의 채널 인증 레코드를 설정하고 ADDRESS 매개변수를 사용하여 허용된 IP 주소 또는 IP 주소 패턴을 지정하십시오.

361 페이지의 『MCAUSER 사용자 ID에 리모트 큐 관리자 맵핑』의 예를 참조하십시오.

### 채널 인증 레코드 사이의 상호 작용

연결을 시도하는 채널이 둘 이상의 채널 인증 레코드와 일치하고 이 시도가 서로 정반대의 효과를 가질 수 있습니다. 예를 들어, 한 채널이 BLOCKUSER 채널 인증 레코드에 의해 차단되어 있는 사용자 ID를 사용하는데 다른 사용자 ID를 설정하는 SSLPEERMAP 레코드와 일치하는 SSL이나 TLS 인증서를 갖고 있을 수 있습니다. 게다가, 채널 인증 레코드가 와일드카드를 사용한다면 단일 IP 주소, 큐 관리자 이름 또는 SSL이나 TLS DN이 여러 개의 패턴과 일치할 수 있습니다. 예를 들어, IP 주소 192.0.2.6은 패턴 192.0.2.0-24, 192.0.2. *, 및 192.0. * .6. 수행되는 조치는 다음과 같이 결정됩니다.

- 사용된 채널 인증 레코드는 다음과 같이 선택됩니다.
  - 채널 이름과 명확히 일치하는 채널 인증 레코드가 와일드카드를 사용하여 채널 이름과 일치하는 채널 인증 레코드보다 우선합니다.
  - SSL 또는 TLS DN을 사용하는 채널 인증 레코드가 사용자 ID, 큐 관리자 이름 또는 IP 주소를 사용하는 레코드보다 높은 우선순위를 갖습니다.
  - 사용자 ID 또는 큐 관리자 이름을 사용하는 채널 인증 레코드가 IP 주소를 사용하는 레코드에 비해 높은 우선순위를 갖습니다.
- 일치하는 채널 인증 레코드가 발견되었는데 MCAUSER를 지정하는 경우 이 MCAUSER가 채널에 지정됩니다.
- 일치하는 채널 인증 레코드가 발견되었는데 이 레코드가 해당 채널이 어떤 액세스 권한도 갖지 못하도록 지정하는 경우 *NOACCESS의 MCAUSER 값이 채널에 지정됩니다. 이 값은 보안 엑시트 프로그램에 의해 나중에 변경할 수 있습니다.
- 일치하는 채널 인증 레코드가 발견되지 않거나 일치하는 채널 인증 레코드가 발견되었는데 이 레코드가 해당 채널의 사용자 ID를 사용하도록 지정하는 경우 MCAUSER 필드가 검사됩니다.
  - MCAUSER 필드가 공백이면 클라이언트 사용자 ID가 채널에 지정됩니다.
  - MCAUSER 필드가 공백이 아니면 채널에 지정됩니다.
- 임의의 보안 엑시트 프로그램이 실행됩니다. 이 엑시트 프로그램은 채널 사용자 ID를 설정하거나 액세스를 차단하도록 판별합니다.
- 연결이 차단되거나 MCAUSER가 *NOACCESS로 설정되면 채널이 종료됩니다.
- 클라이언트 채널을 제외한 어떤 채널에 대해서도 연결이 차단되지 않을 경우 이전 단계에서 판별된 채널 사용자 ID를 차단된 사용자 목록에서 확인합니다.
  - 차단된 사용자 목록에 해당 사용자 ID가 있으면 채널이 종료됩니다.
  - 차단된 사용자 목록에 해당 사용자 ID가 없으면 채널이 실행됩니다.

여러 개의 채널 인증 레코드가 채널 이름, IP 주소, 호스트 이름, 큐 관리자 이름 또는 SSL/TLS DN과 일치할 경우 가장 구체적으로 일치하는 것이 사용됩니다. 일치로 간주되는 경우는 다음과 같습니다.

- 가장 구체적인 일치는 다음과 같이 와일드카드 문자가 없는 이름입니다.
  - 채널 이름 A.B.C
  - IP 주소 192.0.2.6
  - `hursley.ibm.com` 의 호스트 이름
  - 큐 관리자 이름 192.0.2.6
- 가장 일반적인 일치는 다음과 일치하는 단일 별표(*)입니다.
  - 모든 채널 이름
  - 모든 IP 주소
  - 모든 호스트 이름
  - 모든 큐 관리자 이름
- 문자열 시작 부분에 별표가 있는 패턴이 문자열 시작 부분에 정의된 값이 있는 패턴보다 일반적입니다.
  - 채널의 경우 *.B.C가 A.*보다 일반적입니다.
  - IP 주소의 경우 *.0.2.6이 192.*보다 일반적입니다.
  - 호스트 이름의 경우, `*.ibm.com`은(는) `hursley.*`보다 일반적입니다.
  - 큐 관리자 이름의 경우 *QUEUEMANAGER가 QUEUEMANAGER*보다 일반적입니다.
- 문자열의 특정 위치에 별표가 있는 패턴이 문자열의 같은 위치에 정의된 값이 있는 패턴보다 일반적이며, 문자열의 이후 위치에 대해서도 마찬가지입니다.
  - 채널의 경우 A.*.C가 A.B.*보다 일반적입니다.
  - IP 주소의 경우 192.*.2.6이 192.0.*보다 일반적입니다.
  - 호스트 이름의 경우, `hursley.*.com`은(는) `hursley.ibm.*`보다 일반적입니다.
  - 큐 관리자 이름의 경우 Q*MANAGER가 QUEUE*보다 일반적입니다.
- 문자열의 특정 위치에 별표가 있는 패턴이 두 개 이상 있을 경우 별표 뒤의 노드 수가 적은 패턴이 더 일반적입니다.
  - 채널의 경우, A.* 는 A.*.C보다 일반적입니다.
  - IP 주소의 경우, 192 .* 는 192.* .2.*보다 일반적입니다.
  - 호스트 이름의 경우, `hurlsey.*`은(는) `hursley.*.com`보다 일반적입니다.
  - 큐 관리자 이름의 경우 Q*가 Q*MGR보다 일반적입니다.
- IP 주소의 경우:
  - 하이픈(-)으로 표시된 범위는 별표보다 구체적입니다. 따라서, 192.0.2.0-24는 192.0.2.*보다 구체적입니다.
  - 다름 범위의 서브세트인 범위는 더 큰 범위보다 구체적입니다. 따라서, 192.0.2.5-15는 192.0.2.0-24보다 구체적입니다.
  - 겹치는 범위는 허용되지 않습니다. 예를 들어, 192.0.2.0-15와 192.0.2.10-20 모두를 위한 채널 인증 레코드를 가질 수 없습니다.
  - 패턴이 단일 후미 문자 별표로 끝나지 않는 한 패턴의 부분 개수는 필요한 부분 개수보다 적어선 안됩니다. 예를 들어, 192.0.2는 올바르지 않지만 192.0.2 .* 은 유효합니다.
  - 후미 문자 별표는 적절한 부분 구분 기호(IPv4의 경우 점(.), IPv6의 경우 콜론(:))로 나머지 주소와 구분해야 합니다. 예를 들어, 192.0*는 별표가 구분되어 있지 않으므로 올바르지 않습니다.
  - 후미 별표에 인접한 별표가 없다면 패턴에 별표를 추가할 수 있습니다. 예를 들어, 192.* .2.* 는 유효하지만 192.0.*.* 유효하지 않습니다.
  - IPv6 주소 패턴에 이중 콜론 및 후미 별표가 포함되면 주소가 모호해지기 때문에 이와 같이 사용할 수 없습니다. 예를 들어, 2001::*는 2001:0000:*, 2001:0000:0000:* 등으로 확장될 수 있습니다.
- SSL이나 TLS 식별 이름(DN)의 경우 하위 문자열의 우선 순위는 다음과 같습니다.

| 표 7. 하위 문자열의 우선 순위 | | |
|---|---|---|
| **순서** | **DN 하위 문자열** | **이름** |
| 1 | SERIALNUMBER= | 인증서 일련 번호 |
| 2 | MAIL= | 이메일 주소 |
| 3 | Deprecated E= | 이메일 주소(MAIL보다 우선적으로는 더 이상 사용되지 않음) |
| 4 | UID=, USERID= | 사용자 ID |
| 5 | CN= | 공용 이름 |
| 6 | T = | 제목 |
| 7 | OU= | 조직 단위 |
| 8 | DC= | 도메인 컴포넌트 |
| 9 | 오 = | 조직 |
| 10 | STREET= | 상세 주소/주소 두 번째 줄 |
| 11 | L= | 구/군/시 |
| 12 | ST=, SP=, S= | 시/도 이름 |
| 13 | PC = | 우편번호 |
| 14 | C = | 국가 |
| 15 | UNSTRUCTUREDNAME= | 호스트 이름 |
| 16 | UNSTRUCTUREDADDRESS= | IP 주소 |
| 17 | DNQ= | 식별 이름 규정자 |

따라서, SSL 또는 TLS 인증서가 O=IBM 및 C=UK라는 하위 문자열이 포함된 DN과 함께 제시되면, IBM MQ는 C=UK 대신 O=IBM을 위한 채널 인증 레코드를 사용합니다(두 가지 모두 존재할 경우).

DN에는 여러 개의 OU가 포함될 수 있으며 이럴 경우 보다 큰 조직 단위를 먼저 지정하는 계층 구조로 OU를 지정해야 합니다. OU 값을 제외한 다른 모든 면에서 두 가지 DN이 동일하다면 보다 구체적인 DN은 다음과 같이 판별됩니다.

1. 다른 개수의 OU 속성을 갖고 있다면 가장 많은 OU 값을 가진 DN이 더 구체적입니다. 이는 더 많은 조직 단위를 가진 DN이 DN을 보다 자세하게 설명하고 더 많은 일치하는 기준을 제공하기 때문입니다. 최상위 레벨 OU가 와일드카드(OU=*)인 경우라도 OU가 더 많은 DN이 여전히 더 구체적인 것으로 간주됩니다.
2. OU 속성 개수가 동일할 경우 다음 규칙에 따라 좌우 순서로 해당되는 OU 값 쌍을 비교해 맨 왼쪽 OU가 최상위 레벨(가장 덜 구체적)이 됩니다.
   a. 와일드카드 값이 없는 OU는 정확히 한 개의 문자열과만 일치할 수 있기 때문에 가장 구체적입니다.
   b. 시작 또는 끝 부분에 한 개의 와일드카드가 있는 OU(예: OU=ABC* 또는 OU=*ABC)는 그 다음으로 가장 구체적입니다.
   c. 와일드카드가 두 개인 OU(예: OU=*ABC*)가 그 다음으로 가장 구체적입니다.
   d. 별표만으로 구성된 OU(OU=*)는 가장 덜 구체적입니다.
3. 상세도가 동일한 두 개의 속성 값 간에 문자열을 비교할 경우에는 어떤 쪽이든 더 긴 속성 문자열이 더 구체적입니다.
4. 문자열 비교가 상세도와 길이가 동일한 속성 값 두 개와 관련되어 있다면 결과는 DN의 와일드카드를 제외한 문자열 부분을 대소문자를 구분해 비교함으로써 판별됩니다.

두 개의 DN이 해당 DC 값을 제외하고 모든 면에서 동일하면 DC 값에서 가장 왼쪽의 DC가 최하위 레벨(가장 구체적)이라는 점을 제외하면 OU와 동일한 일치 규칙이 적용되고 비교 순서도 그에 따라 달라집니다.

## 채널 인증 레코드 표시

채널 인증 레코드를 표시하려면 MQSC 명령인 **DISPLAY CHLAUTH** 또는 PCF 명령인 **Inquire Channel Authentication Records**를 사용하십시오. 제공된 채널 이름과 일치하는 모든 레코드를 반환하거나 명확하게 일치하는 레코드를 선택할 수 있습니다. 명확한 일치는 채널이 특정 IP 주소, 특정 큐 관리자로부터 연결을 시도하거나 특정 사용자 ID를 사용하여 연결을 시도할 경우 사용할 채널 인증 레코드를 알려주며 선택 사항으로서 지정된 DN이 포함된 SSL/TLS 개인 인증서를 제안합니다.

**관련 개념**

94 페이지의 『원격 메시징의 보안』

이 절에서는 원격 메시징 보안 측면을 다룹니다.

## *CHLAUTH*와 *CONNAUTH*의 상호작용

채널의 단일 대화의 경우, IBM MQ에서 채널 인증 레코드(CHLAUTH) 및 연결 인증(CONNAUTH)이 상호작용 하는 방법에 대해 설명합니다.

## 바인딩의 다른 유형

IBM MQ는 애플리케이션이 연결하는 두 가지 방법을 지원합니다.

**로컬 바인딩**

애플리케이션과 큐 관리자가 동일한 운영 이미지에 있을 때 적용됩니다. CHLAUTH는 이 유형의 애플리케이션 연결과 관련이 없습니다.

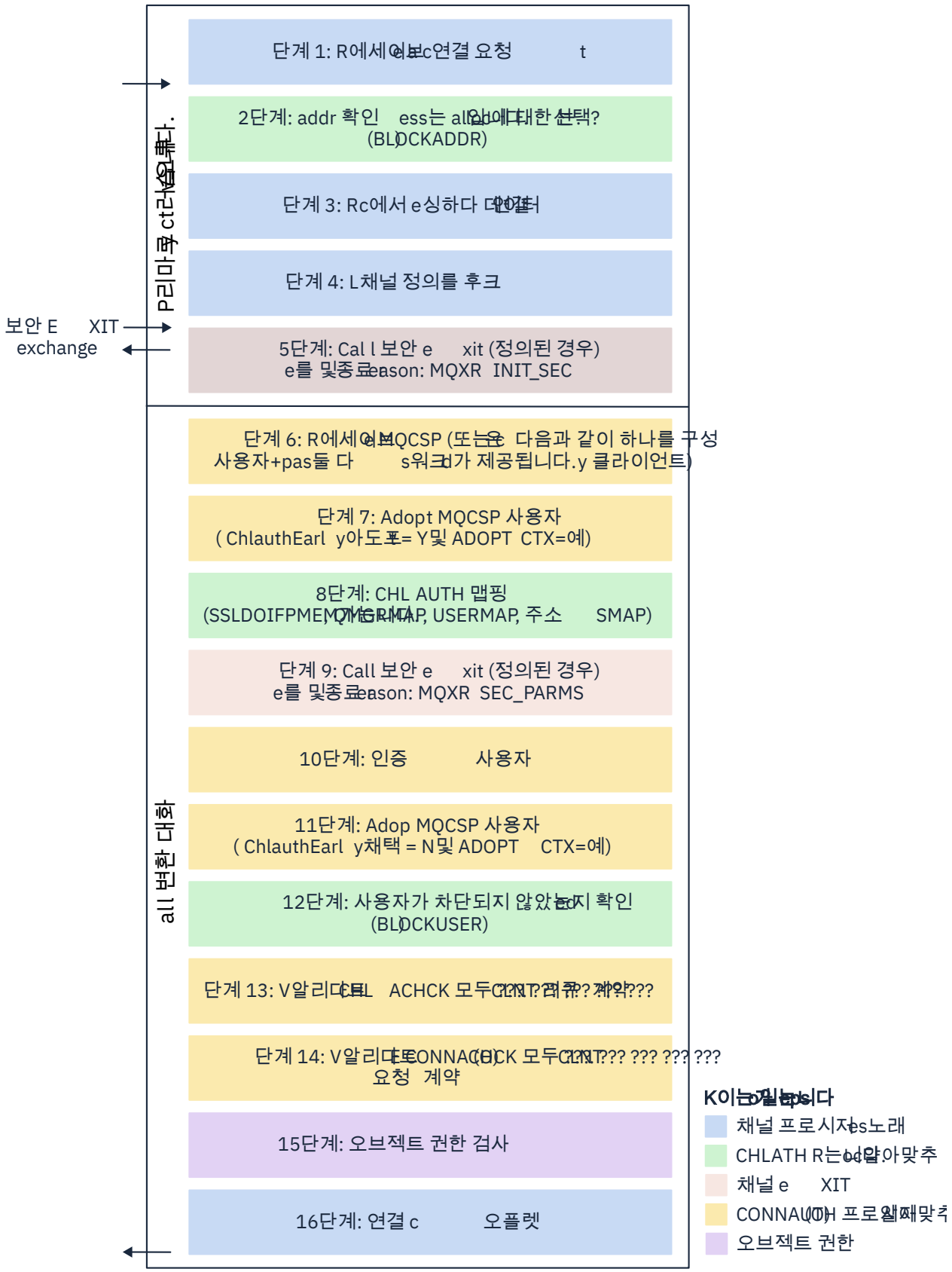
**클라이언트 바인딩**

애플리케이션과 큐 관리자가 통신하기 위해 네트워크를 사용할 때 적용됩니다. 애플리케이션과 큐 관리자는 동일한 시스템 또는 다른 시스템에서 실행될 수 있습니다. IBM MQ에서, 클라이언트 연결은 서버 연결(SVRCONN) 채널의 형태로 처리되며, 이 상황에서는 CONNAUTH와 CHLAUTH가 모두 적용됩니다.

## 채널 수신 측의 바인딩 단계

애플리케이션이 큐 관리자에 연결되면 상당한 양의 검사가 수행되어 채널의 양쪽 끝이 다른 쪽에서 지원하는 것을 이해할 수 있습니다. 채널의 수신 측은 CHLAUTH와 CONNAUTH를 포함하여 클라이언트가 연결될 수 있는지 확인하기 위해 추가 검사를 수행하며, 이 과정은 결과에 영향을 미칠 수 있으므로 보안 엑시트도 포함할 수 있습니다. 이 채널 접속 단계는 또한 결합 단계라고도 합니다.

다음 다이어그램은 서버 측(큐 관리자에서)이 시작될 때 SVRCONN 채널이 수행하는 단계를 나열합니다.



**1단계: 연결 요청 수신**
채널 시작기 또는 리스너는 네트워크의 임의 위치로부터 연결 요청을 수신합니다.

**2단계: 연결이 가능한 주소입니까?**
데이터를 읽기 전에 IBM MQ는 CHLAUTH 규칙에 대해 파트너의 IP 주소를 확인하여 주소가 *BLOCKADDR* 규칙에 있는지 검사합니다. 주소를 찾을 수 없는 경우, 차단하지 않으면 플로우는 다음 단계로 진행됩니다.

**3단계: 채널에서 데이터 읽기**
IBM MQ는 이제 데이터를 버퍼로 읽고, 송신 정보를 처리하기 시작합니다.

**4단계: 채널 정의 검색**
첫 번째 데이터 플로우에서 IBM MQ 는 무엇보다도 송신측이 시작하려고 하는 채널의 이름을 송신합니다. 수신 큐 관리자는 채널에 대해 지정된 모든 설정을 갖는 채널 정의를 검색할 수 있습니다.

**5단계: 보안 엑시트 호출(정의된 경우)**
채널에 보안 엑시트(SCYEXIT)가 정의되어 있으면 엑시트 이유(MQCXP.**ExitReason**)와 함께 호출됩니다. MQXR_INIT_SEC로 설정하십시오.

**6단계: MQCSP 수신**
필요한 경우 클라이언트가 인증 신임 정보를 제공한 경우 하나를 구성하십시오.

클라이언트가 호환 모드로 실행 중인 Java 또는 JMS 애플리케이션인 경우 클라이언트는 큐 관리자에 MQCSP 구조를 전달하지 않습니다. 대신, 애플리케이션이 사용자 ID 및 비밀번호를 제공한 경우에는 MQCSP 구조가 여기서 구성됩니다.

**7단계: MQCSP 사용자 채택(ChlauthEarlyAdopt이(가) *Y*이고 ADOPTCTX=YES인 경우)**
클라이언트가 제공하는 신임 정보가 인증됩니다.

CONNAUTH가 확인된 식별 이름을 간략한 사용자 ID에 맵핑하는 데 LDAP을 사용하고 있는 경우에는 이 맵핑이 이 단계에서 이뤄집니다.

인증이 성공하면 이 사용자 ID가 채널에 의해 채택되며 CHLAUTH 맵핑 단계에서 사용됩니다.

**참고:** IBM MQ 9.0.4에서 **ChlauthEarlyAdopt=** *Y* 매개변수는 새 큐 관리자의 qm.ini 파일의 채널 스탠자에 자동으로 추가됩니다.

**8단계: CHLAUTH 맵핑**
맵핑 규칙 *SSLPEERMAP*, *USERMAP*, *QMGRMAP* 및 *ADDRESSMAP*을 찾기 위해 CHLAUTH 캐시가 다시 조사됩니다.

수신 채널과 가장 일치하는 규칙이 사용됩니다. 규칙에 **USERSRC***(CHANNEL)* 또는 *(MAP)*이 있으면 채널이 바인딩을 지속합니다.

CHLAUTH 규칙이 **USERSRC***(NOACCESS)*를 사용하는 규칙으로 평가되는 경우, 신임 정보가 9단계에서 유효한 신임 정보로 나중에 대체되지 않는 한 애플리케이션은 채널에 대한 연결이 차단됩니다.

**9단계: 보안 엑시트 호출(정의된 경우)**
채널에 보안 엑시트(SCYEXIT)가 정의되어 있으면 엑시트 이유(MQCXP.**ExitReason**)와 함께 호출됩니다. MQXR_SEC_PARMS로 설정하십시오.

MQCSP에 대한 포인터는 MQCXP 구조의 **SecurityParms** 필드에 표시됩니다.

MQCSP 구조에는 사용자 ID(MQCSP.**CSPUserIdPtr**) 및 비밀번호(MQCSP.**CSPPasswordPtr**)에 대한 포인터가 있습니다. V 9.4.0 IBM MQ 9.3.4에서 MQCSP 구조에는 인증 토큰 (MQCSP.**TokenPtr**) 에 대한 포인터도 포함되어 있습니다.

엑시트에서 사용자 ID및 비밀번호, 인증 토큰 을 변경할 수 있습니다. 다음 예는 보안 엑시트가 감사 로그에 사용자 ID 및 비밀번호를 어떻게 기록하는지 보여줍니다.

```
if (pMQCXP -> ExitReason == MQXR_SEC_PARMS)
{
  /* It is not a good idea for security reasons to print out the user ID */
  /* and password but the following is shown for demonstration reasons */
  printf("User ID: %.*s Password: %.*s\n",
         pMQCXP -> SecurityParms -> CSPUserIdLength,
         pMQCXP -> SecurityParms -> CSPUserIdPtr,
         pMQCXP -> SecurityParms -> CSPPasswordLength,
         pMQCXP -> SecurityParms -> CSPPasswordPtr);
```

엑시트는 MQCXP에서 *MQXCC_CLOSE_CHANNEL*을 리턴하여 IBM MQ에 채널 닫기를 지시할 수 있습니다.**Exitresponse** 필드. 그렇지 않으면 채널 처리가 연결 인증 단계로 계속 진행됩니다.

**참고:** 어설션된 사용자가 보안 엑시트에 의해 변경되면 CHLAUTH 맵핑 규칙이 새 사용자에 다시 적용되지 않습니다.

**10단계: 사용자 인증**

인증 단계는 큐 관리자에서 CONNAUTH가 사용으로 설정된 경우 발생합니다.

이를 확인하려면 MQSC 명령 'DISPLAY QMGR CONNAUTH'를 실행하십시오.

z/OS 다음 예는 IBM MQ for z/OS에서 실행 중인 큐 관리자에서 **DISPLAY QMGR CONNAUTH** 명령의 출력을 표시합니다.

```
 CSQM201I !MQ25 CSQMDRTC DISPLAY QMGR DETAILS
 QMNAME(MQ25)
 CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
  END QMGR DETAILS
 CSQ9022I !MQ25 CSQMDRTC ' DISPLAY QMGR' NORMAL COMPLETION
```

Multi 다음 예는 IBM MQ for Multiplatforms에서 실행 중인 큐 관리자에서 '**DISPLAY QMGR CONNAUTH**' 명령의 출력을 표시합니다.

```
     1 : DISPLAY QMGR CONNAUTH
AMQ8408: Display Queue Manager details.
   QMNAME(DEMO)
   CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
```

CONNAUTH 값은 **AUTHINFO** IBM MQ 오브젝트의 이름입니다.

운영 체제 인증(**AUTHTYPE**(*IDPWOS*))이 IBM MQ for Multiplatforms 및 IBM MQ for z/OS 모두에서 유효하므로 예제에서는 운영 체제 인증을 사용합니다.

z/OS 다음 예는 IBM MQ for z/OS에서 실행 중인 큐 관리자에서 **AUTHTYPE**(*IDPWOS*) 가 있는 기본 AUTHINFO 오브젝트를 표시합니다.

```
 CSQM293I !MQ25 CSQMDRTC 1 AUTHINFO FOUND MATCHING REQUEST CRITERIA
 CSQM201I !MQ25 CSQMDRTC DISPLAY AUTHINFO DETAILS
 AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
 AUTHTYPE(IDPWOS)
 QSGDISP(QMGR)
 ADOPTCTX(NO)
 CHCKCLNT(NONE)
 CHCKLOCL(OPTIONAL)
 FAILDLAY(1)
 DESCR()
 ALTDATE(2018-06-04)
 ALTTIME(10.43.04)
   END AUTHINFO DETAILS
 CSQ9022I !MQ25 CSQMDRTC ' DISPLAY AUTHINFO' NORMAL COMPLETION
```

Multi 다음 예는 IBM MQ for Multiplatforms에서 실행 중인 큐 관리자에서 **AUTHTYPE**(*IDPWOS*) 가 있는 기본 AUTHINFO 오브젝트를 표시합니다.

```
 1 : display authinfo(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
AMQ8566: Display authentication information details.
   AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
   AUTHTYPE(IDPWOS)                        ADOPTCTX(NO)
   DESCR( )                                CHCKCLNT(REQDADM)
   CHCKLOCL(OPTIONAL)                      FAILDLAY(1)
   ALTDATE(2015-06-08)                     ALTTIME(16.35.16)
```

AUTHINFO TYPE (IDPWOS) 오브젝트에는 CHCKCLNT라는 속성이 있습니다. 값이 *REQUIRED* 로 변경되면 모든 클라이언트 애플리케이션이 유효한 신임 정보를 제공해야 합니다.

7단계에서 사용자가 인증된 경우 다음과 같은 경우가 아니면 다른 인증 검사가 수행되지 않습니다.

- MQCXP 구조의 SecurityParms 필드에 있는 사용자 ID및 비밀번호또는 인증 토큰 이 9단계에서 보안 엑시트에 의해 변경되었습니다.
- 클라이언트 애플리케이션이 다시 연결 가능한 기능을 요청하는 옵션과 연결되었습니다.

**11단계: MQCSP 사용자의 컨텍스트 채택(ChlauthEarlyAdopt=*N* 및 ADOPTCTX=YES인 경우)**

채널이 MCAUSER에서 실행되는지 여부 또는 애플리케이션이 제공한 사용자 ID를 제어하는 ADOPTCTX 속성을 설정할 수 있습니다.

MQCSP에서 어설션된 사용자 ID 또는 MQXCP 구조의 **SecurityParms** 필드가 성공적으로 인증되고 **ADOPTCTX** 이 예인 경우, MQCXP 구조의 **SecurityParms** 필드에 있는 사용자 ID및 비밀번호또는 인증 토큰, 가 9단계의 보안 엑시트에 의해 변경되지 않았으면 7 및 8 단계의 결과인 사용자의 컨텍스트가 이 애플리케이션에 사용할 컨텍스트로 채택됩니다.

이 확인된 사용자 ID는 IBM MQ 자원을 사용하기 위한 권한 부여에 대해 확인된 사용자 ID입니다.

예를 들어, SVRCONN 채널에 MCAUSER가 설정되어 있지 않으며 클라이언트가 Linux 시스템의 'johndoe'에서 실행 중입니다. 애플리케이션이 MQCSP에서 'fred' 사용자를 지정하여 채널이 활성 MCAUSER로 'johndoe'과(와) 함께 실행을 시작합니다. CONNAUTH를 확인하고 나면 'fred' 사용자가 채택되고 채널이 활성 MCAUSER로 'fred'과(와) 함께 실행됩니다.

**12단계: 사용자 차단 여부 검사(BLOCKUSER)**

CONNAUTH 검사가 성공하면 활성 MCAUSER가 *BLOCKUSER* 규칙에 의해 차단되는지 확인하기 위해 CHLAUTH 캐시를 다시 검사합니다. 사용자가 차단된 경우, 채널은 종료됩니다.

**13단계: CHLAUTH CHCKCLNT 요구사항 유효성 검증**

8 단계에서 선택된 CHLAUTH 규칙이 추가로 CHCKCLNT값을 REQUIRED 또는 REQDADM으로 지정하는 경우 요구사항을 충족하기 위해 유효한 CONNAUTH 사용자 ID가 제공되었는지 확인하기 위해 유효성 검증이 수행됩니다.

- CHCKCLNT (REQUIRED) 가 설정된 경우 사용자는 7 또는 10단계에서 인증되어야 합니다. 그렇지 않으면 연결이 거부됩니다.
- CHCKCLNT (REQDADM) 가 설정된 경우, 이 연결이 권한 부여된 것으로 판별되면 7 또는 10 단계에서 사용자가 인증되어야 합니다. 그렇지 않으면 연결이 거부됩니다.
- CHCKCLNT (ASQMGR) 가 설정되면 이 단계를 건너뜁니다.

**참고:**

1. CHCKCLNT (REQUIRED) 또는 CHCKCLNT (REQDADM) 가 설정되었지만 CONNAUTH가큐 관리자에서 사용으로 설정되지 않은 경우, 구성의 충돌로 인해 MQRC_SECURITY_ERROR (2063) 리턴 코드와 함께 연결이 실패합니다.
2. 사용자는 이 단계에서 다시 인증되지 않습니다.

**14단계: CONNAUTH CHCKCLNT 요구사항의 유효성 검증.**

인증 단계는 큐 관리자에서 CONNAUTH가 사용으로 설정된 경우 발생합니다.

수신 연결에 설정된 요구사항을 판별하기 위해 CONNAUTH CHCKCLNT값을 확인합니다.

- CHCKCLNT (NONE) 이 설정되면 이 단계를 건너뜁니다.
- CHCKCLNT (OPTIONAL) 가 설정되면 이 단계를 건너뜁니다.
- CHCKCLNT (REQUIRED) 가 설정되면 7 또는 10단계에서 사용자가 인증되어야 합니다. 그렇지 않으면 연결이 거부됩니다.
- CHCKCLNT (REQDADM) 가 설정된 경우, 이 연결이 권한 부여된 것으로 판별되면 7 또는 10 단계에서 사용자가 인증되어야 합니다. 그렇지 않으면 연결이 거부됩니다.

**참고:** 사용자는 이 단계에서 다시 인증되지 않습니다.

Multi **15단계: 오브젝트 권한 검사**

활성 MCAUSER에 큐 관리자에 연결하기 위한 적절한 권한이 있는지 확인하기 위해 검사가 수행됩니다.

ALW 자세한 정보는 오브젝트 권한 관리자의 내용을 참조하십시오.

IBM i 자세한 정보는 147 페이지의 『IBM i에서 오브젝트 권한 관리자』 의 내용을 참조하십시오.

**16단계: 연결 완료**
선행 단계를 완료하면, 연결이 완료됩니다.

**관련 개념**
CONNAUTH
연결할 때 애플리케이션에서 제공하는 신임 정보를 인증하도록 큐 관리자를 구성할 수 있습니다.

**관련 참조**
SET CHLAUTH
ALTER AUTHINFO

## *CHLAUTH* 액세스 문제 해결
채널 인증 레코드 (CHLAUTH) 를 사용할 때 특정 액세스 문제를 해결하기 위한 단계 및 예제입니다.

## 시작하기 전에

**참고:** 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 관리IBM MQ MQSC 명령 사용 .

## 이 태스크 정보

CHLAUTH 처리를 위한 세 개의 기본 규칙이 있습니다.

- 모든 MQ-admin* 사용자가 모든 채널에 액세스하지 못함
- 모든 SYSTEM.*에 액세스할 수 없음 .* 모든 사용자에 의한 채널
- SYSTEM.ADMIN.SVRCONN 채널에 대한 액세스 허용(비 MQ-admin 사용자)

처음 두 개의 규칙은 모든 채널에 대한 액세스를 차단합니다. 세 번째 규칙은 더 구체적이고 채널이 SYSTEM.ADMIN.SVRCONN 채널인 경우 다른 두 채널보다 우선하여 해당 채널에 대한 액세스를 허용합니다.

CHLAUTH 규칙은 채널을 시작할 수 있는지 여부를 결정하기 위해 사용되며, MCAUSER를 통해 다른 사용자 ID에 맵핑을 허용합니다. 채널을 시작할 수 없는 경우 일반적으로 다음 오류가 발생합니다.

- RC 2035 MQRC_NOT_AUTHORIZED
- RC 2059 MQRC_Q_MGR_NOT_AVAILABLE
- AMQ4036 액세스가 허용되지 않음
- AMQ9776: 채널이 사용자 ID에 의해 차단됨
- AMQ9777: 채널이 차단됨
- MQJE001: MQException 발생함: 완료 코드 2, 이유 2035
- MQJE036: 큐 관리자가 연결 시도를 거부했음

액세스를 엄격하게 차단한 다음 CHLAUTH 규칙을 추가하여 누가 채널에 액세스하고 시작할 수 있는지 제어해야 합니다.

임시 조치로 나열된 오류를 해결하려면 다음 단계를 완료하십시오.

## 프로시저

- **CHLAUTH 규칙 사용 안함**

  임시 조치로서 위의 오류를 해결하기 위해 CHLAUTH 규칙을 사용하지 않도록 설정할 수 있습니다. 언제든지 규칙을 다시 사용으로 설정할 수 있으며 CHLAUTH 규칙을 사용 안함으로 설정하면 연결 문제가 해결되는 경우 이 문제가 원인임을 알 수 있습니다.

  CHLAUTH 규칙을 사용 안함으로 설정하려면 다음 MQSC 명령을 실행하십시오.

```
ALTER QMGR CHLAUTH (DISABLED)
```

  또한 CHLAUTH를 *WARN*으로 설정하여 규칙 결과를 액세스하고 기록할 수 있다는 점에 주목하십시오.

- **CHLAUTH 규칙 수정 또는 제거**

CHLAUTH 규칙 또는 규칙을 삭제하거나 수정하여 문제를 일으킬 수도 있습니다.

CHLAUTH 규칙을 수정하려면, ACTION(REPLACE)으로 SET CHLAUTH 명령을 사용합니다. 예를 들어, MQ-admin 사용자가 모든 채널에 대해 WARN에 액세스하지 못하게 하는 기본 규칙을 수정하려면 차단되지 않고 다음 MQSC 명령을 실행하십시오.

```
SET CHLAUTH (*) TYPE (BLOCKUSER) USERLIST (*MQADMIN) WARN(YES)
ACTION (REPLACE)
```

CHLAUTH 규칙을 수정하려면, ACTION(REMOVE)으로 SET CHLAUTH 명령을 사용합니다. 예를 들어, MQ-admin 사용자가 모든 채널에 액세스하지 못하게 하는 기본 규칙을 삭제하려면 다음 MQSC 명령을 실행하십시오.

```
SET CHLAUTH (*) TYPE (BLOCKUSER) USERLIST (*MQADMIN) ACTION (REMOVE)
```

- **MATCH를 사용하여 액세스 테스트 (RUNCHECK)**

  CHLAUTH 규칙의 MATCH (*RUNCHECK*) 옵션을 사용하여 CHLAUTH 규칙의 결과를 테스트할 수 있습니다. **MATCH**(*RUNCHECK*) 옵션은 해당 채널이 이 큐 관리자에 연결되는 경우 런타임 시에 특정 인바운드 채널이 일치하는 레코드를 리턴합니다. 다음을 제공해야 합니다.

  - 채널 이름
  - ADDRESS 속성
  - SSLPEER 속성, 인바운드 채널이 SSL 또는 TLS를 사용하는 경우에만
  - QMNAME, 인바운드 채널이 큐 관리자 채널인 경우 또는
  - CLNTUSER 속성, 인바운드 채널이 클라이언트 채널인 경우

  다음 예에서는 MQSC 명령을 실행하여 기본 규칙이 있는 CHLAUTH 규칙으로 인해 MQ-admin 사용자 johndoe 가 CHAN1이라는 채널에 액세스하는지 확인합니다.

```
DISPLAY CHLAUTH (CHAN1) MATCH (RUNCHECK) CLNTUSER ('johndoe') ADDRESS
('192.168.1.138')

AMQ8878: Display channel authentication record details.
CHLAUTH(*) TYPE(BLOCKUSER)
USERLIST(*MQADMIN)
```

  사용자 johndoe의 경우, *MQADMIN 사용자에 대한 BLOCKUSER 규칙으로 인해 차단됩니다.

  다음 예에서는 MQSC 명령을 실행하여 기본 규칙이 있는 CHLAUTH 규칙으로 인해 MQ-admin 사용자가 아닌 alice 사용자가 CHAN1이라는 채널에 액세스하는지 확인합니다.

```
DISPLAY CHLAUTH (CHAN1) MATCH (RUNCHECK) CLNTUSER ('alice') ADDRESS
('192.168.1.138')

AMQ9783: Channel will run using MCAUSER('alice').
```

  사용자 alice의 경우, 채널이 실행되고, 채널은 MCAUSER로 alice를 전달합니다. MCAUSER는 IBM MQ 오브젝트 권한을 확인하는 데 사용되는 사용자 ID입니다.

**관련 참조**

SET CHLAUTH

표시

사용자를 위한 새 *CHLAUTH* 규칙 작성

사용자에 대한 일부 공통 시나리오 및 이를 수행하기 위한 예제 CHLAUTH 규칙입니다.

## 시작하기 전에

**참고:** 이 태스크의 단계에서는 MQSC 명령을 실행해야 합니다. 이를 수행하는 방법은 플랫폼에 따라 다릅니다. 보다 관리IBM MQ MQSC 명령 사용 .

## 이 태스크 정보

CHLAUTH 처리를 위한 세 개의 기본 규칙이 있습니다.

- 모든 MQ-admin* 사용자가 모든 채널에 액세스하지 못함
- 모든 SYSTEM.*에 액세스할 수 없음 .* 모든 사용자에 의한 채널
- SYSTEM.ADMIN.SVRCONN 채널에 대한 액세스 허용(비 MQ-admin 사용자)

처음 두 개의 규칙은 모든 채널에 대한 액세스를 차단합니다. 세 번째 규칙은 더 구체적이고 채널이 SYSTEM.ADMIN.SVRCONN 채널인 경우 다른 두 채널보다 우선하여 해당 채널에 대한 액세스를 허용합니다.

사용자에 대한 새 CHLAUTH 규칙을 작성하려면 다음 시나리오 중 하나 이상을 구성하십시오.

## 프로시저

- **특정 MQ-admin 사용자에 대한 액세스 제어**

  a) 관리 관점에서 독점적으로 사용되는 (즉, IBM MQ Explorer에서 연결하기 위해) 서버 연결 채널을 설정하십시오.

  이 사용법을 위한 특정 채널이 있고 연결이 지정된 IP 주소 중 하나가 아닌 경우, 연결을 허용하려는 위치의 IP 주소를 정의하고 'mqm' ID에 대해 액세스를 차단합니다.

  b) ADMIN.CHAN라는 IBM MQ Explorer 및 MQ-admin 사용자를 위한 SVRCONN 채널을 작성하십시오.

  다음 MQSC 명령을 실행합니다.

```
DEFINE CHANNEL (ADMIN.CHAN) CHLTYPE (SVRCONN) TRPTYPE (TCP)
```

  c) 테스트의 경우, MQ-admin 그룹에 속한 사용자 정의와 그렇지 않은 사용자가 있는지 확인하십시오.

  이 시나리오의 경우, mqadm은 MQ-admin 그룹에 있고 alice는 포함되어 있지 않습니다.

  d) 기본 CHLAUTH 규칙 이 제 위치에 있는지 확인하십시오.

  e) 특정 사용자가 특정 IP 주소에서 MQ-admin으로 ADMIN.CHAN에 액세스할 수 있도록 허용하려면 세 개의 규칙을 추가하십시오.

  - 주소에서 NOACCESS 설정
  - *MQADMIN BLOCKUSER를 대체하는 사용자 nobody만 차단하도록 이 채널의 BLOCKUSER 설정
  - 주소의 특정 서브넷에서 사용자 mqadm에 대한 액세스를 허용하고 mqadm 사용자 권한에 대해 맵핑

  이를 수행하려면 다음 MQSC 명령을 실행하십시오.

```
SET CHLAUTH (ADMIN.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
SET CHLAUTH('ADMIN.CHAN') TYPE(BLOCKUSER) +
DESCR('Rule to override *MQADMIN blockuser on this channel') +
USERLIST('nobody') ACTION(replace)
SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
CLNTUSER('mqadm') USERSRC(MAP) MCAUSER('mqadm') +
ADDRESS('192.168.1.*') +
DESCR('Allow mqadm as mqadm on local subnet') ACTION(ADD)
```

  이 시점에서 사용자 mqadm은 지정된 IP 주소 범위에서, ADMIN.CHAN 채널을 액세스하고 시작할 수 있습니다.

  f) 옵션: 언제든지 MQSC 명령 MATCH (RUNCHECK) 를 실행하여 다음 각 명령의 결과를 볼 수 있습니다.

```
DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('mqadm') ADDRESS
('192.168.1.138')
AMQ8878: Display channel authentication record details.
```

```
CHLAUTH(ADMIN.CHAN) TYPE(USERMAP)
ADDRESS(192.168.1.*) CLNTUSER(mqadm)
MCAUSER(mqadm)
```

```
DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('alice') ADDRESS
('192.168.1.138')
AMQ8878: Display channel authentication record details.
CHLAUTH(ADMIN.CHAN) TYPE(ADDRESSMAP)
ADDRESS(*) USERSRC(NOACCESS)
```

이 시점에서, CHLAUTH 레코드가 있는 사용자만 ADMIN.CHAN을 사용하여 액세스 할 수 있습니다.

- **특정 사용자 및 IBM MQ 클라이언트 애플리케이션에 대한 액세스 제어**

  이 시나리오의 경우 기본 CHLAUTH 규칙 은 특정 사용자에 대해 IBM MQ 권한을 설정해야 한다고 가정하고 setmqaut를 사용하여 올바른 IBM MQ 권한을 제공하는 것이 적합합니다.

  이 사나리오에서, 권한은 `MQ-admin` 사용자가 아닌 사용자 `mqapp1`에 대해 설정됩니다.

  a) 다음 MQSC 명령을 사용하여 SVRCONN 채널, APP1.CHAN-특정 애플리케이션 및 특정 사용자가 사용합니다.

```
DEFINE CHANNEL (APP1.CHAN) CHLTYPE (SVRCONN) TRPTYPE (TCP)
```

  b) 적절한 위치의 기본 CHLAUTH 규칙으로 사용자 `mqapp1`은 APP1.CHAN 채널을 시작할 수 있습니다.

  IBM MQ 클라이언트 애플리케이션에서 발생하는 사용자 ID는 IBM MQ 오브젝트 권한 확인에 사용됩니다. 이 경우 `mqapp1` 사용자가 IBM MQ 클라이언트 앱을 실행 중이라고 가정하면 이는 IBM MQ 오브젝트 권한 검사에 사용됩니다. 그러므로, `mqapp1`이 애플리케이션에 필요한 IBM MQ 오브젝트에 접근할 수 있다면, 아무런 문제가 없으며, 그렇지 않을 경우 권한 오류가 발생합니다.

  `mqapp1` 사용자 ID에 대한 특정 CHLAUTH 규칙을 생성하여 보안을 강화할 수 있지만, 기본 규칙에서 `MQ-admin` 그룹의 멤버는 이 채널에 액세스할 수 없습니다.

  다음 MQSC 명령을 실행하십시오.

```
SET CHLAUTH (APP1.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
SET CHLAUTH('APP1.CHAN') TYPE(USERMAP) +
CLNTUSER('mqapp1') USERSRC(MAP) MCAUSER('mqapp1') +
DESCR('Allow mqapp1 as mqapp1 on local subnet') ACTION(ADD)
```

- **해당 사용자의 인증서 식별 이름 (DN) 을 사용하여 특정 사용자에 대한 액세스 제어**

  이 시나리오의 경우 사용자는 큐 관리자에 플로우되는 인증서가 있어야 합니다. DN은 CHLAUTH 규칙의 SSLPEER 설정과 일치하며, SSLPEER는 와일드카드 문자를 사용할 수 있습니다.

  일치하는 경우, 사용자는 IBM MQ 오브젝트 권한을 확인하기 위해 다른 MCAUSER에 맵핑 될 수도 있습니다. MCAUSER 맵핑은 IBM MQ 오브젝트 권한 관리자(OAM)에서 관리해야 하는 사용자 수를 최소화 할 수 있습니다.

  a) 사용중인 인증서가 있는 TLS 채널이 있으며 다음과 같은 규칙이 필요합니다.

  - 특정 채널에 대한 모든 사용자 차단
  - IBM MQ OAM 액세스에 대해 해당 사용자의 클라이언트를 사용하는 특정 SSLPEER이 있는 사용자만 허용하십시오.

  다음 MQSC 명령을 실행하십시오.

```
.
# block all users on any IP address.
SET CHLAUTH('SSL1.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*')
USERSRC(NOACCESS) DESCR(''block all'') WARN(NO) ACTION(ADD)
.
# override - no MQM admin rule (allow mqm group /mqm admin users to
connect.
SET CHLAUTH('SSL1.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody')
DESCR('override no mqm admin rule') WARN(NO) ACTION(ADD)
.
```

```
# allow particular SSLPEER, use client id coming in from channel
SET CHLAUTH('SSL1.SVRCONN') TYPE(SSLPEERMAP)
SSLPEER('CN=JOHNDOE,O=IBM,C=US') USERSRC(CHANNEL) ACTION(ADD)
```

채널에 연결하는 클라이언트 사용자 ID는 IBM MQ 오브젝트의 IBM MQ OAM 권한에 사용되므로 사용자 ID는 적절한 IBM MQ 권한을 가져야 합니다.

b) 옵션: 다른 IBM MQ 사용자 ID에 맵핑하십시오.

USERSRC(CHANNEL)를 USERSRC(MAP) MCAUSER('mquser1') 로 대체하여 이전 MQSC 명령을 다시 실행하십시오.

- **특정 사용자를 mqm 사용자에 맵핑**

  이는 특정 MQ-admin 사용자에 대한 액세스 제어에 대한 추가 또는 수정입니다.

  MQSC 명령을 사용하여 다음 CHLAUTH 규칙을 추가하여 IBM MQ OAM에서 IBM MQ 오브젝트 권한이 설정된 mqm 사용자 또는 MQ-admin 사용자 ID에 특정 사용자를 맵핑하십시오.

```
SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
CLNTUSER ('johndoe') USERSRC(MAP) MCAUSER ('mqm') +
ADDRESS('192.168.1-100.*') +
DESCR ('Allow johndoe as MQ-admin on local subnet') ACTION (ADD)
```

이는 특정 채널 ADMIN.CHAN을 위한 mqm 사용자에 대해 johndoe 사용자를 허용하고 맵핑합니다.

**관련 개념**

62 페이지의 『채널에 대한 새 CHLAUTH 규칙 작성』
사용자 고유의 CHLAUTH 규칙을 작성하는 데 도움을 주기 위해 채널에 대한 몇 가지 공통 시나리오 및 이를 이루기 위한 CHLAUTH 규칙 예제를 다음과 같이 제공합니다.

**관련 태스크**

58 페이지의 『CHLAUTH 액세스 문제 해결』
채널 인증 레코드 (CHLAUTH) 를 사용할 때 특정 액세스 문제를 해결하기 위한 단계 및 예제입니다.

**관련 참조**

SET CHLAUTH

표시

## 채널에 대한 새 *CHLAUTH* 규칙 작성

사용자 고유의 CHLAUTH 규칙을 작성하는 데 도움을 주기 위해 채널에 대한 몇 가지 공통 시나리오 및 이를 이루기 위한 CHLAUTH 규칙 예제를 다음과 같이 제공합니다.

이 주제는 다음 시나리오를 포함합니다.

- 62 페이지의 『특정 IP 주소 범위에서 특정 채널에 대한 액세스만 허용하십시오.』
- 63 페이지의 『특정 채널의 경우 모든 사용자를 차단하지만 특정 사용자가 연결할 수 있도록 허용하십시오.』
- 63 페이지의 『수신자 및 송신자 채널에 대해 CHLAUTH 사용』

### 특정 IP 주소 범위에서 특정 채널에 대한 액세스만 허용하십시오.

이 시나리오의 경우 다음과 같이 수행하십시오.

- 임의 위치에서 채널에 대한 액세스 없음 설정
- 특정 IP 주소 또는 주소 범위에서 액세스 허용

```
runmqsc:
SET CHLAUTH('APP2.CHAN') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
WARN(NO) ACTION(ADD)
SET CHLAUTH('APP2.CHAN') TYPE(ADDRESSMAP) ADDRESS('9.95.100.1-5')
USERSRC(MAP) MCAUSER('mqapp2') ACTION(ADD)
```

이것은 연결이 지정된 특정 IP 주소 범위로부터 오는 경우 APP2.CHAN 채널만 시작할 수 있도록 허용합니다.

MCAUSER로 연결하는 사용자는 mqapp2에 맵핑되며, 따라서 해당 사용자에 대해 IBM MQ OAM 권한을 부여 받습니다.

### 특정 채널의 경우 모든 사용자를 차단하지만 특정 사용자가 연결할 수 있도록 허용하십시오.

CHLAUTH 처리를 위한 세 개의 기본 규칙이 있습니다.

- 모든 MQ-admin* 사용자가 모든 채널에 액세스하지 못함
- 모든 SYSTEM.*에 액세스할 수 없음 .* 모든 사용자에 의한 채널
- SYSTEM.ADMIN.SVRCONN 채널에 대한 액세스 허용(비 MQ-admin 사용자)

처음 두 개의 규칙은 모든 채널에 대한 액세스를 차단합니다. 세 번째 규칙은 더 구체적이고 채널이 SYSTEM.ADMIN.SVRCONN 채널인 경우 다른 두 채널보다 우선하여 해당 채널에 대한 액세스를 허용합니다.

이 시나리오의 경우 MY.SVRCONN 채널에 대한 액세스에는 기본 CHLAUTH 규칙이 있습니다.

다음을 추가해야 합니다.

```
# block all users
SET CHLAUTH('MY.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR(''block all'') WARN(NO) ACTION(ADD)

# override - no MQM admin rule
SET CHLAUTH('MY.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody') DESCR('override
no mqm admin rule') WARN(NO) ACTION(ADD)

# allow johndoe userid
SET CHLAUTH('MY.SVRCONN') TYPE(USERMAP) CLNTUSER('johndoe')
USERSRC(CHANNEL) DESCR('allow johndoe userid') ACTION(ADD)
```

이 코드의 첫 부분은 모든 사용자의 MY.SVRCONN에 대한 연결을 차단한 다음, 특정 사용자 ID johndoe에서 연결이 이루어질 때 MY.SVRCONN 채널만 시작할 수 있도록 허용합니다.

johndoe 채널에 연결하는 사용자는 IBM MQ 오브젝트의 IBM MQ OAM 권한에 사용됩니다. 그러므로 사용자 ID는 적절한 IBM MQ 권한이 있어야 합니다.

원하는 경우 다음을 사용하여 다른 IBM MQ 사용자 ID에 맵핑할 수 있습니다.

```
USERSRC(MAP) MCAUSER('mquser1')
```

instead of USERSRC(CHANNEL).

### 수신자 및 송신자 채널에 대해 CHLAUTH 사용

CHLAUTH 규칙을 사용하여 수신자 채널에 대한 액세스를 제한하려면 수신자 및 송신자 채널에 추가 보안을 추가할 수 있습니다. CHLAUTH 규칙을 추가하거나 변경하는 경우, 업데이트된 CHLAUTH 규칙은 채널을 시작할 때만 적용되므로 채널이 이미 실행중인 경우, CHLAUTH 업데이트를 적용하려면 중지하고 다시 시작해야 합니다.

CHLAUTH 규칙은 모든 채널에 사용될 수 있지만 일부 제한이 있습니다. 예를 들면, USERMAP 규칙은 SVRCONN 채널에만 적용됩니다.

이 예는 TO.MYSVR1 채널을 시작하기 위해 특정 IP 주소로에서의 연결만 허용합니다.

```
# First you could lock down the channel by disallowing all
# for channel 'TO.MYSVR1', RCVR channel
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')

# Then you could allow this channel to be started
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('192.168.1.134') USERSRC(MAP)
MCAUSER('mqapp') ACTION(ADD)
```

이 예는 특정 큐 관리자만 연결을 허용합니다.

```
# Lock down all access:
```

```
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')

# Then allow access from queue manager MYSVR2 and from a particular ipaddress:
SET CHLAUTH('TO.MYSVR1') TYPE(QMGRMAP) QMNAME('MYSVR2') USERSRC(MAP)
MCAUSER('mqapp') ADDRESS('192.168.1.134') ACTION(ADD)
```

**관련 태스크**

58 페이지의 『CHLAUTH 액세스 문제 해결』
채널 인증 레코드 (CHLAUTH) 를 사용할 때 특정 액세스 문제를 해결하기 위한 단계 및 예제입니다.

59 페이지의 『사용자를 위한 새 CHLAUTH 규칙 작성』
사용자에 대한 일부 공통 시나리오 및 이를 수행하기 위한 예제 CHLAUTH 규칙입니다.

**관련 참조**

SET CHLAUTH

표시

*CHLAUTH* 백스톱 규칙 작성
큐 관리자에 대한 인바운드 연결을 제어하려 할 때는 두 가지 옵션이 있습니다. 허용되지 않는 모든 연결을 나열하거나, 모든 연결을 허용하지 않은 후 허용되는 연결을 나열하는 것입니다. 여기서는 두 번째 옵션을 설명합니다.

## 이 태스크 정보

두 번째 옵션을 사용하는 이유는, 허용되지 않는 모든 연결을 나열하려 했으나 누락된 항목이 있어 연결이 허용되는 경우, 목록에서 누락된 항목으로 인해 허용하지 않아야 할 연결이 연결되어 잠재적 보안 침해를 발생시킬 수 있기 때문입니다.

반대로 처음부터 모든 연결을 허용하지 않은 후 허용되는 연결을 나열하면, 이 목록에서 항목이 누락되어도 보안 침해가 발생하지 않습니다. 엔터프라이즈에서 추가 연결을 목록에 추가해야 하는 경우 이는 비교적 간단한 작업이지만, 잠재적 보안 침해 위험은 없습니다.

가장 먼저 할 일은 더 구체적인 규칙과 일치하지 않는 연결을 포착하는 안전망*(back-stop)* 규칙을 작성하는 것입니다. 이 규칙은 모든 원격 연결이 사용자의 큐 관리자에 전혀 연결할 수 없도록 하는 효과가 있습니다.

그러나 이 접근법이 과도하다고 생각하는 경우에는 경고 모드의 안전망 규칙을 설정할 수 있습니다. 64 페이지의 『2』 단계를 참조하십시오.

## 프로시저

1. 큐 관리자에 연결하려는 원격 연결을 차단하는 안전망 규칙을 작성하려는 경우에는 다음 명령을 실행하십시오.

   ```
   SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*')
   USERSRC(NOACCESS) DESCR('Back-stop rule')
   ```

   이제 모든 원격 연결을 차단했으므로, 특정 연결을 허용하는 더 구체적인 규칙을 배치하기 시작할 수 있습니다. 예를 들면, 다음과 같습니다.

   ```
   SET CHLAUTH('APPL1.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('9.20.1-3.*') USERSRC(CHANNEL)
   SET CHLAUTH('SYSTEM.ADMIN.*') TYPE(SSLPEERMAP) SSLPEER('O=IBM') USERSRC(CHANNEL)
   SET CHLAUTH('TO.QM2') TYPE(QMGRMAP) QMNAME('QM1') USERSRC(MAP) MCAUSER('QM1USER')
   SET CHLAUTH('*.SVRCONN') TYPE(USERMAP) CLNTUSER('johndoe') MCAUSER('johndoe@yourdomain')
   SET CHLAUTH('*') TYPE(SSLPEERMAP) SSLPEER('CN="John Doe"') ADDRESS('9.*') MCAUSER('johndoe')
   ```

2. 경고 모드의 안전망 규칙을 작성하려면 다음 명령을 실행하십시오.

   ```
   SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*')
   USERSRC(NOACCESS) DESCR('Back-stop rule') WARN(YES)
   ```

   이제 계속해서 모든 양성 규칙(postive rule)을 작성할 수 있습니다. 필요한 모든 규칙을 작성했으면 다음 명령을 실행하여 채널 이벤트를 켜십시오.

```
ALTER QMGR CHLEV(EXCEPTION)
```

그리고 SYSTEM.ADMIN.CHANNEL.EVENT 큐에서 **Reason**이 MQRC_CHANNEL_BLOCKED_WARNING으로 설정된 이벤트를 모니터하십시오.

이러한 이벤트는 안전망 규칙과 일치한 연결을 자세히 설명하지만, 이 명령이 경고 모드로 실행되고 있으므로 현재 실제로 차단된 것은 아닙니다.

이러한 각 이벤트를 검토하고 이 연결을 허용하기 위해 양성 규칙을 배치해야 하는지, 또는 이것이 안전망 규칙과 올바르게 일치한 것인지 판별하십시오. 사용자는 모든 인바운드 채널을 확인하고 이들에 대해 모든 양성 규칙이 배치될 때까지 이 모드를 실행하면서 작성되는 이벤트를 검토할 수 있습니다.

이 시점에서, 사용자는 다음 명령을 실행하여 일치하는 연결을 실제로 차단하도록 안전망 규칙을 변경할 수 있습니다.

```
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*')
USERSRC(NOACCESS) DESCR('Back-stop rule') WARN(NO)
ACTION(REPLACE)
```

권한이 없는 *IBM MQ* 관리자 작성
CHLAUTH를 사용하여 권한이 없는 IBM MQ 관리자를 작성하는 방법을 설명합니다.

## 이 태스크 정보

이 태스크의 컨텍스트에서는 다음 용어가 사용됩니다.

**권한이 있는 사용자**
조작을 수행할 수 있는 액세스 권한이 명시적으로 부여되지 않아도 해당 조작을 수행할 권한이 있는 사용자를 의미합니다. mqm 그룹의 사용자는 이러한 권한이 있는 사용자의 예입니다.

***IBM MQ* 관리자**
IBM MQ에 대해 관리 명령을 실행해야 하는 사용자를 의미합니다 (예: **DEFINE QLOCAL** 또는 **START CHANNEL**).

다음 단계는 권한이 없는 IBM MQ 관리자를 작성합니다.

## 프로시저

1. 엔터프라이즈에서 사용하는 플랫폼에 적합한 명령을 사용하여 큐 관리자 시스템에 사용자 ID를 작성하십시오.
   이 예에서는 사용자 이름 `alice`가 사용됩니다.
2. 다음 프로시저를 수행하여 이 새 사용자에게 모든 IBM MQ 관리 명령을 실행할 수 있는 권한을 부여하십시오.
   a) 권한이 있는 사용자를 사용하여 IBM MQ Explorer를 시작하십시오.
   b) 적절한 큐 관리자를 선택한 후 오브젝트 권한 및 역할 기반 권한 추가를 선택하여 역할 기반 마법사로 이동하십시오.
   c) 팝업되는 마법사 패널에서, 첫 번째 단계에서 작성한 사용자 ID를 입력하거나 그룹에 대해 작업하려는 경우 권한이 없는 IBM MQ 관리자로 설정할 사용자 또는 사용자 세트의 그룹 이름을 입력하십시오.
   d) 전체 관리 액세스 권한에 대한 마법사를 설정하십시오.
   e) 권한이 없는 IBM MQ 관리자가 큐에서 메시지를 찾아볼 수 있도록 허용하려면 해당 선택란도 선택하십시오.
   f) 마법사의 맨 아래에 있는 미리 보기 패널에서 명령을 검토하십시오.
   이러한 명령을 잘라내어 붙여넣어서 자체 스크립트를 빌드할 수 있습니다.

   자체 스크립트로 이를 수행하려는 한 가지 이유는 이 사용자에게 부여하는 액세스의 양을 줄이기 위해서입니다. 모든 오브젝트에 대한 액세스를 부여하는 것보다 특정 오브젝트 그룹에 대한 액세스만 부여하는 것을 선호할 수 있습니다.

   마법사에서 **확인**을 누르면 표시된 대로 명령이 실행됩니다.

g) 권한이 없는 IBM MQ 관리자에게도 원격 액세스가 필요한 경우 이 허용에 대해 원격 액세스를 허용하도록 일부 CHLAUTH 규칙을 설정해야 합니다.

엔터프라이즈에서 64 페이지의 『CHLAUTH 백스톱 규칙 작성』의 지침을 사용하고 있다고 가정하면 사용 가능 규칙을 추가하기만 하면 됩니다.

작성하는 규칙은 원격 IBM MQ 관리자를 인증하도록 선택하는 방법에 따라 다릅니다.

취약한 TCP/IP 인증을 사용하는 경우 다음과 같은 CHLAUTH 규칙을 설정할 수 있습니다.

```
  SET CHLAUTH(admin-channel-name)      TYPE(ADDRESSMAP)
ADDRESS('1.2.3.4')      USERSRC(MAP) MCAUSER('alice')
DESCR('Admin Channel - Weak TCP/IP authentication')
```

9. TLS 인증을 사용하는 경우 다음과 같은 CHLAUTH 규칙을 설정할 수 있습니다.

```
SET CHLAUTH(admin-channel-name)      TYPE(SSLPEERMAP)
SSLPEER('CN=Alice') ADDRESS('1.2.3.4')      USERSRC(MAP) MCAUSER('alice')
DESCR('Admin Channel - TLS authentication'
```

이제 사용자가 `admin-channel-name`에 연결하고 CHLAUTH 규칙을 일치시키면 큐 관리자에서 사용자 ID `alice`로 명령을 실행할 수 있으므로 권한이 있는 원격 액세스가 필요하지 않습니다.

## 연결 인증

연결 인증을 사용하면 애플리케이션이 큐 관리자에 연결할 때 인증 신임 정보를 제공할 수 있습니다. 큐 관리자가 신임 정보의 유효성을 검증합니다. 신임 정보에 제공된 사용자 ID는 애플리케이션이 액세스하는 자원에 대한 권한 검사에서 사용하기 위해 채택될 수도 있습니다.

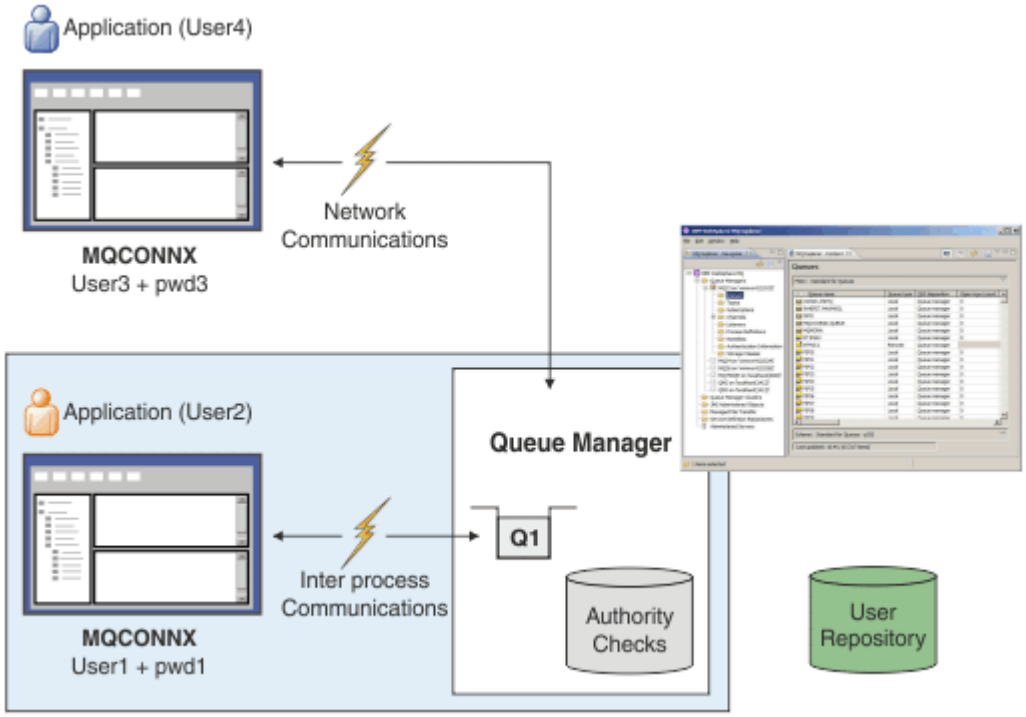
애플리케이션은 큐 관리자에 연결할 때 인증을 위해 사용자 ID및 비밀번호를 제공할 수 있습니다.

V 9.4.0 IBM MQ 9.3.4부터 IBM MQ client 애플리케이션은 인증의 대체 방법으로 인증 토큰을 제공할 수도 있습니다.

애플리케이션에서 제공하는 신임 정보의 유효성을 검증하도록 큐 관리자를 구성할 수 있습니다.

애플리케이션에서 제공하는 사용자 ID및 비밀번호는 큐 관리자 구성의 사용자 저장소를 사용하여 검사합니다. 사용자 ID및 비밀번호를 검사하는 데 사용되는 저장소에 대한 자세한 정보는 사용자 저장소를 참조하십시오.

V 9.4.0 인증 토큰은 토큰의 서명을 유효성 검증하기 위해 큐 관리자의 토큰 인증 키 저장소에 있는 인증서 및 대칭 키를 사용하여 유효성 검증됩니다. 인증 토큰을 사용하여 사용자를 인증하는 방법에 대한 자세한 정보는 305 페이지의 『인증 토큰에 대한 작업』의 내용을 참조하십시오.




다이어그램에서 하나의 애플리케이션은 클라이언트로서, 또 하나의 애플리케이션은 로컬 바인딩을 사용하여 총 두 개의 애플리케이션이 큐 관리자와 연결합니다. 애플리케이션은 다양한 API를 사용하여 큐 관리자에 연결할 수 있지만 모두 사용자 ID및 비밀번호를 제공할 수 있습니다. 다이어그램에서 애플리케이션을 실행 중인 사용자 ID인 `User2` 및 `User4`은 IBM MQ에 제공되는 일반적인 운영 체제 사용자 ID이고, 애플리케이션이 제공하는 사용자 ID인 `User1` 및 `User3`와는 다를 수도 있습니다.

큐 관리자는 구성 명령(다이어그램에서 IBM MQ Explorer가 사용 중임)을 수신하고 자원의 열기를 관리하고 이러한 자원에 액세스하기 위한 권한을 검사합니다. IBM MQ에는 애플리케이션이 액세스할 권한을 필요로 할 수도 있는 여러 가지 자원이 있습니다. 다이어그램은 출력을 위한 큐 열기를 보여주지만 동일한 원리가 또 다른 자원에도 적용됩니다.

**관련 개념**

67 페이지의 『연결 인증: 구성』
연결할 때 애플리케이션에서 제공하는 신임 정보를 인증하도록 큐 관리자를 구성할 수 있습니다.

72 페이지의 『연결 인증: 애플리케이션 변경사항』

73 페이지의 『연결 인증: 사용자 저장소』
각 큐 관리자에 대해 사용자 ID 및 비밀번호를 인증하기 위해 서로 다른 유형의 인증 정보 오브젝트를 선택할 수 있습니다.

## 연결 인증: 구성

연결할 때 애플리케이션에서 제공하는 신임 정보를 인증하도록 큐 관리자를 구성할 수 있습니다.

### 큐 관리자에서 연결 인증 켜기

큐 관리자 오브젝트에서 **CONNAUTH** 속성은 인증 정보(AUTHINFO) 오브젝트의 이름으로 설정될 수 있습니다. AUTHINFO 오브젝트의 **AUTHTYPE** 속성은 오브젝트의 유형을 지정합니다. 연결 인증에 사용되는 AUTHINFO 오브젝트는 다음 두 가지 유형 중 하나일 수 있습니다.

**IDPWOS**

큐 관리자는 로컬 운영 체제를 사용하여 연결 애플리케이션에서 제공하는 사용자 ID및 비밀번호를 인증합니다.

Linux AIX V 9.4.0 IBM MQ 9.3.4부터 이 유형의 AUTHINFO 오브젝트는 AIX 또는 Linux 에서 실행되는 큐 관리자가 인증 토큰의 유효성을 검증하도록 허용합니다. 연결 인증을 구성하는 데 사

용되는 AUTHINFO 오브젝트 외에도 qm.ini 파일의 **AuthInfo** 스탠자를 사용하여 인증 토큰을 승인하도록 큐 관리자를 구성해야 합니다. 인증 토큰을 승인하도록 큐 관리자를 구성하는 방법에 대한 자세한 정보는 312 페이지의 『로컬 키 저장소를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』의 내용을 참조하십시오.

**IDPWLDAP**
큐 관리자는 LDAP 서버를 사용하여 연결 애플리케이션에서 제공하는 사용자 ID및 비밀번호를 인증합니다.

**참고:** 큐 관리자의 **CONNAUTH** 속성에 다른 유형의 인증 정보 오브젝트를 지정할 수 없습니다.

IDPWOS 및 IDPWLDAP 유형의 AUTHINFO 오브젝트는 여러 속성에서 유사합니다. 여기에 설명된 속성은 두 유형의 오브젝트 모두에 공통입니다.

다음 예제 MQSC 명령은 다음 조작을 사용하여 연결 인증을 켭니다.

1. USE.PW라는 AUTHINFO 오브젝트를 정의하십시오.
2. 이 AUTHINFO 오브젝트를 참조하도록 큐 관리자 **CONNAUTH** 속성을 대체하십시오.
3. **REFRESH SECURITY** 명령을 실행하여 큐 관리자의 연결 인증 구성을 새로 고치십시오. 큐 관리자가 연결 인증 구성에 대한 변경사항을 인식하기 전에 **REFRESH SECURITY** 명령을 실행해야 합니다.

```
DEFINE AUTHINFO(USE.PW) +
AUTHTYPE(IDPWOS) +
FAILDLAY(10) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED)

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

로컬로 바인드된 애플리케이션이 작성하는 연결에 대해 신임 정보가 검사되는지 여부를 제어하려면 AUTHINFO 속성 **CHCKLOCL** (로컬 연결 검사) 을 사용하십시오. 클라이언트 애플리케이션이 작성한 연결에 대해 신임 정보가 검사되는지 여부를 제어하려면 AUTHINFO 속성 **CHCKCLNT** (클라이언트 연결 검사) 를 사용하십시오.

**CHCKLOCL** 는 NONE 및 OPTIONAL값을 승인하고, **CHCKCLNT** 는 구성될 인증 요구사항에 대해 NONE 값을 허용합니다.

**NONE**
애플리케이션에서 제공하는 인증 신임 정보는 확인되지 않습니다.

**OPTIONAL**
애플리케이션에서 제공하는 신임 정보가 유효한지 확인합니다. 그러나 애플리케이션이 반드시 인증 신임 정보를 제공해야 하는 것은 아닙니다. 이 옵션은 마이그레이션 등의 경우에 유용할 수 있습니다.

다음을 수행하면 아래와 같이 표시됩니다.

- 사용자 이름 및 비밀번호를 제공하십시오. 이는 인증됩니다.
- 사용자 이름 및 비밀번호를 제공하지 마십시오. 연결이 허용됩니다.
- 사용자 이름을 제공하지만 오류를 수신하는 비밀번호를 제공하지 마십시오.

**중요사항:** OPTIONAL 은 채널 인증 (CHLAUTH) 규칙에서 보다 제한적인 옵션을 설정하려는 경우에도 설정할 수 있는 최소값입니다.

NONE 을 선택하고 클라이언트 연결이 **CHCKCLNT** 가 REQUIRED (또는 z/OS이외의 플랫폼에서는 REQDADM) 로 설정된 CHLAUTH 레코드와 일치하면 연결에 실패합니다. 멀티플랫폼에서는 AMQ9793 메시지를 수신하고 z/OS에서는 CSQX793E 메시지를 수신합니다.

채널 인증 규칙을 사용하여 일부 클라이언트 연결에 대해 더 제한적인 **CHCKCLNT** 옵션을 설정하는 방법에 대한 자세한 정보는 69 페이지의 『구성 세분화』의 내용을 참조하십시오.

**REQUIRED**
모든 애플리케이션이 올바른 신임 정보를 제공해야 합니다. 다음 참고사항도 참조하십시오.

**REQDADM**

권한이 있는 사용자는 올바른 신임 정보를 제공해야 하지만 권한이 없는 사용자는 OPTIONAL 설정으로 처리됩니다. 다음 참고사항도 참조하십시오. z/OS (이 설정은 z/OS 시스템에서는 사용할 수 없습니다.)

**참고:**

**CHCKLOCL** 를 REQUIRED 또는 REQDADM 으로 설정하는 것은 사용자가 **runmqsc** 명령에서 사용자 ID를 지정하기 위해 **-u** 매개변수를 지정하지 않는 한 **runmqsc** (오류 AMQ8135: 권한 없음) 를 사용하여 큐 관리자를 로컬에서 관리할 수 없음을 의미합니다. 해당 매개변수를 설정하면 **runmqsc** 가 콘솔에서 사용자의 비밀번호를 입력하도록 프롬프트를 표시합니다.

마찬가지로, 로컬 시스템에서 IBM MQ Explorer 를 실행하는 사용자는 큐 관리자에 연결하려고 시도할 때 AMQ4036 오류가 표시됩니다. 사용자 ID및 비밀번호를 지정하려면 로컬 큐 관리자 오브젝트를 마우스 오른쪽 단추로 클릭하고 **연결 세부사항 > 특성 ...** 을 선택하십시오. 선택하십시오. **사용자 ID** 섹션에서 사용할 사용자 ID및 비밀번호를 입력한 후 **확인**을 클릭하십시오.

**CHCKCLNT**를 사용하는 원격 연결에도 비슷한 고려사항이 적용됩니다.

큐 관리자 **CONNAUTH** 속성은 IBM MQ 8.0이전 버전에서 마이그레이션되었지만 *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* . 이 기본 **AUTHINFO** 정의의 **CHCKCLNT** 는 기본적으로 REQDADM 으로 설정됩니다.

따라서 권한이 있는 사용자 ID를 사용하여 연결하는 기존 클라이언트는 올바른 신임 정보를 제공해야 합니다.

**경고:** 클라이언트 애플리케이션에 대한 MQCSP 구조의 신임 정보는 때때로 일반 텍스트로 네트워크를 통해 전송됩니다. 클라이언트 신임 정보가 보호되도록 하려면 29 페이지의 『MQCSP 비밀번호 보호』의 내용을 참조하십시오.

## 구성 세분화

AUTHINFO 오브젝트의 **CHCKLOCL** 및 **CHCKCLNT** 속성은 큐 관리자에 대한 모든 연결의 인증 요구사항을 설정합니다. 이러한 속성 외에도 채널 인증 (CHLAUTH) 규칙의 **CHCKCLNT** 속성을 사용하면 CHLAUTH 규칙과 일치하는 특정 클라이언트 연결에 대해 보다 엄격한 인증 요구사항을 설정할 수 있습니다.

예를 들어, AUTHINFO 오브젝트에서 전체 **CHCKCLNT** 값을 OPTIONAL로 설정한 후 CHLAUTH 규칙에서 **CHCKCLNT** 를 REQUIRED 또는 REQDADM 으로 설정하여 특정 채널에 대해 보다 엄격하게 업그레이드할 수 있습니다. 기본적으로 CHLAUTH 규칙은 **CHCKCLNT(ASQMGR)** 로 정의되므로 이 단위를 사용할 필요가 없습니다. 예를 들어, 이러한 MQSC 명령은 AUTHINFO 오브젝트의 **CHCKCLNT** 속성을 대체하는 하나의 CHLAUTH 규칙과 다음을 수행하지 않는 하나의 CHLAUTH 규칙을 정의합니다.

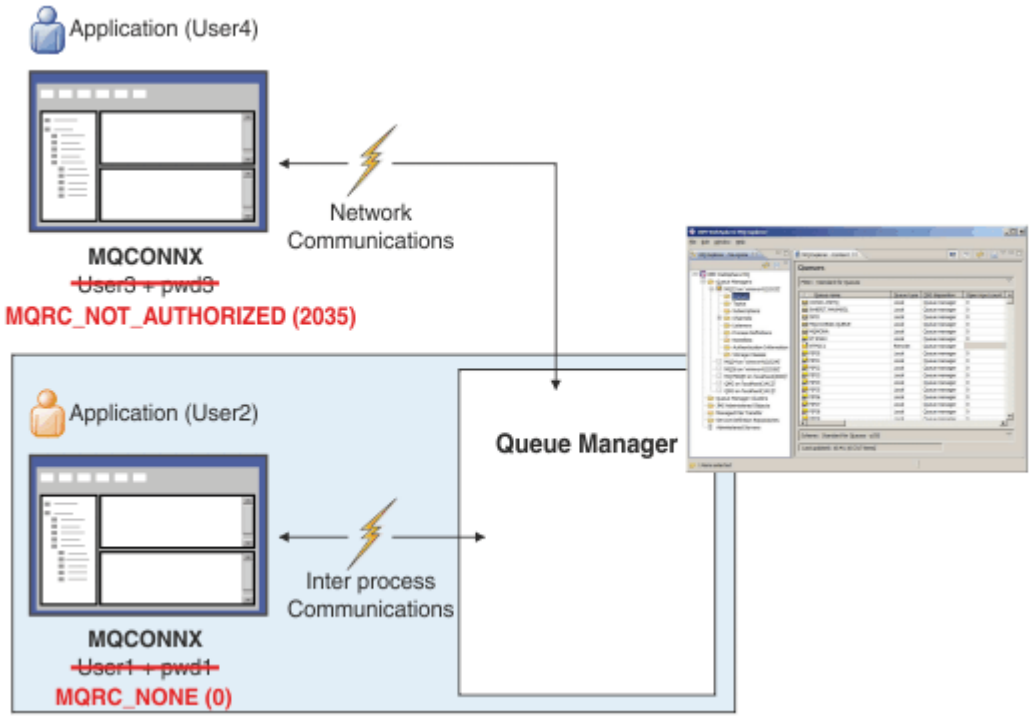
```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)

SET CHLAUTH('*') TYPE(ADDRESSMAP) +
ADDRESS('*') USERSRC(CHANNEL) +
CHCKCLNT(REQUIRED)

SET CHLAUTH('*') TYPE(SSLPEERMAP) +
SSLPEER('CN=*') USERSRC(CHANNEL)
```

CHLAUTH 규칙에 대한 자세한 정보는 47 페이지의 『채널 인증 레코드』의 내용을 참조하십시오.

## 오류 알림




다음 상황에서 오류가 기록됩니다.

- 애플리케이션은 필요할 때 인증 신임 정보를 제공하지 않습니다.
- 애플리케이션이 올바르지 않은 인증 신임 정보를 제공합니다. 이 상황은 구성에서 애플리케이션이 신임 정보를 제공하는 것이 선택사항임을 나타내는 경우에도 오류로 처리됩니다.

**참고: CHCKLOCL** 또는 **CHCKCLNT** 가 NONE으로 설정되면 애플리케이션에서 제공하는 올바르지 않은 신임 정보가 발견되지 않습니다.

애플리케이션에 오류를 리턴하기 전에 **FAILDLAY** 속성에 지정된 시간(초) 동안 실패한 인증이 보류됩니다. 이 지연은 연결을 반복적으로 시도하는 애플리케이션으로부터 일부 보호를 제공합니다.

오류는 다음과 같은 여러 가지 방법으로 기록됩니다.

**애플리케이션**
MQRC_NOT_AUTHORIZED (2035) 이유 코드가 애플리케이션에 리턴됩니다.

**관리자**
IBM MQ 관리자에게 오류 로그에 보고된 이벤트가 표시됩니다. 오류 메시지는 예를 들어, 사용자에게 연결 권한이 없기 때문이 아니라 신임 정보가 올바르지 않기 때문에 연결이 거부되었음을 표시합니다.

**모니터링 도구**
권한 이벤트를 켜는 경우 SYSTEM.ADMIN.QMGR.EVENT 큐의 이벤트 메시지를 사용하여 모니터링 도구에 실패를 알릴 수도 있습니다. 권한 이벤트를 켜려면 다음 MQSC 명령을 실행하십시오.

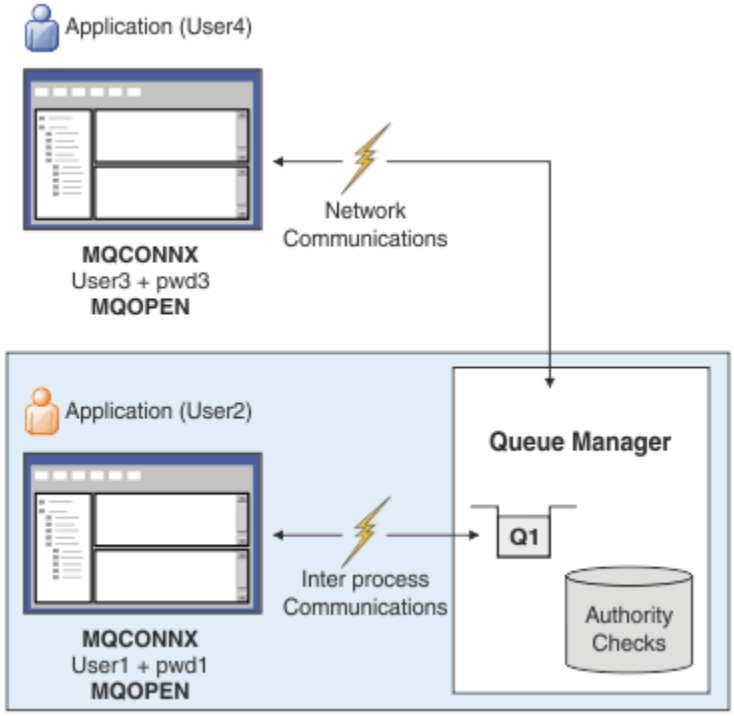
```
ALTER QMGR AUTHOREV(ENABLED)
```

이 "권한 부여되지 않음" 이벤트는 유형 1연결 이벤트이며 다른 유형 1이벤트와 동일한 필드를 제공하며 추가 필드인 제공된 MQCSP 사용자 ID를 제공합니다. 애플리케이션이 비밀번호를 제공한 경우 이벤트 메시지에 포함되지 않습니다. 이는 이벤트 메시지에 두 개의 사용자 ID가 있음을 의미합니다.

- 애플리케이션이 실행 중인 사용자 ID입니다.
- 애플리케이션이 제공한 신임 정보의 사용자 ID입니다.

이 이벤트 메시지에 대한 자세한 정보는 권한 부여되지 않음 (유형 1)을 참조하십시오.

## 권한 부여를 위해 사용자 채택




애플리케이션이 제공하는 신임 정보를 연결에 대한 컨텍스트로 채택하도록 큐 관리자를 구성할 수 있습니다. 신임을 채택하는 것은 인증 신임에 제공된 사용자 ID가 관리 화면에 표시되고 메시지에 표시되는 권한 검사에 사용됨을 의미합니다. AUTHINFO 오브젝트의 **ADOPTCTX** 속성은 신임 정보가 애플리케이션의 컨텍스트로 채택되는지 여부를 제어합니다. 예를 들어, 다음 MQSC 명령은 연결 인증에 사용되는 USE.PWD 라는 AUTHINFO 오브젝트를 정의하고 **ADOPTCTX** 속성을 YES로 설정합니다.

```
DEFINE AUTHINFO(USE.PWD) +
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)

ALTER QMGR CONNAUTH(USE.PWD)
```

**ADOPTCTX** 속성에 대해 다음 값을 지정할 수 있습니다.

**ADOPTCTX(YES)**
애플리케이션에서 제공하는 신임 정보는 연결 지속 기간 동안 애플리케이션 컨텍스트로 채택됩니다. 애플리케이션에 대한 모든 권한 검사는 인증된 신임 정보의 사용자 ID를 사용하여 수행됩니다.

> **주의: ADOPTCTX(YES)** 및 로컬 운영 체제 사용자 ID를 사용하는 경우 채택되는 사용자 ID가 IBM MQ의 사용자 ID에 대한 요구사항을 충족하는지 확인해야 합니다. 자세한 정보는 83 페이지의 『사용자 ID』의 내용을 참조하십시오.

**ADOPTCTX(NO)**
애플리케이션에서 제공하는 신임 정보는 연결 시 인증에만 사용됩니다. 애플리케이션이 실행 중인 사용자 ID는 향후 권한 검사에 계속 사용됩니다. 이 옵션은 마이그레이션할 때 또는 채널 인증 레코드와 같은 다른 메커니즘을 사용하여 메시지 채널 에이전트 사용자 ID(MCAUSER)를 지정하려는 경우에 유용할 수 있습니다.

## 채널 인증과의 상호작용

채널 인증 규칙을 사용하여 클라이언트에서 수신된 사용자 ID를 기반으로 애플리케이션 연결에 대한 컨텍스트로 사용되는 사용자 ID를 변경할 수 있습니다. 채널 인증 규칙을 사용하여 연결과 연관된 사용자 ID를 변경하는 예는 362 페이지의 『MCAUSER 사용자 ID에 클라이언트 사용자 ID 맵핑』의 내용을 참조하십시오.

연결 인증 및 채널 인증 규칙이 처리되는 순서는 IBM MQ 클라이언트 애플리케이션 연결의 보안 컨텍스트를 결정하는 데 중요한 요소입니다 `qm.ini` 파일의 **channels** 스탠자에 있는 **ChlauthEarlyAdopt** 매개변수는 큐 관리자가 애플리케이션에서 제공하는 신임 정보의 컨텍스트를 채택하고 채널 인증 규칙을 적용하는 순서를 제어합니다. **ChlauthEarlyAdopt**에 대한 자세한 정보는 채널 스탠자의 속성을 참조하십시오.

**주의:** 인증 정보 오브젝트에서 **ADOPTCTX(YES)** 매개변수를 사용하는 경우, **ChlauthEarlyAdopt** 매개변수가 Y로 설정된 경우에만 애플리케이션이 제공하는 신임 정보에서 채택된 컨텍스트를 채널 인증 규칙으로 변경할 수 있습니다.

연결 인증 및 채널 인증의 상호작용 및 클라이언트 애플리케이션이 큐 관리자에 연결할 때 검사가 발생하는 순서에 대한 자세한 정보는 53 페이지의 『CHLAUTH와 CONNAUTH의 상호작용』의 내용을 참조하십시오.

**관련 개념**

66 페이지의 『연결 인증』
연결 인증을 사용하면 애플리케이션이 큐 관리자에 연결할 때 인증 신임 정보를 제공할 수 있습니다. 큐 관리자가 신임 정보의 유효성을 검증합니다. 신임 정보에 제공된 사용자 ID는 애플리케이션이 액세스하는 자원에 대한 권한 검사에서 사용하기 위해 채택될 수도 있습니다.

72 페이지의 『연결 인증: 애플리케이션 변경사항』

73 페이지의 『연결 인증: 사용자 저장소』
각 큐 관리자에 대해 사용자 ID 및 비밀번호를 인증하기 위해 서로 다른 유형의 인증 정보 오브젝트를 선택할 수 있습니다.

## 연결 인증: 애플리케이션 변경사항

메시지 큐 인터페이스 (MQI) 를 사용하는 애플리케이션은 MQCONNX가 호출될 때 연결 보안 매개변수 (MQCSP) 구조에서 사용자 ID및 비밀번호를 제공할 수 있습니다. 다른 애플리케이션 프로그래밍 인터페이스에서 MQCSP 구조는 일반적으로 IBM MQ 라이브러리에 의해 애플리케이션 대신 구성됩니다.

V 9.4.0 IBM MQ 9.3.4부터 AIX 또는 Linux 시스템에서 실행되는 큐 관리자에 연결하는 클라이언트 애플리케이션은 식별의 대체 수단으로 MQCSP 구조에서 인증 토큰을 보낼 수도 있습니다.

사용자 ID및 비밀번호또는 인증 토큰 은 큐 관리자와 함께 제공되는 오브젝트 권한 관리자 (OAM) 또는 z/OS 시스템에서 큐 관리자와 함께 제공되는 권한 서비스 컴포넌트에 대한 검사를 위해 전달됩니다. 사용자 고유의 사용자 정의 인터페이스를 작성할 필요가 없습니다.

애플리케이션이 클라이언트로 실행 중인 경우, 사용자 ID및 비밀번호또는 인증 토큰 도 처리를 위해 클라이언트 측 및 서버 측 보안 엑시트에 전달됩니다. 또한 채널 인스턴스의 메시지 채널 에이전트 사용자 ID (MCAUSER) 속성 을 설정하는 데 사용할 수도 있습니다.

**경고:** 클라이언트 애플리케이션에 대한 MQCSP 구조의 신임 정보는 때때로 일반 텍스트로 네트워크를 통해 전송됩니다. 클라이언트 애플리케이션 신임 정보가 보호되는지 확인하려면 29 페이지의 『MQCSP 비밀번호 보호』의 내용을 참조하십시오.

XAOPEN 문자열을 사용하여 사용자 ID및 비밀번호를 제공하면 애플리케이션 코드를 변경하지 않아도 됩니다.

**참고:**

IBM WebSphere MQ 6.0부터 보안 엑시트를 사용하여 MQCSP를 설정할 수 있습니다. 그러므로, 이 레벨 이상에 있는 클라이언트는 업그레이드할 필요가 없습니다.

그러나 IBM MQ 8.0이전의 IBM MQ 버전에서 MQCSP는 애플리케이션이 제공한 사용자 ID및 비밀번호에 제한을 두지 않았습니다. 이러한 값을 IBM MQ에서 제공하는 기능과 함께 사용하는 경우 이러한 기능을 사용하는 데 적용되는 한계가 있지만 사용자 고유의 엑시트에만 전달하는 경우에는 해당 한계가 적용되지 않습니다.

**관련 개념**

66 페이지의 『연결 인증』
연결 인증을 사용하면 애플리케이션이 큐 관리자에 연결할 때 인증 신임 정보를 제공할 수 있습니다. 큐 관리자가 신임 정보의 유효성을 검증합니다. 신임 정보에 제공된 사용자 ID는 애플리케이션이 액세스하는 자원에 대한 권한 검사에서 사용하기 위해 채택될 수도 있습니다.

67 페이지의 『연결 인증: 구성』
연결할 때 애플리케이션에서 제공하는 신임 정보를 인증하도록 큐 관리자를 구성할 수 있습니다.

73 페이지의 『연결 인증: 사용자 저장소』
각 큐 관리자에 대해 사용자 ID 및 비밀번호를 인증하기 위해 서로 다른 유형의 인증 정보 오브젝트를 선택할 수 있습니다.

## 연결 인증: 사용자 저장소

각 큐 관리자에 대해 사용자 ID 및 비밀번호를 인증하기 위해 서로 다른 유형의 인증 정보 오브젝트를 선택할 수 있습니다.

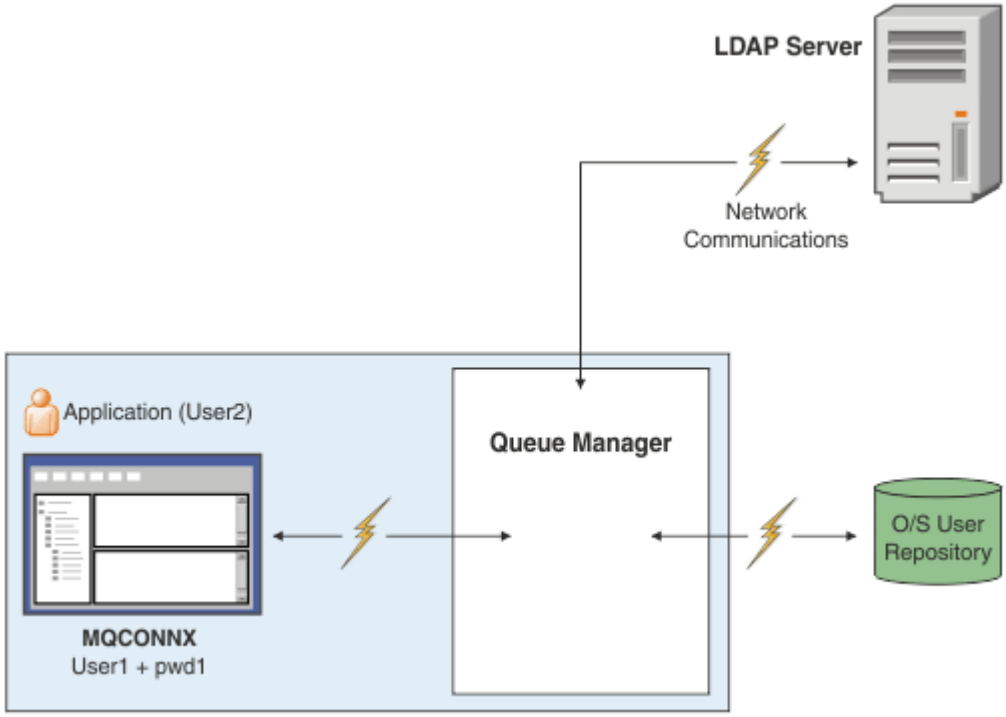



그림 7. 인증 정보 오브젝트 유형

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
DEFINE AUTHINFO(USE.LDAP) +
AUTHTYPE(IDPWLDAP) +
CONNAME('ldap1(389),ldap2(389)') +
LDAPUSER('CN=QMGR1') +
LDAPPWD('passw0rd') SECCOMM(YES)
```

다이어그램에 표시된 대로 두 가지 유형의 인증 정보 오브젝트가 있습니다.

- IDPWOS는 큐 관리자가 사용자 ID 및 비밀번호를 인증하기 위해 로컬 운영 체제를 사용함을 나타내는 데 사용됩니다. 로컬 운영 체제를 사용하기로 선택한 경우에는 앞의 주제에 설명된 대로 공통 속성을 설정해야 합니다.
- IDPWLDAP는 큐 관리자가 사용자 ID 및 비밀번호를 인증하기 위해 LDAP 서버를 사용함을 나타내는 데 사용됩니다. LDAP 서버를 사용하기로 선택한 경우에는 이 주제에서 더 많은 정보가 제공됩니다.

큐 관리자의 **CONNAUTH** 속성에서 적합한 오브젝트에 이름을 지정하여 각 큐 관리자가 사용할 단 한가지 유형의 인증 정보 오브젝트만을 선택할 수 있습니다.

## 인증을 위해 LDAP 서버 사용

**CONNAME** 필드를 큐 관리자의 LDAP 서버의 주소로 설정하십시오. LDAP 서버의 추가 주소를 쉼표로 구분된 목록으로 제공할 수 있고, 이는 LDAP 서버가 이 기능 자체를 제공하지 않는 경우 중복성에 도움을 줄 수 있습니다.

큐 관리자가 LDAP 서버에 액세스하고 사용자 레코드에 대한 정보를 찾아볼 수 있도록 필수 LDAP 서버 ID 및 비밀번호를 **LDAPUSER** 및 **LDAPPWD** 필드에 설정하십시오.

## LDAP 서버로의 보안 연결

채널과는 달리 LDAP 서버와 통신하기 위해 TLS의 사용을 켜기 위한 **SSLCIPH** 매개변수가 없습니다. 이 경우 IBM MQ는 LDAP 서버에서 많은 구성이 수행될 수 있도록 LDAP 서버에 클라이언트의 역할을 수행합니다. IBM MQ에서 일부 기존 매개변수는 해당 연결이 작동하는 방법을 구성하는 데 사용됩니다.

LDAP 서버에 대한 연결성이 TLS를 사용하는지 여부를 제어하려면 **SECCOMM** 필드를 설정하십시오.

이 속성에 추가로, 큐 관리자 속성 **SSLFIPS** 및 **SUITEB**는 선택된 암호 스펙의 세트를 제한합니다. LDAP 서버에 큐 관리자를 식별하는 데 사용되는 인증서 `ibmwebspheremq` *`qmgr-name`* 또는 **CERTLABL** 속성의 값입니다. 자세한 내용은 디지털 인증서 레이블을 참조하십시오.

## LDAP 사용자 저장소

LDAP 사용자 저장소를 사용할 때 단지 큐 관리자에 LDAP 서버를 찾을 위치를 알려주는 것보다 더 많은 구성을 큐 관리자에서 수행해야 합니다.

LDAP 서버에 정의된 사용자 ID에는 이를 고유하게 식별하는 계층 구조가 있습니다. 그러므로 애플리케이션은 큐 관리자에 연결하고 해당 사용자 ID를 완전한 계층 구조 사용자 ID로 제공할 수 있습니다.

그러나 애플리케이션이 제공해야 하는 정보를 단순화하기 위해 계층의 첫 번째 부분은 모든 ID에 공통됨을 가정하고 애플리케이션이 제공한 단축된 ID 앞에 이를 자동으로 추가하도록 큐 관리자를 구성할 수 있습니다. 그러면 큐 관리자는 LDAP 서버에 완전한 ID를 제공할 수 있습니다.

LDAP 계층에서 LDAP이 ID를 검색하는 초기 지점으로 BASEDNU를 설정하십시오. BASEDNU를 설정하는 경우 LDAP 계층에서 ID를 검색할 때 하나의 결과만 리턴되는지 확인해야 합니다.

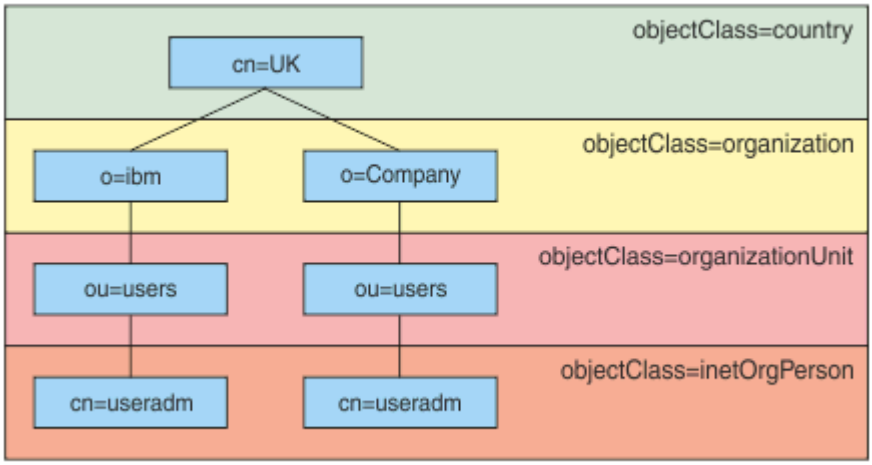



그림 *8. LDAP* 계층 예

예를 들어, 74 페이지의 그림 8에서 BASEDNU는 "ou=users,o=ibm,c=UK" 또는 ",o=ibm,c=UK"로 설정할 수 있습니다. 그러나 "cn=useradm"을 포함하는 식별 이름이 "o=ibm" 분기 및 "o=Company" 분기 모두에 있으므로, BASEDNU는 "c=UK"로 설정할 수 없습니다. 성능 및 보안상의 이유로, LDAP 계층 중 필요한 사용자 ID 모두를 참조할 수 있는 지점에서 가장 높은 지점을 사용합니다. 이 예에서는 "ou=users,o=ibm,c=UK"가 이 지점입니다.

애플리케이션은 LDAP 속성 이름(예: CN=)을 제공하지 않고 큐 관리자에 사용자 ID를 제출할 수도 있습니다. USRFIELD를 LDAP 속성 이름에 설정한 경우에는 이 값은 애플리케이션에서 나오는 사용자 ID에 접두부로서 추가됩니다. 이는 운영 체제 사용자 ID에서 LDAP 사용자 ID로 이동할 때 유용한 마이그레이션 보조 도구일 수 있습니다. 그러면 애플리케이션이 두 경우 모두 동일한 문자열을 제공할 수 있고 애플리케이션을 변경하는 것을 피할 수 있습니다.

그러므로 LDAP 서버에 제공된 완전한 사용자 ID는 다음과 비슷합니다.

```
USRFIELD = ID_from_application BASEDNU
```

**관련 개념**

66 페이지의 『연결 인증』

연결 인증을 사용하면 애플리케이션이 큐 관리자에 연결할 때 인증 신임 정보를 제공할 수 있습니다. 큐 관리자가 신임 정보의 유효성을 검증합니다. 신임 정보에 제공된 사용자 ID는 애플리케이션이 액세스하는 자원에 대한 권한 검사에서 사용하기 위해 채택될 수도 있습니다.

67 페이지의 『연결 인증: 구성』

연결할 때 애플리케이션에서 제공하는 신임 정보를 인증하도록 큐 관리자를 구성할 수 있습니다.

72 페이지의 『연결 인증: 애플리케이션 변경사항』

## 사용자 *ID* 및 비밀번호(*mqccred*)를 삽입하기 위한 클라이언트측 보안 엑시트

사용자 ID 또는 비밀번호를 보내는 데 필요한 클라이언트 애플리케이션이 있지만 아직 소스를 변경할 수 없는 경우 사용할 수 있는 **mqccred** 라는 IBM MQ 8.0 와 함께 제공되는 보안 엑시트가 있습니다. **mqccred**는 클라이언트 애플리케이션 대신에 `.ini` 파일로부터 사용자 ID 및 비밀번호를 제공합니다. 이 사용자 ID 및 비밀번호는 이들을 인증할(그렇게 구성된 경우) 큐 관리자에 전송됩니다.

### 개요

**mqccred**는 클라이언트 애플리케이션과 같은 시스템에서 실행되는 보안 엑시트입니다. 이는 사용자 ID 및 비밀번호 정보를 클라이언트 애플리케이션 대신에 제공하도록 허용합니다(해당 정보가 애플리케이션 자체에 의해 제공되는 중이 아닌 경우). 사용자 ID 및 비밀번호 정보는 연결 보안 매개변수(MQCSP)로서 알려진 구조로 제공되고 연결 인증이 구성된 경우 큐 관리자에 의해 인증됩니다.

사용자 ID 및 비밀번호 정보는 클라이언트 시스템에서 `.ini` 파일로부터 검색됩니다. 파일의 비밀번호는 클라이언트 애플리케이션을 실행 중인 사용자 ID(및 그러므로 엑시트)만이 이를 읽을 수 있도록 **runmqccred** 명령을 사용하고 또한 `.ini`에서 파일 권한을 확인하여 난독화를 통해 보호됩니다.

### 위치

**mqccred**가 설치됩니다:

**Windows 플랫폼**
*installation_directory*\Tools\c\Samples\mqccred\ 디렉토리에서

**AIX and Linux 플랫폼**
*installation_directory*/samp/mqccred 디렉토리에서

**참고:** 엑시트:

1. 보안 채널 엑시트로서 순수하게 작동하고, 채널에 정의된 유일한 이러한 엑시트여야 합니다.
2. 이는 일반적으로 CCDT(Client Channel Definition Table)를 통해 이름이 지정되지만 Java 클라이언트는 JNDI 오브젝트에 직접 언급된 엑시트를 가질 수 있거나, 엑시트가 MQCD 구조를 수동으로 구성하는 애플리케이션에 대해 구성되어 있을 수도 있습니다.
3. **mqccred** 및 **mqccred_r** 프로그램을 `var/mqm/exits` 디렉토리로 복사해야 합니다.

   예를 들어 64비트 AIX 또는 Linux 시스템에서 다음 명령을 실행하십시오.

   ```
   cp installation_directory/samp/mqccred/lib64/* /var/mqm/exits
   ```

   자세한 정보는 mqccred를 테스트하는 방법의 단계별 예를 참조하십시오.
4. IBM MQ의 이전 버전에서 실행할 수 있습니다(IBM WebSphere MQ 7.0.1까지).

### 사용자 ID 및 비밀번호 설정

`.ini` 파일은 지정되지 않은 큐 관리자를 위한 글로벌 설정과 함께 각 큐 관리자의 스탠자를 포함합니다. 각 스탠자에는 큐 관리자의 이름, 사용자 ID 및 일반 텍스트 또는 난독화된 비밀번호를 포함합니다.

원하는 어떤 편집기든지 사용하여 `.ini` 파일을 직접 편집하고 스탠자에 일반 텍스트 비밀번호 속성을 추가해야 합니다. 제공된, **runmqccred** 프로그램을 실행하십시오. 이는 `.ini` 파일을 사용하고 **Password** 속성을 비밀번호의 난독화된 양식인 **OPW** 속성으로 바꿉니다.

명령 및 해당 매개변수의 설명은 runmqccred의 내용을 참조하십시오.

`mqccred.ini` 파일에는 사용자 ID 및 비밀번호 정보가 포함됩니다.

템플리트 `.ini` 파일이 엔터프라이즈의 시작점을 제공하기 위해 엑시트와 동일 디렉토리에 제공됩니다.

기본적으로 `$HOME/.mqs/mqccred.ini`에서 이 파일을 찾습니다. 이를 다른 곳에서 찾아보고 싶으면 이를 가리키는 환경 변수 *MQCCRED*를 사용할 수 있습니다.

```
MQCCRED=C:\mydir\mqccred.ini
```

*MQCCRED*를 사용하는 경우에는 변수에는 `.ini` 파일 유형을 포함하여 구성 파일의 전체 이름을 포함해야 합니다. 이 파일에는 비밀번호(난독화되었더라도)가 포함되므로 권한 없는 사용자가 이를 읽을 수 없도록 하기 위해 운영 체제 권한을 사용하여 파일을 보호할 것으로 기대됩니다. 올바른 파일 권한이 없는 경우에는 엑시트는 성공적으로 실행되지 않습니다.

애플리케이션이 이미 MQCSP 구조를 제공한 경우에는 엑시트는 일반적으로 이를 예상하고 `.ini` 파일로부터 정보를 삽입하지 않습니다. 그러나 스탠자에서 **Force** 속성을 사용하여 이를 대체할 수 있습니다.

**Force**를 *TRUE* 값으로 설정하면 애플리케이션 제공 사용자 ID 및 비밀번호가 제거되고 이들이 ini 파일 버전으로 바뀝니다.

파일의 기본값을 설정하기 위해 해당 파일의 글로벌 절에서 **Force** 속성을 설정할 수도 있습니다.

**Force**의 기본값은 *FALSE*입니다.

모든 큐 관리자 또는 각 개별 큐 관리자에 대해 사용자 ID 및 비밀번호를 제공할 수 있습니다. 이는 `mqccred.ini` 파일의 예입니다.

```
# comments are permitted
AllQueueManagers:
User=abc
OPW=%^&aervrgtsr

QueueManager:
Name=QMA
User=user1
OPW=H&^dbgfh

Force=TRUE

QueueManager:
Name=QMB
User=user2
password=passw0rd
```

**참고:**

1. 개별 큐 관리자 정의는 글로벌 설정에 우선합니다.
2. 속성은 대소문자를 구분합니다.

## 제한조건

이 엑시트가 사용 중이면 애플리케이션을 실행하는 사용자의 로컬 사용자 ID는 클라이언트에서 서버로 플로우하지 않습니다. 사용 가능한 유일한 ID 정보는 ini 파일 컨텐츠에서 나옵니다.

그러므로 **ADOPTCTX(YES)**를 사용하도록 큐 관리자를 구성하거나 사용 가능한 메커니즘 중 하나를 사용하여 (예: 47 페이지의 『채널 인증 레코드』) 인바운드 연결 요청을 적합한 사용자 ID로 맵핑해야 합니다.

**중요사항:** 새 비밀번호를 추가하거나 이전 비밀번호를 업데이트하는 경우에는 **runmqccred** 명령은 일반 텍스트 비밀번호만을 처리하고 난독화된 비밀번호는 변경되지 않은 상태로 남겨둡니다.

### 디버그

엑시트는 사용 가능할 때 표준 IBM MQ 추적에 작성합니다.

디버깅 구성 문제를 돕기 위해 엑시트는 stdout에 직접 작성할 수도 있습니다.

채널 보안 엑시트 데이터(**SCYDATA**) 구성은 일반적으로 채널에는 필요하지 않습니다. 그러나 사용자는 다음을 지정할 수 있습니다.

**오류**
구성 파일을 찾을 수 없는 것과 같은 오류 조건에 대한 인쇄 정보만.

**DEBUG**
이러한 오류 조건 및 일부 추가적인 추적 명령문을 표시합니다.

**NOCHECKS**
파일 권한에 대한 제한조건 및 `.ini` 파일이 보호되지 않은 비밀번호를 포함하지 않아야 하는 추가 제한조건을 무시합니다.

하나 이상의 이러한 요소를 쉼표로 구분하여 임의의 순서대로 **SCYDATA** 필드에 넣을 수 있습니다. 예를 들어, `SCYDATA=(NOCHECKS,DEBUG)`입니다.

항목은 대소문자를 구분하여 대문자로 입력되어야 함을 유의하십시오.

## `mqccred` 사용

파일을 설정한 후에는 `SCYEXIT('mqccred(ChlExit)')` 속성을 포함하도록 클라이언트 연결 채널 정의를 업데이트하여 채널 엑시트를 호출할 수 있습니다.

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

**관련 참조**
SCYDATA
SCYEXIT
runmqccred

## *Java* 클라이언트와의 연결 인증

연결 인증은 큐 관리자가 제공된 사용자 ID 및 비밀번호를 사용하여 애플리케이션을 인증할 수 있도록 큐 관리자를 구성할 수 있는 IBM MQ의 기능입니다. 애플리케이션이 클라이언트 전송을 사용 중인 Java 애플리케이션인 경우 호환 모드 또는 MQCSP 인증 모드에서 연결 인증을 실행할 수 있습니다.

다음 방법 중 하나를 사용하여 애플리케이션에서 인증될 사용자 ID 및 비밀번호를 지정합니다.

- IBM MQ classes for Java 애플리케이션의 경우 `MQEnvironment` 클래스 또는 `com.ibm.mq.MQQueueManager` 구성자로 전달되는 특성 해시 테이블에서
- IBM MQ classes for JMS 애플리케이션의 경우 `createConnection(String username, String Password)` 또는 `createContext(String username, String password)` 메소드에 대한 인수로

## MQCSP 인증 모드

이 모드에서는 인증될 사용자 ID 및 비밀번호뿐만 아니라 애플리케이션이 실행되는 클라이언트 측 사용자 ID가 큐 관리자로 송신됩니다. IBM MQ classes for Java 및 IBM MQ classes for JMS는 인증될 사용자 ID 및 비밀번호를 MQCSP 구조의 큐 관리자에 송신합니다.

사용자 ID 및 비밀번호는 MQCXP 구조 내의 서버 연결 보안 엑시트에서 사용할 수 있습니다. MQCSP 구조 주소는 채널에 대한 MQCXP 구조의 **SecurityParms** 필드에서 찾을 수 있습니다.

MQCSP 인증 모드의 이점은 다음과 같습니다.

- 인증될 사용자 ID의 최대 길이가 1027바이트입니다.
- 인증을 위한 비밀번호의 최대 길이가 256자입니다.
- 큐 관리자에 대한 연결 인증을 제어하는 데 사용되는 인증 정보 오브젝트가 ADOPTCTX(NO)로 구성된 경우 애플리케이션이 실행되는 클라이언트 측 사용자 ID를 사용하여 IBM MQ 자원을 사용하기 위한 액세스 권한 검사를 수행할 수 있습니다.

### 호환 모드

IBM MQ 8.0 이전에는 Java 클라이언트가 클라이언트 연결 채널을 통해 사용자 ID와 비밀번호를 서버 연결 채널로 송신하고, 이들을 MQCD 구조의 **RemoteUserIdentifier** 및 **RemotePassword** 필드에 있는 보안 엑시트에 제공할 수 있었습니다. 호환성 모드에서 이 동작은 유지됩니다.

이 모드를 연결 인증과 결합하여 사용할 수 있고, 이전에 동일 작업을 수행하는 데 사용되었던 보안 엑시트로부터 멀리 마이그레이션할 수도 있습니다.

이 모드의 제한사항은 다음과 같습니다.

- 사용자 ID 및 비밀번호의 길이가 12자 이하여야 합니다. 12자보다 긴 사용자 ID는 12자로 잘립니다. 이로 인해 이유 코드 MQRC_NOT_AUTHORIZED로 연결에 실패할 수 있습니다.
- 애플리케이션이 실행되는 클라이언트 측 사용자 ID가 큐 관리자로 송신되지 않습니다. 큐 관리자에 대한 연결 인증을 제어하는 데 사용되는 인증 정보 오브젝트에 ADOPTCTX(YES)를 설정하거나 TLS 인증서를 기반으로 하는 채널 인증 규칙과 같은 다른 방법을 사용하여 IBM MQ 자원 사용 권한을 확인하는 채널 MCA 사용자 ID를 설정해야 합니다.

### 기본 인증 모드

IBM MQ classes for Java 또는 IBM MQ classes for JMS 클라이언트 애플리케이션에서 사용되는 기본 인증 모드는 애플리케이션이 사용자 ID 및 비밀번호를 지정하는지 여부에 따라 다릅니다.

- 사용자 ID및 비밀번호가 지정되면 기본적으로 MQCSP 인증이 사용됩니다.
- 사용자 ID는 지정되었지만 비밀번호는 지정되지 않은 경우 기본적으로 호환 모드가 사용됩니다.
- 사용자 ID가 지정되지 않으면 항상 호환 모드가 사용됩니다.

사용자 ID가 지정된 경우 78 페이지의 『인증 모드 선택』에 설명된 대로 애플리케이션에서 각 개별 연결에 대한 특정 인증 모드를 선택하거나 애플리케이션이 시작되기 전에 글로벌로 설정할 수 있습니다.

**참고:** IBM MQ classes for JMS를 사용하는 애플리케이션은 IBM MQ 9.3.0의 기본 인증 모드 변경사항에 영향을 받을 수 있습니다. IBM MQ classes for JMS를 IBM MQ 9.3.0로 업그레이드하면 이전에 기본적으로 호환 모드를 사용한 애플리케이션이 대신 MQCSP 인증을 사용합니다. 이로 인해 이전에 큐 관리자에 성공적으로 연결된 애플리케이션이 이유 코드 2035(MQRC_NOT_AUTHORIZED)를 포함하는 JMSException으로 연결에 실패할 수 있습니다. 이러한 경우 78 페이지의 『인증 모드 선택』에 설명된 방법 중 하나를 사용하여 애플리케이션이 호환 모드를 사용하도록 지정하십시오.

로컬 바인딩을 사용하여 큐 관리자에 연결하는 Java 애플리케이션은 항상 MQCSP 인증 모드를 사용합니다.

### 인증 모드 선택

큐 관리자에 연결할 때 사용자 ID를 지정하는 Java 클라이언트 애플리케이션에서 사용되는 인증 모드는 다음 방법 중 하나를 사용하여 지정할 수 있습니다. 이러한 방법이 우선순위 내림차순으로 나열되어 있습니다. 이러한 방법 중 하나를 사용하여 인증 모드를 지정하지 않으면 기본 인증 모드가 사용됩니다.

**참고:** 이러한 메소드를 사용하여 인증 모드를 선택하는 방법은 IBM MQ 9.3.0에서 명확하게 설명되었습니다. 경우에 따라 Java 클라이언트 애플리케이션에서 사용하는 인증 모드는 IBM MQ classes for Java 또는 IBM MQ classes for JMS 가 IBM MQ 9.3.0로 업그레이드될 때 변경될 수 있습니다. 이로 인해 이전에 큐 관리자에 성공적으로 연결된 애플리케이션이 이유 코드 2035(MQRC_NOT_AUTHORIZED)를 포함하는 JMSException으로 연결에 실패할 수 있습니다. 이러한 경우 다음 방법 중 하나를 사용하여 필요한 인증 모드를 선택하십시오.

- 큐 관리자에 연결하기 전에 애플리케이션에서 적절한 특성을 설정하여 각 개별 연결에 대한 인증 모드를 지정하십시오.

- IBM MQ classes for Java를 사용할 때 `com.ibm.mq.MQQueueManager` 구성자에 전달되는 Hashtable 특성에서 *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* 특성을 설정하십시오.
- IBM MQ classes for JMS를 사용하는 경우 연결을 작성하기 전에 적합한 연결 팩토리에서 *JmsConstants.USER_AUTHENTICATION_MQCSP* 특성을 설정하십시오.

이러한 특성의 값을 다음 값 중 하나로 설정하십시오.

**true**
큐 관리자의 인증을 받을 때 MQCSP 인증 모드를 사용합니다.

**false**
큐 관리자의 인증을 받을 때 호환 모드를 사용합니다.

- 애플리케이션을 시작할 때 *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java 시스템 특성을 설정하여 애플리케이션에서 설정된 모든 클라이언트 연결에 대한 인증 모드를 지정하십시오. 특성의 값을 다음 값 중 하나로 설정하십시오.

  **Y**
  큐 관리자의 인증을 받을 때 MQCSP 인증 모드를 사용합니다.

  **N**
  큐 관리자의 인증을 받을 때 호환 모드를 사용합니다.

  예를 들어, 다음 명령은 호환 모드를 선택하도록 특성을 설정하고 Java 애플리케이션을 시작합니다.

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```

- 애플리케이션이 시작된 환경에서 *com.ibm.mq.jmqi.useMQCSPauthentication* 환경 변수를 설정하여 동일한 환경에서 시작된 애플리케이션에서 설정된 모든 클라이언트 연결에 대한 인증 모드를 지정하십시오. 환경 변수의 값을 다음 값 중 하나로 설정하십시오.

  **Y**
  큐 관리자의 인증을 받을 때 MQCSP 인증 모드를 사용합니다.

  **N**
  큐 관리자의 인증을 받을 때 호환 모드를 사용합니다.

- 클라이언트 구성 파일의 JMQI 스탠자에 **useMQCSPauthentication** 속성을 지정하여 특정 IBM MQ MQI client 클라이언트 구성 파일을 사용하는 모든 애플리케이션의 인증 모드를 지정하십시오. 속성의 값을 다음 값 중 하나로 설정하십시오.

  **YES**
  큐 관리자의 인증을 받을 때 MQCSP 인증 모드를 사용합니다.

  **NO**
  큐 관리자의 인증을 받을 때 호환 모드를 사용합니다.

  **useMQCSPauthentication** 속성에 대한 자세한 정보는 클라이언트 구성 파일의 JMQI 스탠자를 참조하십시오.

## IBM MQ Explorer에서 인증 모드 선택

IBM MQ Explorer는 Java 애플리케이션이므로 호환성 모드와 MQCSP 인증 모드라는 두 개의 모드 또한 적용 가능합니다.

MQCSP 인증 모드가 기본값입니다.

사용자 ID가 제공된 패널에는 호환 모드를 사용 또는 사용 안함으로 설정하는 선택란이 있습니다.

- 이 선택란은 디폴트로 선택되어 있지 않습니다. 호환 모드를 사용하려면 이 선택란을 선택하십시오.

**관련 개념**
66 페이지의 『연결 인증』
연결 인증을 사용하면 애플리케이션이 큐 관리자에 연결할 때 인증 신임 정보를 제공할 수 있습니다. 큐 관리자가 신임 정보의 유효성을 검증합니다. 신임 정보에 제공된 사용자 ID는 애플리케이션이 액세스하는 자원에 대한 권한 검사에서 사용하기 위해 채택될 수도 있습니다.

72 페이지의 『연결 인증: 애플리케이션 변경사항』

73 페이지의 『연결 인증: 사용자 저장소』
각 큐 관리자에 대해 사용자 ID 및 비밀번호를 인증하기 위해 서로 다른 유형의 인증 정보 오브젝트를 선택할 수 있습니다.

### IBM MQ의 메시지 보안

IBM MQ 인프라에서 메시지 보안은 Advanced Message Security에 의해 제공됩니다.

Advanced Message Security(AMS)는 IBM MQ 보안 서비스를 확장하여 메시지 레벨에서 데이터 서명 및 암호화를 제공합니다. 확장된 서비스를 통해 메시지 데이터가 원래 큐에 배치된 시기와 검색된 시기 사이에서 수정되지 않도록 보장됩니다. 또한, AMS는 메시지 데이터 송신자가 서명된 메시지를 대상 큐에 배치할 권한이 있는지 확인합니다.

**관련 개념**

557 페이지의 『Advanced Message Security』
Advanced Message Security(AMS)는 엔드 애플리케이션에 영향을 미치지 않으면서 IBM MQ 네트워크를 통해 플로우하는 민감한 데이터에 대한 높은 레벨의 보호를 제공하는 IBM MQ의 컴포넌트입니다.

## 보안 요구사항 계획

이 주제 모음에서는 IBM MQ 환경에서 보안을 계획할 때 고려해야 하는 사항에 대해 설명합니다.

플랫폼 범위에서 다양한 애플리케이션에 대해 IBM MQ를 사용할 수 있습니다. 보안 요구사항은 각 애플리케이션별로 다를 수 있습니다. 어떤 경우에는 보안이 중요한 고려사항입니다.

IBM MQ는 TLS(Transport Layer Security)에 대한 지원을 비롯하여 링크 레벨 보안 서비스 범위를 제공합니다.

IBM MQ 설치를 계획할 때 특정 보안 측면을 고려해야 합니다.

- Multi 멀티플랫폼에서 이런 측면을 무시하고 아무것도 수행하지 않으면 IBM MQ를 사용할 수 없습니다.
- z/OS z/OS에서 이런 측면을 무시하면 IBM MQ 자원이 보호되지 않습니다. 즉, 모든 사용자가 모든 IBM MQ 자원에 액세스하여 변경할 수 있습니다.

### IBM MQ 관리 권한

IBM MQ 관리자는 다음을 수행할 권한이 필요합니다.

- IBM MQ 관리를 위한 명령 발행
- IBM MQ Explorer 사용
- IBM i IBM i 관리 패널 및 명령 사용
- z/OS z/OS에서 조작 및 제어판 사용
- z/OS z/OS에서 IBM MQ 유틸리티 프로그램 CSQUTIL 사용
- z/OS z/OS에서 큐 관리자 데이터 세트 액세스

자세한 정보는 다음을 참조하십시오.

- ALW 375 페이지의 『AIX, Linux, and Windows 에서 IBM MQ 를 관리할 수 있는 권한』
- IBM i 84 페이지의 『IBM i 에서 IBM MQ 를 관리할 수 있는 권한』
- z/OS 85 페이지의 『Authority to administer IBM MQ on z/OS』

### IBM MQ 오브젝트에 대해 작업할 권한

애플리케이션은 MQI 호출을 발행하여 다음 IBM MQ 오브젝트에 액세스할 수 있습니다.

- 큐 관리자

- 큐
- 프로세스
- 이름 목록
- 토픽

애플리케이션은 프로그래밍 가능 명령 형식(PCF) 명령을 사용해서도 해당 IBM MQ 오브젝트에 액세스하고 채널 및 인증 정보 오브젝트에도 액세스할 수 있습니다. 해당 오브젝트는 IBM MQ에서 보호 가능하기 때문에 애플리케이션에 연관된 사용자 ID는 이에 대한 액세스 권한이 필요합니다.

자세한 정보는 87 페이지의 『애플리케이션에서 IBM MQ를 사용하기 위한 권한 부여』의 내용을 참조하십시오.

### 채널 보안

메시지 채널 에이전트(MCA)에 연관된 사용자 ID는 다양한 IBM MQ 자원 액세스 권한이 필요합니다. 예를 들어, MCA는 큐 관리자에 연결할 수 있어야 합니다. 송신 MCA인 경우 채널에 대해 전송 큐를 열 수 있어야 합니다. 수신 MCA의 경우 목적지 큐를 열 수 있어야 합니다. 애플리케이션과 연관된 사용자 ID는 채널, 채널 시작기, 리스너를 관리하려면 관련 PCF 명령을 사용하기 위한 권한이 필요합니다. 그렇지만 대부분의 애플리케이션에는 이러한 액세스 권한이 필요하지 않습니다.

자세한 정보는 106 페이지의 『채널 권한 부여』의 내용을 참조하십시오.

### 추가 고려사항

특정 IBM MQ 기능이나 기본 제품 확장을 사용 중인 경우에만 다음과 같은 보안 관점도 고려해야 합니다.

- 117 페이지의 『큐 관리자 클러스터의 보안』
- 118 페이지의 『IBM MQ 발행/구독에 대한 보안』

## 식별 및 인증 계획

사용하려는 사용자 ID, 인증 제어에 레벨을 적용하는 방법 및 적용하는 레벨을 결정하십시오.

다른 운영 체제는 다른 길이의 사용자 ID를 지원한다는 점에 유의하여 IBM MQ 애플리케이션 사용자 식별 방법을 결정해야 합니다. 채널 인증 레코드를 사용하여 임의 사용자 ID에서 다른 사용자 ID로 맵핑하거나 일부 연결 속성을 기반으로 사용자 ID를 지정할 수 있습니다. TLS를 사용하는 IBM MQ 채널은 디지털 인증서를 식별 및 인증 메커니즘으로 사용합니다. 각 디지털 인증서는 채널 인증 레코드를 사용하여 특정 ID로 맵핑 가능한 주제 식별 이름을 포함합니다. 또한, 키 저장소의 CA 인증서는 IBM MQ 인증에 사용될 수 있는 디지털 인증서를 판별합니다. 자세한 정보는 다음을 참조하십시오.

- 361 페이지의 『MCAUSER 사용자 ID에 리모트 큐 관리자 맵핑』
- 362 페이지의 『MCAUSER 사용자 ID에 클라이언트 사용자 ID 맵핑』
- 362 페이지의 『MCAUSER 사용자 ID에 SSL 또는 TLS 식별 이름 맵핑』
- 364 페이지의 『MCAUSER 사용자 ID에 IP 주소 맵핑』

### 클라이언트 애플리케이션에 대한 인증 계획

네 개의 레벨(통신 레벨, 보안 엑시트, 채널 인증 레코드, 보안 엑시트에 전달된 ID)에서 인증 제어를 적용할 수 있습니다.

고려해야 하는 보안 레벨에는 네 개의 레벨이 있습니다. 다이어그램은 서버에 연결된 IBM MQ MQI client를 보여줍니다. 보안은 다음 텍스트에 설명되어 있는 것처럼 네 개의 레벨에서 적용됩니다. MCA는 메시지 채널 에이전트입니다.

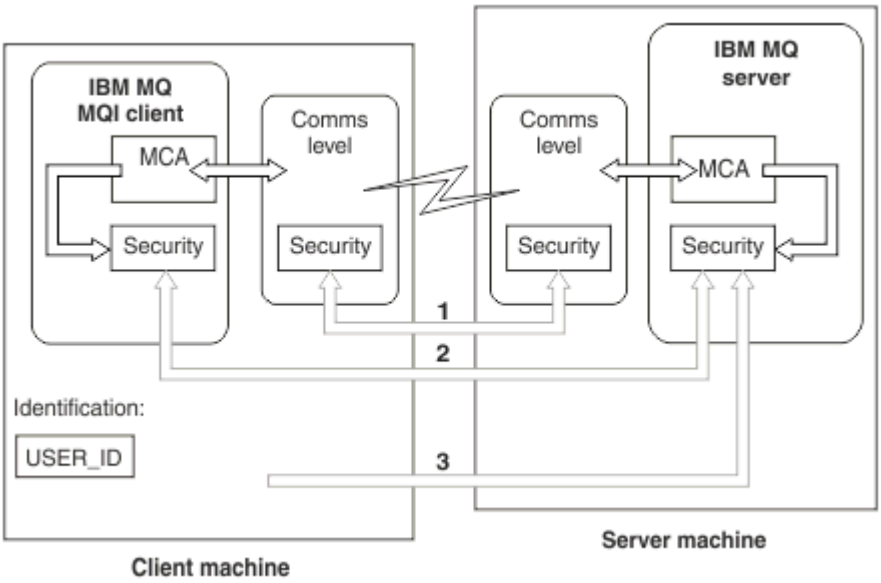



그림 9. 클라이언트/서버 연결에서의 보안

1. 통신 레벨

   화살표 1을 보십시오. 통신 레벨에서 보안을 구현하려면 TLS를 사용하십시오. 자세한 정보는 17 페이지의 『암호화 보안 프로토콜: TLS』의 내용을 참조하십시오.

2. 채널 인증 레코드

   화살표 2 및 3을 참조하십시오. 보안 레벨에서 IP 주소 또는 TLS 식별 이름을 사용하여 인증을 제어할 수 있습니다. 사용자 ID는 차단될 수도 있으며 확인된 사용자 ID는 올바른 사용자 ID에 맵핑될 수 있습니다. 전체 설명은 47 페이지의 『채널 인증 레코드』에서 제공됩니다.

3. 연결 인증

   화살표 3을 참조하십시오. 클라이언트는 사용자 ID및 비밀번호또는 인증 토큰을 전송합니다. 자세한 정보는 67 페이지의 『연결 인증: 구성』의 내용을 참조하십시오.

4. 채널 보안 엑시트

   화살표 2를 참조하십시오. 클라이언트 대 서버 통신의 채널 보안 엑시트는 서버 대 서버 통신과 같은 방식으로 작동할 수 있습니다. 프로토콜 독립 엑시트 쌍은 클라이언트와 서버 모두에 상호 인증을 제공하도록 작성될 수 있습니다. 전체 설명은 채널 보안 엑시트 프로그램에서 제공됩니다.

5. 채널 보안 엑시트에 전달된 ID

   화살표 3을 참조하십시오. 클라이언트 대 서버 통신에서 채널 보안 엑시트는 쌍으로 작동할 필요가 없습니다. IBM MQ 클라이언트 측의 엑시트는 생략할 수 있습니다. 이 경우 사용자 ID는 채널 디스크립터(MQCD)에 저장되며 서버 측 보안 엑시트는 필요한 경우 이를 대체할 수 있습니다.

   IBM MQ MQI clients에서는 식별을 지원하기 위해 추가 정보도 보냅니다.

   - 서버로 전달되는 사용자 ID는 클라이언트에 현재 로그온된 사용자 ID입니다.
   - 현재 로그온된 사용자의 보안 ID입니다.

   사용자 ID 및 가능한 경우 보안 ID 값은 IBM MQ MQI client ID를 설정하기 위해 서버 보안 엑시트에서 사용 가능합니다.

IBM MQ 8.0에서 MQCSP 구조에 포함된 비밀번호를 송신할 수 있습니다.

Linux AIX V 9.4.0 IBM MQ 9.3.4부터 AIX 또는 Linux 시스템에서 실행 중인 IBM MQ 큐 관리자에 연결하는 IBM MQ MQI clients 도 MQCSP 구조에서 인증 토큰을 송신할 수 있습니다.

**경고:** 일부 경우에는 클라이언트 애플리케이션에 대한 MQCSP 구조의 비밀번호 또는 인증 토큰 이 일반 텍스트로 네트워크를 통해 전송됩니다. 클라이언트 애플리케이션 비밀번호 및 인증 토큰 이 적절하게 보호되는지 확인하려면 29 페이지의 『MQCSP 비밀번호 보호』의 내용을 참조하십시오.

### 사용자 *ID*

클라이언트 애플리케이션에 대한 사용자 ID를 작성할 때 이 사용자 ID는 허용되는 최대 길이보다 길어서는 안 됩니다. 예약된 사용자 ID인 UNKNOWN 및 NOBODY는 사용해서는 안 됩니다. 클라이언트가 연결되는 서버가 IBM MQ for Windows 서버인 경우에는 at 기호(@) 사용을 이스케이프해야 합니다. 허용되는 사용자 ID 길이는 서버에 사용되는 플랫폼에 따라 다릅니다.

- Linux z/OS AIX z/OS, AIX and Linux에서 사용자 ID의 최대 길이는 12자입니다.
- IBM i IBM i에서 사용자 ID의 최대 길이는 10자입니다.
- Windows Windows에서 IBM MQ MQI client 및 IBM MQ 서버가 모두 Windows에 있고 서버에서 클라이언트 사용자 ID가 정의된 도메인에 액세스할 권한이 있으면 사용자 ID의 최대 길이는 20자입니다. 하지만 IBM MQ 서버가 Windows 서버가 아닌 경우 사용자 ID는 12자로 잘립니다.
- MQCSP 구조를 사용하여 신임 정보를 전달하는 경우 사용자 ID의 최대 길이는 1024자입니다. MQCSP 구조 사용자 ID는 권한 부여를 위해 IBM MQ에서 사용하는 최대 사용자 ID 길이를 피하는 데 사용할 수 없습니다. MQCSP 구조에 대한 자세한 정보는 302 페이지의 『MQCSP 구조를 사용하여 사용자 식별 및 인증』의 내용을 참조하십시오.

AIX and Linux 시스템에서 기본값은 사용자 ID가 인증에 사용되고 그룹이 권한 부여에 사용되는 것입니다. 그러나 사용자 ID에 대해 권한을 부여하도록 이러한 시스템을 구성할 수 있습니다. 자세한 정보는 331 페이지의 『AIX and Linux의 OAM 사용자 기반 권한』의 내용을 참조하십시오. Windows 시스템은 인증 및 권한 부여 모두에 사용자 ID를 사용하고 권한 부여에 그룹을 사용할 수 있습니다.

그룹에 주의하지 않고 서비스 계정을 작성하고 모든 사용자 ID를 다르게 권한 부여하면 모든 사용자가 다른 모든 사용자의 정보에 액세스할 수 있습니다.

### 제한된 사용자 ID

사용자 ID UNKNOWN 및 그룹 NOBODY는 IBM MQ에서 특별한 의미를 가집니다. 운영 체제에서 UNKNOWN이라는 사용자 ID와 NOBODY라는 그룹을 작성하면 의도하지 않은 결과가 발생할 수 있습니다.

### IBM MQ for Windows 서버에 연결할 때 사용자 ID

Windows

IBM MQ for Windows 서버는 클라이언트가 @ 문자를 포함하는 사용자 ID (예: abc@d) 로 실행 중인 경우 IBM MQ MQI client 의 연결을 지원하지 않습니다. 클라이언트에서 MQCONN 호출에 대한 리턴 코드는 MQRC_NOT_AUTHORIZED입니다.

그렇지만 두 개의 @ 문자를 사용하는 사용자 ID(예: abc@@d)는 지정할 수 있습니다. id@domain 형식 사용은 사용자 ID가 올바른 도메인으로 일관성있게 해결(abc@@d@domain)되도록 보장하는 선호 방식입니다.

## 권한 부여 계획

관리 권한이 있는 사용자를 계획하고 애플리케이션의 사용자가 IBM MQ MQI client에서 연결하는 오브젝트를 포함하여 IBM MQ 오브젝트를 적절하게 사용할 수 있도록 권한을 부여하는 방법을 계획하십시오.

IBM MQ를 사용하려면 개인 또는 애플리케이션에 액세스가 부여되어야 합니다. 필요한 액세스는 수행하는 역할 및 수행해야 하는 태스크에 따라 다릅니다. IBM MQ에서의 권한 부여는 다음과 같은 두 개의 기본 범주로 구분됩니다.

- 관리 조작을 수행하기 위한 권한 부여
- 애플리케이션에서 IBM MQ를 사용하기 위한 권한 부여

두 조작 클래스 모두 동일한 컴포넌트로 제어되며 개인에게는 두 조작 범주를 수행하는 권한이 부여될 수 있습니다.

다음 토픽은 고려해야 하는 특정 권한 부여 영역에 대한 자세한 정보를 제공합니다.

## IBM MQ 관리 권한

IBM MQ 관리자는 다양한 기능 수행 권한이 필요합니다. 이 권한은 다른 플랫폼에서 다양한 방식으로 확보합니다.

IBM MQ 관리자는 다음을 수행할 권한이 필요합니다.

- IBM MQ 관리하는 명령 발행
- Windows Linux IBM MQ Explorer을(를) 사용하십시오.
- z/OS z/OS에서 조작 및 제어판 사용
- z/OS z/OS에서 IBM MQ 유틸리티 프로그램, CSQUTIL을 사용하십시오.
- z/OS z/OS에서 큐 관리자 데이터 세트 액세스

자세한 정보는 운영 체제에 해당하는 주제를 참조하십시오.

### ALW *AIX, Linux, and Windows* 시스템에서 *IBM MQ* 를 관리할 수 있는 권한

IBM MQ 관리자는 mqm 그룹의 구성원입니다. 이 그룹은 모든 IBM MQ 자원에 대한 액세스를 가지며 IBM MQ 제어 명령을 발행할 수 있습니다. 관리자는 다른 사용자에게 특정 권한을 부여할 수 있습니다.

AIX, Linux, and Windows 시스템에서 IBM MQ 관리자가 되려면 사용자가 *mqm* 그룹의 구성원이어야 합니다. 이 그룹은 IBM MQ를 설치하면 자동으로 작성됩니다. 사용자가 제어 명령을 발행하도록 허용하려면 사용자를 mqm 그룹에 추가해야 합니다. 여기에는 AIX and Linux의 루트 사용자도 포함됩니다.

mqm 그룹의 구성원이 아닌 사용자에게 관리 권한을 부여할 수 있지만 IBM MQ 제어 명령을 발행할 수 없고 액세스가 부여된 명령만 실행할 권리가 있습니다.

또한 Windows 시스템에서 SYSTEM및 관리자 계정에는 IBM MQ 자원에 대한 전체 액세스 권한이 있습니다.

mqm 그룹의 모든 구성원은 시스템에서 실행 중인 모든 큐 관리자를 관리할 수 있는 것을 포함하여 시스템에서 모든 IBM MQ 자원에 대한 액세스를 가집니다. 이 액세스는 사용자를 mqm 그룹에서 제거해야만 취소할 수 있습니다. Windows 시스템에서 관리자 그룹의 구성원은 모든 IBM MQ 자원에 대한 액세스 권한도 가집니다.

관리자는 제어 명령 **runmqsc**를 사용하여 MQSC(IBM MQ Script) 명령을 발행할 수 있습니다. MQSC 명령을 리모트 큐 관리자에게 송신하는 데 간접적인 모드로 **runmqsc**가 사용되면, 각 MQSC 명령은 이스케이프 PCF 명령 안에 캡슐화됩니다. 관리자는 리모트 큐 관리자가 처리하는 MQSC 명령에 대한 필수 권한이 있어야 합니다.

IBM MQ Explorer는 PCF 명령을 발행하여 관리 태스크를 수행합니다. 관리자는 IBM MQ Explorer를 사용하여 로컬 시스템에서 큐 관리자를 관리하는 데 추가 권한이 필요하지 않습니다. IBM MQ Explorer가 또 다른 시스템에서 큐 관리자를 관리하는 데 사용되면 관리자에게는 PCF 명령이 리모트 큐 관리자에 의해 처리되기 위해서 필요한 권한이 있어야 합니다.

PCF 및 MQSC 명령이 처리될 때 수행되는 권한 검사에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 큐 관리자, 큐, 채널, 프로세스, 이름 목록, 인증 정보 오브젝트에서 운영되는 명령에 대해서는 87 페이지의 『애플리케이션에서 IBM MQ를 사용하기 위한 권한 부여』의 내용을 참조하십시오.
- 채널, 채널 이니시에이터, 리스너 및 클러스터에 대해 실행되는 명령에 대해서는 채널 보안을 참조하십시오.
- z/OS IBM MQ for z/OS에서 명령 서버로 처리되는 MQSC 명령에 대해서는 86 페이지의 『Command security and command resource security on z/OS』의 내용을 참조하십시오.

IBM MQ for AIX, Linux, and Windows 시스템 관리에 필요한 권한에 관한 자세한 정보는 관련 정보를 참조하십시오.

### IBM i *IBM i* 에서 *IBM MQ* 를 관리할 수 있는 권한

IBM i에서 IBM MQ 관리자가 되려면 *QMQMADM* 그룹의 구성원이어야 합니다. 이 그룹에는 AIX, Linux, and Windows 시스템에 있는 mqm 그룹의 속성과 비슷한 속성이 있습니다. 특히, QMQMADM 그룹은 IBM MQ for IBM i 설치 시에 작성되고 QMQMADM 그룹 구성원은 시스템의 모든 IBM MQ 자원에 대한 액세스 권한을 가집니다. *ALLOBJ 권한이 있는 사용자인 경우 모든 IBM MQ 자원에도 액세스할 수 있습니다.

관리자는 CL 명령을 사용하여 IBM MQ를 관리할 수 있습니다.이들 명령 중 하나는 다른 사용자에게 권한을 부여하는 데 사용되는 GRTMQMAUT입니다. 다른 명령인 STRMQMMQSC로 관리자가 로컬 큐 관리자에게 MQSC 명령을 발행할 수 있습니다.

IBM MQ for IBM i에서 제공되는 CL 명령의 두 그룹이 있습니다.

**그룹 1**
이 범주의 명령을 발행하려면 사용자가 QMQMADM 그룹의 구성원이거나 *ALLOBJ 권한이 있어야 합니다. 예를 들면 GRTMQMAUT 및 STRMQMMQSC가 이 범주에 속합니다.

**그룹 2**
이 범주의 명령을 발행하려면 사용자가 QMQMADM 그룹의 구성원이거나 *ALLOBJ 권한이 있어야 합니다. 대신 두 레벨의 권한이 필요합니다.

- 명령을 사용하려면 사용자에게는 IBM i 권한이 필요합니다. 이 권한은 GRTOBJAUT 명령을 사용하여 부여가 가능합니다.
- 명령에 연관된 임의 IBM MQ 오브젝트에 액세스하려면 IBM MQ 권한이 필요합니다. 이 권한은 GRTMQMAUT 명령으로 부여됩니다.

다음 예에서는 이 그룹의 명령을 보여줍니다.

- CRTMQMQ, MQM 큐 작성
- CHGMQMPRC, MQM 프로세스 변경
- DLTMQMNL, MQM 이름 목록 삭제
- DSPMQMAUTI, MQM 인증 정보 표시
- CRTMQMCHL, MQM 채널 작성

이 명령 그룹에 대한 자세한 정보는 87 페이지의 『애플리케이션에서 IBM MQ를 사용하기 위한 권한 부여』의 내용을 참조하십시오.

그룹 1과 그룹 2 명령의 전체 목록은 148 페이지의 『IBM i 의 IBM MQ 오브젝트에 대한 액세스 권한』의 내용을 참조하십시오.

IBM i에서 IBM MQ 를 관리하는 데 필요한 권한에 대한 자세한 정보는 IBM i 관리 를 참조하십시오.

## z/OS *Authority to administer IBM MQ on z/OS*

This collection of topics describes various aspects of the authority you need to administer IBM MQ for z/OS.

### z/OS *Authority checks on z/OS*

IBM MQ for z/OS uses the System Authorization Facility (SAF) to route requests for authority checks to an external security manager (ESM) such as the z/OS Security Server Resource Access Control Facility ( RACF ). IBM MQ does no authority checks of its own.

It is assumed that you are using RACF as your ESM. If you are using a different ESM, you might need to interpret the information provided for RACF in a way that is relevant to your ESM.

You can specify whether you want authority checks turned on or off for each queue manager individually or for every queue manager in a queue sharing group. This level of control is called *subsystem security*. If you turn subsystem security off for a particular queue manager, no authority checks are carried out for that queue manager.

If you turn subsystem security on for a particular queue manager, authority checks can be performed at two levels:

**Queue sharing group level security**
Authority checks use RACF profiles that are shared by all queue managers in the queue sharing group. This means that there are fewer profiles to define and maintain, making security administration easier.

**Queue manager level security**
Authority checks use RACF profiles specific to the queue manager.

You can use a combination of queue sharing group and queue manager level security. For example, you can arrange for profiles specific to a queue manager to override those of the queue sharing group to which it belongs.

Subsystem security, queue sharing group level security, and queue manager level security are turned on or off by defining *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to IBM MQ.

z/OS *Command security and command resource security on z/OS*

Command security relates to the authority to issue a command; command resource authority relates to the authority to perform an operation on a resource. Both are implemented y using RACF classes.

Authority checks are carried out when an IBM MQ administrator issues an MQSC command. This is called *command security*.

To implement command security, you must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. The name of a profile for command security contains the name of an MQSC command.

Some MQSC commands perform an operation on an IBM MQ resource, such as the DEFINE QLOCAL command to create a local queue. When an administrator issues an MQSC command, authority checks are carried out to determine whether the requested operation can be performed on the resource specified in the command. This is called *command resource security*.

To implement command resource security, you must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. The name of a profile for command resource security contains the name of an IBM MQ resource and its type (QUEUE, PROCESS, NAMELIST, TOPIC, AUTHINFO, or CHANNEL).

Command security and command resource security are independent. For example, when an administrator issues the command:

```
DEFINE QLOCAL(MOON.EUROPA)
```

the following authority checks are performed:

- Command security checks that the administrator is authorized to issue the DEFINE QLOCAL command.
- Command resource security checks that the administrator is authorized to perform an operation on the local queue called MOON.EUROPA.

Command security and command resource security can be turned on or off by defining switch profiles.

z/OS *MQSC commands and the system command input queue on z/OS*

Use this topic to understand how the command server processes MQSC commands directed to the system command input queue on z/OS.

Command security and command resource security are also used when the command server retrieves a message containing an MQSC command from the system command input queue. The user ID that is used for the authority checks is the one found in the *UserIdentifier* field in the message descriptor of the message containing the MQSC command. This user ID must have the required authorities on the queue manager where the command is processed. For more information about the *UserIdentifier* field and how it is set, see Message context.

Messages containing MQSC commands are sent to the system command input queue in the following circumstances:

- The operations and control panels send MQSC commands to the system command input queue of the target queue manager. The MQSC commands correspond to the actions you choose on the panels. The *UserIdentifier* field in each message is set to the TSO user ID of the administrator.
- The COMMAND function of the IBM MQ utility program, CSQUTIL, sends the MQSC commands in the input data set to the system command input queue of the target queue manager. The COPY and EMPTY

functions send DISPLAY QUEUE and DISPLAY STGCLASS commands. The *UserIdentifier* field in each message is set to the job user ID.

- The MQSC commands in the CSQINPX data sets are sent to the system command input queue of the queue manager to which the channel initiator is connected. The *UserIdentifier* field in each message is set to the channel initiator address space user ID.

  No authority checks are performed when MQSC commands are issued from the CSQINP1 and CSQINP2 data sets. You can control who is allowed to update these data sets using RACF data set protection.
- Within a queue sharing group, a channel initiator might send START CHANNEL commands to the system command input queue of the queue manager to which it is connected. A command is sent when an outbound channel that uses a shared transmission queue is started by triggering. The *UserIdentifier* field in each message is set to the channel initiator address space user ID.
- An application can send MQSC commands to a system command input queue. By default, the *UserIdentifier* field in each message is set to the user ID associated with the application.
- On AIX, Linux, and Windows systems, the **runmqsc** control command can be used in indirect mode to send MQSC commands to the system command input queue of a queue manager on z/OS. The *UserIdentifier* field in each message is set to the user ID of the administrator who issued the **runmqsc** command.

z/OS *Access to the queue manager data sets on z/OS*

IBM MQ for z/OS administrators need authority to access the queue manager data sets. Use this topic to understand which data sets need RACF protection.

These data sets include:

- The data sets referred to by CSQINP1, CSQINP2, and CSQINPT in the started task procedure of the queue manager.
- The queue manager's page sets, active log data sets, archive log data sets, and bootstrap data sets (BSDSs)
- The data sets referred to by CSQXLIB and CSQINPX in the channel initiator's started task procedure

You must protect the data sets so that no unauthorized user can start a queue manager or gain access to any queue manager data. To do this, use RACF data set protection.

## 애플리케이션에서 IBM MQ를 사용하기 위한 권한 부여

애플리케이션이 오브젝트에 액세스하는 경우 애플리케이션에 연관되는 사용자 ID에는 적절한 권한이 있어야 합니다.

애플리케이션은 MQI 호출을 발행하여 다음 IBM MQ 오브젝트에 액세스할 수 있습니다.

- 큐 관리자
- 큐
- 프로세스
- 이름 목록
- 토픽

애플리케이션은 PCF 명령을 사용하여 IBM MQ 오브젝트를 관리할 수도 있습니다. PCF 명령이 처리되면 PCF 메시지를 넣는 사용자 ID의 권한 컨텍스트를 사용합니다.

이 컨텍스트에서 애플리케이션에는 사용자 및 벤더가 작성한 애플리케이션및 IBM MQ for z/OS와 함께 제공되는 애플리케이션이 포함됩니다.

z/OS IBM MQ for z/OS에서 제공하는 애플리케이션에는 다음이 포함됩니다.

- 조작 및 제어판
- IBM MQ 유틸리티 프로그램, CSQUTIL
- 데드-레터 큐 핸들러 유틸리티인 CSQUDLQH

IBM MQ classes for Java, IBM MQ classes for JMS, IBM MQ classes for .NET 또는 C/C++ 및 .NET용 메시지 서비스 클라이언트를 사용하는 애플리케이션은 MQI를 간접적으로 사용합니다.

MCA는 MQI 호출을 발행하고 MCA에 연관된 사용자 ID는 해당 IBM MQ 오브젝트에 액세스하는 권한이 있어야 합니다. 이 사용자 ID와 필요한 권한에 대한 자세한 정보는 106 페이지의 『채널 권한 부여』의 내용을 참조하십시오.

z/OS z/OS에서 애플리케이션은 MQSC 명령을 사용하여 해당 IBM MQ 오브젝트에 액세스할 수는 있지만 명령 보안 및 명령 자원 보안은 이런 상황에서 권한 검사를 제공합니다. z/OS 자세한 정보는 86 페이지의 『Command security and command resource security on z/OS』 및 86 페이지의 『MQSC commands and the system command input queue on z/OS』의 내용을 참조하십시오.

IBM i IBM i에서 그룹 2에서 CL 명령을 발행하는 사용자의 경우 명령에 연관된 IBM MQ 오브젝트 액세스 권한이 필요할 수도 있습니다. 자세한 정보는 88 페이지의 『권한 검사가 수행되는 시기』의 내용을 참조하십시오.

## 권한 검사가 수행되는 시기

권한 검사는 애플리케이션이 큐 관리자, 큐, 프로세스 또는 이름 목록에 액세스를 시도할 때 수행됩니다.

IBM i에서 권한 검사는 이 IBM MQ 오브젝트 중 임의의 오브젝트에 액세스하는 그룹 2에서 CL 명령을 발행할 때 수행될 수도 있습니다. 다음과 같은 상황에서 검사가 수행됩니다.

**애플리케이션이 MQCONN 또는 MQCONNX 호출을 사용하여 큐 관리자에 연결하는 경우**

큐 관리자가 애플리케이션에 연관된 사용자 ID를 운영 체제에 요청하는 경우. 그러면 큐 관리자는 사용자 ID에 연결 권한이 있는지 및 이후 검사를 위한 사용자 ID를 보유하고 있는지를 검사합니다.

사용자는 IBM MQ에 사인온할 필요는 없습니다. IBM MQ에서는 사용자가 근본적인 운영 체제에 사인온하고 이미 인증되었다고 간주합니다.

**애플리케이션이 MQOPEN 또는 MQPUT1 호출을 사용하여 IBM MQ 오브젝트를 여는 경우**

모든 권한 검사는 나중에 액세스할 때가 아니라 오브젝트를 열 때 수행됩니다. 예를 들어, 권한 검사는 애플리케이션이 큐를 열 때 수행됩니다. 애플리케이션이 큐에 메시지를 넣거나 큐에서 메시지를 가져올 때는 수행되지 않습니다.

애플리케이션이 오브젝트를 열 때, 오브젝트에서 수행해야 하는 조작의 유형을 지정합니다. 예를 들어, 애플리케이션이 큐를 열어 큐 안의 메시지를 찾아보거나 메시지를 가져오기만 하고, 메시지를 넣지는 않을 수 있습니다. 조작의 유형별로 큐 관리자는 애플리케이션에 연관된 사용자 ID에 해당 조작을 수행할 권한이 있는지 검사합니다.

애플리케이션이 큐를 열 때 권한 검사가 오브젝트 디스크립터의 `ObjectName` 필드에 이름 지정된 오브젝트에 대해 수행됩니다. `ObjectName` 필드는 `MQOPEN` 또는 `MQPUT1` 호출에서 사용됩니다. 오브젝트가 알리어스 큐 또는 리모트 큐 정의인 경우, 권한 검사는 오브젝트 자체에 대해 수행됩니다. 알리어스 큐나 리모트 큐 정의가 해결되는 큐에서 수행되지 않습니다. 이는 사용자가 여기에 액세스할 권한이 필요하지 않음을 의미합니다. 큐 작성 권한을 권한이 있는 사용자로 제한하십시오. 그렇게 하지 않으면 사용자가 알리어스를 작성하는 것만으로 일반 액세스 제어를 무시할 수도 있습니다.

애플리케이션은 리모트 큐를 명확하게 참조할 수 있습니다. 이는 오브젝트 디스크립터의 `ObjectName` 및 `ObjectQMgrName` 필드를 리모트 큐 및 리모트 큐 관리자 이름으로 설정합니다. 리모트 큐 관리자와 이름이 동일한 전송 큐에 대해 권한 검사가 수행됩니다.

- z/OS z/OS에서는 리모트 큐 관리자 이름과 일치하는 RACF 큐 프로파일에 대한 검사가 수행되며 이 전송 큐가 로컬로 정의되어 있는지 여부에 상관없이 검사가 수행됩니다.
- Multi 멀티플랫폼에서, 클러스터링이 사용되고 있는 경우 리모트 큐 관리자 이름과 일치하는 RQMNAME 프로파일에 대해 검사가 수행됩니다.

애플리케이션은 오브젝트 디스크립터의 `ObjectName` 필드를 클러스터 큐 이름으로 설정하여 클러스터 큐를 명시적으로 참조할 수 있습니다. 권한 검사는 클러스터 전송 큐 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`에 대해 수행됩니다.

동적 큐의 권한은 도출된 모델 큐를 기반으로 하지만 동일할 필요는 없습니다. 참고 1을 참조하십시오.

큐 관리자가 권한 검사에 사용하는 사용자 ID는 운영 체제에서 확보합니다. 사용자 ID는 애플리케이션이 큐 관리자에 연결될 때 확보합니다. 적절히 권한이 부여된 애플리케이션은 대체 사용자 ID를 지정하는 MQOPEN 호출을 발행할 수 있습니다. 그런 다음 대체 사용자 ID에 대해 액세스 제어 검사를 수행합니다. 대체 사용자 ID의 사용은 애플리케이션과 연관된 사용자 ID를 변경하지 않으며 액세스 제어 검사에 사용된 사용자 ID만 변경됩니다.

**애플리케이션에서 MQSUB 호출을 사용하여 토픽을 구독할 때**

애플리케이션이 토픽을 구독할 때 이는 수행해야 할 조작의 유형을 지정합니다. 이는 구독을 작성, 기존 구독 수정 또는 기존 구독을 변경하지 않고 재개합니다. 조작의 유형마다 큐 관리자는 애플리케이션에 연관된 사용자 ID에 해당 조작을 수행할 권한이 있는지 검사합니다.

애플리케이션이 토픽을 구독할 때 권한 검사는 토픽 트리에 있는 토픽 오브젝트에 대해 수행됩니다. 토픽 오브젝트는 애플리케이션이 구독하는 토픽 트리 지점 이상에 있습니다. 권한 검사에는 둘 이상의 토픽 오브젝트에 대한 검사가 포함될 수 있습니다. 큐 관리자가 권한 검사에 사용하는 사용자 ID는 운영 체제에서 확보합니다. 사용자 ID는 애플리케이션이 큐 관리자에 연결될 때 확보합니다.

큐 관리자는 관리 큐가 아니라 구독자 큐에서 권한 검사를 수행합니다.

**애플리케이션이 MQCLOSE 호출을 사용하여 영구적 동적 큐를 삭제하는 경우**

MQCLOSE 호출에 지정되는 오브젝트 핸들은 영구적 동적 큐를 작성한 MQOPEN 호출에서 리턴되는 것과 동일할 필요는 없습니다. 다른 경우, 큐 관리자는 MQCLOSE 호출을 발행한 애플리케이션에 연관된 사용자 ID를 검사합니다. 사용자 ID가 큐를 삭제할 권한이 있는지 검사합니다.

제거를 위해 구독을 닫은 애플리케이션이 이를 작성하지 않으면 제거에 필요한 적절한 권한이 필요합니다.

**IBM MQ 오브젝트에서 운영되는 PCF 명령이 명령 서버로 처리되는 경우**

이 규칙에는 PCF 명령이 인증 정보 오브젝트에서 운영되는 경우를 포함합니다.

권한 검사에 사용되는 사용자 ID는 PCF 명령의 메시지 디스크립터에서 UserIdentifier 필드에 있는 사용자 ID입니다. 이 사용자 ID는 명령이 처리되는 큐 관리자에서 필수 권한을 가져야 합니다. 이스케이프 PCF 명령 안에 캡슐화되어 있는 동등한 MQSC 명령은 같은 방식으로 처리됩니다. UserIdentifier 필드에 대한 자세한 정보 및 설정 방법에 대한 정보는 90 페이지의 『메시지 컨텍스트』의 내용을 참조하십시오.

IBM i **IBM i에서 사용자가 IBM MQ 오브젝트를 조작하는 그룹 2의 CL 명령을 발행하는 경우**

이 규칙에는 그룹 2의 CL 명령이 인증 정보 오브젝트에서 운영되는 경우가 포함됩니다.

검사는 사용자가 명령에 연관된 IBM MQ 오브젝트에서 조작할 권한이 있는지 판별하기 위해 수행됩니다. 사용자가 QMQMADM 그룹의 구성원이거나 *ALLOBJ 권한이 있는 경우를 제외하고는 검사가 수행됩니다. 필요한 권한은 오브젝트에 대해 명령이 수행하는 조작 유형에 따라 다릅니다. 예를 들어, **CHGMQMQ**(MQM 큐 변경) 명령은 명령이 지정하는 큐의 속성을 변경할 수 있는 권한을 필요로 합니다. 반대로, **DSPMQMQ**(MQM 큐 표시) 명령은 명령이 지정하는 큐의 속성을 표시할 수 있는 권한을 필요로 합니다.

여러 명령이 둘 이상의 오브젝트에 대해 작용합니다. 예를 들어, **DLTMQMQ**(MQM 큐 삭제) 명령을 발행하려면 다음 권한이 필요합니다.

- 명령에 의해 지정된 큐 관리자에 연결할 수 있는 권한
- 명령에 의해 지정된 큐를 삭제할 수 있는 권한

일부 명령은 오브젝트에 전혀 작용하지 않습니다. 이 경우 사용자는 이러한 명령 중 하나를 실행하기 위해 IBM i 권한만 필요합니다. **STRMQMLSR**, MQM 리스너 시작은 이러한 명령의 예입니다.

## 대체 사용자 권한

애플리케이션이 오브젝트를 열거나 토픽을 구독하는 경우, 애플리케이션은 MQOPEN, MQPUT1 또는 MQSUB 호출에 사용자 ID를 제공할 수 있습니다. 큐 관리자가 이 사용자 ID를 애플리케이션에 연관된 사용자 ID 대신 권한 검사에 사용하도록 요청할 수 있습니다.

다음 두 조건이 모두 충족된 경우에만 애플리케이션이 오브젝트를 열 수 있습니다.

- 애플리케이션에 연관된 사용자 ID가 권한 검사를 위해 다른 사용자 ID를 제공할 수 있는 권한을 가집니다. 애플리케이션은 대체 사용자 권한이 있는 것으로 언급됩니다.
- 애플리케이션에서 제공한 사용자 ID에는 요청된 조작의 유형에 맞게 오브젝트를 열거나 토픽을 구독할 권한이 있습니다.

## 메시지 컨텍스트

메시지 컨텍스트 정보를 사용하여 메시지를 검색하는 애플리케이션이 메시지의 진원지를 알 수 있습니다. 정보는 메시지 디스크립터의 필드에 보유되고 필드는 세 개의 논리 파트로 분할됩니다.

해당 파트는 다음과 같습니다.

**ID 컨텍스트(identity context)**
이 필드는 메시지를 큐에 넣는 애플리케이션의 사용자 정보를 포함합니다.

**원본 컨텍스트**
이 필드는 애플리케이션 자체에 대한 정보 및 메시지가 큐에 추가되는 시기에 대한 정보를 포함합니다.

**사용자 컨텍스트**
이러한 필드에는 애플리케이션에서 큐 관리자가 전달해야 하는 메시지를 선택하는 데 사용할 수 있는 메시지 특성이 포함됩니다.

애플리케이션이 메시지를 큐에 넣을 때, 애플리케이션은 큐 관리자에게 메시지에 컨텍스트 정보를 생성하도록 요청할 수 있습니다. 이는 기본 조치입니다. 또는, 컨텍스트 필드가 정보를 포함하지 않도록 지정할 수 있습니다. 애플리케이션과 연관된 사용자 ID는 이를 수행하기 위한 특별한 권한을 필요로 하지 않습니다.

애플리케이션이 ID 컨텍스트 필드를 메시지에 설정할 수 있으면, 큐 관리자가 원본 컨텍스트를 생성할 수 있거나, 모든 컨텍스트 필드를 설정할 수 있습니다. 애플리케이션이 검색한 메시지로부터 큐에 넣고 있는 메시지로 ID 컨텍스트 필드를 전달하거나, 모든 컨텍스트 필드를 전달할 수 있습니다. 그렇지만 애플리케이션에 연관된 사용자 ID는 컨텍스트 정보를 설정하거나 전달하는 데 권한을 필요로 하지 않습니다. 애플리케이션이 메시지를 넣기 직전의 큐를 열 때 컨텍스트 정보를 설정하거나 전달하려고 하는지를 지정하고, 그에 대한 권한이 이 때에 검사됩니다.

다음은 컨텍스트 필드 각각에 대한 간단한 설명입니다.

**ID 컨텍스트**

**UserIdentifier**
메시지를 넣는 애플리케이션에 연관된 사용자 ID. 큐 관리자가 이 필드를 설정하면, 애플리케이션이 큐 관리자에 연결될 때 운영 체제에서 확보한 사용자 ID로 설정됩니다.

**AccountingToken**
메시지의 결과로 완료된 작업에 청구하는 데 사용할 수 있는 정보.

**ApplIdentityData**
애플리케이션에 연관된 사용자 ID가 ID 컨텍스트 필드를 설정하거나 모든 컨텍스트 필드를 설정할 수 있는 권한을 가지면, 애플리케이션이 이 필드를 ID에 관련된 임의 값으로 설정할 수 있습니다. 큐 관리자가 이 필드를 설정하면, 공백으로 설정됩니다.

**원본 컨텍스트**

**PutApplType**
메시지를 넣는 애플리케이션 유형(예: CICS® 트랜잭션).

**PutApplName**
메시지를 넣는 애플리케이션 이름.

**PutDate**
메시지를 넣은 날짜.

**PutTime**
메시지를 넣은 시간.

**ApplOriginData**
애플리케이션에 연관된 사용자 ID가 모든 컨텍스트 필드를 설정할 수 있는 권한을 가지면, 애플리케이션이 이 필드를 원본에 관련된 임의 값으로 설정할 수 있습니다. 큐 관리자가 이 필드를 설정하면, 공백으로 설정됩니다.

**사용자 컨텍스트**

다음 값은 **MQINQMP** 또는 **MQSETMP**에 대해 지원됩니다.

**MQPD_USER _CONTEXT**

특성은 사용자 컨텍스트와 연관됩니다.

MQSETMP 호출을 사용하여 사용자 컨텍스트와 연관된 특성을 설정할 수 있는 특수 권한은 필요하지 않습니다.

V7.0 또는 후속 큐 관리자에서 사용자 컨텍스트에 연관된 특성은 MQOO_SAVE_ALL_CONTEXT에 대해 설명된 대로 저장됩니다. MQOO_PASS_ALL_CONTEXT가 지정된 MQPUT를 통해 특성이 저장된 컨텍스트에서 새 메시지로 복사됩니다.

**MQPD_NO_CONTEXT**

특성은 메시지 컨텍스트와 연관되지 않습니다.

인식할 수 없는 값은 MQRC_PD_ERROR로 거부됩니다. 이 필드의 초기값은 **MQPD_NO_CONTEXT**입니다.

각 컨텍스트 필드에 대한 자세한 정보는 MQMD - 메시지 디스크립터를 참조하십시오. 메시지 컨텍스트 사용 방법에 대한 자세한 정보는 메시지 컨텍스트를 참조하십시오.

## IBM i ALW IBM i *IBM i , AIX, Linux, and Windows* 시스템에서 *IBM MQ* 오브젝트에 대해 작업할 수 있는 권한

IBM MQ에서 제공되는 권한 서비스 컴포넌트는 오브젝트 권한 관리자(OAM)라고 합니다. 이는 인증 및 권한 검사를 통해 액세스 제어를 제공합니다.

**인증.**

IBM MQ에서 제공되는 OAM이 수행하는 인증 검사는 기본이며 특정 상황에서만 수행됩니다. 매우 안전한 환경에서 예상되는 엄격한 요구사항을 충족시키기 위한 것은 아닙니다.

OAM은 애플리케이션이 큐 관리자에 연결되고 다음 조건에 해당될 때 인증 검사를 수행합니다.

- MQCSP 구조가 연결 애플리케이션에 연결 애플리케이션에서 제공되었습니다. 그리고
- MQCSP 구조의 *AuthenticationType* 속성에 MQCSP_AUTH_USER_ID_AND_PWD 값이 지정되었습니다. 그리고
- 구성된 AUTHINFO 오브젝트의 CHCKLOCL 또는 CHKCCLNT값이 'NONE' 이 아닙니다.

OAM의 인증 단계에서는 운영 체제 서비스를 사용하여 비밀번호의 유효성을 검증합니다. 운영 체제 서비스는 사용자 이름에서 올바르지 않은 비밀번호 테스트를 너무 많이 시도하지 않았는지 확인하는 등의 추가 검사를 수행하도록 구성되었을 수 있습니다.

새 인증 서비스 컴포넌트를 작성하거나 공급업체로부터 확보하는 경우 대체 인증 메커니즘을 사용할 수 있습니다.

**권한 부여.**

권한 검사는 포괄적이며 대부분의 일반 요구사항을 충족시키기 위해 수행됩니다.

권한 검사는 애플리케이션이 큐 관리자, 큐, 프로세스, 토픽 또는 이름 목록 액세스를 위해 MQI 호출을 발행하는 경우에 수행됩니다. 이는 명령 서버로 명령이 수행 중인 경우처럼 다른 상황에서도 수행됩니다.

IBM i IBM i, AIX, Linux, and Windows 시스템에서, 권한 서비스는 애플리케이션이 큐 관리자, 큐, 프로세스, 토픽 또는 이름 목록인 IBM MQ 오브젝트에 액세스하는 MQI 호출을 발생할 때 액세스 제어를 제공합니다. 여기에는 대체 사용자 권한 및 컨텍스트 정보 설정 또는 전달 권한에 대한 검사가 포함됩니다.

Windows Windows에서 OAM은 관리 그룹 구성원에서 모든 IBM MQ 오브젝트 액세스 권한을 부여하며 UAC가 사용되는 경우도 포함됩니다. 또한 Windows 시스템에서 SYSTEM 계정에는 IBM MQ 자원에 대한 전체 액세스 권한이 있습니다.

권한 서비스는 PCF 명령이 다음 IBM MQ 오브젝트 중 하나 또는 인증 정보 오브젝트에서 수행되는 경우에도 권한 검사를 제공합니다. 이스케이프 PCF 명령 안에 캡슐화되어 있는 동등한 MQSC 명령은 같은 방식으로 처리됩니다.

IBM i IBM i에서 사용자가 QMQMADM 그룹의 구성원이거나 *ALLOBJ 권한을 가진 경우를 제외하고는 권한 서비스는 IBM MQ 오브젝트 또는 인증 정보 오브젝트에서 조작되는 그룹 2에서 CL 명령을 사용자가 발행하는 경우에도 권한 검사를 제공합니다.

권한 서비스는 설치 가능 서비스이며, 이는 하나 이상의 설치 가능 서비스 구성요소에 의해 구현된다는 것을 의미합니다. 각 컴포넌트는 문서화된 인터페이스를 사용하여 호출됩니다. 이를 사용하면 사용자 및 벤더가 IBM MQ 제품에서 제공되는 컴포넌트를 기능 보강하거나 대체하기 위해 컴포넌트를 제공할 수 있습니다.

IBM MQ에서 제공되는 권한 서비스 컴포넌트는 오브젝트 권한 관리자(OAM)라고 합니다. OAM은 작성하는 각 큐 관리자에 자동으로 사용 가능하게 되어 있습니다.

OAM은 액세스를 제어 중인 각 IBM MQ 오브젝트에 대해 액세스 제어 목록(ACL)을 유지보수합니다. AIX and Linux 시스템에서 그룹 ID만 ACL에 표시 가능합니다. 이는 그룹의 모든 구성원에 같은 권한이 있다는 것을 의미합니다. IBM i IBM i 및 Windows 시스템에서는 사용자 ID와 그룹 ID 모두 ACL에 표시 가능합니다. 이는 권한이 개별 사용자 및 그룹에 부여될 수 있음을 의미합니다.

12자 제한은 그룹 및 사용자 ID 모두에 적용됩니다. 일반적으로, UNIX 플랫폼은 사용자 ID의 길이를 12자로 제한합니다. AIX 및 Linux에서는 이러한 제한이 약화되었지만, IBM MQ는 계속해서 모든 UNIX 플랫폼에 대해 12자 제한을 적용합니다. 12자보다 많은 사용자 ID를 사용하면 IBM MQ는 이를 "UNKNOWN" 값으로 바꿉니다. 사용자 ID를 "UNKNOWN" 값으로 정의하지 마십시오.

OAM은 사용자를 인증하고 적절한 ID 컨텍스트 필드를 변경할 수 있습니다. MQCONNX 호출에서 연결 보안 매개변수 구조(MQCSP)를 지정하여 이를 사용 가능하게 할 수 있습니다. 이 구조는 적절한 ID 컨텍스트 필드를 설정하는 OAM 인증 사용자 함수(MQZ_AUTHENTICATE_USER)에 전달됩니다. MQCONNX 연결이 IBM MQ 클라이언트에서의 연결인 경우, MQCSP의 정보가 클라이언트 연결 및 서버 연결 채널을 통해 클라이언트가 연결되는 큐 관리자로 플로우됩니다. 보안 엑시트가 해당 채널에 정의되면 MQCSP가 각 보안 엑시트로 전달되고 엑시트에 의해 변경될 수 있습니다. 보안 엑시트가 MQCSP를 작성할 수도 있습니다. 이 컨텍스트에서 보안 엑시트 사용에 대한 자세한 정보는 채널 보안 엑시트 프로그램을 참조하십시오.

**경고:** 일부 경우에 클라이언트 애플리케이션에 대한 MQCSP 구조의 비밀번호는 일반 텍스트로 네트워크 전체에 전송됩니다. 클라이언트 애플리케이션 비밀번호가 적절하게 보호될 수 있도록 하려면 IBM MQCSP 비밀번호 보호를 참조하십시오.

AIX, Linux, and Windows 시스템에서 제어 명령인 **setmqaut**를 통해 권한을 부여하고 취소하며 ACL을 유지 보수합니다. 예를 들어,

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

명령은 그룹 VOYAGER의 구성원이 큐 관리자 JUPITER가 소유하는 큐 MOON.EUROPA에 있는 메시지를 볼 수 있게 합니다. 구성원들이 큐에서 메시지를 가져오게도 합니다. 이 권한을 나중에 취소하려면 다음 명령을 입력하십시오.

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

:NONE.

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

이는 그룹 VOYAGER의 구성원이 MOON. 문자로 시작되는 이름의 임의 큐에 메시지를 넣도록 허용합니다. 모온.* 일반 프로파일의 이름입니다.일반 프로파일 을 사용하면 단일 **setmqaut** 명령을 사용하여 오브젝트 세트에 대한 권한을 부여할 수 있습니다.

제어 명령 **dspmqaut**는 사용자 또는 그룹이 지정된 오브젝트에 대해 갖는 현재 권한을 표시할 수 있습니다.제어 명령 **dmpmqaut**도 일반 프로파일에 연관된 현재 권한을 표시하는 데 사용 가능합니다.

IBM i IBM i에서 관리자는 CL 명령 GRTMQMAUT를 사용하여 권한을 부여하고 CL 명령 RVKMQMAUT를 사용하여 권한을 취소합니다. 일반 프로파일도 마찬가지로 사용될 수 있습니다.예를 들어, CL 명령:

```
GRTMQMAUT MQMNAME(JUPITER) OBJTYPE(*Q) OBJ('MOON.*') USER(VOYAGER) AUT(*PUT)
```

**setmqaut** 명령의 이전 예제와 동일한 기능을 제공합니다. 이 명령을 사용하면 그룹 보이저의 구성원이 MOON. 문자로 시작하는 이름을 가진 모든 큐에 메시지를 넣을 수 있습니다.

IBM i CL 명령 DSPMQMAUT는 사용자 또는 그룹이 지정된 오브젝트에 대해 갖는 현재 권한을 표시합니다. CL 명령 WRKMQMAUT와 WRKMQMAUTD는 오브젝트 및 일반 프로파일과 연관된 현재 권한에 대해서도 작업할 수 있습니다.

권한 검사를 하지 않는 경우(예를 들어, 테스트 환경), OAM을 사용하지 않을 수 있습니다.

Multi *PCF*를 사용하여 *OAM* 명령에 액세스

IBM i, AIX, Linux, and Windows 시스템에서는 PCF 명령을 사용하여 OAM 관리 명령에 액세스할 수 있습니다.

PCF 명령과 대응되는 OAM 명령은 다음과 같습니다.

표 *8. PCF* 명령 및 대응되는 *OAM* 명령

| **PCF 명령** | **OAM 명령** |
|---|---|
| 권한 레코드 조회 | dmpmqaut |
| 엔티티 권한 조회 | dspmqaut |
| 권한 레코드 설정 | setmqaut |
| 권한 레코드 삭제 | -remove 옵션이 포함된 setmqaut |

**setmqaut** 및 **dmpmqaut** 명령은 mqm 그룹의 구성원으로 제한됩니다. 대응되는 PCF 명령은 큐 관리자에 대해 dsp 및 chg 권한이 부여된 그룹의 사용자만 실행할 수 있습니다.

해당 명령 사용에 대한 자세한 정보는 프로그래밍 가능 명령 형식(PCF) 소개를 참조하십시오.

### z/OS *Authority to work with IBM MQ objects on z/OS*

On z/OS, there are seven categories of authority check associated with calls to the MQI. You must define certain RACF profiles and give appropriate access to these profiles. Use the *RESLEVEL* profile to control how many users IDs are checked.

The seven categories of authority check associated with calls to the MQI:

**Connection security**
The authority checks that are performed when an application connects to a queue manager

**Queue security**
The authority checks that are performed when an application opens a queue or deletes a permanent dynamic queue

**Process security**
The authority checks that are performed when an application opens a process object

**Namelist security**
The authority checks that are performed when an application opens a namelist object

**Alternate user security**
The authority checks that are performed when an application requests alternate user authority when opening an object

**Context security**
The authority checks that are performed when an application opens a queue and specifies that it intends to set or pass the context information in the messages it puts on the queue

**Topic security**
The authority checks that are performed when an application opens a topic

Each category of authority check is implemented in the same way that command security and command resource security are implemented. You must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. For queue security, the level of access determines the types of operation the application can perform on a queue. For context security, the level of access determines whether the application can:

- Pass all the context fields
- Pass all the context fields and set the identity context fields
- Pass and set all the context fields

Each category of authority check can be turned on or off by defining switch profiles.

All the categories, except connection security, are known collectively as *API-resource security*.

By default, when an API-resource security check is performed as a result of an MQI call from an application using a batch connection, only one user ID is checked. When a check is performed as a result of an MQI call from a CICS or IMS application, or from the channel initiator, two user IDs are checked.

By defining a *RESLEVEL profile*, however, you can control whether zero, one, or two users IDs are checked. The number of user IDs that are checked is determined by the user ID associated with the type of connection when an application connects to the queue manager and the access level that user ID has to the RESLEVEL profile. The user ID associated with each type of connection is:

- The user ID of the connecting task for batch connections
- The CICS address space user ID for CICS connections
- The IMS region address space user ID for IMS connections
- The channel initiator address space user ID for channel initiator connections

For more information about the authority to work with IBM MQ objects on z/OS, see "Authority to administer IBM MQ on z/OS" on page 85.

## 원격 메시징의 보안

이 절에서는 원격 메시징 보안 측면을 다룹니다.

IBM MQ 기능을 사용할 권한을 가진 사용자를 제공해야 합니다. 이 사용자는 오브젝트와 정의에 대해 취해질 조치에 따라 구성됩니다. 예를 들면, 다음과 같습니다.

- 큐 관리자는 권한이 부여된 사용자가 시작하고 중지할 수 있습니다.
- 애플리케이션은 큐 관리자에 연결해야 하며 큐를 사용할 권한이 있어야 합니다.
- 권한 부여된 사용자가 메시지 채널을 작성하고 제어해야 합니다.
- 오브젝트가 라이브러리에 유지되며 이 라이브러리에 대한 액세스가 제한될 수 있습니다.

원격 사이트의 메시지 채널 에이전트는 이 원격 사이트에서 이를 수행하도록 권한을 가진 사용자로부터 생성되어 전달 중인 메시지를 확인합니다. 또한, MCA를 원격으로 시작할 수 있으므로, MCA를 시작하려고 시도하는 원격 프로세스가 이를 수행하도록 권한 부여되었는지 확인해야 할 수 있습니다. 다음 네 가지 방법으로 이를 처리할 수 있습니다.

1. RCVR, RQSTR 또는 CLUSRCVR 채널 정의의 PutAuthority 속성을 적절히 사용하여, 큐에 수신 메시지를 넣을 때 권한 검사를 위해 사용되는 사용자를 제어하십시오. MQSC 명령 참조에서 DEFINE CHANNEL 명령 설명을 참조하십시오.
2. 원하지 않는 연결 시도를 거부하거나 원격 IP 주소, 원격 사용자 ID, 제공된 TLS 주제 식별 이름(DN) 또는 리모트 큐 관리자 이름을 기반으로 하는 MCAUSER 값을 설정하도록 채널 인증 레코드를 구현하십시오.
3. 해당 메시지 채널에 권한이 있는지 확인하는 사용자 엑시트 보안 검사를 구현하십시오. 해당 채널을 호스트하는 설치 보안이 모든 사용자가 적절하게 권한 부여되어 개별 메시지를 검사하지 않아도 되는지 확인하십시오.
4. 사용자 엑시트 메시지 프로세싱을 구현하여 개별 메시지에 대한 권한을 면밀히 조사했는지 확인하십시오.

### IBM i *IBM MQ for IBM i* 오브젝트 보안

이 절에서는 원격 메시징 보안 측면을 다룹니다.

IBM MQ for IBM i 기능을 사용할 권한을 사용자에게 제공해야 합니다. 이 권한은 오브젝트 및 정의에 관련되어 수행되는 조치에 따라 구성됩니다. 예를 들면, 다음과 같습니다.

- 큐 관리자는 권한이 부여된 사용자가 시작하고 중지할 수 있습니다.
- 애플리케이션은 큐 관리자에 연결해야 하며, 큐를 사용할 권한이 있어야 합니다.
- 메시지 채널은 권한이 있는 사용자가 작성하고 제어해야 합니다.

원격 사이트의 메시지 채널 에이전트는 전달 중인 메시지가 이 원격 사이트에서 메시지를 발행하는 권한이 있는 사용자에게 도출되었는지 확인해야 합니다. 또한, MCA를 원격으로 시작할 수 있으므로, MCA를 시작하려고 시도하는 원격 프로세스가 이를 수행하도록 권한 부여되었는지 확인해야 할 수 있습니다. 다음 네 가지 방법으로 이를 처리할 수 있습니다.

- 메시지가 허용 가능한 컨텍스트 권한을 포함해야 함을 채널 정의에서 지정하십시오. 그렇지 않으면 제거됩니다. .
- 원하지 않는 연결 시도를 거부하거나 원격 IP 주소, 원격 사용자 ID, 제공된 TLS 식별 이름(DN) 또는 리모트 큐 관리자 이름 중 하나를 기반으로 하는 MCAUSER 값을 설정하도록 채널 인증 레코드를 구현하십시오.
- 해당 메시지 채널에 권한이 있는지 확인하는 사용자 엑시트 보안 검사 구현하십시오. 해당 채널을 호스트하는 설치 보안이 모든 사용자가 적절하게 권한 부여되어 개별 메시지를 검사하지 않아도 되는지 확인하십시오.
- 사용자 엑시트 메시지 처리를 구현하여 개별 메시지에 대한 권한을 면밀히 조사했는지 확인하십시오.

다음은 IBM MQ for IBM i에서 보안을 운영하는 방법에 대한 몇 가지 사실입니다.

- 사용자는 IBM i에서 식별되고 인증됩니다.
- 애플리케이션이 호출하는 큐 관리자 서비스는 큐 관리자 사용자 프로파일의 권한으로 실행되지만 사용자의 프로세스에 있습니다.
- 사용자 명령에서 호출하는 큐 관리자 서비스는 큐 관리자 사용자 프로파일의 권한으로 실행됩니다.

### Linux AIX *AIX and Linux*에서 오브젝트의 보안

이 ID가 IBM MQ 관리 명령을 사용하는 경우, 관리 사용자는 시스템에서 mqm 그룹의 파트여야 합니다(루트 포함).

항상 "mqm" 사용자 ID로 amqcrsta를 실행해야 합니다.

## AIX and Linux에서 사용자 ID

큐 관리자는 모든 대문자 또는 대소문자 혼합 사용자 ID를 소문자로 변환합니다. 그런 다음 큐 관리자는 사용자 ID를 메시지의 컨텍스트 부분에 삽입하거나 해당 권한을 검사합니다. 따라서 권한은 소문자 ID만을 기반으로 합니다.

### Windows *Windows* 시스템에서 오브젝트 보안

이 ID가 IBM MQ 관리 명령을 사용할 경우 관리 사용자는 Windows 시스템에서 mqm 그룹 및 관리자 그룹 모두의 일부여야 합니다.

## Windows 시스템의 사용자 ID

Windows 시스템에서 메시지 엑시트가 설치되지 않은 경우 큐 관리자는 모든 대문자 또는 대소문자 혼합 사용자 ID를 소문자로 변환합니다. 그런 다음 큐 관리자는 사용자 ID를 메시지의 컨텍스트 부분에 삽입하거나 해당 권한을 검사합니다. 따라서 권한은 소문자 ID만을 기반으로 합니다.

## 시스템 전체에 걸친 사용자 *ID*

AIX, Linux, and Windows 시스템 이외의 플랫폼에서는 메시지의 사용자 ID로 대문자를 사용합니다. AIX, Linux, and Windows 시스템에서 메시지에 소문자 사용자 ID를 사용할 수 있으려면 메시지 채널 에이전트(MCA)가 영문자를 적절하게 변환해야 합니다.

AIX, Linux, and Windows 시스템에서 메시지에 소문자 사용자 ID를 사용할 수 있으려면 해당 플랫폼에서 메시지 채널 에이전트(MCA)를 통해 다음을 변환해야 합니다.

**송신 측에서**
메시지 엑시트가 설치되지 않은 경우, 모든 사용자 ID의 영문자는 대문자로 변환됩니다.

**수신 측에서**
메시지 엑시트가 설치되지 않은 경우, 모든 사용자 ID의 영문자는 소문자로 변환됩니다.

다른 어떤 이유로 AIX, Linux, and Windows에서 메시지 엑시트를 제공하는 경우, 자동 변환이 수행되지 않습니다.

## 사용자 정의 권한 서비스 사용

IBM MQ에서는 설치 가능한 권한 서비스를 제공합니다. 대체 서비스 설치를 선택할 수 있습니다.

IBM MQ에서 제공되는 권한 서비스 컴포넌트는 오브젝트 권한 관리자(OAM)라고 합니다. OAM이 필요한 권한 기능을 제공하지 않으면 자체적으로 권한 서비스 컴포넌트를 작성할 수 있습니다. 권한 서비스 컴포넌트로 구현해야 하는 설치 가능한 서비스 함수에 대해서는 설치 가능 서비스 인터페이스 참조 정보에서 설명됩니다.

## 클라이언트의 액세스 제어

액세스 제어는 사용자 ID를 기반으로 합니다. 관리할 사용자 ID가 많고 사용자 ID는 다른 형식일 수 있습니다. 서버 연결 채널 특성 MCAUSER를 클라이언트에서 사용되는 특수 사용자 ID 값으로 설정할 수 있습니다.

IBM MQ의 액세스 제어는 사용자 ID를 기반으로 합니다. MQI 호출을 수행하는 프로세스의 사용자 ID가 일반적으로 사용됩니다. MQ MQI 클라이언트의 경우, 서버 연결 MCA는 MQ MQI 클라이언트를 대신하여 MQI 호출을 작성합니다. 서버 연결 MCA가 MQI 호출 작성에 사용할 대체 사용자 ID를 선택할 수 있습니다. 대체 사용자 ID는 클라이언트 워크스테이션이나 클라이언트의 액세스를 구성 및 제어하는 데 선택하는 모든 것과 연관될 수 있습니다. 사용자 ID에는 MQI 호출을 발행하기 위해 필요한 권한이 서버에서 할당되어야 합니다. 클라이언트가 서버 연결 MCA의 권한으로 MQI 호출을 작성할 수 있게 하려면 대체 사용자 ID를 선택하는 것이 좋습니다.

표 *9.* 서버 연결 채널에서 사용되는 사용자 *ID*

| 사용자 ID | 사용되는 경우 |
| --- | --- |
| 보안 엑시트로 설정되는 사용자 ID | **CHLAUTH TYPE(BLOCKUSER)** 규칙으로 차단되는 경우를 제외하고 사용됨. 자세한 정보는 97 페이지의 『보안 엑시트에서 사용자 ID 설정』 절을 참조하십시오. |
| CHLAUTH 규칙으로 설정되는 사용자 ID | 보안 엑시트로 겹쳐써지는 경우를 제외하고는 사용됨. 자세한 정보는 채널 인증 레코드를 참조하십시오. |
| SVRCONN 인증 정의의 **MCAUSER** 속성에 정의되는 사용자 ID | 보안 엑시트 또는 CHLAUTH 규칙으로 겹쳐써지는 경우를 제외하고는 사용됨. |
| 클라이언트 시스템에서 플로우되는 사용자 ID | 다른 방법으로 설정된 사용자 ID가 없는 경우 사용됨. |
| 서버 연결 채널을 시작한 사용자 ID | 다른 방법으로 설정된 사용자 ID가 없고 클라이언트 사용자 ID가 플로우되지 않는 경우 사용됨. 자세한 정보는 97 페이지의 『채널 프로그램을 실행하는 사용자 ID』 절을 참조하십시오. |

서버 연결 MCA가 원격 사용자를 대신하여 MQI 호출을 작성하기 때문에 원격 클라이언트 대신 MQI 호출을 발행하는 서버 연결 MCA의 보안 포함 및 잠재적으로 많은 수의 사용자 액세스를 관리하는 방법을 고려하는 것이 중요합니다.

- 한 가지 방법은 서버 연결 MCA가 자체 권한으로 MQI 호출을 발행하는 것입니다. 하지만 일반적으로 서버 연결 MCA가 자체의 강력한 액세스 성능으로 클라이언트 사용자 대신 MQI 호출을 발행하는 것은 바람직하지 않음에 유념하십시오.
- 다른 방법은 클라이언트에서 플로우되는 사용자 ID를 사용하는 방법입니다. 서버 연결 MCA가 클라이언트 사용자 ID의 액세스 성능을 사용하여 MQI 호출을 발행할 수 있습니다. 이 방식은 고려해야 하는 몇 가지 질문을 작성합니다.
  1. 각 플랫폼의 사용자 ID 형식이 서로 다릅니다. 클라이언트의 사용자 ID 형식이 서버에서 허용 가능한 형식과 다른 경우 이는 문제가 될 수 있습니다.
  2. 서로 다르며 사용자 ID가 변경되는 잠재적으로 많은 클라이언트가 있을 수 있습니다. 서버에서 ID를 정의하거나 관리해야 합니다.
  3. 사용자 ID는 신뢰됩니까? 모든 사용자 ID(로그온한 사용자의 ID가 아니어도 됨)를 클라이언트에서 사용될 수 있습니다. 예를 들어, 클라이언트는 보안을 이유로 서버에서만 의도적으로 정의된 전체 `mqm` 권한이 있는 ID를 플로우할 수 있습니다.
- 바람직한 방법은 서버에서 클라이언트 식별 토큰을 정의하여 클라이언트 연결된 애플리케이션의 성능을 제한하는 것입니다. 이는 일반적으로 서버 연결 채널 특성 MCAUSER를 클라이언트에서 사용되는 특수 사용자 ID 값으로 설정하고 서버의 권한 레벨과 다른 레벨로 클라이언트에서 사용되는 몇 개의 ID를 정의하여 수행합니다.

### 보안 엑시트에서 사용자 ID 설정

IBM MQ MQI clients에 대해 MQI 호출을 발행하는 프로세스는 서버 연결 MCA입니다. 서버 연결 MCA에서 사용하는 사용자 ID는 MQCD의 `MCAUserIdentifier` 또는 `LongMCAUserIdentifier` 필드에 포함된 것입니다. 이러한 필드의 컨텐츠는 다음으로 설정됩니다.

- 보안 엑시트에 의해 설정된 값
- 클라이언트의 사용자 ID
- MCAUSER(서버 연결 채널 정의에서)

보안 엑시트는 호출될 때 인식되는 값을 대체할 수 있습니다.

- 서버 연결 채널 MCAUSER 속성이 공백으로 설정되지 않으면 MCAUSER 값이 사용됩니다.
- 서버 연결 채널 MCAUSER 속성이 공백이면 클라이언트에서 수신된 사용자 ID가 사용됩니다.
- 서버 연결 채널 MCAUSER 속성이 공백이면 클라이언트에서 사용자 ID가 수신되지 않고 서버 연결 채널을 시작한 사용자 ID가 사용됩니다.

클라이언트측 보안 엑시트가 사용 중일 때 IBM MQ 클라이언트는 확인된 사용자 ID를 서버에 보내지 않습니다.

### 채널 프로그램을 실행하는 사용자 ID

서버 연결 채널을 시작한 사용자 ID에서 사용자 ID 필드가 도출되면 다음 값이 사용됩니다.

- z/OS z/OS의 경우 z/OS 시작 프로시저 테이블에서 채널 시작기 시작 태스크에 지정된 사용자 ID.
- TCP/IP (비 z/OS )의 경우, `inetd.conf` 항목의 사용자 ID 또는 리스너를 시작한 사용자 ID입니다.
- SNA (비 z/OS )의 경우, SNA 서버 항목의 사용자 ID 또는 수신 접속 요청 (없는 경우) 또는 리스너를 시작한 사용자 ID.
- NetBIOS 또는 SPX의 경우 리스너를 시작한 사용자 ID.

MCAUSER 속성을 공백으로 설정한 서버 연결 채널 정의가 있으면 클라이언트에서 이 채널 정의를 사용하여 큐 관리자에 연결할 수 있습니다. 이때 클라이언트에서 사용하는 액세스 권한은 클라이언트에서 제공한 사용자 ID로 결정됩니다. 큐 관리자가 실행되고 있는 시스템에서 권한 없는 네트워크 연결을 허용할 경우 보안이 노출될 수 있습니다. IBM MQ 기본 서버 연결 채널(SYSTEM.DEF.SVRCONN)에는 MCAUSER 속성이 공백으로 설정되어 있습니다. 무단 액세스를 방지하려면 기본 정의의 MCAUSER 속성을 IBM MQ MQ 오브젝트에 대한 액세스 권한이 없는 사용자 ID로 업데이트하십시오.

### 사용자 ID의 대소문자

`runmqsc`를 사용하여 채널을 정의할 때 사용자 ID가 작은따옴표 내에 포함되지 않으면 MCAUSER 속성이 대문자로 변경됩니다.

ALW AIX, Linux, and Windows의 서버의 경우, 클라이언트에서 수신한 `MCAUserIdentifier` 필드의 컨텐츠는 소문자로 변경됩니다.

IBM i IBM i의 서버의 경우, 클라이언트에서 수신한 `LongMCAUserIdentifier` 필드 컨텐츠는 대문자로 변경됩니다.

Linux AIX AIX and Linux 시스템의 서버의 경우, 클라이언트에서 수신한 `LongMCAUserIdentifier` 필드의 컨텐츠는 소문자로 변경됩니다.

기본적으로 IBM MQ JMS 바인딩 애플리케이션이 사용될 때 전달되는 사용자 ID는 애플리케이션이 실행 중인 JVM의 사용자 ID입니다.

`createQueueConnection` 메소드를 통해 사용자 ID를 전달할 수도 있습니다.

## 기밀성 계획

데이터 기밀 보관 방법을 계획합니다.

애플리케이션 레벨 또는 링크 레벨에서 기밀성을 구현할 수 있습니다. TLS를 사용하도록 선택할 수 있습니다. 이 경우 디지털 서명을 사용하도록 계획해야 합니다. 또한, 표준 기능이 요구사항에 맞지 않는 경우에는 채널 엑시트 프로그램을 사용할 수도 있습니다.

**관련 개념**

98 페이지의 『링크 레벨 보안과 애플리케이션 레벨 보안 비교』
이 토픽에는 다양한 측면의 링크 레벨 보안 및 애플리케이션 레벨 보안에 대한 정보가 들어 있고 두 보안 레벨을 비교합니다.

102 페이지의 『채널 엑시트 프로그램』
채널 엑시트 프로그램은 MCA의 처리 순서에서 정의된 위치에서 호출되는 프로그램입니다. 사용자와 벤더는 자체적으로 채널 엑시트 프로그램을 작성할 수 있습니다. 일부는 IBM에서 제공됩니다.

108 페이지의 『SSL/TLS을 사용하는 채널 보호』
IBM MQ에서의 TLS 지원은 큐 관리자 인증 정보 오브젝트 및 다양한 MQSC 명령을 사용합니다. 또한, 디지털 인증서 사용도 고려해야 합니다.

## 링크 레벨 보안과 애플리케이션 레벨 보안 비교

이 토픽에는 다양한 측면의 링크 레벨 보안 및 애플리케이션 레벨 보안에 대한 정보가 들어 있고 두 보안 레벨을 비교합니다.

링크 레벨 및 애플리케이션 레벨 보안은 98 페이지의 그림 10에서 설명됩니다.

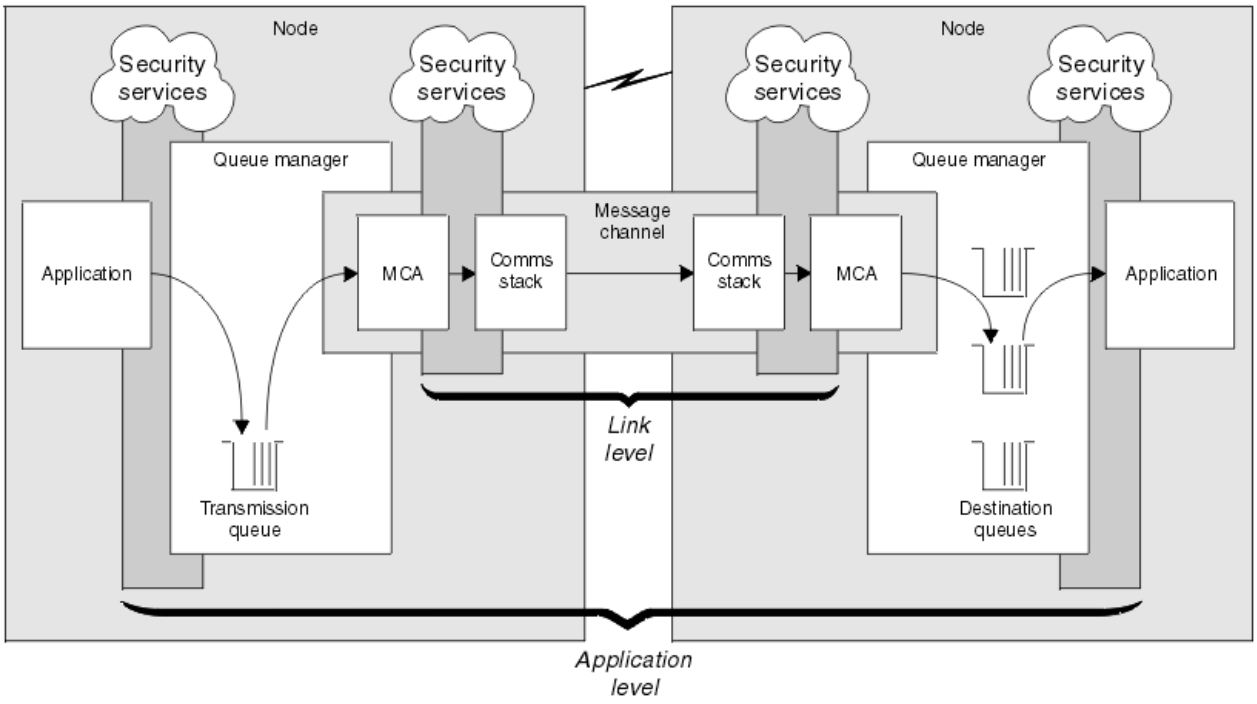



그림 *10.* 링크 레벨 보안 및 애플리케이션 레벨 보안

## 큐의 보호 메시지

링크 레벨 보안은 메시지가 하나의 큐 관리자에서 다른 큐 관리자로 전송되는 동안 이를 보호할 수 있습니다. 비보안 네트워크에서 메시지가 전송될 때 특히 중요합니다. 그러나 이는 메시지가 소스 큐 관리자, 목적지 큐 관리자 또는 중간 큐 관리자에서 큐에 저장되어 있는 동안에는 메시지를 보호할 수 없습니다.

z/OS z/OS 데이터 세트 암호화는 큐에 저장된 메시지에 어느 정도의 보호를 제공할 수 있지만, 이는 로컬 큐 관리자의 저장 데이터로 한정됩니다. 데이터 세트 암호화를 사용하는 IBM MQ for z/OS의 저장 데이터에 대한 기밀성 섹션을 참조하십시오. 참조하십시오.

애플리케이션 레벨 보안은 메시지가 큐에 저장되어 있는 동안 이를 보호할 수 있으며 분산 큐잉이 사용되지 않는 경우에도 적용됩니다. 이는 링크 레벨 보안과 애플리케이션 레벨 보안의 주요 차이점이며 98 페이지의 그림 10에 설명되어 있습니다.

## 큐 관리자가 제어되고 신뢰되는 환경에서 실행 중이 아님

큐 관리자가 제어되고 신뢰되는 환경에서 실행 중인 경우 IBM MQ에서 제공되는 액세스 제어 메커니즘은 해당 큐에 저장된 메시지 보호에 충분한 것으로 간주됩니다. 이는 특히 로컬 큐잉만이 관련되고 메시지가 큐 관리자를 떠나지 않는 경우에 해당합니다. 이 경우 애플리케이션 레벨 보안은 불필요한 것으로 간주될 수도 있습니다.

애플리케이션 레벨 보안은 또한 메시지가 제어되고 신뢰되는 환경에서 실행 중인 또 다른 큐 관리자로 전송되거나 이러한 큐 관리자로부터 받는 경우에 불필요한 것으로 간주될 수도 있습니다. 메시지가 제어되고 신뢰되는 환경에서 실행 중이 아닌 큐 관리자로 전송되거나 이러한 큐 관리자로부터 받은 경우에 애플리케이션 레벨 보안의 필요성은 더욱 커집니다.

## 비용의 차이점

애플리케이션 레벨 보안은 관리 및 성능의 관점에서 링크 레벨 보안보다 비용이 더 많이 들어갈 수도 있습니다.

구성 및 유지보수하는 데 잠재적인 제한조건이 더 많기 때문에 관리 비용이 더 클 수 있습니다. 예를 들면, 특정 사용자가 특정 유형의 메시지만을 송신하고 메시지를 특정 목적지에만 송신하는지를 확인해야 할 수도 있습니다. 반대로, 특정 사용자가 특정 유형의 메시지만을 받고 특정 소스로부터의 메시지만 받도록 해야 할 수도 있습니다. 하나의 메시지 채널에서 링크 레벨 보안 서비스를 관리하는 대신에, 그 채널에서 메시지를 교환하는 모든 쌍의 사용자에 대해 규칙을 구성하고 유지보수해야 할 수 있습니다.

애플리케이션이 메시지를 넣거나 가져올 때마다 보안 서비스가 호출되는 경우 성능에 영향을 미칠 수도 있습니다.

조직은 링크 레벨 보안이 구현하기가 더 쉬우므로 이를 먼저 고려하는 경향이 있습니다. 조직은 링크 레벨 보안이 모든 요구사항을 만족하지 못하는 경우 애플리케이션 레벨 보안을 고려합니다.

## 컴포넌트 가용성

일반적으로 분산 환경에서는 보안 서비스에는 둘 이상의 시스템에 있는 컴포넌트가 필요합니다. 예를 들어, 메시지가 한 시스템에서 암호화되고 또 다른 시스템에서 복호화될 수도 있습니다. 이는 링크 레벨 보안 및 애플리케이션 레벨 보안 둘 모두에 적용됩니다.

사용 중인 다른 플랫폼들에 대해 각각이 다른 레벨의 보안 기능을 가지는 이기종 환경에서, 보안 서비스의 필수 구성요소들이 그들이 필요로 하는 모든 플랫폼에서, 사용하기 쉬운 형식으로 사용할 수 없을 수도 있습니다. 이는 링크 레벨 보안에서보다 애플리케이션 레벨 보안에서 더 큰 문제입니다. 특히 다양한 소스로부터 컴포넌트를 구매하여 사용자 고유의 애플리케이션 레벨 보안을 제공하려는 경우에 더욱 그렇습니다.

## 데드-레터 큐의 메시지

메시지가 애플리케이션 레벨 보안에 의해 보호되는 경우 어떤 이유에서든지 메시지가 해당 목적지에 도달하지 못하고 데드-레터 큐에 놓이는 경우 문제가 발생할 수 있습니다. 메시지 디스크립터 및 데드 레터 헤더의 정보에서 메시지를 처리하는 방법을 알아낼 수 없는 경우에는 애플리케이션 데이터의 컨텐츠를 검사해야 할 수도 있습니다. 애플리케이션 데이터가 암호화되고 의도한 수신인만이 이를 암호 해독할 수 있는 경우 이를 수행할 수 없습니다.

## 애플리케이션 레벨 보안이 할 수 없는 일

애플리케이션 레벨 보안은 완벽한 솔루션은 아닙니다. 애플리케이션 레벨 보안을 구현하더라도 여전히 몇몇 링크 레벨 보안 서비스가 필요할 수도 있습니다. 예를 들면, 다음과 같습니다.

- 채널이 시작될 때, 두 MCA가 여전히 상호 인증해야 할 수 있습니다. 이는 링크 레벨 보안 서비스만이 수행할 수 있습니다.
- 애플리케이션 레벨 보안은 임베디드 메시지 디스크립터를 포함하는 전송 큐 헤더, MQXQH를 보호할 수 없습니다. 메시지 데이터가 아닌 IBM MQ 채널 프로토콜 플로우의 데이터를 보호할 수도 없습니다. 링크 레벨 보안만이 이 보호를 제공할 수 있습니다.
- 애플리케이션 레벨 보안 서비스가 MQI 채널의 서버 측에서 호출된 경우 서비스는 채널을 통해 전송된 MQI 호출의 매개변수를 보호할 수 없습니다. 특히 MQPUT, MQPUT1 또는 MQGET 호출의 애플리케이션 데이터는 보호되지 않습니다. 이 경우, 링크 레벨 보안만 보호를 제공할 수 있습니다.

## 링크 레벨 보안

링크 레벨 보안은 MCA, 통신 서브시스템 또는 함께 작동하는 두 개의 결합에 의해 직접 또는 간접적으로 호출되는 보안 서비스를 의미합니다.

링크 레벨 보안은 98 페이지의 그림 10에서 설명됩니다.

다음은 링크 레벨 보안 서비스의 몇 개 예입니다.

- 메시지 채널의 양 측에 있는 MCA는 해당 파트너를 인증할 수 있습니다. 이는 채널이 시작되고 통신 연결이 설정되면 메시지가 플로우를 시작하기 전에 수행됩니다. 어느 쪽에서든지 인증이 실패하면, 채널이 닫히고 메시지가 전송되지 않습니다. 이는 식별과 인증 서비스의 예입니다.
- 메시지는 채널의 송신 측에서 암호화되고 수신 측에서 복호화됩니다. 이는 기밀성 서비스의 예입니다.
- 네트워크에서 메시지가 전송되는 동안 컨텐츠가 의도적으로 수정되었는지를 판별하기 위해 채널의 수신 측에서 메시지가 검사될 수 있습니다. 이는 데이터 무결성 서비스의 예입니다.

## IBM MQ에서 제공되는 링크 레벨 보안

IBM MQ에서의 기밀성 및 데이터 무결성의 프로비저닝을 위한 기본 방법은 TLS를 사용하는 것입니다. IBM MQ에서 TLS를 사용하는 것에 관한 자세한 정보는 22 페이지의 『IBM MQ의 TLS 보안 프로토콜』의 내용을 참조하십시오. 인증을 위해 IBM MQ는 채널 인증 레코드 사용 기능을 제공합니다. 채널 인증 레코드는 개별 채널이나 채널 그룹의 레벨에서 연결 시스템에 부여되는 액세스에 대해 세밀한 제어를 제공합니다. 자세한 정보는 47 페이지의 『채널 인증 레코드』의 내용을 참조하십시오.

### 사용자 고유의 링크 레벨 보안 제공

사용자 고유의 링크 레벨 보안 서비스를 제공할 수 있습니다. 사용자 고유의 채널 엑시트 프로그램을 작성하는 것이 사용자 고유의 링크 레벨 보안 서비스를 제공하는 기본적인 방법입니다.

채널 엑시트 프로그램은 102 페이지의 『채널 엑시트 프로그램』에 소개되어 있습니다. 동일한 주제에서는 IBM MQ for Windows와 함께 제공되는 채널 엑시트 프로그램(SSPI 채널 엑시트 프로그램)에 대해서도 설명합니다. 이 채널 엑시트 프로그램은 소스 코드를 수정하여 요구사항을 충족시킬 수 있도록 소스 형식으로 제공됩니다. 이 채널 엑시트 프로그램 또는 다른 벤더에서 제공되는 채널 엑시트 프로그램이 요구사항에 맞지 않는 경우, 사용자 고유로 설계하고 작성할 수 있습니다. 이 주제에서는 채널 엑시트 프로그램이 보안 서비스를 제공할 수 있는 방법을 제안합니다. 채널 엑시트 프로그램 작성 방법에 대한 정보는 채널 엑시트 프로그램 작성을 참조하십시오.

### 보안 엑시트를 사용하는 링크 레벨 보안

보안 엑시트는 보통 쌍으로 작동합니다. 채널의 각 끝에서 하나씩 작동합니다. 이들은 채널이 시동될 때 초기 데이터 조정이 끝난 직후 호출됩니다.

보안 엑시트는 식별 및 인증, 액세스 제어, 기밀성을 제공하기 위해 사용할 수 있습니다.

### 메시지 엑시트를 사용하는 링크 레벨 보안

메시지 엑시트는 MQI 채널이 아니라 메시지 채널에서만 사용할 수 있습니다. 메시지 엑시트는 임베드된 메시지 디스크립터를 포함하는 전송 큐 헤더인 MQXQH와 메시지 안의 애플리케이션 데이터에 모두 액세스할 수 있습니다. 메시지 엑시트는 메시지의 컨텐츠를 수정하고 그 길이를 변경할 수 있습니다.

메시지 엑시트는 메시지의 일부가 아니라 전체에 액세스해야 하는 경우에도 사용될 수 있습니다.

메시지 엑시트는 식별 및 인증, 액세스 제어, 기밀성, 데이터 무결성, 비거절, 보안 이외의 다른 이유로 제공하는 데 사용할 수 있습니다.

### 송신 및 수신 엑시트를 사용하는 링크 레벨 보안

송신 및 수신 엑시트는 메시지 및 MQI 채널 모두에서 사용 가능합니다. 이는 채널에서 플로우되는 모든 유형의 데이터에 대해 호출되고, 양방향 모두의 플로우에 대해 호출됩니다.

송신 및 수신 엑시트가 각 전송 세그먼트에 액세스할 수 있습니다. 해당 컨텐츠를 수정하고 그 길이를 변경할 수 있습니다.

메시지 채널에서 MCA가 메시지를 분할하여 둘 이상의 전송 세그먼트로 송신해야 하면, 메시지의 일부가 있는 각 전송 세그먼트에 대해 송신 엑시트가 호출되고, 수신하는 쪽에서는 각 전송 세그먼트에 대해 수신 엑시트가 호출됩니다. MQI 채널에서 MQI 호출의 입력이나 출력 매개변수가 너무 커서 하나의 전송 세그먼트로 송신될 수 없는 경우에도 같은 일이 발생합니다.

MQI 채널에서는 전송 세그먼트의 바이트 10이 MQI 호출을 식별하고, 전송 세그먼트에 호출의 입력이나 출력 매개변수가 있는지 여부를 표시합니다. 송신 및 수신 엑시트가 MQI 호출에 보호되어야 할 수 있는 애플리케이션 데이터가 있는지 여부를 판별하기 위해 이 바이트를 조사할 수 있습니다.

송신 엑시트가 처음으로 호출되면, 필요한 모든 자원을 확보하고 초기화하기 위해 MCA에게 전송 세그먼트를 보유하는 버퍼 내 지정된 양의 공간을 예약하게 할 수 있습니다. 전송 세그먼트를 처리하기 위해 나중에 호출되는 경우, 이 공간을 사용하여 암호화된 키나 디지털 서명 등을 추가할 수 있습니다. 채널의 다른 쪽에 있는 해당하는 수신 엑시트는 송신 엑시트로 추가된 데이터를 제거하고, 전송 세그먼트를 처리하는 데 사용할 수 있습니다.

송신 및 수신 엑시트는 처리 중인 데이터 구조를 알 필요가 없기 때문에 각 전송 세그먼트를 2진 오브젝트로 처리할 수 있다는 점에서 최적입니다.

송신 및 수신 엑시트는 기밀성 및 데이터 무결성을 제공하는 데 사용할 수 있으며 보안 이외의 용도로 사용하기 위해서도 제공 가능합니다.

**관련 태스크**

송신 또는 수신 엑시트 프로그램에서 API 호출 식별

## 애플리케이션 레벨의 보안

애플리케이션 레벨 보안은 애플리케이션과 애플리케이션이 연결된 큐 관리자 사이의 인터페이스에서 호출되는 해당 보안 서비스를 나타냅니다.

해당 서비스는 애플리케이션이 큐 관리자에 MQI 호출을 발행할 때 호출됩니다. 서비스는 애플리케이션, 큐 관리자, IBM MQ를 지원하는 다른 제품 또는 같이 작동하는 이들의 결합으로 간접 또는 직접 호출될 수 있습니다. 애플리케이션 레벨 보안은 98 페이지의 그림 10에서 설명됩니다.

애플리케이션 레벨 보안은 엔드-투-엔드 보안 또는 메시지 레벨의 보안이라고도 합니다.

다음은 애플리케이션 레벨 보안 서비스의 몇 가지 예입니다.

- 애플리케이션이 메시지를 큐에 넣으면 메시지 디스크립터는 애플리케이션에 연관된 사용자 ID를 포함합니다. 그렇지만 사용자 ID 인증에 사용할 수 있는 암호화된 비밀번호와 같은 데이터는 없습니다. 보안 서비스는 이 데이터를 추가할 수 있습니다. 메시지가 결국 수신 애플리케이션에서 검색되는 경우, 서비스의 다른 컴포넌트는 메시지와 같이 전달되는 데이터를 사용하여 사용자 ID를 인증할 수 있습니다. 이는 식별과 인증 서비스의 예입니다.
- 메시지는 애플리케이션이 큐에 이를 넣을 때 암호화되고 수신 애플리케이션이 이를 검색할 때 복호화됩니다. 이는 기밀성 서비스의 예입니다.
- 메시지는 수신 애플리케이션이 이를 검색할 때 검사 가능합니다. 이 검사는 애플리케이션 전송에 의해 메시지를 큐에 처음 넣은 후에 컨텐츠가 의도적으로 수정되었는지 여부를 판별합니다.이는 데이터 무결성 서비스의 예입니다.

*Advanced Message Security* 계획

Advanced Message Security(AMS)는 엔드 애플리케이션에 영향을 미치지 않으면서 IBM MQ 네트워크를 통해 플로우하는 민감한 데이터에 대한 높은 레벨의 보호를 제공하는 IBM MQ의 컴포넌트입니다.

특히 기밀 또는 지불 관련 정보(예: 환자 보고서 또는 신용카드 정보)와 같은 아주 민감하거나 높은 가치의 정보를 이동하는 경우, 정보 보안에 특히 주의해야 합니다. 회사 내에서 이동하는 정보의 해당 무결성을 유지하고 권한없는 액세스로부터 보호하는 것은 매우 위험하고 책임이 따르는 일입니다. 이 때 보안 규제를 준수해야 하며 그렇지 못할 경우에는 커다란 불이익이 있을 수 있습니다.

IBM MQ에 대해 자체적으로 보안 확장을 개발할 수도 있습니다. 그렇지만 이런 솔루션의 경우 전문가의 기술이 필요하며 유지보수가 복잡하고 비용이 많이 들 수 있습니다. Advanced Message Security는 기업 내에서 정보를 이동할 때 거의 모든 상용 IT 시스템 유형 사이에서 발생하는 이런 문제 해결에 사용할 수 있습니다.

Advanced Message Security는 다음과 같은 방식으로 IBM MQ의 보안 기능을 확장합니다.

- 메시지 암호화 또는 디지털 서명을 사용하여 애플리케이션 레벨, 지점간 메시징 인프라에 대한 엔드-투-엔드 보호를 제공합니다.
- 복잡한 보호 코드를 작성하거나 기존 애플리케이션을 수정하거나 다시 컴파일하지 않고도 포괄적인 보호를 제공합니다.
- PKI(Public Key Infrastructure) 기술을 사용하여 메시지에 대해 인증, 권한 부여, 기밀성, 데이터 무결성 서비스를 제공합니다.

- 메인프레임 및 분산 서버에 대한 보안 정책 관리를 제공합니다.
- IBM MQ 서버 및 클라이언트 모두를 지원합니다.
- 엔드-투-엔드 안전 메시징 솔루션을 제공하기 위해 Managed File Transfer와 통합됩니다.

자세한 정보는 557 페이지의 『Advanced Message Security』의 내용을 참조하십시오.

사용자 고유의 애플리케이션 레벨 보안 제공
사용자 고유의 애플리케이션 레벨 보안 서비스를 제공할 수 있습니다. 애플리케이션 레벨 보안을 구현할 수 있도록 IBM MQ는 두 개의 엑시트인 API 엑시트와 API 교차 엑시트를 제공합니다.

API 엑시트와 API 교차 엑시트는 ID 및 인증, 액세스 제어, 기밀성, 데이터 무결성 및 부인 방지 서비스 및 보안과 무관한 기타 기능을 제공할 수 있습니다.

API 엑시트나 API 교차 엑시트가 시스템 환경에서 지원되지 않으면, 사용자 고유의 애플리케이션 레벨 보안을 제공하는 다른 방법을 고려하는 것을 원할 수 있습니다. 한 가지 방법은 MQI를 캡슐화하는 상위 레벨 API를 개발하는 것입니다. 그러면 프로그래머가 이 API를 MQI 대신 사용하여 IBM MQ 애플리케이션을 작성합니다.

상위 레벨 API를 사용하는 가장 보편적인 이유는 다음과 같습니다.

- MQI의 향상된 기능들을 프로그래머에게 숨기기 위해.
- MQI를 사용할 때 표준을 강화하기 위해.
- MQI에 함수를 추가하기 위해. 이 추가된 함수들이 보안 서비스일 수 있습니다.

일부 벤더 제품은 이 기술을 사용하여 IBM MQ에 대한 애플리케이션 레벨 보안을 제공합니다.

보안 서비스를 이 방법으로 제공하려고 계획하고 있으면, 데이터 변환에 대한 다음 사항들을 알아두십시오.

- 디지털 서명 등의 보안 토큰이 메시지의 애플리케이션 데이터에 추가되었으면, 데이터 변환을 수행하고 있는 어떠한 코드이든지 이 토큰이 있다는 것을 알아야 합니다.
- 보안 토큰이 애플리케이션 데이터의 2진 이미지로부터 도출되었을 수 있습니다. 따라서, 데이터를 변환하기 전에 모든 토큰 검사가 완료되어야 합니다.
- 메시지의 애플리케이션 데이터가 암호화되었으면, 데이터 변환 전에 복호화되어야 합니다.

## 채널 엑시트 프로그램

채널 엑시트 프로그램은 MCA의 처리 순서에서 정의된 위치에서 호출되는 프로그램입니다. 사용자와 벤더는 자체적으로 채널 엑시트 프로그램을 작성할 수 있습니다. 일부는 IBM에서 제공됩니다.

여러 유형의 채널 엑시트 프로그램이 있으나, 네 개만이 링크 레벨 보안을 제공하는 역할을 합니다.

- 보안 엑시트
- 메시지 엑시트
- 송신 엑시트
- 수신 엑시트

이러한 네 가지 유형의 채널 엑시트 프로그램은 103 페이지의 그림 11에 설명되어 있으며 다음 주제에서도 설명됩니다.

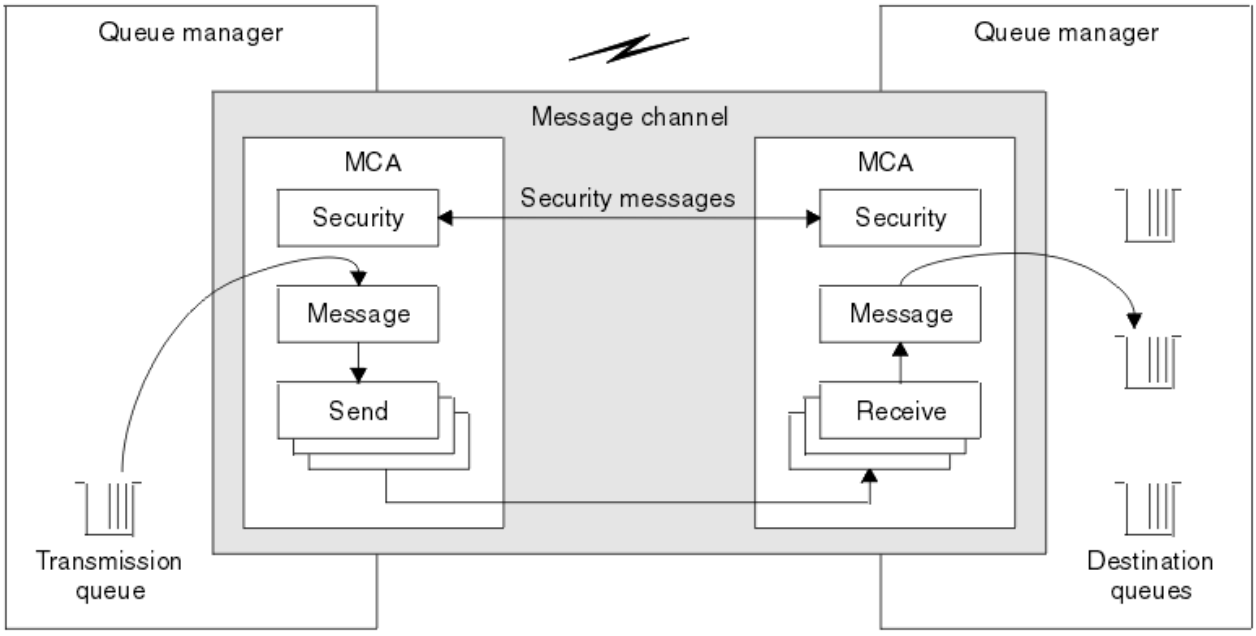



그림 *11.* 메시지 채널의 보안, 메시지, 송신, 수신

**관련 개념**

메시지 채널에 대한 채널 엑시트 프로그램

## 보안 엑시트 개요

보안 엑시트는 일반적으로 쌍으로 작동합니다. 이는 메시지가 플로우되기 전에 호출되고 MCA가 해당 파트너를 인증할 수 있도록 하기 위해 사용됩니다.

일반적으로 보안 엑시트가 채널의 양쪽 끝에서 하나씩 쌍으로 작동합니다. 이를 초기 데이터 협상이 채널 시작 시에 완료된 후 메시지가 플로우를 시작하기 전에 즉시 호출됩니다. 보안 엑시트는 기본적으로 채널 양쪽 끝의 MCA가 해당 파트너를 인증하도록 하기 위해 사용됩니다. 보안과는 전혀 상관없는 기능이더라도 보안 엑시트가 다른 기능을 수행하는 것을 막는 것은 아무것도 없습니다.

보안 엑시트는 보안 메시지를 송신하여 서로 통신할 수 있습니다. 보안 메시지의 형식은 정의되어 있지 않고 사용자가 결정합니다. 보안 메시지 교환의 가능한 한 가지 결과는 보안 엑시트 중 하나가 더 이상 수행되지 않도록 결정될 수도 있다는 점입니다. 이런 경우 채널은 닫히고 메시지가 플로우되지 않습니다. 채널의 한쪽 끝에만 보안 엑시트가 있는 경우에도 여전히 엑시트가 호출되고 채널을 계속할지 또는 닫을지 여부를 선택할 수 있습니다.

보안 엑시트는 메시지와 MQI 채널 둘 다에서 호출될 수 있습니다. 보안 엑시트의 이름은 채널 양쪽의 채널 정의에서 매개변수로 지정됩니다.

보안 엑시트에 대한 자세한 정보는 100 페이지의 『보안 엑시트를 사용하는 링크 레벨 보안』의 내용을 참조하십시오.

## 메시지 엑시트

메시지 엑시트는 메시지 채널에서만 작동되며 보통 쌍으로 작동합니다. 메시지 엑시트는 전체 메시지에 대해 작동될 수 있으며 다양하게 변경할 수 있습니다.

채널의 송신 및 수신 측에 있는 메시지 엑시트는 보통 쌍으로 작동합니다. 채널의 송신 측에 있는 메시지 엑시트는 MCA가 전송 큐에서 메시지를 받은 후에 호출됩니다. 채널의 수신 측에 있는 메시지 엑시트는 MCA가 그 목적지 큐에 메시지를 넣기 전에 호출됩니다.

메시지 엑시트는 임베드된 메시지 디스크립터를 포함하는 전송 큐 헤더인 MQXQH와 메시지 안의 애플리케이션 데이터에 모두 액세스할 수 있습니다. 메시지 엑시트는 메시지의 컨텐츠를 수정하고 그 길이를 변경할 수 있습니다. 메시지를 압축, 압축 풀기, 암호화 또는 복호화로 인해 길이가 변경될 수 있습니다. 또는 메시지에 데이터를 추가하거나, 데이터를 제거한 결과일 수도 있습니다.

메시지 엑시트는 반드시 보안을 위해서 뿐만 아니라, 메시지의 일부가 아닌 전체 메시지에 액세스해야 하는 어떠한 목적을 위해서든지 사용될 수 있습니다.

메시지 엑시트는 현재 처리하고 있는 메시지가 해당 목적지로 더 이상 진행하면 안된다는 것을 판별할 수 있습니다. 그런 후, MCA가 데드-레터 큐에 메시지를 넣습니다. 메시지 엑시트는 채널을 닫을 수도 있습니다.

메시지 엑시트는 MQI 채널이 아니라 메시지 채널에서만 호출될 수 있습니다. 이는 MQI 호출의 입력 및 출력 매개변수가 IBM MQ MQI client 애플리케이션과 큐 관리자 사이에서 플로우될 수 있도록 하기 위해 MQI 채널이 사용되기 때문입니다.

메시지 엑시트의 이름은 양측 채널에 있는 채널 정의에서 매개변수로 지정됩니다. 연속적으로 실행되도록 메시지 엑시트 목록을 지정할 수도 있습니다.

메시지 엑시트에 대한 자세한 정보는 100 페이지의 『메시지 엑시트를 사용하는 링크 레벨 보안』의 내용을 참조하십시오.

### 송신 및 수신 엑시트

송신 및 수신 엑시트는 일반적으로 쌍으로 작동합니다. 이는 전송 세그먼트에서 운영되며 처리 중인 데이터 구조가 적절하지 않은 경우에 사용하기에 적합합니다.

채널의 한 쪽에 있는 송신 엑시트와 다른 쪽에 있는 수신 엑시트는 보통 쌍으로 작동합니다. 송신 엑시트는 MCA가 통신 연결에서 데이터를 송신하기 위해 통신 송신을 발행하기 직전에 호출됩니다. 수신 엑시트는 MCA가 통신 수신 후에 제어를 다시 확보하고 통신 연결에서 데이터를 수신한 직후에 호출됩니다. MQI 채널을 통해 공유 대화가 사용 중인 경우, 송신 및 수신 엑시트의 다른 인스턴스가 각 대화에 대해 호출됩니다.

IBM MQ 채널 프로토콜은 제어 정보와 메시지 데이터 모두를 포함하는 메시지 채널의 두 MCA 사이를 플로우합니다. 유사하게 MQI 채널에서 플로우는 제어 정보와 MQI 호출 매개변수 모두를 포함합니다. 송신 및 수신 엑시트는 모든 데이터 유형에 대해 호출됩니다.

메시지 데이터는 메시지 채널에서 한 방향으로만 플로우되지만 MQI 채널에서 MQI 호출의 입력 매개변수는 한 방향으로 플로우되고 출력 매개변수는 다른 방향으로 플로우됩니다. 메시지와 MQI 채널 모두에서 제어 정보는 두 방향으로 모두 플로우합니다. 그 결과 송신 및 수신 엑시트는 채널의 양 측에서 모두 호출 가능합니다.

두 MCA 사이에서 하나의 플로우로 전송되는 데이터 단위를 전송 세그먼트라 합니다. 송신 및 수신 엑시트가 각 전송 세그먼트에 액세스할 수 있습니다. 해당 컨텐츠를 수정하고 그 길이를 변경할 수 있습니다. 그렇지만 송신 엑시트는 전송 세그먼트의 처음 8바이트를 변경하면 안됩니다. 이 8바이트는 IBM MQ 채널 프로토콜 헤더 파트를 양식화합니다. 송신 엑시트가 전송 세그먼트의 길이를 늘일 수 있는 양에도 제한이 있습니다. 특히, 송신 엑시트는 채널이 시동될 때 두 개의 MCA 사이에서 협상된 최대 길이 이상으로 그 길이를 늘릴 수 없습니다.

메시지 채널에서, 메시지가 너무 커서 하나의 전송 세그먼트로 송신할 수 없으면, 송신 MCA가 메시지를 분할하고 둘 이상의 전송 세그먼트로 이를 송신합니다. 결과적으로, 메시지의 일부를 포함하는 각 전송 세그먼트에 대해 송신 엑시트가 호출되고, 수신 측에서는 각 전송 세그먼트에 대해 수신 엑시트가 호출됩니다. 수신 MCA는 수신 엑시트가 전송 세그먼트를 처리한 후에 해당 세그먼트에서 메시지를 다시 구성합니다.

이와 유사하게 MQI 채널에서는 MQI 호출의 입력이나 출력 매개변수가 너무 큰 경우 둘 이상의 전송 세그먼트로 송신됩니다. 예를 들어, 애플리케이션 데이터가 충분히 클 경우, MQPUT, MQPUT1, 또는 MQGET 호출에서 둘 이상의 세그먼트로 분할될 수 있습니다.

이런 점들을 고려해볼 때, 송신 및 수신 엑시트는 처리 중인 데이터 구조를 알 필요가 없기 때문에 각 전송 세그먼트를 2진 오브젝트로 처리하는 것이 적절합니다.

송신 또는 수신 엑시트는 채널을 종료할 수 있습니다.

송신 엑시트와 수신 엑시트 이름은 채널 양 측의 채널 정의에서 매개변수로 지정됩니다. 연속적으로 실행되도록 송신 엑시트를 지정할 수도 있습니다. 이와 유사한 방법으로 수신 엑시트 목록을 지정할 수 있습니다.

송신 및 수신 엑시트에 대한 자세한 정보는 100 페이지의 『송신 및 수신 엑시트를 사용하는 링크 레벨 보안』의 내용을 참조하십시오.

## 데이터 무결성 계획

데이터 무결성 보존 방법을 계획합니다.

애플리케이션 레벨 또는 링크 레벨에서 데이터 무결성을 구현할 수 있습니다.

애플리케이션 레벨에서 표준 기능이 요구사항에 맞지 않는 경우 API 엑시트 프로그램을 사용할 수 있습니다. 권한 없는 수정으로부터 보호하기 위해 Advanced Message Security(AMS)를 사용하여 메시지에 디지털로 서명하도록 선택할 수 있습니다.

링크 레벨에서 TLS를 사용하도록 선택할 수 있으며 이 경우 디지털 인증서 사용을 계획해야 합니다. 또한, 표준 기능이 요구사항에 맞지 않는 경우에는 채널 엑시트 프로그램을 사용할 수도 있습니다.

**관련 개념**

108 페이지의 『SSL/TLS을 사용하는 채널 보호』
IBM MQ에서의 TLS 지원은 큐 관리자 인증 정보 오브젝트 및 다양한 MQSC 명령을 사용합니다. 또한, 디지털 인증서 사용도 고려해야 합니다.

10 페이지의 『데이터 무결성』
데이터 무결성 서비스는 데이터가 권한 없이 수정되었는지를 감지합니다.

101 페이지의 『Advanced Message Security 계획』
Advanced Message Security(AMS)는 엔드 애플리케이션에 영향을 미치지 않으면서 IBM MQ 네트워크를 통해 플로우하는 민감한 데이터에 대한 높은 레벨의 보호를 제공하는 IBM MQ의 컴포넌트입니다.

**관련 참조**

API 엑시트 참조

채널 엑시트 호출 및 데이터 구조

# 감사 계획

감사해야 하는 데이터 및 감사 정보 캡처 및 처리 방법을 결정합니다. 시스템이 올바르게 구성되었는지 검사하는 방법을 고려하십시오.

활동 모니터링에는 몇 개의 측면이 있습니다. 고려해야 하는 측면은 감사자 요구사항으로 정의되는 경우가 있으며 해당 요구사항은 HIPAA(Health Insurance Portability and Accountability Act) 또는 SOX(Sarbanes-Oxley)와 같은 규제 표준으로 구동되는 경우가 있습니다. IBM MQ에서는 이런 표준 준수를 위한 기능을 제공합니다.

예외에만 관심이 있는지 또는 모든 시스템 작동에 대해서 관심이 있는지를 고려하십시오.

일부 감사 측면은 가동 모니터링으로 고려할 수도 있습니다. 감사에 대한 한 가지 탁월함은 실시간 경보만 보는 것이 아니라 히스토리 데이터로 볼 수 있다는 점입니다. 모니터링은 모니터링 및 성능 절에서 설명됩니다.

## 감사하려는 데이터

다음 절에서 설명하는 것처럼 감사해야 하는 데이터 또는 활동 유형을 고려하십시오.

**IBM MQ 인터페이스를 사용하여 IBM MQ에 수행되는 변경**
도구 이벤트, 특히 명령 이벤트 및 구성 이벤트를 발행하도록 IBM MQ를 구성하십시오.

**해당 제어를 벗어난 IBM MQ에 대한 변경**
일부 변경은 IBM MQ 작동 방식에 영향을 줄 수 있지만 IBM MQ에서 직접 모니터할 수는 없습니다. 이런 변경의 예에는 구성 파일 `mqs.ini`, `qm.ini` 및 `mqclient.ini` 변경, 큐 관리자 작성 및 삭제, 사용자 엑시트 프로그램과 같은 2진 파일 설치, 파일 권한 변경이 포함됩니다. 이런 활동을 모니터하려면 운영 체제 레벨에서 실행 중인 도구를 사용해야 합니다. 다른 도구를 사용할 수 있으며 이는 다른 운영 체제에 적합합니다. *sudo*와 같이 관련된 도구로 작성되는 로그가 있을 수도 있습니다.

**IBM MQ의 조작 제어**
큐 관리자 시작 및 중지와 같은 활동을 감사하기 위해 운영 체제를 사용해야 할 수도 있습니다. IBM MQ가 도구 이벤트를 발행하도록 구성 가능한 경우도 있습니다.

**IBM MQ 내에서 애플리케이션 활동**
큐 열기 및 메시지 넣기 및 가져오기와 같은 애플리케이션 조치를 감사하려면 적절한 이벤트를 발행하도록 IBM MQ를 구성하십시오.

**불법 침입 경보**
시도된 보안 결함을 감사하려면 권한 이벤트를 발행하도록 시스템을 구성하십시오. 채널 이벤트는 활동을 표시하는 데 유용할 수도 있으며 특히 채널이 예상치 못하게 종료되는 경우입니다.

## 감사 데이터 캡처, 표시, 아카이브 계획

필요한 다수의 요소가 IBM MQ 이벤트 메시지로 보고됩니다. 해당 메시지를 읽고 형식화할 수 있는 도구를 선택해야 합니다. 장기 기억장치 및 분석에 관심이 있는 경우, 이를 데이터베이스와 같은 보조 기억장치 메커니즘으로 이동해야 합니다. 이런 메시지를 처리하지 않으면 계속 이벤트 큐에 남아 있어서 큐를 채울 수도 있습니다. 일부

이벤트(예: 보안 실패가 발생하는 경우 경보 발행)를 기반으로 조치를 자동으로 수행하는 도구를 구현하도록 결정할 수도 있습니다.

### 시스템이 올바르게 구성되었는지 확인

테스트 세트가 IBM MQ Explorer에서 제공됩니다. 이를 사용하여 오브젝트 정의에서 문제점을 검사하십시오.

또한, 시스템 구성이 예상과 맞는지 주기적으로 검사하십시오. 명령 및 구성 이벤트가 임의의 변경이 수행될 때 보고될 수는 있지만 구성을 덤프하고 이를 알려진 좋은 사본과 비교하는 것도 유용합니다.

## 토폴로지에 의한 보안 계획

이 절에서는 특정 상황, 즉 채널, 큐 관리자 클러스터, 발행/구독, 멀티캐스트 애플리케이션에서의 보안 및 방화벽을 사용하는 경우의 보안에 대해 설명합니다.

자세한 정보는 다음 하위 주제를 참조하십시오.

### 채널 권한 부여

채널을 통해 메시지를 송수신하는 경우, 다양한 IBM MQ 자원에 대해 액세스를 제공해야 합니다. 메시지 채널 에이전트(MCA)는 큐 관리자 사이에서 메시지를 이동하는 필수 IBM MQ 애플리케이션이기 때문에 제대로 작동하기 위해 다양한 IBM MQ 자원에 대한 액세스가 필요합니다.

MCA에 대해 PUT 시에 메시지를 수신하기 위해 MCA에 연관된 사용자 ID 또는 메시지에 연관된 사용자 ID를 사용할 수 있습니다.

CONNECT 시에 **CHLAUTH** 채널 인증 레코드를 사용하여 어설션된 사용자 ID를 대체 사용자에 맵핑할 수 있습니다.

IBM MQ에서 채널은 TLS 지원으로 보호 가능합니다.

MCAUSER 속성이 사용되지 않는 송신자 채널을 제외한 송신 및 수신 채널에 연관된 사용자 ID의 경우 다음 자원에 대한 액세스가 필요합니다.

- 송신 채널에 연관된 사용자 ID에는 큐 관리자, 전송 큐, 데드-레터 큐에 대한 액세스가 필요하며 채널 엑시트에 필요한 다른 모든 자원에 대한 액세스도 필요합니다.
- 수신자 채널의 MCAUSER 사용자 ID에는 *+setall* 권한이 필요합니다. 이유는 수신자 채널이 원격 송신자 채널에 수신된 데이터를 사용하여 전체 컨텍스트 필드가 포함된 전체 MQMD를 작성해야 하기 때문입니다. 따라서, 큐 관리자의 경우 이 활동을 수행하는 사용자에게 *+setall* 권한이 있도록 요구합니다. 이 *+setall* 권한은 다음을 위해 사용자에 부여되어야 합니다.
  - 수신자 채널이 유효하게 메시지를 넣는 모든 큐.
  - 큐 관리자 오브젝트. 자세한 정보는 컨텍스트에 대한 권한 부여를 참조하십시오.
- 발신자가 COA 보고 메시지를 요청한 수신자 채널의 MCAUSER 사용자 ID에는 보고 메시지를 리턴하는 전송 큐에 대한 *+setall* 권한이 필요합니다. 이 권한이 없으면 AMQ8077 오류 메시지가 로깅됩니다.
- 수신 채널에 연관된 사용자 ID로 대상 큐를 열어 메시지를 큐에 넣을 수 있습니다. 이 때 메시지 큐잉 인터페이스(MQI)가 사용되기 때문에 IBM MQ Object Authority Manager(OAM)를 사용 중이지 않은 경우 추가 액세스 제어 검사가 필요할 수 있습니다. MCA에 연관된 사용자 ID(이 주제에 설명된 대로) 또는 메시지에 연관된 사용자 ID(MQMD UserIdentifier 필드에서)에 대해 권한 검사가 수행되는지 여부를 지정할 수 있습니다.

  적용되는 채널 유형에 대해 채널 정의의 **PUTAUT** 매개변수는 해당 검사에 사용되는 사용자 ID를 지정합니다.
  - 채널은 기본적으로 큐 관리자의 서비스 계정을 사용하는데 이 계정은 전체 관리 권한을 가지고 있어서 특별한 권한이 필요하지 않습니다.
  - 서버 연결 채널의 경우, 관리 연결은 CHLAUTH 규칙에 의해 기본적으로 차단되며 명시적인 프로비져닝이 필요합니다.
  - 수신자, 요청자, 클러스터 수신자 유형의 채널을 사용하면 인접 큐 관리자의 로컬 관리가 허용되며 관리에서 이 액세스를 제한하기 위해 단계를 수행하는 경우는 제외됩니다.
  - 수신자 채널의 MCAUSER 사용자 ID에 대해 *dsp* 및 *ctrlx* 권한을 부여할 필요가 없습니다.

- IBM MQ 8.0.0 Fix Pack 4 이전에는 IBM MQ 관리 권한이 없는 사용자 ID를 사용할 경우, 채널에 대한 **dsp** 및 **ctrlx** 권한을 작업하려는 채널의 사용자 ID에 부여해야 합니다.

  IBM MQ 8.0.0 Fix Pack 4부터, 채널이 스스로 다시 동기화하고 순서를 정정 할 때 권한 검사가 없습니다.

  그러나 수동으로 RESET CHANNEL 명령을 실행하는 것은 여전히 모든 릴리스에서 **+dsp** 및 **+ctrlx** 권한이 필요합니다.

  **주의:** 메시지 배치(batch) 확인에 채널 재설정이 필요한 경우 IBM MQ에서 채널을 조회하며 이를 위해서는 **+dsp** 권한이 필요합니다.
- MCAUSER 속성은 SDR 채널 유형에는 사용되지 않습니다.
- 메시지에 연관된 사용자 ID를 사용하는 경우 사용자 ID는 원격 시스템의 사용자 ID일 수 있습니다. 원격 시스템 사용자 ID는 대상 시스템에서 인식되어야 합니다. 다음 명령은 원격 시스템에서 사용자 ID에 권한을 부여할 때 발행할 수 있는 명령 유형의 예입니다.

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect +inq +setall
```

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

여기서 *Profile*은 채널입니다.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

여기서 *Profile*은 데드-레터 큐입니다(설정된 경우).

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

여기서 *Profile*은 권한 부여된 큐의 목록입니다.

**주의:** 명령 큐 및 다른 민감한 시스템 큐에 메시지를 넣기 위해 사용자 ID에 권한을 부여하는 경우에는 주의해야 합니다.

MCA에 연관된 사용자 ID는 MCA 유형에 따라 다릅니다. 두 가지 유형의 MCA가 있습니다.

**호출자 MCA**
채널을 시작하는 MCA. 호출자 MCA는 개별 프로세스, 채널 시작기 스레드 또는 프로세스 풀 스레드로 시작 가능합니다. 사용되는 사용자 ID는 상위 프로세스(채널 시작기)에 연관된 사용자 ID 또는 MCA를 시작하는 프로세스에 연관된 사용자 ID입니다.

**응답자 MCA**
응답자 MCA는 호출자 MCA 요청 결과로 시작되는 MCA입니다. 응답자 MCA는 개별 프로세스, 리스너 스레드 또는 프로세스 풀 스레드로 시작 가능합니다. 사용자 ID는 다음 유형 중 하나일 수 있습니다(환경 설정의 순서대로).

1. APPC에서 호출자 MCA는 응답자 MCA에 사용되는 사용자 ID를 표시할 수 있습니다. 이는 네트워크 사용자 ID라고 하며 개별 프로세스로 시작되는 채널에만 적용됩니다. 채널 정의의 USERID 매개변수를 사용하여 네트워크 사용자 ID를 설정하십시오.
2. **USERID** 매개변수가 사용되지 않는 경우 응답자 MCA의 채널 정의는 MCA가 사용해야 하는 사용자 ID를 지정할 수 있습니다. 채널 정의의 **MCAUSER** 매개변수를 사용하여 사용자 ID를 설정하십시오.
3. 사용자 ID가 이전 두 메소드 중 하나로 설정되지 않은 경우, MCA로 시작되는 프로세스의 사용자 ID 또는 상위 프로세스(리스너)의 사용자 ID가 사용됩니다.

**관련 개념**
47 페이지의 『채널 인증 레코드』
채널 레벨에서 연결 시스템에 부여된 액세스 권한에 대한 보다 정밀한 제어를 실행하려면 채널 인증 레코드를 사용하십시오.

**관련 참조**
채널 인증 레코드 특성

### 채널 시작기 정의 보호
mqm 그룹의 구성원만 채널 시작기를 조작할 수 있습니다.

IBM MQ 채널 시작기는 IBM MQ 오브젝트가 아닙니다. 이에 대한 액세스는 OAM으로 제어되지 않습니다. IBM MQ는 해당 사용자 ID가 mqm 그룹의 구성원인 경우를 제외하고는 사용자나 애플리케이션이 해당 오브젝트를 조작하는 것을 허용하지 않습니다. PCF 명령 **StartChannelInitiator**를 발행하는 애플리케이션이 있는 경우, PCF 메시지의 메시지 디스크립터에 지정된 사용자 ID는 대상 큐 관리자의 mqm 그룹 구성원이어야 합니다.

또한, 사용자 ID는 이스케이프 PCF 명령을 통하거나 간접 모드에서 `runmqsc`를 사용하여 대응하는 MQSC 명령을 발행하도록 대상 시스템에서 MQM 그룹의 구성원이어야 합니다.

### 전송 큐
큐 관리자는 자동으로 원격 메시지를 전송 큐에 넣으며 이를 위해 특수 권한이 필요하지는 않습니다.

그렇지만 메시지를 전송 큐에 직접 넣어야 하는 경우, 특수 권한이 필요합니다. 124 페이지의 표 12의 내용을 참조하십시오.

### 채널 엑시트
채널 인증 레코드가 적합하지 않는 경우, 추가된 보안에 대해 채널 엑시트를 사용할 수 있습니다. 보안 엑시트는 두 보안 엑시트 프로그램 사이에서 안전한 연결을 양식화합니다. 하나의 프로그램은 메시지 채널 에이전트(MCA)를 송신하기 위해 사용되며 다른 하나는 MCA 수신에 사용됩니다.

채널 엑시트에 대한 자세한 정보는 102 페이지의 『채널 엑시트 프로그램』의 내용을 참조하십시오.

### *SSL/TLS*을 사용하는 채널 보호
IBM MQ에서의 TLS 지원은 큐 관리자 인증 정보 오브젝트 및 다양한 MQSC 명령을 사용합니다. 또한, 디지털 인증서 사용도 고려해야 합니다.

### 디지털 인증서 및 키 저장소

큐 관리자 인증서 레이블 속성(**CERTLABL**)을 주요 채널에 사용되는 개인 인증서 이름으로 설정하고 다른 인증서가 필요한 해당 채널에서 인증서 레이블을 설정하여 이를 예외에 대해 대체하는 것은 바람직합니다.

큐 관리자의 기본 인증서 설정과 다른 인증서가 포함된 다수의 채널이 필요한 경우, 여러 개의 큐 관리자에서 채널을 분리하거나 큐 관리자 앞에 QMIPT 프록시를 사용하여 다른 인증서를 표시하도록 해야 합니다.

모든 채널에 대한 다른 인증서를 사용할 수는 있지만 키 저장소에 너무 많은 인증서를 저장하는 경우 TLS 채널 시작 시에 성능이 저하될 수 있습니다. 키 저장소에 있는 인증서 수를 약 50미만으로 유지하고 IBM Global Security Kit (GSKit) 성능이 더 큰 키 저장소에서 급격히 저하되므로 100을 최대로 고려하십시오.

동일한 큐 관리자에 여러 개의 인증서를 허용하면 여러 개의 CA 인증서가 동일한 큐 관리자에서 사용될 가능성이 높아집니다. 그러면 별도의 인증 기관에서 발행되는 인증서에 대해 인증서의 해당 식별 이름 네임스페이스 충돌 가능성이 높아집니다.

전문 인증 기관이 더 주의하는 것과는 달리 회사 내 인증 기관에서는 이름 지정 규칙에서 명확성이 떨어지는 경우가 많아서 임의 CA 사이에서 의도치 않은 일치가 발생하는 경우가 있습니다.

해당 식별 이름뿐만 아니라 인증서 발행인 식별 이름도 검사해야 합니다. 이를 수행하기 위해 채널 인증 SSLPEERMAP 레코드를 사용하고 **SSLPEER** 및 **SSLCERTI** 필드 모두를 해당 DN 및 발행인 DN에 각각 일치하도록 설정하십시오.

### 자체 서명 및 CA 서명 인증서

애플리케이션을 개발 및 테스트하는 경우와 프로덕션에서 이를 사용하는 경우 모두에서 디지털 인증서 사용을 계획하는 것이 중요합니다. CA 서명 인증서 또는 자체 서명 인증서를 큐 관리자 및 클라이언트 애플리케이션 사용법에 따라 사용할 수 있습니다.

**CAN 서명 인증서**
: 프로덕션 시스템의 경우 인증서를 신뢰 가능한 인증 기관(CA)에서 확보하십시오. 외부 CA에서 인증서를 확보하는 경우 서비스 비용을 지불합니다.

**자체 서명 인증서**

애플리케이션을 자체적으로 개발 중인 경우 자체 서명 인증서 또는 로컬 CA 발행 인증서를 플랫폼별로 사용할 수 있습니다.

AIX, Linux, and Windows 시스템에서 자체 서명 인증서를 사용할 수 있습니다. 지시사항은 506 페이지의 『AIX, Linux, and Windows에서 자체 서명된 개인 인증서 작성』의 내용을 참조하십시오.

IBM i 시스템에서 로컬 CA에서 서명된 인증서를 사용할 수 있습니다. 지시사항은 269 페이지의 『IBM i에서 서버 인증서 요청』의 내용을 참조하십시오.

z/OS에서 자체 서명 또는 로컬 CA 서명 인증서를 사용할 수 있습니다. 지시사항은 292 페이지의 『Creating a self-signed personal certificate on z/OS』 또는 293 페이지의 『Requesting a personal certificate on z/OS』의 내용을 참조하십시오.

자체 서명 인증서는 다음과 같은 이유로 프로덕션에서 사용하기에 적합하지 않습니다.

- 자체 서명 인증서는 폐기할 수 없습니다. 폐기하는 경우 개인 키가 손상된 후에 공격자가 ID를 모방할 수 있습니다. CA는 손상된 인증서를 폐기할 수 있으며 폐기된 후에는 더 이상 사용할 수 없습니다. 따라서 CA 서명 인증서는 프로덕션 환경에서 사용하기에 더 안전하며 자체 서명 인증서는 테스트 시스템에서 더 편리합니다.
- 자체 서명 인증서는 만기되지 않습니다. 이는 테스트 환경에서 안전하고 더 편리하지만 프로덕션 환경인 경우 계속 열려 있기 때문에 결국 보안 위반을 초래하게 됩니다. 자체 서명 인증서는 폐기할 수 없다는 점이 이런 위험을 초래합니다.
- 자체 서명 인증서는 개인 인증서 및 루트(또는 신뢰 앵커) CA 인증서 모두로 사용할 수 있습니다. 자체 서명 개인 인증서를 사용하는 사용자는 다른 개인 인증서 서명에 이를 사용할 수도 있습니다. 일반적으로 CA에서 발행되는 개인 인증서에서는 가능하지 않으며 이는 중요한 노출이 됩니다.

## CipherSpec 및 디지털 인증서

지원되는 CipherSpec 서브세트만 디지털 인증서의 지원되는 모든 유형에 사용할 수 있습니다. 따라서, 디지털 인증서에 대해 적합한 CipherSpec을 선택해야 합니다. 이와 유사하게 조직의 보안 정책에서 특정 CipherSpec을 사용하도록 요구하면 적합한 디지털 인증서를 확보해야 합니다.

CipherSpec과 디지털 인증서 사이의 관계에 대한 자세한 정보는 43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』의 내용을 참조하십시오.

## 인증서 유효성 검증 정책

IETF RFC 5280 표준은 준수 애플리케이션 소프트웨어가 위장 공격을 방지하기 위해 구현해야 하는 일련의 인증서 유효성 검증 규칙을 지정합니다. 인증서 유효성 검증 규칙 세트는 인증서 유효성 검증 정책이라고도 합니다. IBM MQ에서의 인증서 유효성 검증 정책에 대한 자세한 정보는 42 페이지의 『IBM MQ의 인증서 유효성 검증 정책』의 내용을 참조하십시오.

## 인증서 폐기 검사 계획

다른 인증 권한의 여러 인증서를 허용하면 불필요한 추가 인증서 폐기 검사가 수행됩니다.

특히, 특정 CA의 폐기 서버 사용을 명시적으로 구성하는 경우(예를 들어, AUTHINFO 오브젝트 또는 인증 정보 레코드(MQAIR) 구조 사용) 다른 CA의 인증서로 표시되면 폐기 검사가 실패합니다.

명시적인 인증서 폐기 서버 구성을 하면 안됩니다. 대신, 각 인증서가 자체 폐기 서버 위치를 인증서 확장자(예: CRL 분배 위치 또는 OCSP AuthorityInfoAccess)에 포함하는 암시적 검사를 사용해야 합니다.

자세한 정보는 OCSPCheckExtensions 및 CDPCheckExtensions를 참조하십시오.

## TLS 지원에 대한 명령 및 속성

TLS(Transport Layer Security) 프로토콜은 도청, 도용, 위장에 대한 보호를 사용하여 채널 보안을 제공합니다. TLS에 대한 IBM MQ 지원을 사용하면 채널 정의에서 특정 채널이 TLS 보안을 사용하도록 지정할 수 있습니다. 또한, 사용하려는 암호화 알고리즘과 같은 필요한 보안 유형의 세부사항을 지정할 수도 있습니다.

- 다음 MQSC 명령이 TLS를 지원합니다.

**ALTER AUTHINFO**
인증 정보 오브젝트 속성을 수정합니다.

**DEFINE AUTHINFO**
인증 정보 오브젝트를 작성합니다.

**DELETE AUTHINFO**
인증 정보 오브젝트를 삭제합니다.

**DISPLAY AUTHINFO**
특정 인증 정보 오브젝트 속성을 표시합니다.

- 다음 큐 관리자 매개변수가 TLS를 지원합니다.

  **CERTLABL**
  사용하려는 개인 인증서 레이블을 정의합니다.

  **키 RPWD**
  AIX, Linux, and Windows 시스템에서 IBM MQ 가 키 저장소에 액세스하는 데 사용하는 비밀번호를 정의합니다. 이 필드는 암호 보호 시스템을 사용하여 암호화됩니다.

  **SSLCRLNL**
  SSLCRLNL 속성은 향상된 TLS 인증서 검사를 허용하기 위해 인증서 폐기 위치를 제공하는 데 사용되는 인증 정보 오브젝트 이름 목록을 지정합니다.

  **SSLCRYP**
  AIX, Linux, and Windows 시스템에서 **SSLCryptoHardware** 큐 관리자 속성을 설정합니다. 이 속성은 시스템에 있는 암호화 하드웨어를 구성하는 데 사용할 수 있는 매개변수 문자열의 이름입니다.

  **SSLEV**
  TLS를 사용하는 채널이 TLS 연결 설정에 실패할 경우 TLS 이벤트 메시지를 보고할 것인지를 판별합니다.

  **SSLFIPS**
  암호화가 암호화 하드웨어가 아니라 IBM MQ에서 수행되는 경우에 FIPS 준수 알고리즘만 사용되는지 여부를 지정합니다. 암호화 하드웨어가 구성된 경우, 하드웨어 제품에서 제공되는 암호화 모듈이 사용되고 이는 특정 레벨에 대한 FIPS를 준수할 수 있습니다. 이는 사용 중인 하드웨어 제품에 따라 다릅니다.

  **SSLKEYR**
  AIX, Linux, and Windows 시스템의 경우 키 저장소를 큐 관리자와 연관시키십시오. GSKit 를 사용하면 AIX, Linux, and Windows 시스템에서 TLS 보안을 사용할 수 있습니다.

  **SSLRKEYC**
  보안 키가 재협상되기 전에 TLS 대화 내에서 송수신되는 바이트 수. 바이트 수에는 MCA에서 전송된 제어 정보가 포함됩니다.

- 다음 채널 매개변수가 TLS를 지원합니다.

  **CERTLABL**
  사용하려는 개인 인증서 레이블을 정의합니다.

  **SSLCAUTH**
  IBM MQ에서 TLS 클라이언트의 인증서 필요 여부 및 유효성 검증 여부를 정의합니다.

  **SSLCIPH**
  암호화 강도 및 기능(CipherSpec)(예: TLS_RSA_WITH_AES_128_CBC_SHA)을 지정합니다. CipherSpec은 채널의 양 측과 일치해야 합니다.

  **SSLPEER**
  허용된 파트너의 식별 이름(고유 ID)을 지정합니다.

이 절에서는 인증 정보 오브젝트를 지원하는 **setmqaut**, **dspmqaut**, **dmpmqaut**, **rcrmqobj**, **rcdmqimg**, **dspmqfls** 명령을 설명합니다. 또한 AIX, Linux, and Windows에서 키 및 인증서를 관리하는 데 사용할 수 있는 명령에 대해서도 설명합니다. 다음 절을 참조하십시오.

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj

- rcdmqimg
- dspmqfls
- 504 페이지의 『AIX, Linux, and Windows에서 키 및 인증서 관리』

TLS를 사용하는 채널 보안 개요는 다음을 참조하십시오.

- 22 페이지의 『IBM MQ의 TLS 보안 프로토콜』

TLS에 연관된 MQSC 명령에 대한 자세한 내용은 다음을 참조하십시오.

- ALTER AUTHINFO
- DEFINE AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

TLS와 연관된 PCF 명령에 대한 자세한 내용은 다음을 참조하십시오.

- 인증 정보 오브젝트 변경, 복사 및 작성
- 인증 정보 오브젝트 삭제
- 인증 정보 오브젝트 조회

### z/OS *IBM MQ for z/OS server connection channel*

The IBM MQ for z/OS SVRCONN channel is not secure without implementing channel authentication, or adding a security exit using TLS. SVRCONN channels do not have a security exit defined by default.

## Security concerns

SVRCONN channels are not secure as initially defined, SYSTEM.DEF.SVRCONN for example. To secure a SVRCONN channel you must set up channel authentication using the SET CHLAUTH command, or install a security exit and implement TLS.

You must use a publicly available sample security exit, write a security exit yourself, or purchase a security exit.

There are several samples available that you can use as a good starting point for writing your own SVRCONN channel security exit.

In IBM MQ for z/OS, the member CSQ4BCX3 in your hlq.SCSQC37S library is a security exit sample written in the C language. Sample CSQ4BCX3 is also shipped pre-compiled in your hlq.SCSQAUTH library.

You can implement the CSQ4BCX3 sample exit by copying the compiled member hlq.SCSQAUTH(CSQ4BCX3) into a load library that is allocated to the CSQXLIB DD in your CHIN Proc. Note that the CHIN requires the load library to be set as "Program Controlled".

Alter your SVRCONN channel to set CSQ4BCX3 as the security exit.

When a client connects using that SVRCONN channel, CSQ4BCX3 will authenticate using the **RemoteUserIdentifier** and **RemotePassword** pair from MQCD or, from IBM MQ for z/OS 9.1.4, the **CSPUserIdPtr** and **CSPPasswordPtr** pair from the MQCSP. If authentication is successful it will copy **RemoteUserIdentifier** into **MCAUserIdentifier**, changing the identity context of the thread.

For Long Term Support and Continuous Delivery before IBM MQ for z/OS 9.1.4, when a client connects using that SVRCONN channel, CSQ4BCX3 will authenticate using the **RemoteUserIdentifier** and **RemotePassword** pair from MQCD. If authentication is successful it will copy **RemoteUserIdentifier** into **MCAUserIdentifier**, changing the identity context of the thread.

If you are writing an IBM MQ Java client you can use pop-ups to query the user and set MQEnvironment.userID and MQEnvironment.password. These values will be passed when the connection is made.

Now that you have a functional security exit, there is the additional concern that the userid and password are being transmitted in plain text across the network when the connection is made, as are the contents

of any subsequent IBM MQ messages. You can use TLS to encrypt this initial connection information as well as the contents of any IBM MQ messages.

## Example

To secure the IBM MQ Explorer SVRCONN channel SYSTEM.ADMIN.SVRCONN complete the following steps:

1. Copy hlq.SCSQAUTH(CSQ4BCX3) into a load library that is allocated to the CSQXLIB DD in the CHINIT Proc.
2. Verify that load library is Program Controlled.
3. Alter the SYSTEM ADMIN.SVRCONN to use security exit CSQ4BCX3.
4. In IBM MQ Explorer, right-click the z/OS Queue Manager name, select **Connection Details** > **Properties** > **Userid** and enter your z/OS user ID.
5. Connect to the z/OS Queue Manager by entering a password.

## Additional information

For exit CSQ4BCX3 to run in a Program Controlled environment, everything loaded into the CHIN address space must be loaded from a Program Controlled library, for example, all libraries in STEPLIB and any libraries named on CSQXLIB DD. To set a load library as Program Controlled issue RACF commands. In the following example the load library name is MY.TEST.LOADLIB.

```
RALTER PROGRAM * ADDMEM('MY.TEST.LOADLIB'//NOPADCHK)
SETROPTS WHEN(PROGRAM)REFRESH
```

To alter the SVRCONN channel to implement CSQ4BCX3, issue the following IBM MQ command:

```
ALTER CHANNEL(SYSTEM ADMIN.SVRCONN) CHLTYPE(SVRCONN) SCYEXIT(CSQ4BCX3)
```

In the example above, the SVRCONN channel name being used is SYSTEM ADMIN.SVRCONN.

See “채널 엑시트 프로그램” on page 102 for more information about channel exits.

**Related tasks**

Writing channel exit programs on z/OS

## *SNA LU 6.2* 보안 서비스

SNA LU 6.2는 세션 레벨 암호화, 세션 레벨 인증, 대화 레벨 인증을 제공합니다.

**참고:** 이 주제 모음에서는 SNA(Systems Network Architecture)에 대해 기본적으로 이해하고 있는 것으로 간주합니다. 이 절에서 언급하는 다른 문서에는 관련 개념 및 용어에 대한 간략한 소개가 포함됩니다. SNA에 대한 더 광범위한 기술적인 소개가 필요하면, *Systems Network Architecture Technical Overview*, GC30-3073을 참조하십시오.

SNA LU 6.2는 세 개의 보안 서비스를 제공합니다.

- 세션 레벨 암호화
- 세션 레벨 인증
- 대화 레벨 인증

세션 레벨 암호화와 세션 레벨 인증에서는 S NA가 데이터 암화화 표준*(DES)* 알고리즘을 사용합니다. DES 알고리즘은 데이터를 암호화하고 복호화하기 위해 대칭 키를 사용하는 블록 암호 알고리즘입니다. 블록 및 키 모두 길이는 8바이트입니다.

세션 레벨 암호화

세션 레벨 암호화는 DES 알고리즘을 사용하여 세션 데이터를 암호화하고 복호화합니다. 따라서, SNA LU 6.2 채널에서 링크 레벨 기밀성 서비스를 제공하는 데 사용될 수 있습니다.

논리 장치(LU)는 필수(또는 필요한) 데이터 암호화, 선택 데이터 암호화 또는 데이터 암호화 없음을 제공합니다.

필수 암호화 세션에서 LU는 모든 아웃바운드 데이터 요청 장치를 암호화하고 모든 인바운드 데이터 요청 장치를 복호화합니다.

선택 암호화 세션에서 LU는 송신 트랜잭션 프로그램(TP)로 지정된 데이터 요청 장치만 암호화합니다. 송신 LU는 요청 헤더에 표시기를 설정하여 데이터가 암호화되었다는 것을 신호합니다. 이 표시기를 검사하여 수신 LU는 요청 장치를 수신 TP로 전달하기 전에 복호화해야 하는 요청 장치를 알려줄 수 있습니다.

SNA 네트워크에서 IBM MQ MCA는 트랜잭션 프로그램입니다. MCA는 송신하는 어떠한 데이터에 대한 암호화도 요청하지 않습니다. 선택 데이터 암호화가 옵션이 아니므로, 필수 데이터 암호화나 데이터 암호화 없음만이 세션에서 가능합니다.

필수 데이터 암호화 구현 방법에 대한 정보는 SNA 서브시스템 문서를 참조하십시오. z/OS의 Triple DES 24바이트암호화와 같이플랫폼에서 사용할 수 있는 더 강력한 암호화 양식에 대한 정보는 동일한 문서를 참조하십시오.

세션 레벨 암호화에 대한 일반적인 정보는 *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808을 참조하십시오.

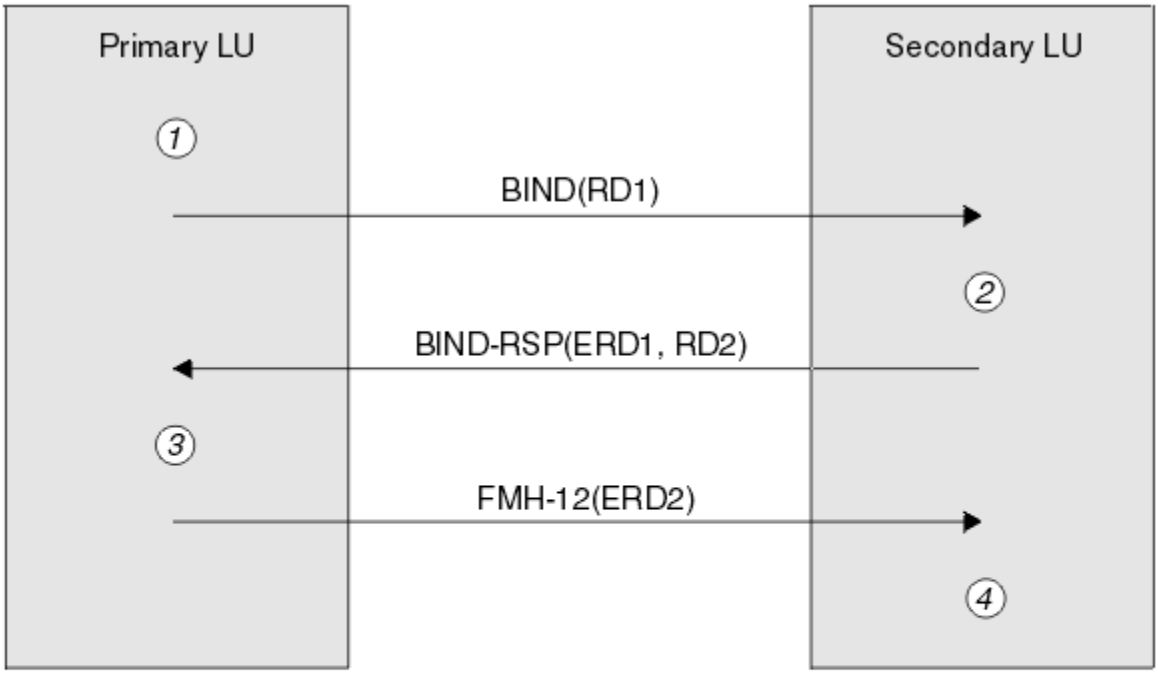
### 세션 레벨 인증

세션 레벨 인증은 두 개의 LU가 세션을 활성화하고 있는 동안에 서로 인증하게 하는 세션 레벨 보안 프로토콜입니다. *LU-LU* 확인이라고도 합니다.

LU는 네트워크에서 시스템으로의 효율적인 "게이트웨이"이기 때문에, 어떤 상황에서는 이런 레벨의 인증이 충분하다고 고려할 수도 있습니다. 예를 들면, 큐 관리자가, 제어되고 신뢰되는 환경에서 실행 중인 리모트 큐 관리자와 메시지를 교환해야 하면, LU가 인증된 후에는 원격 시스템의 나머지 구성요소의 ID를 신뢰할 준비가 되어 있을 수 있습니다.

각 LU가 그 파트너의 비밀번호를 확인하여 세션 레벨 인증이 수행됩니다. 하나의 비밀번호가 LU의 각 쌍 사이에서 설정되기 때문에 비밀번호를 *LU-LU* 비밀번호라고 합니다. LU-LU 비밀번호가 설정되는 방식은 구현에 따라 다르고 SNA의 범위 밖에 있습니다.

114 페이지의 그림 12는 세션 레벨 인증에 대한 플로우를 나타냅니다.




Legend:

| | | |
|---|---|---|
| BIND | = | BIND request unit |
| BIND-RSP | = | BIND response unit |
| ERD | = | Encrypted random data |
| FMH-12 | = | Function Management Header 12 |
| RD | = | Random data |

그림 *12.* 세션 레벨 인증 플로우

세션 레벨 인증의 프로토콜은 다음과 같습니다. 프로시저에 있는 번호는 114 페이지의 그림 12의 번호에 해당합니다.

1. 1차 LU가 무작위 데이터 값(RD1)을 생성하고 BIND 요청으로 2차 LU에게 송신합니다.
2. 2차 LU가 무작위 데이터가 있는 BIND 요청을 수신하면, DES 알고리즘을 사용하여 데이터를 암호화하고 LU-LU 비밀번호의 사본을 키로서 가집니다. 그러면 2차 LU가 2차 무작위 데이터 값(RD2)를 생성하고 이를 암호화된 데이터(ERD1)와 함께 BIND 응답의 1차 LU로 송신합니다.
3. 1차 LU가 BIND 응답을 수신하면, 원래 생성했던 무작위 데이터에서 암호화된 데이터의 자체 버전을 처리합니다. DES 알고리즘을 사용하여 이렇게 하고 LU-LU 비밀번호의 사본을 키로서 가집니다. 그런 후, BIND 응답에서 수신한 암호화된 데이터와 함께 그 버전을 비교합니다. 두 값이 같으면, 1차 LU는 2차 LU가 같은 비밀번호를 가지고 있다는 것을 알게 되고 2차 LU가 인증됩니다. 두 값이 일치하지 않으면, 1차 LU가 세션을 종료합니다.

   그런 다음, 1차 LU는 BIND 응답에서 수신한 무작위 데이터를 암호화하고 암호화된 데이터(ERD2)를 FMH-12(함수 관리 헤더 12)로 2차 LU에게 송신합니다.
4. 2차 LU가 FMH-12를 수신하면, 원래 생성했던 무작위 데이터에서 암호화된 데이터의 자체 버전을 처리합니다. 그런 후, FMH-12로 수신한 암호화된 데이터와 함께 해당 버전을 비교합니다. 두 값이 같으면, 1차 LU가 인증됩니다. 두 값이 일치하지 않으면, 2차 LU가 세션을 종료합니다.

중간자 공격에 대해 더 나은 보호를 제공하는 프로토콜의 개선된 버전에서, 2차 LU는 LU-LU 비밀번호의 사본을 키로서 사용하여 RD1, RD2와 2차 LU의 전체 이름으로부터 DES 메시지 인증 코드(MAC)를 처리합니다. 2차 LU가 ERD1 대신에 BIND 응답으로 MAC을 1차 LU에게 송신합니다.

1차 LU는 BIND 응답에서 수신한 MAC과 비교하는 자신의 MAC 버전을 처리하여 2차 LU를 인증합니다. 그런 후, 1차 LU는 RD1과 RD2의 두 번째 MAC을 처리하고, MAC을 ERD2 대신에 FMH-12로 2차 LU에게 송신합니다.

2차 LU는 FMH-12에서 수신한 MAC과 비교하는 자신의 2차 MAC 버전을 처리하여 1차 LU를 인증합니다.

세션 레벨 인증 구성 방법에 대한 정보는 SNA 서브시스템 문서를 참조하십시오. 세션 레벨 인증에 대해 더 일반적인 정보는 *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808을 참조하십시오.

대화 레벨 인증
로컬 TP가 파트너 TP로 대화를 할당하려고 할 때, 로컬 LU가 파트너 LU로 첨부 요청을 송신하며, 파트너 LU가 파트너 TP를 첨부하도록 요청합니다. 어떤 상황에서는 첨부 요청이 보안 정보를 포함할 수 있으며, 파트너 LU가 로컬 TP를 인증하는 데 이를 사용할 수 있습니다. 이는 대화 레벨 인증 또는 일반 사용자 확인이라고 합니다.

다음 주제에서는 IBM MQ가 대화 레벨 인증에 대한 지원을 제공하는 방법에 대해 설명합니다.

대화 레벨 인증에 대한 자세한 정보는 *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808을 참조하십시오.

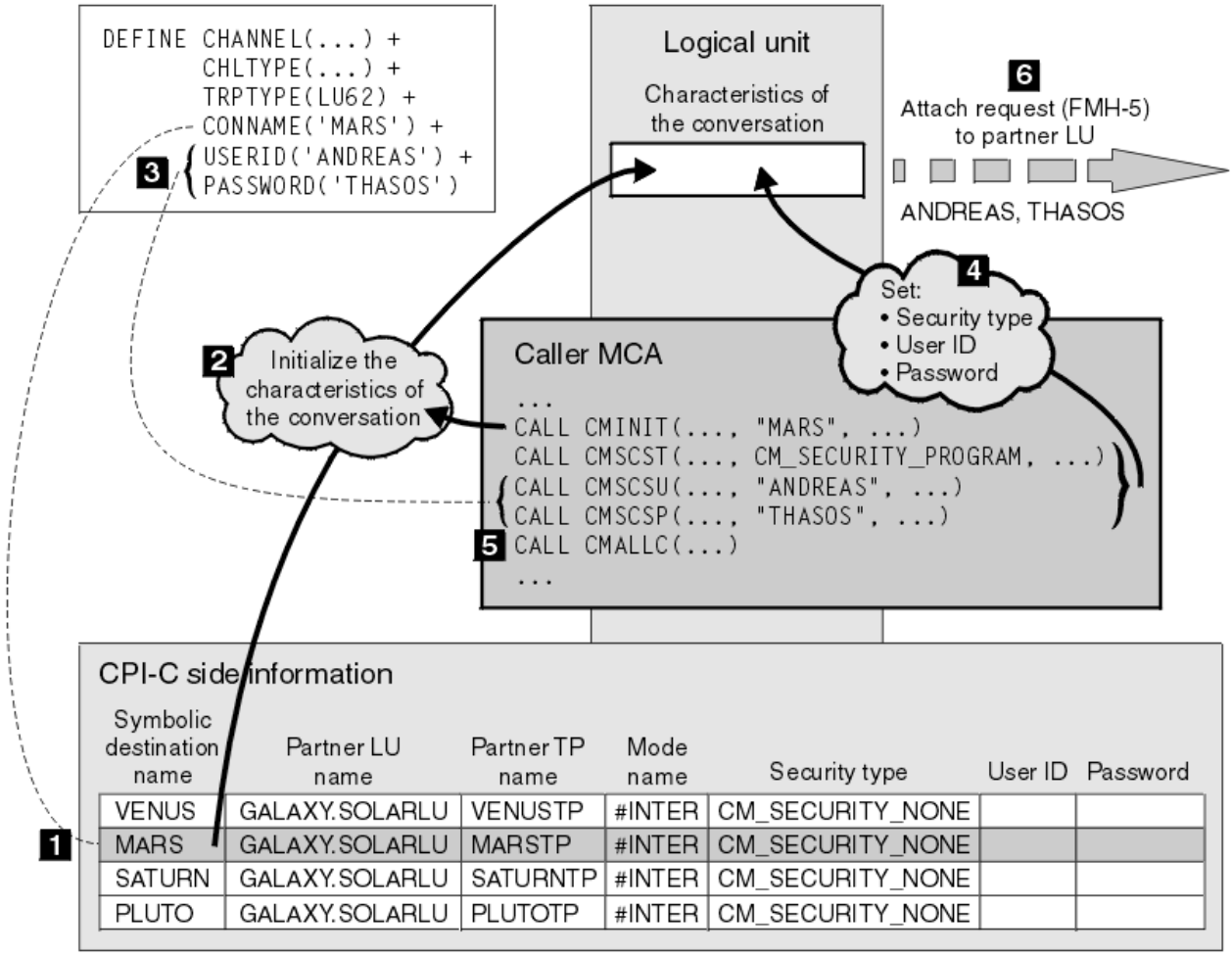
z/OS z/OS에 특정한 정보는 z/OS MVS Planning: APPC/MVS Management를 참조하십시오.

CPI-C에 대한 자세한 정보는 CPI 통신 사용을 참조하십시오.

APPC/MVS TP 대화 호출 가능 서비스에 대한 자세한 정보는 APPC/MVS TP 대화 호출 가능 서비스를 참조하십시오.

Multi 멀티플랫폼에서 대화 레벨 인증 지원
이 주제를 사용하여 멀티플랫폼에서 대화 레벨 인증이 작동하는 방식을 대략적으로 파악하십시오.

멀티플랫폼에서의 대화 레벨 인증 지원은 115 페이지의 그림 13에 설명되어 있습니다. 다이어그램에 있는 번호는 그 뒤에 오는 설명에 있는 번호에 해당합니다.




| Symbolic destination name | Partner LU name | Partner TP name | Mode name | Security type | User ID | Password |
|---|---|---|---|---|---|---|
| VENUS | GALAXY.SOLARLU | VENUSTP | #INTER | CM_SECURITY_NONE | | |
| MARS | GALAXY.SOLARLU | MARSTP | #INTER | CM_SECURITY_NONE | | |
| SATURN | GALAXY.SOLARLU | SATURNTP | #INTER | CM_SECURITY_NONE | | |
| PLUTO | GALAXY.SOLARLU | PLUTOTP | #INTER | CM_SECURITY_NONE | | |

그림 *13.* 대화 레벨 인증에 대한 *IBM MQ* 지원

멀티플랫폼에서 MCA는 CPI-C(Common Programming Interface Communications) 호출을 사용하여 SNA 네트워크에서 파트너 MCA와 통신합니다. 채널의 호출자 측에 있는 채널 정의에서, CONNAME 매개변수의 값은 목적지 기호 이름이며, 이는 CPI-C 부가 정보 항목(1)을 식별합니다. 이 항목은 다음을 지정합니다.

- 파트너 LU의 이름
- 응답 MCA인 파트너 TP의 이름
- 대화에 사용할 모드의 이름

부가 정보 항목은 다음 보안 정보도 지정할 수 있습니다.

- 보안 유형

  흔히 구현되는 보안 유형은 CM_SECURITY_NONE, CM_SECURITY_PROGRAM, CM_SECURITY_SAME이지만, 다른 것이 CPI-C 스펙에 정의되어 있습니다.
- 사용자 ID
- 비밀번호

호출자 MCA가 호출에서 매개변수 중 하나로서 CONNAME의 값을 사용하여 CPI-C 호출 CMINIT를 발행함으로써 응답자 MCA에 대화를 할당하기 위해 준비합니다. CMINIT 호출이 로컬 LU의 특성으로, MCA가 대화에 사용하려고 하는 부가 정보 항목을 식별합니다. 로컬 LU는 대화의 특성을 초기화하기 위해 이 항목에 있는 값을 사용합니다(2).

그런 다음 호출자 MCA가 채널 정의에서 USERID와 PASSWORD 매개변수의 값을 검사합니다(3). USERID가 설정되면, 호출자 MCA가 다음 CPI-C 호출을 발행합니다(4).

- CMSCST - 대화의 보안 유형을 CM_SECURITY_PROGRAM으로 설정.
- CMSCSU - 대화의 사용자 ID를 USERID의 값으로 설정.
- CMSCSP - 대화의 비밀번호를 PASSWORD의 값으로 설정. PASSWORD가 설정되지 않으면, CMSCSP는 호출되지 않습니다.

이 호출이 설정하는 보안 유형, 사용자 ID와 비밀번호는 이전에 부가 정보 항목에서 얻은 모든 값을 대체합니다.

그런 후, 호출자 MCA가 CPI-C 호출 CMALLC를 발행하여 대화를 할당합니다(5). 이 호출에 응답하여, 로컬 LU가 첨부 요청(함수 관리 헤더 5, 즉 FMH-5)을 파트너 LU로 송신합니다(6).

파트너 LU가 사용자 ID와 비밀번호를 승인하면, USERID와 PASSWORD의 값이 첨부 요청에 포함됩니다. 파트너 LU가 사용자 ID와 비밀번호를 승인하지 않으면, 값이 첨부 요청에 포함되지 않습니다. LU들이 세션을 양식화하기 위해 바인딩할 때 로컬 LU는 파트너 LU가 사용자 ID와 비밀번호를 정보 교환의 일부로서 승인할 것인지를 알아냅니다.

첨부 요청의 이후 버전에서는 비밀번호 대체가 명확한 비밀번호 대신에 LU 사이에서 플로우될 수 있습니다. 비밀번호 대체는 비밀번호에서 양식화된 DES 메시지 인증 코드(MAC) 또는 SHA-1 메시지 요약입니다. 비밀번호 대체는 두 LU가 모두 그것을 지원하는 경우에만 사용될 수 있습니다.

파트너 LU가 사용자 ID와 비밀번호가 있는 첨부 요청을 수신하면, 식별과 인증을 목적으로 사용자 ID와 비밀번호를 사용할 수 있습니다. 액세스 제어 목록을 참조하여, 파트너 LU는 사용자 ID가 대화를 할당하고 응답 MCA를 첨부할 수 있는 권한을 가지는지도 판별할 수 있습니다.

또한, 응답 MCA는 첨부 요청에 포함된 사용자 ID에서 실행될 수 있습니다. 이런 경우에, 사용자 ID는 응답 MCA에 대한 디폴트 사용자 ID가 되고 MCA가 큐 관리자에 연결하려고 할 때 권한 검사에 사용됩니다. 이는 MCA가 큐 관리자의 자원에 액세스하려고 할 때 계속해서 권한 검사를 위해 사용될 수도 있습니다.

첨부 요청의 사용자 ID와 비밀번호가 식별, 인증, 액세스 제어를 위해 사용될 수 있는 방법은 구현에 따라 다릅니다. 사용자의 SNA 서브시스템에만 적용되는 정보는 해당 문서를 참조하십시오.

USERID가 설정되지 않으면, 호출자 MCA가 CMSCST, CMSCSU, CMSCSP를 호출하지 않습니다. 이런 경우에, 첨부 요청에서 플로우되는 보안 정보는 부가 정보 항목에 지정된 것과 파트너 LU가 승인하는 것에 의해서만 판별됩니다.

*Conversation level authentication and IBM MQ for z/OS*

Use this topic to gain an overview of how conversation level authentication works, on z/OS.

On IBM MQ for z/OS, MCAs do not use CPI-C. Instead, they use APPC/MVS TP Conversation Callable Services, an implementation of Advanced Program-to-Program Communication (APPC), which has some

CPI-C features. When a caller MCA allocates a conversation, a security type of SAME is specified on the call. Therefore, because an APPC/MVS LU supports persistent verification only for inbound conversations, not for outbound conversations, there are two possibilities:

- If the partner LU trusts the APPC/MVS LU and will accept an already verified user ID, the APPC/MVS LU sends an attach request containing:
  - The channel initiator address space user ID
  - A security profile name, which, if RACF is used, is the name of the current connect group of the channel initiator address space user ID
  - An already verified indicator
- If the partner LU does not trust the APPC/MVS LU and will not accept an already verified user ID, the APPC/MVS LU sends an attach request containing no security information.

On IBM MQ for z/OS, the USERID and PASSWORD parameters on the DEFINE CHANNEL command cannot be used for a message channel and are valid only at the client connection end of an MQI channel. Therefore, an attach request from an APPC/MVS LU never contains values specified by these parameters.

## 큐 관리자 클러스터의 보안

큐 관리자 클러스터가 사용하기는 편리하지만 보안에는 주의해야 합니다.

큐 관리자 클러스터는 일부 방식에서 논리적으로 연관된 큐 관리자의 네트워크입니다. 클러스터의 멤버인 큐 관리자를 클러스터 큐 관리자라 합니다.

클러스터 큐 관리자에 속하는 큐를 같은 클러스터의 다른 큐 관리자가 알도록 할 수 있습니다. 이런 큐를 클러스터 큐라고 합니다. 클러스터의 모든 큐 관리자는 다음 중에서 아무것도 알지 못해도 클러스터 큐에 메시지를 송신할 수 있습니다.

- 각 클러스터 큐에 대한 명시적인 리모트 큐 정의
- 각 리모트 큐 관리자로 송신 및 수신을 위해 명백히 정의된 채널
- 각 아웃바운드 채널에 대한 별도의 전송 큐

두 개 이상의 큐 관리자가 복제되는 클러스터를 작성할 수 있습니다. 즉, 큐 관리자가 클러스터 큐로 선언된 모든 로컬 큐를 비롯하여 동일한 로컬 큐의 인스턴스를 포함하고 동일한 서버 애플리케이션의 인스턴스를 지원할 수 있습니다.

클러스터 큐 관리자에 연결된 애플리케이션이 메시지를 복제된 큐 관리자 각각에서 인스턴스를 포함하는 클러스터 큐로 송신하면 IBM MQ는 이를 송신할 큐 관리자를 결정합니다. 다수의 애플리케이션이 메시지를 클러스터 큐에 송신하는 경우, IBM MQ는 큐의 인스턴스를 포함하는 각 큐 관리자에서 워크로드 밸런스를 유지합니다. 복제된 큐 관리자를 호스트하는 시스템 중 하나가 실패하면 IBM MQ는 실패한 시스템에 재시작될 때까지 나머지 큐 관리자에서 워크로드 밸런스를 계속 유지합니다.

큐 관리자 클러스터를 사용하는 경우 다음의 보안 문제점을 고려해야 합니다.

- 선택된 큐 관리자만 큐 관리자로 메시지를 송신하도록 허용
- 리모트 큐 관리자의 선택된 사용자만 큐 관리자의 큐로 메시지를 송신하도록 허용
- 큐 관리자에 연결된 애플리케이션이 선택된 리모트 큐로만 메시지를 송신하도록 허용

클러스터를 사용하고 있지 않아도 이 사항들은 관련이 있으며, 클러스터를 사용하고 있으면 더욱 중요합니다.

애플리케이션이 한 클러스터 큐로 메시지를 송신하면, 추가 리모트 큐 정의나 전송 큐 또는 채널 없이도 다른 모든 클러스터 큐로 메시지를 송신할 수 있습니다. 따라서, 큐 관리자의 클러스터 큐에 대한 액세스를 제한하고 애플리케이션이 메시지를 송신할 수 있는 클러스터 큐를 제한해야 하는지를 고려하는 것이 더욱 중요할 수 있습니다.

다음은 큐 관리자 클러스터를 사용하고 있는 경우에만 관련이 있는 일부 추가 보안 고려사항입니다.

- 선택된 큐 관리자만 클러스터를 조인하도록 허용
- 필요하지 않은 큐 관리자는 클러스터에서 강제로 제거

이 모든 고려사항에 대한 자세한 정보는 클러스터 안전 유지를 참조하십시오. z/OS IBM MQ for z/OS 고유의 고려사항은 248 페이지의 『Security in queue manager clusters on z/OS』의 내용을 참조하십시오.

**관련 태스크**

449 페이지의 『큐 관리자가 메시지 수신하는 것을 방지』
엑시트 프로그램을 사용하여 클러스터 큐 관리자가 수신할 권한이 없는 메시지를 수신하는 것을 막을 수 있습니다.

## IBM MQ 발행/구독에 대한 보안

IBM MQ 발행/구독을 사용 중인 경우에는 추가 보안 고려사항이 있습니다.

발행/구독 시스템에는 발행자와 구독자의 두 애플리케이션 유형이 있습니다. 발행자는 IBM MQ 메시지의 양식으로 정보를 제공합니다. 발행자가 메시지를 발행하면, 메시지에 있는 정보의 주제를 식별하는 토픽을 지정합니다.

구독자는 발행되는 정보의 사용자입니다. 구독자는 토픽을 구독하여 관심 토픽을 지정합니다.

큐 관리자는 IBM MQ 발행/구독에서 제공되는 애플리케이션입니다. 이는 발행자에서 발행된 메시지를 수신하고 구독자에서 구독 요청을 수신하여 발행된 메시지를 구독자에게 라우트합니다. 구독자는 구독된 해당 토픽의 메시지만 수신합니다.

자세한 정보는 발행/구독 보안을 참조하십시오.

## 멀티캐스트 보안

이 정보를 사용하여 보안 프로세스가 IBM MQ 멀티캐스트에 필요한 이유를 이해하십시오.

IBM MQ Multicast는 기본 제공 보안이 필요하지 않습니다. 보안 검사는 MQOPEN 시에 큐 관리자에서 처리되고 MQMD 필드 설정은 클라이언트에서 처리됩니다. 네트워크에 있는 일부 애플리케이션은 IBM MQ 애플리케이션이 아닐 수 있습니다(예: LLM 애플리케이션. 자세한 정보는 IBM MQ LLM(Low Latency Messaging)과의 멀티캐스트 상호 운용성 참조). 따라서 수신 애플리케이션이 컨텍스트 필드의 유효성을 확인할 수 없으므로 자체 보안 프로시저를 구현해야 할 수도 있습니다.

고려해야 하는 보안 프로세스는 세 가지입니다.

**액세스 제어**

IBM MQ의 액세스 제어는 사용자 ID를 기반으로 합니다. 이 주제에 대한 자세한 정보는 96 페이지의 『클라이언트의 액세스 제어』의 내용을 참조하십시오.

**네트워크 보안**

격리된 네트워크는 허위 메시지를 방지하는 가능한 보안 옵션일 수 있습니다. 멀티캐스트 그룹 주소의 애플리케이션이 고유 통신 기능을 사용하여 악의적인 메시지(동일한 멀티캐스트 그룹 주소의 애플리케이션에서 오기 때문에 MQ 메시지가 구분할 수 없음)를 발행할 수 있습니다.

멀티캐스트 그룹 주소의 클라이언트가 동일한 멀티캐스트 그룹 주소의 다른 클라이언트에 대한 메시지를 수신할 수도 있습니다.

멀티캐스트 네트워크를 격리하면 올바른 클라이언트 및 애플리케이션만 액세스를 가지도록 할 수 있습니다. 이 보안 예방책을 사용하여 악의적인 메시지가 수신되는 것과 기밀 정보가 송신되는 것을 방지할 수 있습니다.

멀티캐스트 그룹 네트워크 주소에 대한 정보는 멀티캐스트 트래픽에 적절한 네트워크 설정을 참조하십시오.

**디지털 서명**

디지털 서명은 메시지의 표현을 암호화하여 형성됩니다. 암호화는 서명인의 개인 키를 사용하고, 효율성을 위해 보통 메시지 자체가 아니라 메시지 요약에 대해 작동합니다. MQPUT 이전에는 메시지 디지털 서명이 좋은 보안 해결책이었지만 이 프로세스는 메시지의 볼륨이 큰 경우 성능에 좋지 않은 영향을 줄 수도 있습니다.

디지털 서명은 서명 중인 데이터에 따라 다릅니다. 두 개의 다른 메시지가 같은 엔티티에 의해 디지털로 서명되면 두 서명은 서로 다르지만 둘 다 같은 공개 키, 즉, 이 메시지에 서명한 엔티티의 공개 키를 사용하여 확인됩니다.

이 절의 앞에서 언급한 것처럼 멀티캐스트 그룹 주소의 애플리케이션이 고유 통신 기능을 사용하여 악의적인 메시지를 발행할 수 있으며, 이는 MQ 메시지로부터 구별할 수 없습니다. 디지털 서명은 원본 증명을 제공하며 송신자만 개인 키를 알기 때문에 송신자가 메시지의 진원지인 강력한 증거를 제시합니다.

이 주제에 대한 자세한 정보는 10 페이지의 『암호화 개념』의 내용을 참조하십시오.

## 방화벽 및 IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru 는 방화벽을 통해 통신을 단순화할 수 있습니다.

MQIPT 를 사용하면 두 개의 큐 관리자가 직접 TCP/IP 연결 없이도 메시지를 교환하거나 IBM MQ 클라이언트 애플리케이션이 큐 관리자에 연결할 수 있습니다. 이 구조는 방화벽이 두 시스템 간의 직접 TCP/IP 연결을 금지하는 경우에 유용합니다. MQIPT 를 프록시로 사용하면 방화벽을 통해 IBM MQ 채널 데이터를 더 간단하고 관리하기 쉽게 전달할 수 있습니다. MQIPT 는 또한 HTTP내에서 TLS (Transport Layer Security) 및 터널 IBM MQ 데이터를 사용하여 인터넷을 통해 전송되는 IBM MQ 데이터를 보호할 수 있습니다.

자세한 정보는 IBM MQ Internet Pass-Thru의 내용을 참조하십시오.

## z/OS IBM MQ for z/OS security implementation checklist

This topic gives a step-by-step procedure you can use to work out and define the security implementation for each of your IBM MQ queue managers.

RACF provides definitions for the IBM MQ security classes in its supplied static Class Descriptor Table (CDT). As you work through the checklist, you can determine which of these classes your setup requires. You must ensure that they are activated as described in "RACF security classes" on page 174.

Refer to other sections for details, in particular "Profiles used to control access to IBM MQ resources" on page 183.

If you require security checking, follow this checklist to implement it:

1. Activate the RACF MQADMIN (uppercase profiles) or MXADMIN (mixed case profiles) class.
   - Do you want security at queue sharing group level, queue manager level, or a combination of both?

     See, "Profiles to control queue sharing group or queue manager level security" on page 178.
2. Do you need connection security?
   - **Yes**: Activate the MQCONN class. Define appropriate connection profiles at either queue manager level or queue sharing group level in the MQCONN class. Then permit the appropriate users or groups access to these profiles.

     **Note:** Only users of the MQCONN API request or CICS or IMS address space user IDs need to have access to the corresponding connection profile.
   - **No**: Define an hlq.NO.CONNECT.CHECKS profile at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class.
3. Do you need security checking on commands?
   - **Yes**: Activate the MQCMDS class. Define appropriate command profiles at either queue manager level or queue sharing group level in the MQCMDS class. Then permit the appropriate users or groups access to these profiles.

     If you are using a queue sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator. See "Setting up IBM MQ for z/OS resource security" on page 240.
   - **No**: Define an hlq.NO.CMD.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
4. Do you need security on the resources used in commands?
   - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define appropriate profiles for protecting resources on commands at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the appropriate users or groups access to these

profiles. Set the CMDUSER parameter in CSQ6SYSP to the default user ID to be used for command security checks.

If you are using a queue sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator. See "Setting up IBM MQ for z/OS resource security" on page 240.

- **No**: Define an hlq.NO.CMD.RESC.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.

5. Do you need queue security?
   - **Yes**: Activate the MQQUEUE or MXQUEUE class. Define appropriate queue profiles for the required queue manager or queue sharing group in the MQQUEUE or MXQUEUEclass. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.QUEUE.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
6. Do you need process security?
   - **Yes**: Activate the MQPROC or MXPROC class. Define appropriate process profiles at either queue manager or queue sharing group level and permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.PROCESS.CHECKS profile for the appropriate queue manager or queue sharing group in the MQADMIN or MXADMIN class.
7. Do you need namelist security?
   - **Yes**: Activate the MQNLIST or MXNLISTclass. Define appropriate namelist profiles at either queue manager level or queue sharing group level in the MQNLIST or MXNLIST class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.NLIST.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
8. Do you need topic security?
   - **Yes**: Activate the MXTOPIC class. Define appropriate topic profiles at either queue manager level or queue sharing group level in the MXTOPIC class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.TOPIC.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
9. Do any users need to protect the use of the `MQOPEN` or `MQPUT1` options relating to the use of context?
   - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define hlq.CONTEXT.queuename profiles at the queue, queue manager, or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.CONTEXT.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
10. Do you need to protect the use of alternative user IDs?
    - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define the appropriate hlq.ALTERNATE.USER. *`alternateuserid`* profiles for the required queue manager or queue sharing group and permit the required users or groups access to these profiles.
    - **No**: Define the profile hlq.NO.ALTERNATE.USER.CHECKS for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
11. Do you need to tailor which user IDs are to be used for resource security checks through RESLEVEL?
    - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define an hlq.RESLEVEL profile at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the required users or groups access to the profile.

- **No**: Ensure that no generic profiles exist in the MQADMIN or MXADMIN class that can apply to hlq.RESLEVEL. Define an hlq.RESLEVEL profile for the required queue manager or queue sharing group and ensure that no users or groups have access to it.

12. Do you need to 'timeout' unused user IDs from IBM MQ ?
    - **Yes**: Determine what timeout values you would like to use and issue the MQSC ALTER SECURITY command to change the TIMEOUT and INTERVAL parameters.
    - **No**: Issue the MQSC ALTER SECURITY command to set the INTERVAL value to zero.

    **Note:** Update the CSQINP1 initialization input data set used by your subsystem so that the MQSC ALTER SECURITY command is issued automatically when the queue manager is started.
13. Do you use distributed queuing?
    - **Yes**: Use channel authentication records. For more information, see "채널 인증 레코드" on page 47.
    - You can also determine the appropriate MCAUSER attribute value for each channel, or provide suitable channel security exits.
14. Do you want to use Transport Layer Security (TLS)?
    - **Yes**: To specify that any user presenting an TLS personal certificate containing a specified DN is to use a specific MCAUSER, set a channel authentication record of type SSLPEERMAP. You can specify a single distinguished name or a pattern including wildcards.
    - Plan your TLS infrastructure. Install the System SSL feature of z/OS. In RACF, set up your certificate name filters (CNFs), if you are using them, and your digital certificates. Set up your SSL key ring. Ensure that the SSLKEYR queue manager attribute is nonblank and points to your SSL key ring. Also ensure that the value of SSLTASKS is at least 2.
    - **No**: Ensure that SSLKEYR is blank, and SSLTASKS is zero.

    For further details about TLS, see "IBM MQ의 TLS 보안 프로토콜" on page 22.
15. Do you use clients?
    - **Yes**: Use channel authentication records.
    - You can also determine the appropriate MCAUSER attribute value for each server-connection channel, or provide suitable channel security exits if required.
16. Check your switch settings.

    IBM MQ issues messages when the queue manager is started that display your security settings. Use these messages to determine whether your switches are set correctly.
17. Do you send passwords from client applications?
    - **Yes**: Ensure that the z/OS feature is installed and Integrated Cryptographic Service Facility (ICSF) is started for the best protection.
    - **No**: You can ignore the error message reporting that ICSF has not started.

    For further information about ICSF see "Using the Integrated Cryptographic Service Facility (ICSF)" on page 248

# 보안 설정

이 토픽 콜렉션에는 다른 운영 체제 및 클라이언트 사용에 특정되는 정보가 포함됩니다.

## ALW AIX, Linux, and Windows에서 보안 설정

AIX, Linux, and Windows 시스템 특정 보안 고려사항입니다.

IBM MQ 큐 관리자는 잠재적으로 중요한 정보를 전송하기 때문에 권한 시스템을 사용하여 권한 없는 사용자가 큐 관리자에 액세스하지 못하도록 해야 합니다. 다음과 같은 보안 제어 유형을 고려하십시오.

**IBM MQ 관리 사용자**
IBM MQ 관리를 위해 명령을 발행할 수 있는 사용자 세트를 정의할 수 있습니다.

**IBM MQ 오브젝트 사용자**
MQI 호출 및 PCF 명령을 사용하여 다음을 수행할 수 있는 사용자(주로 애플리케이션)를 정의할 수 있습니다.

- 큐 관리자 연결 사용자
- 오브젝트(큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트)에 액세스할 수 있는 사용자 및 해당 오브젝트에 대해 필요한 액세스 유형.
- IBM MQ 메시지에 액세스하는 사용자.
- 메시지와 연관된 컨텍스트 정보에 액세스할 수 있는 사용자.

**채널 보안**
메시지를 원격 시스템에 송신하는 데 사용되는 채널이 필수 자원에 액세스할 수 있도록 해야 합니다.

표준 조작 기능을 사용하여 프로그램 라이브러리, MQI 링크 라이브러리 및 명령에 대한 액세스를 부여할 수 있습니다. 그렇지만 큐 및 기타 큐 관리자 데이터가 포함된 디렉토리는 IBM MQ에 대해 개인용입니다. MQI 자원에 권한을 부여하거나 취소하는 경우에는 표준 운영 체제 명령을 사용하지 마십시오.

## ALW AIX, Linux, and Windows에서의 권한 작동 방법

이 절의 주제에 있는 권한 스펙 표는 권한이 작동하는 방법 및 적용되는 제한사항을 정확하게 정의합니다.

표는 다음과 같은 상황에 적용됩니다.

- MQI 호출을 발행한 애플리케이션
- 이스케이프 PCF로 MQSC 명령을 발행하는 관리 프로그램
- PCF 명령을 발행하는 관리 프로그램

이 절에서 정보는 다음을 지정하는 표의 세트로 표시됩니다.

**수행할 조치**
MQI 옵션, MQSC 명령 또는 PCF 명령

**액세스 제어 오브젝트**
큐, 프로세스, 큐 관리자, 이름 목록, 인증 정보, 채널, 클라이언트 연결 채널, 리스너 또는 서비스.

**필수 권한**
MQZAO_ 상수로 표현됩니다.

표에서 접두부가 MQZAO_인 상수는 특정 엔티티를 위한 `setmqaut` 명령에 대한 권한 목록의 키워드에 대응합니다. 예를 들어, MQZAO_BROWSE는 키워드 `+browse`에 대응하고 MQZAO_SET_ALL_CONTEXT는 키워드 `+setall` 등에 대응합니다. 이런 상수는 제품에서 제공하는 헤더 파일 cmqzc.h에 정의됩니다.

## ALW *MQI* 호출에 대한 권한 부여

**MQCONN**, **MQOPEN**, **MQPUT1**, **MQCLOSE**에서 권한 검사가 필요할 수도 있습니다. 이 주제의 표에는 각 호출에 필요한 권한이 요약되어 있습니다.

실행 중인(또는 권한으로 간주 가능한) 사용자 ID에 관련 권한이 부여된 경우에만 애플리케이션은 특정 MQI 호출 및 옵션을 발행하도록 허용됩니다.

MQI 호출에 권한 검사가 필요할 수 있습니다. **MQCONN**, **MQOPEN**, **MQPUT1**, **MQCLOSE**.

**MQOPEN** 및 **MQPUT1**의 경우, 권한 검사는 이름이 해결된 후의 이름에서가 아니라 열려 있는 오브젝트 이름에 대해 수행됩니다. 예를 들면, 애플리케이션에는 알리어스가 해석되는 기본 큐를 열 수 있는 권한은 부여되지 않고 알리어스 큐를 열 수 있는 권한이 부여될 수 있습니다. 큐 관리자 알리어스 정의가 직접 열리는 경우(즉, 해당 이름이 오브젝트 디스크립터의 *ObjectName* 필드에 표시)를 제외하고 큐 관리자 알리어스가 아닌 이름을 해결하는 프로세스 중에 발생하는 첫 번째 정의에서 검사가 수행되는 것이 규칙입니다. 열려는 오브젝트에 대한 권한은 항상 필요합니다. 큐 관리자 오브젝트의 권한 부여를 통해 확보되는 추가 큐 독립 권한이 필요한 경우도 있습니다.

123 페이지의 표 10, 123 페이지의 표 11, 124 페이지의 표 12, 124 페이지의 표 13에는 각 호출에 필요한 권한이 요약되어 있습니다. 표에서 적용할 수 없음은 권한 검사가 이 조작과 관련이 없음을 의미하며, 검사 안함은 권한 검사를 수행하지 않음을 의미합니다.

**참고:** 이 표에서는 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 또는 인증 정보 오브젝트에 대해 언급하지 않습니다. 이는 동일한 권한이 다른 오브젝트에도 적용되는 MQOO_INQUIRE를 제외하고 이러한 오브젝트에 적용되는 권한이 없기 때문입니다.

특수 권한 MQZAO_ALL_MQI에는 관리 권한으로 분류되는 MQZAO_DELETE 및 MQZAO_DISPLAY를 제외하고 오브젝트 유형과 관련된 표의 모든 권한이 포함됩니다.

임의 메시지 컨텍스트 옵션을 수정하려면 호출을 발행할 수 있는 해당 권한이 있어야 합니다. 예를 들어, MQOO_SET_IDENTITY_CONTEXT 또는 MQPMO_SET_IDENTITY_CONTEXT를 사용하려면 `+setid` 권한이 있어야 합니다.

표 *10. MQCONN* 호출에 필요한 보안 권한

| **다음에 대한 권한 필요** | **큐 오브젝트(124 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
|---|---|---|---|
| **MQCONN** | 적용할 수 없음 | 적용할 수 없음 | MQZAO_CONNECT |

표 *11. MQOPEN* 호출에 대해 필요한 보안 권한

| **다음에 대한 권한 필요** | **큐 오브젝트(124 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
|---|---|---|---|
| MQOO_INQUIRE | MQZAO_INQUIRE | MQZAO_INQUIRE | MQZAO_INQUIRE |
| MQOO_BROWSE | MQZAO_BROWSE | 적용할 수 없음 | 검사 안함 |
| MQOO_INPUT_* | MQZAO_INPUT | 적용할 수 없음 | 검사 안함 |
| MQOO_SAVE_ALL_CONTEXT **(124 페이지의 『2』)** | MQZAO_INPUT | 적용할 수 없음 | 적용할 수 없음 |
| MQOO_OUTPUT(정상 큐) **(124 페이지의 『3』)** | MQZAO_OUTPUT | 적용할 수 없음 | 적용할 수 없음 |
| MQOO_PASS_IDENTITY_CONTEXT **(124 페이지의 『4』)** | MQZAO_PASS_IDENTITY_CONTEXT | 적용할 수 없음 | 검사 안함 |
| MQOO_PASS_ALL_CONTEXT **(124 페이지의 『4』, 125 페이지의 『5』)** | MQZAO_PASS_ALL_CONTEXT | 적용할 수 없음 | 검사 안함 |
| MQOO_SET_IDENTITY_CONTEXT **(124 페이지의 『4』, 125 페이지의 『5』)** | MQZAO_SET_IDENTITY_CONTEXT | 적용할 수 없음 | MQZAO_SET_IDENTITY_CONTEXT **(125 페이지의 『6』)** |
| MQOO_SET_ALL_CONTEXT **(124 페이지의 『4』, 125 페이지의 『7』)** | MQZAO_SET_ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT **(125 페이지의 『6』)** |
| MQOO_OUTPUT(전송 큐) **(125 페이지의 『8』)** | MQZAO_SET_ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT **(125 페이지의 『6』)** |
| MQOO_SET | MQZAO_SET | 적용할 수 없음 | 검사 안함 |

| 표 *11. MQOPEN* 호출에 대해 필요한 보안 권한 *(계속)* | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(124 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQOO_ALTERNATE_ USER_AUTHORITY | **( 125 페이지의 『9』 )** | **( 125 페이지의 『9』 )** | MQZAO_ALTERNATE_ USER_AUTHORITY **(125 페이지의 『9』, 125 페이지의 『10』)** |

| 표 *12. MQPUT1* 호출에 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(124 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQPMO_PASS_ IDENTITY_CONTEXT | MQZAO_PASS_ IDENTITY_CONTEXT **(125 페이지의 『11』)** | 적용할 수 없음 | 검사 안함 |
| MQPMO_PASS_ALL _CONTEXT | MQZAO_PASS_ ALL_CONTEXT **(125 페이지의 『11』)** | 적용할 수 없음 | 검사 안함 |
| MQPMO_SET_ IDENTITY_CONTEXT | MQZAO_SET_ IDENTITY_CONTEXT **(125 페이지의 『11』)** | 적용할 수 없음 | MQZAO_SET_ IDENTITY_CONTEXT **(125 페이지의 『6』)** |
| MQPMO_SET_ ALL_CONTEXT | MQZAO_SET_ ALL_CONTEXT **(125 페이지의 『11』)** | 적용할 수 없음 | MQZAO_SET_ ALL_CONTEXT **(125 페이지의 『6』)** |
| (전송 큐) **(125 페이지의 『8』)** | MQZAO_SET_ ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ ALL_CONTEXT **(125 페이지의 『6』)** |
| MQPMO_ALTERNATE_ USER_AUTHORITY | **( 125 페이지의 『12』 )** | 적용할 수 없음 | MQZAO_ALTERNATE_ USER_AUTHORITY **(125 페이지의 『10』)** |

| 표 *13. MQCLOSE* 호출에 대해 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(124 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQCO_DELETE | MQZAO_DELETE **(125 페이지의 『13』)** | 적용할 수 없음 | 적용할 수 없음 |
| MQCO_DELETE _PURGE | MQZAO_DELETE **(125 페이지의 『13』)** | 적용할 수 없음 | 적용할 수 없음 |

**표에 대한 참고:**

1. 모델 큐를 여는 경우:
   - 열기 액세스 유형의 경우 모델 큐를 열 수 있는 권한 외에 MQZAO_DISPLAY 권한이 모델 큐에 필요합니다.
   - 동적 큐 작성에 MQZAO_CREATE 권한은 필요하지 않습니다.
   - 모델 큐를 여는 데 사용되는 사용자 ID는 모든 큐 특정 권한을 작성된 동적 큐에 대해 자동 부여합니다 (MQZAO_ALL에 해당).
2. MQOO_INPUT_*도 지정해야 합니다. 이것은 로컬, 모델 또는 알리어스 큐에도 유효합니다.
3. 이 검사는 전송 큐를 제외한 모든 출력의 경우에 대해서도 수행됩니다(125 페이지의 『8』 참고 참조).
4. MQOO_OUTPUT도 지정해야 합니다.

5. 이 옵션으로 MQOO_PASS_IDENTITY_CONTEXT도 포함해야 합니다.
6. 이 권한은 큐 관리자 오브젝트와 특정 큐 모두에 필요합니다.
7. 이 옵션으로 MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT, MQOO_SET_IDENTITY_CONTEXT도 포함해야 합니다.
8. 이 검사는 MQUS_TRANSMISSION의 *Usage* 큐 속성을 갖고 있고 출력을 위해 직접 열려 있는 로컬 또는 모델 큐에 대해 수행됩니다. 리모트 큐가 열려 있는 경우(리모트 큐 관리자 이름과 리모트 큐 이름을 지정하거나 리모트 큐의 로컬 정의 이름을 지정하여) 적용되지 않습니다.
9. MQOO_INQUIRE(모든 오브젝트 유형에 대해) 또는 MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT 또는 MQOO_SET(큐에 대해) 중 하나 이상도 지정해야 합니다. 검사는 특정 이름의 오브젝트 권한에 대해 제공되는 대체 사용자 ID 및 MQZAO_ALTERNATE_USER_IDENTIFIER 검사에 대한 현재 애플리케이션 권한을 사용하여 지정된 다른 옵션에 대해 수행됩니다.
10. 이 권한은 모든 *AlternateUserId* 지정을 허용합니다.
11. 큐에 MQUS_TRANSMISSION의 *Usage* 큐 속성이 없는 경우, MQZAO_OUTPUT 검사가 수행됩니다.
12. 검사는 특정 이름의 큐 권한에 대해 제공되는 대체 사용자 ID 및 MQZAO_ALTERNATE_USER_IDENTIFIER 검사에 대한 현재 애플리케이션 권한을 사용하여 지정된 다른 옵션에 대해 수행됩니다.
13. 검사는 다음 두 가지 사항에 모두 해당하는 경우에만 수행됩니다.
    - 영구적 동적 큐가 닫히고 삭제 중인 경우.
    - 사용되고 있는 오브젝트 핸들을 리턴한 MQOPEN 호출에 의해 큐가 작성되지 않은 경우.

    그렇지 않으며 검사가 수행되지 않습니다.

### ALW 이스케이프 *PCF*에서 *MQSC* 명령의 권한 부여

이 정보는 Escape PCF에 포함된 각 MQSC 명령에 필요한 권한을 요약합니다.

적용할 수 없음은 이 조작이 이 오브젝트 유형에 연관되지 않았음을 나타냅니다.

명령을 제출하는 프로그램이 실행 중인 사용자 ID는 다음 권한도 있어야 합니다.

- 큐 관리자에 대한 MQZAO_CONNECT 권한
- PCF 명령을 수행하기 위한 큐 관리자의 MQZAO_DISPLAY 권한
- 이스케이프 PCF 명령 텍스트 내에서 MQSC 명령을 발행하는 권한

**ALTER *object***

| **오브젝트** | **필수 권한** |
|---|---|
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | MQZAO_CHANGE |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |
| 통신 정보 | MQZAO_CHANGE |

**CLEAR *object***

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CLEAR |
| 토픽 | MQZAO_CLEAR |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | 적용할 수 없음 |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |
| 통신 정보 | 적용할 수 없음 |

**DEFINE *object* NOREPLACE (130 페이지의 『1』)**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CREATE (130 페이지의 『2』) |
| 토픽 | MQZAO_CREATE (130 페이지의 『2』) |
| 프로세스 | MQZAO_CREATE (130 페이지의 『2』) |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CREATE (130 페이지의 『2』) |
| 인증 정보 | MQZAO_CREATE (130 페이지의 『2』) |
| 채널 | MQZAO_CREATE (130 페이지의 『2』) |
| 클라이언트 연결 채널 | MQZAO_CREATE (130 페이지의 『2』) |
| 리스너 | MQZAO_CREATE (130 페이지의 『2』) |
| 서비스 | MQZAO_CREATE (130 페이지의 『2』) |
| 통신 정보 | MQZAO_CREATE (130 페이지의 『2』) |

**DEFINE *object* REPLACE (130 페이지의 『1』, 130 페이지의 『3』)**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |
| 통신 정보 | MQZAO_CHANGE |

**DELETE *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DELETE |
| 토픽 | MQZAO_DELETE |
| 프로세스 | MQZAO_DELETE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_DELETE |
| 인증 정보 | MQZAO_DELETE |
| 채널 | MQZAO_DELETE |
| 클라이언트 연결 채널 | MQZAO_DELETE |
| 리스너 | MQZAO_DELETE |
| 서비스 | MQZAO_DELETE |
| 통신 정보 | MQZAO_DELETE |

**DISPLAY *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DISPLAY |
| 토픽 | MQZAO_DISPLAY |
| 프로세스 | MQZAO_DISPLAY |
| 큐 관리자 | MQZAO_DISPLAY |
| 이름 목록 | MQZAO_DISPLAY |
| 인증 정보 | MQZAO_DISPLAY |
| 채널 | MQZAO_DISPLAY |
| 클라이언트 연결 채널 | MQZAO_DISPLAY |
| 리스너 | MQZAO_DISPLAY |
| 서비스 | MQZAO_DISPLAY |
| 통신 정보 | MQZAO_DISPLAY |

**START *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | MQZAO_CONTROL |
| 서비스 | MQZAO_CONTROL |
| 통신 정보 | 적용할 수 없음 |

**STOP *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | MQZAO_CONTROL |
| 서비스 | MQZAO_CONTROL |
| 통신 정보 | 적용할 수 없음 |

**채널 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| PING CHANNEL | 채널 | MQZAO_CONTROL |
| RESET CHANNEL | 채널 | MQZAO_CONTROL_EXTENDED |
| RESOLVE CHANNEL | 채널 | MQZAO_CONTROL_EXTENDED |

**구독 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| ALTER SUB | 토픽 | MQZAO_CONTROL |
| DEFINE SUB | 토픽 | MQZAO_CONTROL |
| DELETE SUB | 토픽 | MQZAO_CONTROL |
| DISPLAY SUB | 토픽 | MQZAO_DISPLAY |

**보안 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| SET AUTHREC | 큐 관리자 | MQZAO_CHANGE |
| DELETE AUTHREC | 큐 관리자 | MQZAO_CHANGE |
| DISPLAY AUTHREC | 큐 관리자 | MQZAO_DISPLAY |
| DISPLAY AUTHSERV | 큐 관리자 | MQZAO_DISPLAY |
| DISPLAY ENTAUTH | 큐 관리자 | MQZAO_DISPLAY |
| SET CHLAUTH | 큐 관리자 | MQZAO_CHANGE |
| DISPLAY CHLAUTH | 큐 관리자 | MQZAO_DISPLAY |
| REFRESH SECURITY | 큐 관리자 | MQZAO_CHANGE |

**상태 표시**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| DISPLAY CHSTATUS | 큐 관리자 | MQZAO_DISPLAY<br>채널 유형이 CLUSSDR이면 전송 큐에 `+inq` 권한(또는 MQZAO_INQUIRE)이 필요합니다. |
| DISPLAY LSSTATUS | 큐 관리자 | MQZAO_DISPLAY |
| DISPLAY PUBSUB | 큐 관리자 | MQZAO_DISPLAY |
| DISPLAY SBSTATUS | 큐 관리자 | MQZAO_DISPLAY |
| DISPLAY SVSTATUS | 큐 관리자 | MQZAO_DISPLAY |
| DISPLAY TPSTATUS | 큐 관리자 | MQZAO_DISPLAY |

**클러스터 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| DISPLAY CLUSQMGR | 큐 관리자 | MQZAO_DISPLAY |
| REFRESH CLUSTER | 'mqm' 그룹 멤버십 필수 | |
| RESET CLUSTER | 'mqm' 그룹 멤버십 필수 | |
| SUSPEND QMGR | 'mqm' 그룹 멤버십 필수 | |
| RESUME QMGR | 'mqm' 그룹 멤버십 필수 | |

**기타 관리 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| PING QMGR | 큐 관리자 | MQZAO_DISPLAY |
| REFRESH QMGR | 큐 관리자 | MQZAO_CHANGE |
| RESET QMGR | 큐 관리자 | MQZAO_CHANGE |
| DISPLAY CONN | 큐 관리자 | MQZAO_DISPLAY |
| STOP CONN | 큐 관리자 | MQZAO_CHANGE |

**참고:**

1. DEFINE 명령의 경우, LIKE 오브젝트(지정된 경우)에 대한 MQZAO_DISPLAY 권한도 필요하며, LIKE가 생략된 경우 적절한 SYSTEM.DEFAULT.xxx 오브젝트에 대한 권한이 필요합니다.
2. MQZAO_CREATE 권한은 특정 오브젝트나 오브젝트 유형에 특정되지 않습니다. 작성 권한은 `setmqaut` 명령에 오브젝트 유형 QMGR을 지정하여 지정된 큐 관리자에 대한 모든 오브젝트에 부여됩니다.
3. 이는 바꿀 오브젝트가 이미 있는 경우에도 적용됩니다. 존재하지 않는 경우, 검사는 DEFINE *object* NOREPLACE로 수행됩니다.

**관련 정보**

클러스터링: REFRESH CLUSTER 사용 우수 사례

## ALW *PCF* 명령의 권한 부여

이 절에서는 각 PCF 명령에 필요한 권한을 요약합니다.

검사 안함은 권한 검사가 수행되지 않는 것을 나타내며 적용할 수 없음은 이 조작이 이 오브젝트 유형에 연관되지 않았음을 나타냅니다.

명령을 제출하는 프로그램이 실행 중인 사용자 ID는 다음 권한도 있어야 합니다.

- 큐 관리자에 대한 MQZAO_CONNECT 권한
- PCF 명령을 수행하기 위한 큐 관리자의 MQZAO_DISPLAY 권한

특수 권한 MQZAO_ALL_ADMIN에는 특정 오브젝트 또는 오브젝트 유형에 특정하지 않는 MQZAO_CREATE를 제외한 오브젝트 유형에 관련되는 다음 목록의 모든 권한을 포함합니다.

***object* 변경**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CHANGE |
| 주제 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | MQZAO_CHANGE |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |
| 통신 정보 | MQZAO_CHANGE |

***object* 지우기**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CLEAR |
| 주제 | MQZAO_CLEAR |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 채널 | 적용할 수 없음 |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |
| 통신 정보 | 적용할 수 없음 |

***object* 복사(대체 제외)(1)**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CREATE **(2)** |
| 주제 | MQZAO_CREATE **(2)** |
| 프로세스 | MQZAO_CREATE **(2)** |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CREATE **(2)** |
| 인증 정보 | MQZAO_CREATE **(2)** |
| 채널 | MQZAO_CREATE **(2)** |
| 클라이언트 연결 채널 | MQZAO_CREATE **(2)** |
| 리스너 | MQZAO_CREATE **(2)** |
| 서비스 | MQZAO_CREATE **(2)** |
| 통신 정보 | MQZAO_CREATE **(136 페이지의 『2』)** |

**Copy *object*(바꾸기 포함)( 1, 4 )**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CHANGE |
| 주제 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |
| 통신 정보 | MQZAO_CHANGE |

***object* 작성(대체 제외)(3)**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CREATE **(2)** |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 주제 | MQZAO_CREATE **(2)** |
| 프로세스 | MQZAO_CREATE **(2)** |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CREATE **(2)** |
| 인증 정보 | MQZAO_CREATE **(2)** |
| 채널 | MQZAO_CREATE **(2)** |
| 클라이언트 연결 채널 | MQZAO_CREATE **(2)** |
| 리스너 | MQZAO_CREATE **(2)** |
| 서비스 | MQZAO_CREATE **(2)** |
| 통신 정보 | MQZAO_CREATE **(2)** |

**Create *object*(바꾸기 포함)( 3, 4 )**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CHANGE |
| 주제 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |
| 통신 정보 | MQZAO_CHANGE |

***object* 삭제**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DELETE |
| 주제 | MQZAO_DELETE |
| 프로세스 | MQZAO_DELETE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_DELETE |
| 인증 정보 | MQZAO_DELETE |
| 채널 | MQZAO_DELETE |
| 클라이언트 연결 채널 | MQZAO_DELETE |
| 리스너 | MQZAO_DELETE |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 서비스 | MQZAO_DELETE |
| 통신 정보 | MQZAO_DELETE |

***object* 조회**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DISPLAY |
| 주제 | MQZAO_DISPLAY |
| 프로세스 | MQZAO_DISPLAY |
| 큐 관리자 | MQZAO_DISPLAY |
| 이름 목록 | MQZAO_DISPLAY |
| 인증 정보 | MQZAO_DISPLAY |
| 채널 | MQZAO_DISPLAY |
| 클라이언트 연결 채널 | MQZAO_DISPLAY |
| 리스너 | MQZAO_DISPLAY |
| 서비스 | MQZAO_DISPLAY |
| 통신 정보 | MQZAO_DISPLAY |

***object* 이름 조회**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 검사 안함 |
| 토픽 | 검사 안함 |
| 프로세스 | 검사 안함 |
| 큐 관리자 | 검사 안함 |
| 이름 목록 | 검사 안함 |
| 인증 정보 | 검사 안함 |
| 채널 | 검사 안함 |
| 클라이언트 연결 채널 | 검사 안함 |
| 리스너 | 검사 안함 |
| 서비스 | 검사 안함 |
| 통신 정보 | 검사 안함 |

***object* 시작**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | MQZAO_CONTROL |
| 서비스 | MQZAO_CONTROL |
| 통신 정보 | 적용할 수 없음 |

***object* 중지**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | MQZAO_CONTROL |
| 서비스 | MQZAO_CONTROL |
| 통신 정보 | 적용할 수 없음 |

**채널 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| 채널 ping | 채널 | MQZAO_CONTROL |
| 채널 재설정 | 채널 | MQZAO_CONTROL_EXTENDED |
| 채널 분석 | 채널 | MQZAO_CONTROL_EXTENDED |

**구독 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| 구독 변경 | 토픽 | MQZAO_CONTROL |
| 구독 작성 | 토픽 | MQZAO_CONTROL |
| 구독 삭제 | 토픽 | MQZAO_CONTROL |
| 구독 조회 | 토픽 | MQZAO_DISPLAY |

**보안 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| 권한 레코드 설정 | 큐 관리자 | MQZAO_CHANGE |
| 권한 레코드 삭제 | 큐 관리자 | MQZAO_CHANGE |
| 권한 레코드 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 권한 서비스 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 엔티티 권한 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 채널 인증 레코드 설정 | 큐 관리자 | MQZAO_CHANGE |
| 채널 인증 레코드 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 보안 새로 고치기 | 큐 관리자 | MQZAO_CHANGE |

**상태 표시**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| 채널 상태 조회 | 큐 관리자 | MQZAO_DISPLAY<br>채널 유형이 CLUSSDR이면 전송 큐에 `+inq` 권한(또는 MQZAO_INQUIRE)이 필요합니다. |
| 채널 리스너 상태 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 발행/구독 상태 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 구독 상태 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 서비스 상태 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 토픽 상태 조회 | 큐 관리자 | MQZAO_DISPLAY |

**클러스터 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| 클러스터 큐 관리자 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 클러스터 새로 고치기 | 'mqm' 그룹 멤버십 필수 | 'mqm' 그룹 멤버십 필수 |
| 클러스터 재설정 | 'mqm' 그룹 멤버십 필수 | 'mqm' 그룹 멤버십 필수 |
| 큐 관리자 클러스터 일시중단 | 'mqm' 그룹 멤버십 필수 | 'mqm' 그룹 멤버십 필수 |
| 큐 관리자 클러스터 재개 | 'mqm' 그룹 멤버십 필수 | 'mqm' 그룹 멤버십 필수 |

**기타 관리 명령**

| 명령 | 오브젝트 | 필수 권한 |
| --- | --- | --- |
| 큐 관리자 ping | 큐 관리자 | MQZAO_DISPLAY |
| 큐 관리자 새로 고치기 | 큐 관리자 | MQZAO_CHANGE |
| 큐 관리자 재설정 | 큐 관리자 | MQZAO_CHANGE |
| 큐 통계 재설정 | 큐 | MQZAO_DISPLAY and MQZAO_CHANGE |

| 명령 | 오브젝트 | 필수 권한 |
|---|---|---|
| 연결 조회 | 큐 관리자 | MQZAO_DISPLAY |
| 연결 중지 | 큐 관리자 | MQZAO_CHANGE |

**참고:**

1. 복사 명령의 경우, From 오브젝트에 대한 MQZAO_DISPLAY 권한도 필요합니다.
2. MQZAO_CREATE 권한은 특정 오브젝트나 오브젝트 유형에 특정되지 않습니다. 작성 권한은 `setmqaut` 명령에 오브젝트 유형 QMGR을 지정하여 지정된 큐 관리자에 대한 모든 오브젝트에 부여됩니다.
3. Create 명령의 경우 적절한 SYSTEM.DEFAULT.* 오브젝트.
4. 이는 바꿀 오브젝트가 이미 있는 경우에도 적용됩니다. 없는 경우, 검사는 복사 또는 작성(바꾸기 없음)에 대해 수행됩니다.

## AIX AIX에서 그룹 작성 및 관리

AIX에서 NIS 또는 NIS+를 사용하지 않는 경우 SMITTY를 사용하여 그룹에 대해 작업하십시오.

### 이 태스크 정보

AIX에서 SMITTY를 사용하여 그룹 작성, 그룹에 사용자 추가, 그룹에 있는 사용자 목록 표시, 그룹에서 사용자 제거를 수행할 수 있습니다.

### 프로시저

1. SMITTY에서 **보안 및 사용자**를 선택하고 Enter를 누르십시오.
2. **그룹**을 선택하고 Enter를 누르십시오.
3. 그룹을 작성하려면 다음 단계를 완료하십시오.
   a) **그룹 추가**를 선택하고 Enter를 누르십시오.
   b) 그룹 이름 및 그룹에 추가하려는 모든 사용자 이름을 쉼표로 구분하여 입력하십시오.
   c) Enter를 눌러 그룹을 작성하십시오.
4. 그룹에 사용자를 추가하려면 다음 단계를 완료하십시오.
   a) **그룹 특성 변경 / 표시**를 선택하고 Enter를 누르십시오.
   b) 그룹 구성원 목록을 표시하려는 그룹 이름을 입력하십시오.
   c) 그룹에 추가하려는 사용자 이름을 쉼표로 구분하여 추가하십시오.
   d) 이름을 그룹에 추가하려면 Enter를 누르십시오.
5. 그룹에 있는 사용자를 표시하려면 다음 단계를 완료하십시오.
   a) **그룹 특성 변경 / 표시**를 선택하고 Enter를 누르십시오.
   b) 그룹 구성원 목록을 표시하려는 그룹 이름을 입력하십시오.
6. 그룹에서 사용자를 제거하려면 다음 단계를 완료하십시오.
   a) **그룹 특성 변경 / 표시**를 선택하고 Enter를 누르십시오.
   b) 그룹 구성원 목록을 표시하려는 그룹 이름을 입력하십시오.
   c) 그룹에서 제거하려는 사용자의 이름을 삭제하십시오.
   d) 그룹에서 이름을 제거하려면 Enter를 누르십시오.

## Linux Linux에서 그룹 작성 및 관리

Linux에서 NIS 또는 NIS+를 사용하지 않는 경우 `/etc/group` 파일을 사용하여 그룹에 대해 작업하십시오.

### 이 태스크 정보

Linux에서 그룹 정보는 /etc/group 파일에 보유됩니다. 명령을 사용하여 그룹 작성, 그룹에 사용자 추가, 그룹에 있는 사용자 목록 표시, 그룹에서 사용자 제거를 수행할 수 있습니다.

### 프로시저

1. 새 그룹을 작성하려면 **groupadd** 명령을 사용하십시오.
   다음 명령을 입력하십시오.

   ```
   groupadd -g group-ID group-name
   ```

   여기서 *group-ID*는 그룹의 숫자 ID이고 *group-name*은 그룹의 이름입니다.
2. 보충 그룹에 멤버를 추가하려면, **usermod** 명령을 사용하여 사용자가 현재 멤버인 보충 그룹과 사용자가 멤버가 되는 보충 그룹을 나열하십시오.
   예를 들어, 사용자가 이미 groupa 그룹의 멤버이고 groupb의 멤버가 되는 경우 다음 명령을 사용하십시오.

   ```
   usermod -G groupa,groupb user-name
   ```

   여기서 *user-name*은 사용자 이름입니다.
3. 그룹의 멤버를 표시하려면 **getent** 명령을 사용하십시오.
   다음 명령을 입력하십시오.

   ```
   getent group group-name
   ```

   여기서, *group-name*은 그룹 이름입니다.
4. 보충 그룹에서 멤버를 제거하려면 **usermod** 명령을 사용하여 사용자를 멤버로 남겨두려는 보충 그룹을 나열하십시오.
   예를 들어, 사용자의 기본 그룹이 users이고 사용자가 mqm, groupa 및 groupb의 멤버이기도 한 경우 mqm 그룹에서 사용자를 제거하려면 다음 명령을 사용하십시오.

   ```
   usermod -G groupa,groupb user-name
   ```

   여기서 *user-name*은 사용자 이름입니다.

## Windows Windows에서 그룹 작성 및 관리

Windows에서 컴퓨터 관리 기능을 사용하여 워크스테이션 또는 멤버 서버 시스템에서 그룹을 관리합니다.

### 이 태스크 정보

도메인 제어기에 대해 사용자 및 그룹은 Active Directory를 통해 관리됩니다. Active Directory 사용에 대한 자세한 정보는 해당하는 운영 체제 지시사항을 참조하십시오.

프린시펄의 그룹 멤버십에 대한 변경은 큐 관리자가 재시작되거나 MQSC 명령 **REFRESH SECURITY**(또는 PCF와 동일)를 실행해야 인식됩니다.

Windows 컴퓨터 관리 패널을 사용하여 사용자 및 그룹에 대해 작업하십시오. 현재 로그온한 사용자에 대한 모든 변경사항은 사용자가 다시 로그온할 때까지 적용되지 않을 수 있습니다.

## Windows *Windows*에서 그룹 작성

제어판을 사용하여 그룹을 작성합니다.

### 프로시저

1. 제어판을 여십시오.
2. **관리 도구**를 두 번 클릭하십시오.
   관리 도구 패널이 열립니다.

3. **컴퓨터 관리**를 두 번 클릭하십시오.
   컴퓨터 관리 패널이 열립니다.
4. **로컬 사용자 및 그룹**을 펼치십시오.
5. **그룹**을 마우스의 오른쪽 단추 클릭하고 **새 그룹...**을 선택하십시오.
   새 그룹 패널이 표시됩니다.
6. 그룹 이름 필드에 적절한 이름을 입력하고 **작성**을 클릭하십시오.
7. **닫기**를 클릭하십시오.

### Windows *Windows*에서 그룹에 사용자 추가

제어판을 사용하여 그룹에 사용자를 추가합니다.

#### 프로시저

1. 제어판을 여십시오.
2. **관리 도구**를 두 번 클릭하십시오.
   관리 도구 패널이 열립니다.
3. **컴퓨터 관리**를 두 번 클릭하십시오.
   컴퓨터 관리 패널이 열립니다.
4. 컴퓨터 관리 패널에서 **로컬 사용자 및 그룹**을 펼치십시오.
5. **사용자**를 선택하십시오.
6. 그룹에 추가하려는 사용자를 두 번 클릭하십시오.
   사용자 특성 패널이 표시됩니다.
7. **멤버** 탭을 선택하십시오.
8. 사용자를 추가할 그룹을 선택하십시오. 필요한 그룹이 표시되지 않는 경우에는 다음을 수행하십시오.
   a) **추가...**를 클릭하십시오.
      그룹 선택 패널이 표시됩니다.
   b) **위치...**를 클릭하십시오.
      위치 패널이 표시됩니다.
   c) 목록에서 사용자를 추가하려는 그룹 위치를 선택하고 **확인**을 클릭하십시오.
   d) 제공된 필드에 그룹 이름을 입력하십시오.

      또는 **고급 ...** 을 클릭하십시오. 그런 다음 **지금 찾기** 를 클릭하여 현재 선택된 위치에서 사용 가능한 그룹을 나열하십시오. 여기에서 사용자를 추가하려는 그룹을 선택하고 **확인**을 클릭하십시오.
   e) **확인**을 클릭하십시오.
      추가한 그룹을 표시하는 사용자 특성 패널이 표시됩니다.
   f) 그룹을 선택하십시오.
9. **확인**을 클릭하십시오.
   컴퓨터 관리 패널이 표시됩니다.

### Windows *Windows*의 그룹에 있는 구성원 표시

제어판을 사용하여 그룹의 구성원을 표시합니다.

#### 프로시저

1. 제어판을 여십시오.
2. **관리 도구**를 두 번 클릭하십시오.
   관리 도구 패널이 열립니다.
3. **컴퓨터 관리**를 두 번 클릭하십시오.
   컴퓨터 관리 패널이 열립니다.

4. 컴퓨터 관리 패널에서 **로컬 사용자 및 그룹**을 펼치십시오.
5. **그룹**을 선택하십시오.
6. 그룹을 두 번 클릭하십시오. 그룹 특성 패널이 표시됩니다.
   그룹 특성 패널이 표시됩니다.

### 결과
그룹 구성원이 표시됩니다.

## Windows *Windows*의 그룹에서 사용자 제거
제어판을 사용하여 그룹에서 사용자를 제거합니다.

### 프로시저
1. 제어판을 여십시오.
2. **관리 도구**를 두 번 클릭하십시오.
   관리 도구 패널이 열립니다.
3. **컴퓨터 관리**를 두 번 클릭하십시오.
   컴퓨터 관리 패널이 열립니다.
4. 컴퓨터 관리 패널에서 **로컬 사용자 및 그룹**을 펼치십시오.
5. **사용자**를 선택하십시오.
6. 그룹에 추가하려는 사용자를 두 번 클릭하십시오.
   사용자 특성 패널이 표시됩니다.
7. **멤버** 탭을 선택하십시오.
8. 사용자를 제거하려는 그룹을 선택하고 **제거**를 클릭하십시오.
9. **확인**을 클릭하십시오.
   컴퓨터 관리 패널이 표시됩니다.

### 결과
이제 그룹에서 사용자를 제거했습니다.

## Windows Windows의 보안에 대한 특별한 고려사항
일부 보안 기능은 Windows의 다른 버전에 대해 다르게 작동합니다.

IBM MQ 보안은 사용자 권한 및 그룹 멤버십에 대한 정보의 운영 체제 API에 대한 호출에 의존합니다. 일부 기능은 Windows 시스템에서 동일하게 작동하지 않습니다. 이 주제 콜렉션에는 Windows 환경에서 IBM MQ 를 실행할 때 이러한 차이가 IBM MQ 보안에 미치는 영향에 대한 설명이 포함되어 있습니다.

## Windows *IBM MQ Windows* 서비스용 로컬 및 도메인 사용자 계정
IBM MQ가 실행 중인 경우 권한 부여된 사용자만 큐 관리자 또는 큐에 액세스할 수 있도록 해야 합니다. 따라서 이러한 액세스를 시도하는 사용자에 대한 정보를 IBM MQ에서 조회하는 데 사용할 수 있는 특수 사용자 계정이 필요합니다.


- 139 페이지의 『Prepare IBM MQ Wizard를 사용하여 특수 사용자 계정 구성』
- 140 페이지의 『IBM MQ에서 Active Directory 사용』
- 140 페이지의 『IBM MQ Windows 서비스에 필요한 사용자 권한』


## Prepare IBM MQ Wizard를 사용하여 특수 사용자 계정 구성

Prepare IBM MQ Wizard는 Windows 서비스가 이를 사용해야 하는 프로세스에 의해 공유될 수 있도록 특수 사용자 계정을 작성합니다(Prepare IBM MQ Wizard로 IBM MQ 구성 참조).

Windows 서비스는 IBM MQ 설치를 위한 클라이언트 프로세스 사이에 공유됩니다. 한 가지 서비스가 각 설치를 위해 작성됩니다. 각 서비스의 이름은 MQ_*InstallationName*이며 표시 이름은 IBM MQ(*InstallationName*) 입니다.

각 서비스가 비대화식 및 대화식 로그온 세션 사이에 공유되어야 하기 때문에, 특수 사용자 계정에서 각각을 실행해야 합니다. 모든 서비스에 대해 하나의 특수 사용자 계정을 사용하거나 다른 특수 사용자 계정을 작성할 수 있습니다. 각 특수 사용자 계정에는 서비스로서  로그온에 대한 사용자 권한이 있어야 합니다. 자세한 정보는 140 페이지의 표 14의 내용을 참조하십시오. 사용자 ID에 서비스를 실행할 권한이 없으면 서비스가 시작되지 않고 Windows 시스템 이벤트 로그에서 오류가 리턴됩니다. 일반적으로 Prepare IBM MQ Wizard를 실행하고 사용자 ID를 올바르게 설정합니다. 그러나 사용자 ID를 수동으로 구성한 경우 해결해야 할 문제가 있을 수 있습니다.

IBM MQ를 설치하고 Prepare IBM MQ Wizard를 처음 실행할 때, 서비스로  로그온을 포함하여 필수 설정 및 권한으로 MUSR_MQADMIN이라는 서비스에 대한 로컬 사용자 계정을 작성합니다.

후속 설치의 경우, Prepare IBM MQ Wizard는 MUSR_MQADMIN*x*라는 사용자 계정을 작성합니다. 여기서 *x*는 존재하지 않는 사용자 ID를 표시하는 다음에 사용 가능한 숫자입니다. MUSR_MQADMIN*x*에 대한 비밀번호는 계정이 작성될 때 무작위로 생성되고 서비스에 대한 로그온 환경을 구성하기 위해 사용됩니다. 생성된 비밀번호는 만기되지 않습니다.

이 IBM MQ 계정은 특정 기간 이후에 계정 비밀번호가 변경되도록 요청하기 위해 시스템에 설정되는 계정 정책에 의해 영향을 받지 않습니다.

비밀번호는 이 일회성 처리 외부에는 알려지지 않고 레지스트리의 보안 파트에 Windows 운영 체제에 의해 저장됩니다.

## IBM MQ에서 Active Directory 사용

사용자 계정이 Active Directory 디렉토리 서비스를 사용 중인 도메인 제어기에서 정의되는 일부 네트워크 구성에서, IBM MQ가 실행 중인 로컬 사용자 계정에는 기타 도메인 사용자 계정의 그룹 멤버십을 조회하기 위해 필요한 권한이 없을 수 있습니다. IBM MQ를 설치할 때 Prepare IBM MQ Wizard는 네트워크 구성에 대한 질문을 하고 테스트를 실행하는 경우인지 여부를 식별합니다.

IBM MQ가 실행 중인 로컬 사용자 계정에 필수 권한이 없는 경우, Prepare IBM MQ Wizard는 특별 사용자 권한을 가지는 도메인 사용자 계정의 계정 세부사항에 대한 사용자를 프롬프트합니다. Windows 도메인 계정을 작성하고 설정하는 자세한 방법은 IBM MQ용 Windows 도메인 계정 작성 및 설정을 참조하십시오. 도메인 사용자 계정이 요구하는 사용자 권한에 대해서는 140 페이지의 표 14의 내용을 참조하십시오.

도메인 사용자 계정에 대한 올바른 계정 세부사항을 Prepare IBM MQ Wizard에 입력하면, 마법사가 새 계정에서 실행하기 위한 IBM MQ Windows 서비스를 구성합니다. 계정 세부사항이 레지스트리의 보안 부분에 보유되고 사용자는 읽을 수 없습니다.

서비스가 실행 중일 때, IBM MQ Windows 서비스가 실행되고 서비스가 실행 중인 한 실행으로 남습니다. Windows 서비스가 실행된 후 서버에 로그온하는 IBM MQ 관리자는 IBM MQ Explorer 를 사용하여 서버에서 큐 관리자를 관리할 수 있습니다. 이는 IBM MQ Explorer를 기존Windows 서비스 프로세스에 연결합니다. 이러한 두 개의 조치가 작동하려면 다른 레벨의 권한이 필요합니다.

- 시작 프로세스에는 실행 권한이 필요합니다.
- IBM MQ 관리자에는 액세스 권한이 필요합니다.

## IBM MQ Windows 서비스에 필요한 사용자 권한

다음 표에는 IBM MQ 설치를 위한 Windows 서비스가 실행되는 로컬 및 도메인 사용자 계정에 필요한 사용자 권한이 나열되어 있습니다.

표 *14. IBM MQ Windows* 서비스에 필요한 사용자 권한

| **권한** | **설명** |
|---|---|
| 배치 작업으로 로그온 | 이 사용자 계정에서 실행하기 위한 IBM MQ Windows 서비스를 사용합니다. |
| 서비스로 로그온 | 사용자가 구성된 계정을 사용하여 로그온하기 위해 IBM MQ Windows 서비스를 설정할 수 있게 합니다. |

| 표 *14. IBM MQ Windows* 서비스에 필요한 사용자 권한 *(계속)* | |
|---|---|
| **권한** | **설명** |
| 시스템 종료 | IBM MQ Windows 서비스가 서비스의 복구가 실패할 때 수행하도록 하기 구성되면 서버를 다시 시작할 수 있게 허용합니다. |
| 할당량 증가 | 운영 체제 `CreateProcessAsUser` 호출에 필요합니다. |
| 운영 체제의 일부로 작동 | 운영 체제 `LogonUser` 호출에 필요합니다. |
| 통과 검사 우회 | 운영 체제 `LogonUser` 호출에 필요합니다. |
| 프로세스 레벨 토큰 대체 | 운영 체제 `LogonUser` 호출에 필요합니다. |

**참고:** ASP 및 IIS 애플리케이션을 실행하는 환경에서는 디버그 프로그램 권리가 필요할 수 있습니다.

사용자의 도메인 사용자 계정은 이러한 Windows 사용자 권한을 로컬 보안 정책 애플리케이션에 나열된 대로 유효한 사용자 권한으로 설정되게 해야 합니다. 그렇지 않으면, 서버에서 로컬 보안 정책 애플리케이션을 로컬로 사용하거나 도메인 보안 애플리케이션 도메인을 사용하여 이를 설정하십시오.

Windows *Windows* 서버 보안 권한

IBM MQ 설치는 로컬 사용자 또는 도메인 사용자가 설치를 수행하는지 여부에 따라 Windows Server에서 다르게 작동합니다.

로컬 사용자가 IBM MQ를 설치하는 경우, Prepare IBM MQ Wizard가 IBM MQ Windows 서비스에 대해 작성된 로컬 사용자가 설치 사용자의 그룹 멤버십 정보를 검색할 수 있음을 감지합니다. Prepare IBM MQ Wizard는 Windows 2000 이상에서 실행 중인 도메인 제어기에 정의된 다른 사용자 계정이 있는지를 판별하기 위해 네트워크 구성에 대한 질문을 합니다. 그런 경우, IBM MQ Windows 서비스는 특정 설정 및 권한의 도메인 사용자 계정으로 실행되어야 합니다. Prepare IBM MQ Wizard에서는 Prepare IBM MQ Wizard로 IBM MQ 구성에서 설명한 대로, 이 사용자의 계정 세부사항을 사용자에게 표시합니다.

도메인 사용자가 IBM MQ를 설치하는 경우, Prepare IBM MQ Wizard가 IBM MQ Windows 서비스에 대해 작성된 로컬 사용자가 설치 사용자의 그룹 멤버십 정보를 검색할 수 없음을 감지합니다. 이런 경우, Prepare IBM MQ Wizard는 항상 사용하려는 IBM MQ Windows 서비스에 대한 도메인 사용자 계정의 계정 세부사항을 사용자에게 프롬프트합니다.

IBM MQ Windows 서비스가 도메인 사용자 계정을 사용해야 하는 경우, IBM MQ는 Prepare IBM MQ Wizard를 사용하여 구성될 때까지 제대로 작동할 수 없습니다. Prepare IBM MQ Wizard 준비 마법사는 Windows 서비스가 적합한 계정으로 구성될 때까지 사용자가 다른 태스크로 계속하는 것을 허용하지 않습니다.

자세한 정보는 IBM MQ의 도메인 계정 작성 및 설정을 참조하십시오.

Windows *IBM MQ* 서비스와 연관된 사용자 이름 변경

새 계정을 작성하고 Prepare IBM MQ Wizard를 사용하여 해당 세부사항을 입력하여 IBM MQ 서비스와 연관된 사용자 이름을 변경할 수 있습니다.

## 이 태스크 정보

IBM MQ를 설치하고 Prepare IBM MQ Wizard를 처음 실행할 때, MUSR_MQADMIN이라는 서비스에 대한 로컬 사용자 계정을 작성합니다. 후속 설치의 경우, Prepare IBM MQ Wizard는 MUSR_MQADMIN*x*라는 사용자 계정을 작성합니다. 여기서 *x*는 존재하지 않는 사용자 ID를 표시하는 다음에 사용 가능한 숫자입니다.

IBM MQ 서비스와 연관된 사용자 이름을 MUSR_MQADMIN 또는 MUSR_MQADMIN*x*에서 다른 것으로 변경해야 할 수도 있습니다. 예를 들어, 큐 관리자가 Db2®와 연관된 경우 이를 수행해야 할 수도 있습니다. DB2는 8자를 초과하는 사용자 이름을 승인하지 않습니다.

## 프로시저

1. 새 사용자 계정을 작성하십시오(예: **NEW_NAME** )

2. Prepare IBM MQ Wizard를 사용하여 새 사용자 계정의 세부사항을 입력하십시오.

**관련 태스크**

Prepare IBM MQ Wizard로 IBM MQ 구성

Windows *IBM MQ Windows* 서비스 로컬 사용자 계정의 비밀번호 변경

컴퓨터 관리 패널을 사용하여 IBM MQ Windows 서비스 로컬 사용자 계정의 비밀번호를 변경할 수 있습니다.

## 이 태스크 정보

IBM MQ Windows 서비스 로컬 사용자 계정의 비밀번호를 변경하려면 다음 단계를 수행하십시오.

## 프로시저

1. 서비스가 실행 중인 사용자를 식별하십시오.
2. 컴퓨터 관리 패널에서 IBM MQ 서비스를 중지하십시오.
3. 개인의 비밀번호를 변경하는 동일한 방법으로 필수 비밀번호를 변경하십시오.
4. 컴퓨터 관리 패널에서 IBM MQ 서비스에 대한 특성으로 이동하십시오.
5. **로그온** 페이지를 선택하십시오.
6. 지정된 계정 이름이 비밀번호가 수정된 사용자와 일치하는지 확인하십시오.
7. 비밀번호를 **비밀번호** 및 **비밀번호 확인** 필드에 입력하고 **확인**을 클릭하십시오.

Windows 도메인 사용자 계정에서 실행 중인 설치를 위한 *IBM MQ Windows* 서비스의 비밀번호 변경

Prepare IBM MQ Wizard 를 사용하여 도메인 사용자 계정의 계정 세부사항을 입력하는 대신 컴퓨터 관리 패널을 사용하여 설치 특정 IBM MQ 서비스에 대한 **로그온** 세부사항을 변경할 수 있습니다.

## 이 태스크 정보

설치를 위한 IBM MQ Windows 서비스가 도메인 사용자 계정에서 실행 중인 경우, 다음과 같이 계정에 대한 비밀번호를 변경할 수 있습니다.

## 프로시저

1. 도메인 제어기에서 도메인 계정에 대한 비밀번호를 변경하십시오. 도메인 관리자가 이를 수행하도록 요청해야 할 수도 있습니다.
2. IBM MQ 서비스에 대한 **로그온** 페이지를 수정하려면 다음 단계를 완료하십시오.
   a) 서비스가 실행 중인 사용자를 식별하십시오.
   b) 컴퓨터 관리 패널에서 IBM MQ 서비스를 중지하십시오.
   c) 개인의 비밀번호를 변경하는 동일한 방법으로 필수 비밀번호를 변경하십시오.
   d) 컴퓨터 관리 패널에서 IBM MQ 서비스에 대한 특성으로 이동하십시오.
   e) **로그온** 페이지를 선택하십시오.
   f) 지정된 계정 이름이 비밀번호가 수정된 사용자와 일치하는지 확인하십시오.
   g) 비밀번호를 **비밀번호** 및 **비밀번호 확인** 필드에 입력하고 **확인**을 클릭하십시오.

   IBM MQ Windows 서비스가 실행되는 사용자 계정은 사용자 인터페이스 애플리케이션에서 실행되거나 시스템 시동, 시스템 종료 또는 서비스 복구에서 자동으로 수행되는 MQSC 명령을 실행합니다. 따라서 이 사용자 계정에는 IBM MQ 관리 권한이 있어야 합니다. 기본적으로 서버의 로컬 mqm 그룹에 추가됩니다. 이 멤버십이 제거되면 IBM MQ Windows 서비스는 작동하지 않습니다. 사용자 권한에 대한 자세한 정보는 140 페이지의 『IBM MQ Windows 서비스에 필요한 사용자 권한』의 내용을 참조하십시오.

   보안 문제점이 IBM MQ Windows 서비스가 실행되는 사용자 계정과 함께 발생하면 오류 메시지 및 설명이 시스템 이벤트 로그에 나타납니다.

**관련 태스크**

Prepare IBM MQ Wizard로 IBM MQ 구성

## Windows *Windows* 서버를 도메인 제어기로 승격시킬 때 고려사항

Windows 서버를 도메인 제어기로 승격시키는 경우, 사용자 및 그룹 권한과 관련된 보안 설정이 적절한지 여부를 고려해야 합니다. 서버와 도메인 제어기 사이에서 Windows 시스템의 상태가 변경되면, IBM MQ에서는 로컬로 정의된 mqm 그룹을 사용하므로 이것이 IBM MQ의 조작에 영향을 줄 수 있음을 고려해야 합니다.

### 도메인 사용자 및 그룹 권한과 관련된 보안 설정

IBM MQ가 보안 정책 구현을 위해 그룹 멤버십 정보에 의존하므로 IBM MQ 운영을 수행하는 사용자 ID가 다른 사용자의 그룹 멤버십을 판별할 수 있는지 여부가 중요합니다.

Windows 서버를 도메인 제어기로 승격하면, 사용자 및 그룹 권한과 관련된 보안 설정 옵션이 표시됩니다. 이 옵션은 임의 사용자가 Active Directory에서 그룹 멤버십을 검색할 수 있도록 하는지 여부를 제어합니다. 로컬 계정에 도메인 사용자 계정이 그룹 멤버십을 조회할 권한을 부여하도록 도메인 제어기가 설정된 경우, 설치 프로세스 중에 IBM MQ에서 작성되는 기본 사용자 ID는 필요에 따라 다른 사용자에 대한 그룹 멤버십을 확보할 수 있습니다. 그러나 로컬 계정이 도메인 사용자 계정의 그룹 멤버십을 조회할 권한을 갖지 않도록 도메인 제어기가 설정된 경우, 도메인에 정의된 사용자에게 큐 관리자 또는 큐에 액세스할 권한이 부여되었는지 확인하는 검사가 IBM MQ에서 완료되지 않아 액세스가 실패합니다. 이 방법으로 설정된 도메인 제어기에서 Windows를 사용하는 경우 필수 권한이 있는 특수 도메인 계정이 사용되어야 합니다.

이런 경우 다음을 알아야 합니다.

- Windows 버전의 보안 권한이 작동하는 방식
- 도메인 mqm 그룹 멤버가 그룹 멤버십을 읽는 방법
- 도메인 사용자에서 실행되도록 IBM MQ Windows 서비스를 구성하는 방법

자세한 정보는 IBM MQ의 사용자 계정 구성을 참조하십시오.

### mqm 그룹에 대한 IBM MQ 액세스

Windows 서버가 도메인 제어기로 승격되거나 도메인 제어기에서 강등되는 경우, IBM MQ는 로컬 qm 그룹에 대한 액세스 권한이 상실됩니다.

서버가 도메인 제어기로 승격되는 경우 범위는 로컬에서 도메인 로컬로 변경됩니다. 시스템이 서버로 강등되면 모든 도메인 로컬 그룹이 제거됩니다. 즉, 서버에서 도메인 제어기로 시스템을 변경 및 서버로 다시 변경하는 경우, 로컬 mqm 그룹에 대한 액세스가 유실됨을 의미합니다. 증상은 로컬 mqm 그룹 부족을 표시하는 오류로 예를 들어 다음과 같습니다.

```
>crtmqm qm0
AMQ8066:Local mqm group not found.
```

이 문제를 해결하려면 표준 Windows 관리 도구를 사용하여 로컬 mqm 그룹을 다시 작성하십시오. 모든 그룹 멤버십 정보가 유실되기 때문에 새로 작성한 로컬 mqm 그룹에서 특권이 있는 IBM MQ 사용자를 다시 복원시켜야 합니다. 시스템이 도메인 멤버인 경우, 특권이 있는 도메인 IBM MQ 사용자 ID에 필요한 권한 레벨을 부여하기 위해 도메인 mqm 그룹을 로컬 mqm 그룹에도 추가해야 합니다.

## Windows *Windows*에서 중첩 그룹에 대한 제한사항

중첩 그룹 사용에는 제한이 있습니다. 이는 도메인 기능 레벨에서 부분적으로 초래되고 일부는 IBM MQ 제한에서 초래됩니다.

Active Directory는 도메인 기능 레벨에 따라 도메인 컨텍스트 내에서 다른 그룹 유형을 지원할 수 있습니다. 기본적으로 Windows 2003 도메인은 "Windows 2000 혼합" 기능 레벨에 있습니다. (Windows Server 2008 및 Windows Server 2012는 Windows 2003 도메인 모델을 따릅니다.) 도메인 기능 레벨은 지원되는 도메인 환경에서 사용자 ID를 구성할 때 허용되는 지원 그룹 유형 및 중첩 레벨을 판별합니다. 그룹 범위 및 포함 기준에 대한 자세한 내용은 Active Directory 문서를 참조하십시오.

Active Directory 요구사항뿐만 아니라 추가 제한이 IBM MQ에서 사용되는 ID에도 있습니다. IBM MQ에서 사용되는 네트워크 API는 도메인 기능 레벨에서 지원하는 모든 구성을 지원하지는 않습니다. 이로 인해 IBM MQ는 도메인 로컬 그룹에 있으며 로컬 그룹에 중첩되는 모든 도메인 ID의 그룹 멤버십을 조회할 수 없습니다. 더구나 글로벌 및 유니버셜 그룹의 다중 중첩은 지원되지 않습니다. 그렇지만 바로 중첩된 글로벌 또는 유니버셜 그룹은 지원됩니다.

## Windows 사용자에게 *IBM MQ*의 원격 사용 권한 부여

IBM MQ에 원격으로 연결될 때 큐 관리자를 작성 및 시작해야 하는 경우, 글로벌  오브젝트  작성 사용자 액세스 권한이 있어야 합니다.

### 이 태스크 정보

**참고:** 관리자에게는 기본적으로 글로벌  오브젝트  작성 사용자 액세스 권한이 있습니다. 따라서 관리자인 경우에는 사용자 권한을 변경하지 않고 원격으로 연결될 때 큐 관리자를 작성하고 시작할 수 있습니다.

터미널 서비스 는 원격 데스크탑 연결을 사용하여 Windows 시스템에 연결하고 큐 관리자 작성, 시작 또는 삭제에 문제가 발생하는 경우, 이는 글로벌  오브젝트  작성에 대한 사용자 액세스 권한이 없기 때문일 수 있습니다.

글로벌  오브젝트  작성 사용자 액세스는 글로벌 네임스페이스에서 오브젝트를 작성하도록 권한이 부여된 사용자로 제한됩니다. 애플리케이션에서 글로벌 오브젝트를 작성하기 위해서는 글로벌 네임스페이스에서 실행되거나 애플리케이션을 실행하는 사용자에게 글로벌  오브젝트  작성 사용자 액세스 권한이 적용되어야 합니다.

터미널 서비스 또는 원격 데스크탑 연결을 사용하여 원격으로 Windows 시스템에 연결하는 경우, 애플리케이션은 자체 보유한 로컬 네임스페이스에서 실행됩니다. IBM MQ Explorer를 사용하거나 **crtmqm** 또는 **dltmqm** 명령을 사용하여 큐 관리자를 작성하거나 삭제하는 경우 또는 **strmqm** 명령을 사용하여 큐 관리자를 시작하려고 하면 권한 부여에 실패합니다. 그러면 프로브 ID가 XY132002인 IBM MQ FDC가 작성됩니다.

IBM MQ Explorer를 사용하거나 **amqmdain qmgr start** 명령을 사용하면 큐 관리자가 제대로 시작되는데, 이는 이 명령으로 큐 관리자가 직접 시작되는 것이 아니기 때문입니다. 대신, 이 명령은 큐 관리자 시작 요청을 글로벌 네임스페이스에서 실행되는 별도의 프로세스로 송신합니다.

IBM MQ에 대한 다양한 관리 방법이 터미널 서비스 사용 시에 작동하지 않으면 글로벌  오브젝트  작성 사용자 권한을 설정하십시오.

### 프로시저

1. 관리 도구 패널을 여십시오.

   **Windows Server 2008 및 Windows Server 2012**
   **제어판** > **시스템 및 유지보수** > **관리 도구**를 사용하여 이 패널에 액세스하십시오.

   **Windows 8.1**
   **관리 도구** > **컴퓨터 관리**를 사용하여 이 패널에 액세스하십시오.
2. **로컬 보안 정책**을 두 번 클릭하십시오.
3. **로컬 정책**을 펼치십시오.
4. **사용자 권한 지정**을 클릭하십시오.
5. 새 사용자 또는 그룹을 글로벌  오브젝트  작성 정책에 추가하십시오.

## Windows *Windows*에서 *SSPI* 채널 엑시트 프로그램

IBM MQ for Windows는 보안 엑시트 프로그램을 제공하며 이는 메시지 및 MQI 채널 모두에서 사용 가능합니다. 엑시트는 소스 및 오브젝트 코드로 제공되며 단방향 및 양방향 인증을 제공합니다.

보안 엑시트는 SSPI(Security Support Provider Interface)를 사용하며 이는 Windows 플랫폼의 통합 보안 기능을 제공합니다.

보안 엑시트는 다음의 식별 및 인증 서비스를 제공합니다.

**단방향 인증**
이는 Windows NT LAN Manager(NTLM) 인증 지원을 사용합니다. NTLM을 사용하여 서버가 해당 클라이언트를 인증합니다. 클라이언트가 서버를 인증하는 것이나, 임의 서버가 다른 서버를 인증하는 것은 허용하지 않습니다. NTLM은 서버가 조작되지 않았다고 가정되는 네트워크 환경을 위해 설계되었습니다. NTLM은 IBM WebSphere MQ 7.0에서 지원되는 모든 Windows 플랫폼에서 지원됩니다.

이 서비스는 일반적으로 MQI 채널에서 사용되어 서버 큐 관리자가 IBM MQ MQI client 애플리케이션을 인증하도록 합니다. 클라이언트 애플리케이션은 실행 중인 프로세스에 연관된 사용자 ID로 식별됩니다.

인증을 수행하려면, 채널의 클라이언트 쪽에 있는 보안 엑시트가 NTLM에서 인증 토큰을 얻고 토큰을 보안 메시지로 채널의 다른 쪽에 있는 해당 파트너에게 송신합니다. 파트너 보안 엑시트는 토큰을 NTLM에게 전달하며, NTLM은 해당 토큰이 인증된 것인지 검사합니다. 파트너 보안 엑시트가 토큰 인증을 만족시키지 않으면 MCA가 채널을 닫도록 지시합니다.

**양방향 또는 상호 인증**

이는 Kerberos 인증 서비스를 사용합니다. Kerberos 프로토콜은 네트워크 환경에 있는 서버가 진짜라고 가정하지 않습니다. 서버가 클라이언트와 다른 서버를 인증할 수 있고, 클라이언트가 서버를 인증할 수 있습니다. Kerberos 는 IBM WebSphere MQ 7.0에서 지원되는 모든 Windows 플랫폼에서 지원됩니다.

이 서비스는 메시지와 MQI 채널 둘 다에서 사용될 수 있습니다. 메시지 채널에서는 두 개의 큐 관리자의 상호 인증을 제공합니다. MQI 채널에서 서버 큐 관리자 및 IBM MQ MQI client 애플리케이션이 서로를 인증하도록 합니다. 큐 관리자는 `ibmMQSeries/` 문자열이 접두부로 붙은 이름에 의해 식별됩니다. 클라이언트 애플리케이션은 실행 중인 프로세스에 연관된 사용자 ID로 식별됩니다.

상호 인증을 수행하기 위해, 시작 측 보안 엑시트가 Kerberos 보안 서버에서 인증 토큰을 취득하여 토큰을 보안 메시지로 파트너에게 송신합니다. 파트너 보안 엑시트는 토큰을 Kerberos 서버에 전달하며 Kerberos 서버는 토큰이 인증된 것인지 검사합니다. Kerberos 보안 서버는 파트너가 시작 측 보안 엑시트에 보안 메시지로 송신하는 두 번째 토큰을 생성합니다. 그런 다음, 시작 측 보안 엑시트가 Kerberos 서버에게 두 번째 토큰이 인증되었는지 검사하도록 요청합니다. 이 교환 동안에, 보안 엑시트 중 어느 것이든지 상대방이 송신한 토큰의 인증에 만족하지 않으면 MCA에게 채널을 닫으라고 지시합니다.

보안 엑시트는 소스와 오브젝트 형식 둘 다로 제공됩니다. 자체 채널 엑시트 프로그램을 쓰기 위한 시작점으로서 소스 코드를 사용하거나, 오브젝트 모듈을 제공된 대로 사용할 수 있습니다. 오브젝트 모듈에는 두 개의 시작점이 있는데, 하나는 NTLM 인증 지원을 사용하는 단방향 인증용이고 다른 하나는 Kerberos 인증 서비스를 사용하는 양방향 인증용입니다.

SSPI 채널 엑시트 프로그램이 작동하는 방법에 대한 자세한 정보 및 구현 방법에 대한 지시사항은 Windows 시스템에서 SSPI 보안 엑시트 사용을 참조하십시오.

## Windows *Windows*에서 보안 템플리트 파일 적용

템플리트를 적용하면 IBM MQ 파일 및 디렉토리에 적용되는 보안 설정에 영향을 줄 수 있습니다. 매우 안전한 템플리트를 사용하는 경우 IBM MQ 설치 전에 이를 적용하십시오.

Windows는 보안 구성 및 분석 MMC 스냅인으로 하나 이상의 컴퓨터에 균등한 보안 설정을 적용할 수 있도록 하는 텍스트 기반의 보안 템플리트 파일을 지원합니다. 특히, Windows는 특정 보안 레벨을 제공하기 위한 보안 설정 범위를 포함하는 몇 개의 템플리트를 제공합니다. 이런 템플리트에는 Compatible, Secure, Highly Secure가 포함됩니다.

이 템플리트 중 하나를 적용하면 IBM MQ 파일 및 디렉토리에 적용되는 보안 설정에 영향을 줄 수 있습니다. Highly Secure 템플리트를 사용하려는 경우, IBM MQ 설치 전에 시스템을 구성하십시오.

Highly Secure 템플리트를 IBM MQ가 이미 설치된 시스템에 적용하는 경우, IBM MQ 파일 및 디렉토리에 이미 설정한 모든 권한이 제거됩니다. 이 권한이 제거되기 때문에 오류 디렉토리에서 관리자, *mqm*, 적용되는 경우 모든 사용자 그룹 액세스가 유실됩니다.

## Windows *IBM MQ* 에 연결하는 *Windows* 애플리케이션에 대한 추가 권한 구성

IBM MQ 프로세스가 실행되는 계정은 애플리케이션 프로세스에 대해 SYNCHRONIZE 액세스가 부여되기 전에 추가 인증이 필요할 수도 있습니다.

### 이 태스크 정보

Windows 애플리케이션이 있는 경우에 문제가 발생할 수 있습니다. 예를 들어, ASP 페이지로 정상적인 보안 레벨 이상으로 실행되도록 구성되는 IBM MQ 연결됩니다.

IBM MQ의 경우 특정 조치를 통합하려면 애플리케이션 프로세스에 대해 SYNCHRONIZE 액세스가 필요합니다. 서버 애플리케이션이 처음으로 큐 관리자 연결을 시도하는 경우, IBM MQ는 IBM MQ 관리자에 대해 SYNCHRONIZE 권한을 부여하기 위해 프로세스를 수정합니다. 그러나 IBM MQ 프로세스가 실행되는 계정에는 요청된 액세스가 부여되기 전에 추가 권한이 필요할 수 있습니다.

IBM MQ 프로세스가 실행되는 사용자 ID에 대해 추가 권한을 구성하려면 다음 단계를 완료하십시오.

### 프로시저

1. 로컬 보안 정책 도구를 시작하고 **보안 설정->로컬 정책->사용자 권한 지정**을 클릭한 후 **디버그 프로그램**을 클릭하십시오.
2. **디버그 프로그램**을 두 번 클릭하고 IBM MQ 사용자 ID를 목록에 추가하십시오.

   시스템이 Windows 도메인에 있고 유효한 정책 설정이 여전히 설정되지 않았다면 로컬 정책 설정이 설정되더라도 사용자 ID는 도메인 보안 정책 도구를 사용하여 도메인 레벨에서 동일한 방식으로 권한 부여되어야 합니다.

## IBM i IBM i에서 보안 설정

IBM i에서 보안은 IBM MQ 오브젝트 권한 관리자(OAM) 및 IBM i 오브젝트 레벨 보안을 사용하여 구현됩니다.

IBM MQ 오브젝트에 대한 액세스 권한을 판별할 때 수행해야 하는 보안 고려사항.

엔터프라이즈의 사용자에 대한 권한을 설정할 때 다음 사항을 고려해야 합니다.

1. IBM i GRTOBJAUT 및 RVKOBJAUT 명령을 사용하여 IBM MQ for IBM i 명령에 대한 권한을 부여하고 취소합니다.

   QMQM 라이브러리에서 특정 비명령(*cmd) 오브젝트가 ***USE**에 대한 ***PUBLIC** 권한을 갖도록 설정됩니다. 권한을 제공하기 위해 권한 목록을 사용하거나 이 오브젝트의 권한을 변경하지 마십시오. 잘못된 모든 권한으로 인해 IBM MQ 기능이 손상될 수 있습니다.
2. IBM MQ for IBM i 설치 중에 다음 특수 사용자 프로파일이 작성됩니다.

   **QMQM**
   내부 제품 전용 기능으로 주로 사용됩니다. 그렇지만 MQCNO_FASTPATH_BINDINGS를 사용하여 신뢰 애플리케이션 실행에 사용할 수 있습니다. MQCONNX 호출을 사용하여 큐 관리자에 연결을 참조하십시오.

   **QMQMADM**
   IBM MQ 관리자의 그룹 프로파일로 사용됩니다. 그룹 프로파일은 CL 명령 및 IBM MQ 자원에 대한 액세스를 제공합니다.

   IBM MQ 명령을 호출하는 프로그램을 제출하기 위해 SBMJOB을 사용하는 경우, USER는 QMQMADM에 명시적으로 설정되면 안됩니다. 대신, USER를 QMQM 또는 그룹으로 QMQMADM을 지정한 다른 사용자 프로파일에 설정합니다.
3. 채널 명령을 리모트 큐 관리자에게 보내려는 경우 사용자 프로파일이 대상 시스템의 QMQMADM 그룹 멤버인지 확인하십시오. PCF 및 MQSC 채널 명령 목록에 대해서는 IBM MQ for IBM i CL 명령을 참조하십시오.
4. 그룹 권한 부여가 OAM에서 처리될 때 사용자와 연관된 그룹 세트가 캐시됩니다.

   **큐 관리자를 다시 시작하거나 RFRMQMAUT를 실행하여 보안을 새로 고칠 때까지, 그룹 세트가 캐시된 후 사용자의 그룹 멤버십에 작성된 변경사항이 인식되지 않습니다**.
5. 특히 예민한 명령과 작업할 권한이 있는 사용자의 수를 제한합니다. 이 명령에는 다음이 포함됩니다.
   - 메시지 큐 관리자 작성(CRTMQM)
   - 메시지 큐 관리자 삭제(DLTMQM)
   - 메시지 큐 관리자 시작(STRMQM)
   - 메시지 큐 관리자 종료(ENDMQM)
   - 명령 서버 시작(STRMQMCSVR)
   - 명령 서버 종료(ENDMQMCSVR)
6. 채널 정의는 보안 엑시트 프로그램 스펙을 포함합니다. 채널 작성 및 수정에는 특별한 고려사항이 필요합니다. 보안 엑시트의 세부사항은 103 페이지의 『보안 엑시트 개요』에서 제공됩니다.
7. 채널 엑시트 및 트리거 모니터 프로그램은 대체 가능합니다. 이러한 대체에 대한 보안은 프로그래머의 책임입니다.

## IBM i IBM i에서 오브젝트 권한 관리자

오브젝트 권한 관리자(OAM)는 큐 및 프로세스 정의를 포함하여 IBM MQ 오브젝트를 조작하기 위해 사용자의 권한을 관리합니다. 또한, 특정 사용자 그룹에 대해 오브젝트에 대한 액세스 권한을 부여하거나 취소하는 명령 인터페이스도 제공합니다. 자원 액세스 허용 여부는 OAM이 결정하며 큐 관리자는 해당 결정을 따릅니다. OAM이 결정하지 못하는 경우 큐 관리자가 해당 자원 액세스를 차단합니다.

OAM을 통해 다음을 제어할 수 있습니다.

- MQI를 통해 IBM MQ 오브젝트 액세스. 애플리케이션 프로그램이 오브젝트에 대해 액세스를 시도하면 OAM은 요청을 수행하는 사용자 프로파일에 요청된 조작에 대한 권한이 있는지를 검사합니다.

  특히, 권한 없는 액세스로부터 큐의 메시지와 큐를 보호할 수 있습니다.
- PCF 및 MQSC 명령 사용 권한.

다른 사용자 그룹은 동일한 오브젝트에 대해 다른 액세스 권한을 가집니다. 예를 들어, 특정 큐에 대해 한 그룹은 넣기 및 가져오기 조작을 모두 수행할 수 있지만 다른 그룹은 큐를 찾아보는 것만 허용될 수도 있습니다(browse 옵션이 포함된 MQGET). 이와 유사하게 일부 그룹은 큐에 대해 가져오기 및 넣기 권한이 있지만 큐를 대체하거나 삭제할 수는 없습니다.

IBM MQ for IBM i 오브젝트에 대한 IBM MQ for IBM i 명령 및 수행 조작

## IBM i IBM i 에 대한 IBM MQ 권한

IBM MQ 오브젝트에 액세스하려면 명령을 발행하고 참조되는 오브젝트에 액세스할 권한이 필요합니다. 관리자도 모든 IBM MQ 자원에 액세스 권한이 있습니다.

IBM MQ 오브젝트 액세스는 다음에 대한 권한으로 제어됩니다.

1. IBM MQ 명령 발행
2. 명령에서 참조되는 IBM MQ 오브젝트 액세스

모든 IBM MQ for IBM i CL 명령은 QMQM 소유자와 같이 제공되고 관리 프로파일(QMQMQADM)에는 *PUBLIC 액세스가 *EXCLUDE로 설정된 *USE 권한이 있습니다.

**참고:** QSRDUPER 프로그램은 IBM MQ for IBM i 라이센스가 있는 프로그램 설치 프로그램이 QSYS에서 명령(*CMD) 오브젝트를 복제하는 데 사용됩니다. IBM i V5R4 이상에서 QSRDUPER 프로그램은 기본 동작이 원래 명령 중복이 아니라 프록시 명령 작성이 되도록 변경되었습니다. 프록시 명령은 명령 실행은 다른 명령으로 경로 재지정하고 PRX 속성을 포함합니다. 복사 중인 명령과 동일한 이름의 프록시 명령이 라이브러리 QSYS에 있는 경우, 프록시 명령에 대한 개인용 권한은 제품 라이브러리의 명령에 부여되지 않습니다. QSYS의 프록시 명령 프롬프트 또는 실행 시도는 제품 라이브러리에서 대상 명령 권한을 검사합니다. 따라서 권한에서 *CMD로의 모든 변경은 제품 라이브러리(QMQM)에서 수행해야 하며 QSYS의 권한은 수정할 필요가 없습니다. 예를 들면, 다음과 같습니다.

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

제품 CL 명령 일부의 권한 구조 변경을 수행하는 IBM MQ 오브젝트에 대한 필수 OAM 권한이 있는 경우 이 변경을 통해 해당 명령을 공개적으로 사용할 수 있게 됩니다.

IBM i에서 IBM MQ 관리자가 되려면 *QMQMADM* 그룹의 구성원이어야 합니다. 이 그룹에는 AIX, Linux, and Windows 시스템에 있는 mqm 그룹의 속성과 같은 속성이 있습니다. 특히, QMQMADM 그룹은 IBM MQ for IBM i 설치 시에 작성되고 QMQMADM 그룹 구성원은 시스템의 모든 IBM MQ 자원에 대한 액세스 권한을 가집니다. *ALLOBJ 권한이 있는 사용자인 경우 모든 IBM MQ 자원에도 액세스할 수 있습니다.

관리자는 CL 명령을 사용하여 IBM MQ를 관리할 수 있습니다. 이들 명령 중 하나는 다른 사용자에게 권한을 부여하는 데 사용되는 GRTMQMAUT입니다. 다른 명령인 STRMQMMQSC로 관리자가 로컬 큐 관리자에게 MQSC 명령을 발행할 수 있습니다.

**관련 개념**

84 페이지의 『IBM i 에서 IBM MQ 를 관리할 수 있는 권한』

## IBM i *IBM i 의 IBM MQ 오브젝트에 대한 액세스 권한*

IBM MQ CL 명령 실행에 필요한 액세스 권한입니다.

IBM MQ for IBM i는 제품의 CL 명령을 두 그룹으로 범주화합니다.

**그룹 1**

해당 명령을 처리하려면 사용자는 QMQMADM 사용자 그룹에 있거나 *ALLOBJ 권한이 있어야 합니다. 이 권한 중 하나가 있는 사용자는 추가 권한 없이도 모든 범주의 모든 명령을 처리할 수 있습니다.

**참고:** 이 권한은 모든 OAM 권한을 대체합니다.

이 명령은 다음과 같이 그룹화할 수 있습니다.

- 명령 서버 명령
  - ENDMQMCSVR, IBM MQ 명령 서버 종료
  - STRMQMCSVR, IBM MQ 명령 서버 시작
- 데드-레터 큐 핸들러 명령
  - STRMQMDLQ, IBM MQ 데드-레터 큐 핸들러 시작
- 리스너 명령
  - ENDMQMLSR, IBM MQ 리스너 종료
  - STRMQMLSR, 비오브젝트 리스너 시작
- 매체 복원 명령
  - RCDMQMIMG, IBM MQ 오브젝트 이미지 기록
  - RCRMQMOBJ, IBM MQ 오브젝트 재작성
  - WRKMQMTRN, IBM MQ Q 트랜잭션에 대한 작업
- 큐 관리자 명령
  - CRTMQM, 메시지 큐 관리자 작성
  - DLTMQM, 메시지 큐 관리자 삭제
  - ENDMQM, 메시지 큐 관리자 종료
  - STRMQM, 메시지 큐 관리자 시작
- 보안 명령
  - GRTMQMAUT, IBM MQ 오브젝트 권한 부여
  - RVKMQMAUT, IBM MQ 오브젝트 권한 취소
- 추적 명령
  - TRCMQM, IBM MQ 작업 추적
- 트랜잭션 명령
  - RSVMQMTRN, IBM MQ 트랜잭션 해결
- 트리거 모니터 명령
  - STRMQMTRM, 트리거 모니터 시작
- IBM MQSC 명령
  - RUNMQSC, IBM MQSC 명령 실행
  - STRMQMMQSC, IBM MQSC 명령 시작

**그룹 2**

2 레벨의 권한이 필요한 나머지 명령:

1. 명령 실행을 위한 IBM i 권한. IBM MQ 관리자는 **GRTOBJAUT** 명령으로 사용자 또는 사용자 그룹에 대한 *PUBLIC(*EXCLUDE) 제한을 대체하여 이를 설정합니다.

   예를 들면, 다음과 같습니다.

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

2. 1단계에서올바른 IBM i 권한이 제공된 경우 명령과 연관된 IBM MQ 오브젝트를 조작하기 위한 IBM MQ 권한.

   이 권한은 **GRTMQMAUT** 명령을 사용하여 IBM MQ 관리자가 설정한 필수 조치에 대해 적절한 OAM 권한이 있는 사용자가 제어합니다.

   예를 들면, 다음과 같습니다.

```
GRTMQMAUT *connect authority to the queue manager + *admchg authority to
        the queue
```

명령은 다음과 같이 그룹화할 수 있습니다.

- 채널 명령
  - CHGMQMCHL, IBM MQ 채널 변경

    큐 관리자에 대해서는 *connect 권한이 필요하고 채널에 대해서는 *admchg 권한이 필요합니다.
  - CPYMQMCHL, IBM MQ 채널 복사

    큐 관리자에 대해서는 *connect 및 *admcrt 권한이 필요하고 복사되는 기본 채널 유형에 대해서는 *admdsp 권한이 필요하며 채널 오브젝트 클래스에 대해서는 *admcrt 권한이 필요합니다.

    예를 들어, 송신자 채널을 복사하려면 SYSTEM.DEF.SENDER 채널에 대해 *admdsp 권한이 필요합니다.
  - CRTMQMCHL, IBM MQ 채널 작성

    큐 관리자에 대해서는 *connect 및 *admcrt 권한이 필요하고 작성되는 기본 채널 유형에 대해서는 *admdsp 권한이 필요하고 채널 오브젝트 클래스에 대해서는 *admcrt 권한이 필요합니다.

    예를 들어, 송신자 채널을 작성하려면 SYSTEM.DEF.SENDER 채널에 대해 *admdsp 권한이 필요합니다.
  - DLTMQMCHL, IBM MQ 채널 삭제

    큐 관리자에 대해서는 *connect 권한이 필요하고 채널에 대해서는 *admdlt 권한이 필요합니다.
  - RSVMQMCHL, IBM MQ 채널 분석

    큐 관리자에 대해서는 *connect 권한이 필요하고 채널에 대해서는 *ctrlx 권한이 필요합니다.
- 명령 표시

  DSP 명령을 처리하려면 사용자에게 나열된 모든 특정 옵션과 같이 큐 관리자를 위한 *connect 및 *admdsp 권한을 부여해야 합니다.
  - DSPMQM, 메시지 큐 관리자 표시
  - DSPMQMAUT, IBM MQ 오브젝트 권한 표시
  - DSPMQMAUTI, IBM MQ 인증 정보 표시 - 인증 정보 오브젝트에 대해 *admdsp
  - DSPMQMCHL, IBM MQ 채널 표시 - 채널에 대해 *admdsp
  - DSPMQMCSVR, IBM MQ 명령 서버 표시
  - DSPMQMNL, IBM MQ이름 목록 표시 - 이름 목록에 대해 *admdsp
  - DSPMQMOBJN, IBM MQ 오브젝트 이름 표시
  - DSPMQMPRC, IBM MQ 프로세스 표시 - 프로세스에 대해 *admdsp
  - DSPMQMQ, IBM MQ 큐 표시 - 큐에 대해 *admdsp
  - DSPMQMTOP, IBM MQ 토픽 표시 - 토픽에 대해 *admdsp
- 명령에 대해 작업

WRK 명령을 처리하고 옵션 패널을 표시하려면 사용자에게 나열된 모든 특정 옵션과 같이 큐 관리자를 위한 *connect 및 *admdsp 권한을 부여해야 합니다.

- WRKMQM, 메시지 큐 관리자에 대해 작업
- WRKMQMAUT, IBM MQ 오브젝트 권한에 대해 작업
- WRKMQMAUTD, IBM MQ 오브젝트 권한 데이터에 대해 작업
- WRKMQMAUTI, IBM MQ 인증 정보에 대해 작업
  - *admchg - IBM MQ 인증 정보 오브젝트 변경 명령의 경우.
  - IBM MQ 인증 정보 오브젝트 작성 및 복사 명령의 경우 *admcrt .
  - IBM MQ 인증 정보 오브젝트 삭제 명령의 경우 *admdlt 입니다.
  - IBM MQ 인증 정보 오브젝트 표시 명령의 경우 *admdsp 입니다.
- WRKMQMCHL, IBM MQ 채널에 대해 작업

  여기에는 다음 권한이 필요합니다.
  - IBM MQ 채널 변경 명령의 경우 *admchg .
  - IBM MQ 채널 지우기 명령의 경우 *admclr 입니다.
  - IBM MQ 채널 작성 및 복사 명령의 경우 *admcrt 입니다.
  - IBM MQ 채널 삭제 명령의 경우 *admdlt 입니다.
  - IBM MQ 채널 표시 명령의 경우 *admdsp 입니다.
  - IBM MQ 채널 시작 명령의 경우 *ctrl 입니다.
  - IBM MQ 채널 종료 명령의 경우 *ctrl 입니다.
  - IBM MQ 채널 ping 명령의 경우 *ctrl 입니다.
  - IBM MQ 채널 재설정 명령의 경우 *ctrlx 입니다.
  - IBM MQ 채널 해석 명령의 경우 *ctrlx 입니다.
- WRKMQMCHST, IBM MQ 채널 상태에 대해 작업

  이 명령에는 채널에 대해 *admdsp 권한이 필요합니다.
- WRKMQMCL, IBM MQ 클러스터에 대해 작업
- WRKMQMCLQ, IBM MQ 클러스터 큐에 대해 작업
- WRKMQMCLQM, WIBM MQ 클러스터 큐 관리자에 대해 작업
- WRKMQMLSR, IBM MQ 리스너에 대해 작업
- WRKMQMMSG, IBM MQ 메시지에 대해 작업

  이 명령에는 큐에 대해 *browse 권한이 필요합니다.
- WRKMQMNL, IBM MQ 이름 목록에 대해 작업

  여기에는 다음 권한이 필요합니다.
  - IBM MQ 이름 목록 변경 명령의 경우 *admchg .
  - IBM MQ 이름 목록 작성 및 복사 명령의 경우 *admcrt 입니다.
  - IBM MQ 이름 목록 삭제 명령의 경우 *admdlt 입니다.
  - IBM MQ 이름 목록 표시 명령의 경우 *admdsp 입니다.
- WRKMQMPRC, IBM MQ 프로세스에 대해 작업

  여기에는 다음 권한이 필요합니다.
  - IBM MQ 프로세스 변경 명령의 경우 *admchg 입니다.
  - IBM MQ 프로세스 작성 및 복사 명령의 경우 *admcrt .
  - IBM MQ 프로세스 삭제 명령의 경우 *admdlt 입니다.

- IBM MQ 프로세스 표시 명령의 경우 *admdsp 입니다.

– WRKMQMQ, IBM MQ 큐에 대해 작업

여기에는 다음 권한이 필요합니다.

- IBM MQ 큐 변경 명령의 경우 *admchg 입니다.
- IBM MQ 큐 지우기 명령의 경우 *admclr 입니다.
- IBM MQ 큐 작성 및 복사 명령의 경우 *admcrt 입니다.
- IBM MQ 큐 삭제 명령의 경우 *admdlt 입니다.
- IBM MQ 큐 표시 명령의 경우 *admdsp 입니다.

– WRKMQMQSTS, IBM MQ 큐 상태에 대해 작업

– WRKMQMTOP, IBM MQ 토픽에 대해 작업

여기에는 다음 권한이 필요합니다.

- IBM MQ 토픽 변경 명령의 경우 *admchg 입니다.
- IBM MQ 토픽 작성 및 복사 명령의 경우 *admcrt 입니다.
- IBM MQ 토픽 삭제 명령의 경우 *admdlt 입니다.
- IBM MQ 토픽 표시 명령의 경우 *admdsp 입니다.

– WRKMQMSUB, IBM MQ 구독에 대해 작업

• 기타 채널 명령

채널 명령을 처리하려면 사용자에게 나열된 특정 권한을 부여해야 합니다.

– ENDMQMCHL, IBM MQ 채널 종료

여기에는 큐 관리자에 대해 *connect 권한 및 채널에 연관된 전송 큐에 대해 *allmqi 권한이 필요합니다.

– ENDMQMLSR, IBM MQ 리스너 종료

여기에는 큐 관리자에 대해 *connect 권한 및 이름 지정된 리스너 오브젝트에 대해 *ctrl 권한이 필요합니다.

– PNGMQMCHL, IBM MQ 채널 ping

여기에는 큐 관리자에 대해 *connect 및 *inq 권한 및 채널 오브젝트에 대해 *ctrl 권한이 필요합니다.

– RSTMQMCHL, IBM MQ 채널 재설정

여기에는 큐 관리자에 대한 *connect 권한이 필요합니다.

– STRMQMCHL, IBM MQ 채널 시작

여기에는 큐 관리자에 대해 *connect 권한 및 채널 오브젝트에 대해 *ctrl 권한이 필요합니다.

– STRMQMCHLI, IBM MQ 채널 시작기 시작

여기에는 큐 관리자에 대해 *connect 및 *inq 권한 및 채널의 전송 큐에 연관된 이니시에이션 큐에 대해 *allmqi 권한이 필요합니다.

– STRMQMLSR, IBM MQ 리스너 시작

여기에는 큐 관리자에 대해 *connect 권한 및 이름 지정된 리스너 오브젝트에 대해 *ctrl 권한이 필요합니다.

• 기타 명령:

다음 명령을 처리하려면 사용자에게 나열된 특정 권한을 부여해야 합니다.

– CCTMQM, 메시지 큐 관리자에 연결

여기에는 IBM MQ 오브젝트 권한이 필요하지 않습니다.

- CHGMQM, 메시지 큐 관리자 변경

  여기에는 큐 관리자에 대한 *connect 및 *admchg 권한이 필요합니다.
- CHGMQMAUTI, IBM MQ 인증 정보 변경

  여기에는 큐 관리자에 대해 *connect 권한 및 인증 정보 오브젝트에 대해 *admchg 및 *admdsp 권한이 필요합니다.
- CHGMQMNL, CIBM MQ 이름 목록 변경

  여기에는 큐 관리자에 대해 *connect 권한 및 이름 목록에 대해 *admchg 권한이 필요합니다.
- CHGMQMPRC, IBM MQ 프로세스 변경

  여기에는 큐 관리자에 대해 *connect 권한 및 프로세스에 대해 *admchg 권한이 필요합니다.
- CHGMQMQ, IBM MQ 큐 변경

  여기에는 큐 관리자에 대해 *connect 권한 및 큐에 대해 *admchg 권한이 필요합니다.
- CLRMQMQ, IBM MQ 큐 지우기

  여기에는 큐 관리자에 대해 *connect 권한 및 큐에 대해 *admclr 권한이 필요합니다.
- CPYMQMAUTI, IBM MQ 인증 정보 복사

  여기에는 큐 관리자에 대해 *connect 권한, 인증 정보 오브젝트에 대해 *admdsp 권한, 인증 정보 오브젝트 클래스에 대한 *admcrt 권한이 필요합니다.
- CPYMQMNL, IBM MQ 이름 목록 복사

  여기에는 큐 관리자에 대한 *connect 및 *admcrt 권한이 필요합니다.
- CPYMQMPRC, IBM MQ 프로세스 복사

  여기에는 큐 관리자에 대한 *connect 및 *admcrt 권한이 필요합니다.
- CPYMQMQ, IBM MQ 큐 복사

  여기에는 큐 관리자에 대한 *connect 및 *admcrt 권한이 필요합니다.
- CRTMQMAUTI, IBM MQ 인증 정보 작성

  여기에는 큐 관리자에 대해 *connect 권한, 인증 정보 오브젝트에 대해 *admdsp 권한, 인증 정보 오브젝트 클래스에 대한 *admcrt 권한이 필요합니다.
- CRTMQMNL, IBM MQ 이름 목록 작성

  여기에는 큐 관리자에 대해 *connect 및 *admcrt 권한 및 기본 이름 목록에 대해 *admdsp 권한이 필요합니다.
- CRTMQMPRC, IBM MQ 프로세스 작성

  여기에는 큐 관리자에 대해 *connect 및 *admcrt 권한 및 기본 프로세스에 대해 *admdsp 권한이 필요합니다.
- CRTMQMQ, IBM MQ 큐 작성

  여기에는 큐 관리자에 대해 *connect 및 *admcrt 권한 및 기본 큐에 대해 *admdsp 권한이 필요합니다.
- CVTMQMDTA, IBM MQ 데이터 유형 변환 명령

  여기에는 IBM MQ 오브젝트 권한이 필요하지 않습니다.
- DLTMQMAUTI, IBM MQ 인증 정보 삭제

  여기에는 큐 관리자에 대해 *connect 권한 및 인증 정보 오브젝트에 대해 *ctrlx 권한이 필요합니다.
- DLTMQMNL, IBM MQ 이름 목록 삭제

  여기에는 큐 관리자에 대해 *connect 권한 및 이름 목록에 대해 *admdlt 권한이 필요합니다.
- DLTMQMPRC, IBM MQ 프로세스 삭제

여기에는 큐 관리자에 대해 *connect 권한 및 프로세스에 대해 *admdlt 권한이 필요합니다.

- DLTMQMQ, IBM MQ 큐 삭제

  여기에는 큐 관리자에 대해 *connect 권한 및 큐에 대해 *admdlt 권한이 필요합니다.

- DSCMQM, 메시지 큐 관리자에서 연결 끊기

  여기에는 IBM MQ 오브젝트 권한이 필요하지 않습니다.

- RFRMQMAUT, 보안 새로 고치기

  여기에는 큐 관리자에 대한 *connect 권한이 필요합니다.

- RFRMQMCL, 클러스터 새로 고치기

  여기에는 큐 관리자에 대한 *connect 권한이 필요합니다.

- RSMMQMCLQM, 클러스터 큐 관리자 재개

  여기에는 큐 관리자에 대한 *connect 권한이 필요합니다.

- RSTMQMCL, 클러스터 재설정

  여기에는 큐 관리자에 대한 *connect 권한이 필요합니다.

- SPDMQMCLQM, 클러스터 큐 관리자 일시중단

  여기에는 큐 관리자에 대한 *connect 권한이 필요합니다.

### IBM i *IBM i*에서의 액세스 권한 부여

이 정보를 사용하여 액세스 권한 부여 명령에 대해 이해하십시오.

GRTMQMAUT 및 RVKMQMAUT 명령에서 AUT 키워드로 정의되는 권한은 다음과 같이 분류할 수 있습니다.

- MQI 호출에 연관된 권한
- 권한 부여 관련 관리 명령
- 컨텍스트 권한
- MQI 호출, 명령 또는 둘 다를 위한 일반 권한

다음 표는 MQI 호출, 컨텍스트 호출, MQSC 및 PCF 명령, 일반 조작에 대해 AUT 매개변수를 사용하여 다른 권한을 나열합니다.

표 *15. MQI* 호출에 대한 권한 부여

| **AUT** | **설명** |
|---|---|
| *ALTUSR | 다른 사용자의 권한이 MQOPEN 및 MQPUT1 호출에 사용되도록 허용합니다. |
| *BROWSE | BROWSE 옵션과 함께 MQGET을 호출하여 큐에서 메시지를 검색합니다. |
| *CONNECT | MQCONN을 호출하여 지정된 큐 관리자에 애플리케이션을 연결합니다. |
| *GET | MQGET을 호출하여 큐에서 메시지를 검색합니다. |
| *INQ | MQINQ를 호출하여 특정 큐에 대한 조회를 작성합니다. |
| *PUB | MQPUT 호출을 사용하여 메시지를 발행하기 위해 토픽을 엽니다. |
| *PUT | MQPUT을 호출하여 특정 큐에 메시지를 넣습니다. |
| *RESUME | MQSUB 호출을 사용하여 구독을 재개합니다. |
| *SET | MQSET을 호출하여 MQI의 큐에서 속성을 설정합니다. 다중 옵션에 대해 큐를 여는 경우, 각 큐에 대해 권한이 있어야 합니다. |
| *SUB | MQSUB 호출을 사용하여 토픽에 대한 구독을 작성, 대체 또는 재개합니다. |

| 표 *16.* 컨텍스트 호출에 대한 권한 부여 | |
|---|---|
| **AUT** | **설명** |
| *PASSALL | 지정한 큐의 모든 컨텍스트를 전달합니다. 모든 컨텍스트 필드가 원래 요청으로부터 복사됩니다. |
| *PASSID | 지정된 큐의 ID 컨텍스트를 전달합니다. ID 컨텍스트는 요청의 컨텍스트와 동일합니다. |
| *SETALL | 지정된 큐에 모든 컨텍스트를 설정합니다. 이것은 특수 시스템 유틸리티에서 사용합니다. |
| *SETID | 지정된 큐에 ID 컨텍스트를 설정합니다. 이것은 특수 시스템 유틸리티에서 사용합니다. |

| 표 *17. MQSC* 및 *PCF* 호출에 대한 권한 부여 | |
|---|---|
| **AUT** | **설명** |
| *ADMCHG | 지정된 오브젝트의 속성을 변경합니다. |
| *ADMCLR | 지정된 오브젝트(PCF Clear 오브젝트 명령에만 해당)를 지웁니다. |
| *ADMCRT | 지정된 유형의 오브젝트를 작성합니다. |
| *ADMDLT | 지정된 오브젝트를 삭제합니다. |
| *ADMDSP | 지정된 오브젝트의 속성을 표시합니다. |

| 표 *18.* 일반 조작에 대한 권한 | |
|---|---|
| **AUT** | **설명** |
| *ALL | 오브젝트에 적용할 수 있는 모든 조작을 사용합니다. all 권한은 오브젝트 유형에 적합한 권한 alladm, allmqi 및 system의 결합과 동등합니다. |
| *ALLADM | 오브젝트에 대해 적용할 수 있는 모든 관리 조작을 수행합니다. |
| *ALLMQI | 오브젝트에 적용할 수 있는 모든 MQI 호출을 사용합니다. |
| *CTRL | 채널, 리스너, 서비스의 시동 및 종료를 제어합니다. |
| *CTRLX | 순서 번호를 재설정하고 인다우트 채널을 해석합니다. |

### IBM i *IBM i*에서 액세스 권한 부여 명령 사용

이 정보를 사용하여 액세스 권한 부여 명령에 대해 학습하고 명령 예제를 사용하십시오.

## GRTMQMAUT 명령 사용

권한이 필요한 경우, GRTMQMAUT 명령을 사용하여 특정 오브젝트에 액세스하는 사용자 프로파일 또는 사용자 그룹에 권한을 부여할 수 있습니다. 다음 예제는 GRTMQMAUT 명령이 사용되는 방법을 설명합니다.

1. ```
   GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
             AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
   ```

   이 예제에서

   - RED.LOCAL.QUEUE는 오브젝트 이름입니다.
   - *LCLQ(로컬 큐)는 오브젝트 유형입니다.
   - GROUPA는 권한이 변경되는 시스템의 사용자 프로파일 이름입니다. 이 프로파일은 다른 사용자의 그룹 프로파일로 사용 가능합니다.
   - *BROWSE 및 *PUT는 지정한 큐에 부여 중인 권한입니다.
   ```

*BROWSE는 큐에서 메시지를 찾기 위한 권한을 추가합니다(browse 옵션을 포함하여 MQGET 발행).

*PUT은 큐에 메시지를 넣는(MQPUT) 권한을 추가합니다.

- saturn.queue.manager는 큐 관리자 이름입니다.

2. 다음 명령은 사용자 JACK 및 JILL에게 기본 큐 관리자에 대해 모든 프로세스 정의에 적용 가능한 모든 권한을 부여합니다.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```

3. 다음 명령은 사용자 GEORGE에게 큐 관리자 TRENT에서 큐 ORDERS에 메시지를 넣는 권한을 부여합니다.

```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

### RVKMQMAUT 명령 사용

권한이 필요한 경우 RVKMQMAUT 명령을 사용하여 특정 오브젝트에 액세스하는 사용자 프로파일 또는 사용자 그룹에 대해 이전에 부여된 권한을 제거할 수 있습니다. 다음 예제는 RVKMQMAUT 명령이 사용되는 방법을 설명합니다.

1.
```
RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
AUT(*PUT) MQMNAME('saturn.queue.manager')
```

   이전 예제에서 부여된 지정한 큐에 메시지를 넣는 권한이 GROUPA에 대해 제거됩니다.

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```

   큐 관리자 PAYROLLQM이 소유하고 문자 PAY로 시작되는 이름의 모든 큐에서 메시지를 가져오는 권한이 시스템의 모든 사용자에서 제거되지만 해당 사용자 또는 해당 사용자가 속하는 그룹이 별도로 권한 부여된 경우는 제외됩니다.

### DSPMQMAUT 명령 사용

MQM 권한 표시(DSPMQMAUT) 명령은 지정한 오브젝트 및 사용자에 대해 사용자가 오브젝트에 대해 가지는 권한 목록을 표시합니다. 다음 예제는 명령이 사용되는 방법에 대해 설명합니다.

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

### RFRMQMAUT 명령 사용

MQM 보안 새로 고치기(RFRMQMAUT) 명령을 사용하면 큐 관리자를 중지하고 재시작하지 않고도 운영 체제 레벨에서 수행된 변경사항을 반영하여 OAM 권한 그룹 정보를 즉시 업데이트할 수 있습니다. 다음 예제는 명령이 사용되는 방법에 대해 설명합니다.

```
RFRMQMAUT MQMNAME(ADMINQM)
```

## IBM i IBM i에서의 권한 스펙 표

이 정보를 사용하여 특정 API 호출, 해당 호출의 특정 옵션을 큐 오브젝트, 프로세스 오브젝트, 큐 관리자 오브젝트에서 사용하기 위해 필요한 권한을 판별합니다.

156 페이지의 표 19에서 시작되는 권한 스펙 표는 권한이 작동하는 방법 및 적용되는 제한사항을 정확하게 정의합니다. 표는 다음과 같은 상황에 적용됩니다.

- MQI 호출을 발행한 애플리케이션
- 이스케이프 PCF로 MQSC 명령을 발행하는 관리 프로그램
- PCF 명령을 발행하는 관리 프로그램

이 절에서 정보는 다음 데이터를 지정하는 표의 세트로 표시됩니다.

**수행할 조치**
MQI 옵션, MQSC 명령 또는 PCF 명령

**액세스 제어 오브젝트**
큐, 프로세스 정의, 큐 관리자, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 또는 인증 정보 오브젝트

**필수 권한**
MQZAO_ 상수로 표현됩니다.

표에서 접두부가 MQZAO_인 상수는 특정 엔티티를 위한 **GRTMQMAUT** 및 **RVKMQMAUT** 명령에 대한 권한 목록의 키워드에 대응합니다. 예를 들어, MQZAO_BROWSE는 키워드 *BROWSE에 대응하고 유사하게 키워드 MQZAO_SET_ALL_CONTEXT는 키워드 *SETALL 등에 대응합니다. 이 상수는 제품에서 제공하는 헤더 파일 cmqzc.h에 정의됩니다.

## MQI 권한

실행 중인(또는 권한으로 간주 가능한) 사용자 ID에 관련 권한이 부여된 경우에만 애플리케이션은 특정 MQI 호출 및 옵션을 발행하도록 허용됩니다.

MQCONN, MQOPEN, MQPUT1, MQCLOSE의 네 MQI 호출에 권한 검사가 필요합니다.

MQOPEN 및 MQPUT1의 경우, 권한 검사가 여는 중인 오브젝트의 이름에 대해 수행되며, 이름이 해석된 후에 발생하는 이름에 대해서는 수행되지 않습니다. 예를 들어, 애플리케이션에는 알리어스가 해석되는 기본 큐를 열 수 있는 권한은 부여되지 않고 알리어스 큐를 열 수 있는 권한이 부여됩니다. 큐 관리자 알리어스 정의가 직접 열리는 경우(즉, 해당 이름이 오브젝트 디스크립터의 *ObjectName* 필드에 표시)를 제외하고 큐 관리자 알리어스가 아닌 이름을 해석하는 프로세스 중에 발생하는 첫 번째 정의에서 검사가 수행되는 것이 규칙입니다. 권한은 열리는 특정 오브젝트에 대해 항상 필요하며, 큐 관리자 오브젝트에 대한 권한 부여를 통해 확보되는 추가 큐 독립 권한이 필요한 경우도 있습니다.

156 페이지의 표 19, 156 페이지의 표 20, 157 페이지의 표 21, 158 페이지의 표 22에는 각 호출에 필요한 권한이 요약되어 있습니다.

**참고:** 이 표는 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 또는 인증 정보 오브젝트는 언급하지 않습니다. 이는 동일한 권한이 다른 오브젝트에도 적용되는 MQOO_INQUIRE를 제외하고 이러한 오브젝트에 적용되는 권한이 없기 때문입니다.

표 *19. MQCONN* 호출에 필요한 보안 권한

| **다음에 대한 권한 필요** | **큐 오브젝트(158 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
|---|---|---|---|
| MQCONN 옵션 | 적용할 수 없음 | 적용할 수 없음 | MQZAO_CONNECT |

표 *20. MQOPEN* 호출에 대해 필요한 보안 권한

| **다음에 대한 권한 필요** | **큐 오브젝트(158 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
|---|---|---|---|
| MQOO_INQUIRE | MQZAO_INQUIRE **(158 페이지의 『2』)** | MQZAO_INQUIRE **(158 페이지의 『2』)** | MQZAO_INQUIRE **(158 페이지의 『2』)** |
| MQOO_BROWSE | MQZAO_BROWSE | 적용할 수 없음 | 검사 안함 |
| MQOO_INPUT_* | MQZAO_INPUT | 적용할 수 없음 | 검사 안함 |

표 *20. MQOPEN* 호출에 대해 필요한 보안 권한 *(계속)*

| 다음에 대한 권한 필요 | 큐 오브젝트(158 페이지의 『1』) | 프로세스 오브젝트 | 큐 관리자 오브젝트 |
|---|---|---|---|
| MQOO_SAVE_ALL_CONTEXT (158 페이지의 『3』) | MQZAO_INPUT | 적용할 수 없음 | 적용할 수 없음 |
| MQOO_OUTPUT(정상 큐) (158 페이지의 『4』) | MQZAO_OUTPUT | 적용할 수 없음 | 적용할 수 없음 |
| MQOO_PASS_IDENTITY_CONTEXT (158 페이지의 『5』) | MQZAO_PASS_IDENTITY_CONTEXT | 적용할 수 없음 | 검사 안함 |
| MQOO_PASS_ALL_CONTEXT (158 페이지의 『5』, 158 페이지의 『6』) | MQZAO_PASS_ALL_CONTEXT | 적용할 수 없음 | 검사 안함 |
| MQOO_SET_IDENTITY_CONTEXT (158 페이지의 『5』, 158 페이지의 『6』) | MQZAO_SET_IDENTITY_CONTEXT | 적용할 수 없음 | MQZAO_SET_IDENTITY_CONTEXT (158 페이지의 『7』) |
| MQOO_SET_ALL_CONTEXT (158 페이지의 『5』, 158 페이지의 『8』) | MQZAO_SET_ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT (158 페이지의 『7』) |
| MQOO_OUTPUT(전송 큐) (158 페이지의 『9』) | MQZAO_SET_ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT (158 페이지의 『7』) |
| MQOO_SET | MQZAO_SET | 적용할 수 없음 | 검사 안함 |
| MQOO_ALTERNATE_USER_AUTHORITY | ( 158 페이지의 『10』 ) | ( 158 페이지의 『10』 ) | MQZAO_ALTERNATE_USER_AUTHORITY (158 페이지의 『10』, 158 페이지의 『11』) |

표 *21. MQPUT1* 호출에 필요한 보안 권한

| 다음에 대한 권한 필요 | 큐 오브젝트(158 페이지의 『1』) | 프로세스 오브젝트 | 큐 관리자 오브젝트 |
|---|---|---|---|
| MQPMO_PASS_IDENTITY_CONTEXT | MQZAO_PASS_IDENTITY_CONTEXT (158 페이지의 『12』) | 적용할 수 없음 | 검사 안함 |
| MQPMO_PASS_ALL_CONTEXT | MQZAO_PASS_ALL_CONTEXT (158 페이지의 『12』) | 적용할 수 없음 | 검사 안함 |
| MQPMO_SET_IDENTITY_CONTEXT | MQZAO_SET_IDENTITY_CONTEXT (158 페이지의 『12』) | 적용할 수 없음 | MQZAO_SET_IDENTITY_CONTEXT (158 페이지의 『7』) |
| MQPMO_SET_ALL_CONTEXT | MQZAO_SET_ALL_CONTEXT (158 페이지의 『12』) | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT (158 페이지의 『7』) |

| 표 *21. MQPUT1* 호출에 필요한 보안 권한 *(계속)* | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(158 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| (전송 큐) **(158 페이지의 『9』)** | MQZAO_SET_ ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ ALL_CONTEXT **(158 페이지의 『7』)** |
| MQPMO_ALTERNATE_ USER_AUTHORITY | **( 158 페이지의 『13』 )** | 적용할 수 없음 | MQZAO_ALTERNATE_ USER_AUTHORITY **(158 페이지의 『11』)** |

| 표 *22. MQCLOSE* 호출에 대해 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(158 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQCO_DELETE | MQZAO_DELETE **(158 페이지의 『14』)** | 적용할 수 없음 | 적용할 수 없음 |
| MQCO_DELETE _PURGE | MQZAO_DELETE **(158 페이지의 『14』)** | 적용할 수 없음 | 적용할 수 없음 |

**표에 대한 참고:**

1. 모델 큐를 열고 있는 경우:
   - 열기 액세스 유형의 경우 모델 큐를 열 수 있는 권한 외에 MQZAO_DISPLAY 권한이 모델 큐에 필요합니다.
   - 동적 큐 작성에 MQZAO_CREATE 권한은 필요하지 않습니다.
   - 모델 큐를 여는 데 사용되는 사용자 ID는 모든 큐 특정 권한을 작성된 동적 큐에 대해 자동 부여합니다 (MQZAO_ALL에 해당).
2. 큐, 프로세스, 이름 목록 또는 큐 관리자 오브젝트가 열고 있는 오브젝트 유형에 따라 검사됩니다.
3. MQOO_INPUT_*도 지정해야 합니다. 이 옵션은 로컬, 모델 또는 알리어스 큐에 대해 유효합니다.
4. 이 검사는 158 페이지의 『9』 참고에서 지정하는 경우를 제외하고는 모든 출력의 경우에 대해 수행됩니다.
5. MQOO_OUTPUT도 지정해야 합니다.
6. 이 옵션으로 MQOO_PASS_IDENTITY_CONTEXT도 포함해야 합니다.
7. 이 권한은 큐 관리자 오브젝트와 특정 큐 모두에 필요합니다.
8. 이 옵션으로 MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT, MQOO_SET_IDENTITY_CONTEXT도 포함해야 합니다.
9. 이 검사는 MQUS_TRANSMISSION의 *Usage* 큐 속성을 갖고 있고 출력을 위해 직접 열려 있는 로컬 또는 모델 큐에 대해 수행됩니다. 리모트 큐가 열려 있는 경우(리모트 큐 관리자 이름과 리모트 큐 이름을 지정하거나 리모트 큐의 로컬 정의 이름을 지정하여) 적용되지 않습니다.
10. MQOO_INQUIRE(모든 오브젝트 유형에 대해) 또는 (큐에 대해) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT 또는 MQOO_SET 중 하나 이상을 지정해야 합니다. 검사는 특정 이름의 오브젝트 권한에 대해 제공되는 대체 사용자 ID 및 MQZAO_ALTERNATE_USER_IDENTIFIER 검사에 대한 현재 애플리케이션 권한을 사용하여 지정된 다른 옵션에 대해 수행됩니다.
11. 이 권한은 모든 *AlternateUserId* 지정을 허용합니다.
12. 큐에 MQUS_TRANSMISSION의 *Usage* 큐 속성이 없는 경우, MQZAO_OUTPUT 검사가 수행됩니다.
13. 검사는 이름 지정된 큐 권한에 대해 제공되는 대체 사용자 ID 및 MQZAO_ALTERNATE_USER_IDENTIFIER 검사에 대한 현재 애플리케이션 권한을 사용하여 지정된 다른 옵션에 대해 수행됩니다.
14. 검사는 다음 두 가지 사항에 모두 해당하는 경우에만 수행됩니다.
    - 영구적 동적 큐가 닫히고 삭제 중인 경우.
    - 사용되고 있는 오브젝트 핸들을 리턴한 MQOPEN에 의해 큐가 작성되지 않은 경우.

그렇지 않으며 검사가 수행되지 않습니다.

**일반 참고:**

1. 특수 권한 MQZAO_ALL_MQI에는 오브젝트 유형에 연관되는 다음 모든 권한이 포함됩니다.
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_SET
   - MQZAO_BROWSE
   - MQZAO_INPUT
   - MQZAO_OUTPUT
   - MQZAO_PASS_IDENTITY_CONTEXT
   - MQZAO_PASS_ALL_CONTEXT
   - MQZAO_SET_IDENTITY_CONTEXT
   - MQZAO_SET_ALL_CONTEXT
   - MQZAO_ALTERNATE_USER_AUTHORITY
2. MQZAO_DELETE(158 페이지의 『14』 참고 참조) 및 MQZAO_DISPLAY는 관리 권한으로 분류됩니다. 따라서 이는 MQZAO_ALL_MQI에 포함되지 않습니다.
3. 검사 안함은 권한 검사가 수행되지 않는 것을 나타냅니다.
4. 적용할 수 없음은 이 조작에 권한 검사가 연관되지 않음을 나타냅니다. 예를 들어, MQPUT 호출을 발행하여 오브젝트를 처리할 수 없습니다.

### IBM i *IBM i*에서 이스케이프 *PCF*의 *MQSC* 명령에 대한 권한

이 권한을 사용하여 관리 명령을 이스케이프 PCF 메시지로 발행할 수 있습니다. 이 메소드는 해당 사용자를 대신하여 실행하기 위해 프로그램이 관리 명령을 큐 관리자에 메시지로 송신하도록 허용합니다.

이 절은 이스케이프 PCF에 포함된 각 MQSC 명령에 필요한 권한을 요약합니다.

적용할 수 없음은 이 조작에 권한 검사가 연관되지 않음을 나타냅니다.

명령을 제출하는 프로그램이 실행 중인 사용자 ID는 다음 권한도 있어야 합니다.

- 큐 관리자에 대한 MQZAO_CONNECT 권한
- PCF 명령을 수행하기 위한 큐 관리자의 DISPLAY 권한
- 이스케이프 PCF 명령 텍스트 내에서 MQSC 명령을 발행하는 권한

**ALTER *object***

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | MQZAO_CHANGE |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |

**CLEAR *object***

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CLEAR |
| 토픽 | MQZAO_CLEAR |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | 적용할 수 없음 |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**DEFINE *object* NOREPLACE (163 페이지의 『1』)**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CREATE (163 페이지의 『2』) |
| 토픽 | MQZAO_CREATE (163 페이지의 『2』) |
| 프로세스 | MQZAO_CREATE (163 페이지의 『2』) |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CREATE (163 페이지의 『2』) |
| 인증 정보 | MQZAO_CREATE (163 페이지의 『2』) |
| 채널 | MQZAO_CREATE (163 페이지의 『2』) |
| 클라이언트 연결 채널 | MQZAO_CREATE (163 페이지의 『2』) |
| 리스너 | MQZAO_CREATE (163 페이지의 『2』) |
| 서비스 | MQZAO_CREATE (163 페이지의 『2』) |

**DEFINE *object* REPLACE (163 페이지의 『1』, 163 페이지의 『3』)**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 서비스 | MQZAO_CHANGE |

**DELETE *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DELETE |
| 토픽 | MQZAO_DELETE |
| 프로세스 | MQZAO_DELETE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_DELETE |
| 인증 정보 | MQZAO_DELETE |
| 채널 | MQZAO_DELETE |
| 클라이언트 연결 채널 | MQZAO_DELETE |
| 리스너 | MQZAO_DELETE |
| 서비스 | MQZAO_DELETE |

**DISPLAY *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DISPLAY |
| 토픽 | MQZAO_DISPLAY |
| 프로세스 | MQZAO_DISPLAY |
| 큐 관리자 | MQZAO_DISPLAY |
| 이름 목록 | MQZAO_DISPLAY |
| 인증 정보 | MQZAO_DISPLAY |
| 채널 | MQZAO_DISPLAY |
| 클라이언트 연결 채널 | MQZAO_DISPLAY |
| 리스너 | |
| 서비스 | |

**PING CHANNEL**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**RESET CHANNEL**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL_EXTENDED |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**RESOLVE CHANNEL**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL_EXTENDED |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**START *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |

| 오브젝트 | 필수 권한 |
|---|---|
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | MQZAO_CONTROL |
| 서비스 | MQZAO_CONTROL |

**STOP *object***

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | MQZAO_CONTROL |
| 서비스 | MQZAO_CONTROL |

**참고:**

1. DEFINE 명령의 경우, LIKE 오브젝트(지정된 경우)에 대한 MQZAO_DISPLAY 권한도 필요하며, LIKE가 생략된 경우 적절한 SYSTEM.DEFAULT.xxx 오브젝트에 대한 권한이 필요합니다.
2. MQZAO_CREATE 권한은 특정 오브젝트나 오브젝트 유형에 특정되지 않습니다. 작성 권한은 GRTMQMAUT 명령에 오브젝트 유형 QMGR을 지정하여 지정된 큐 관리자의 모든 오브젝트에 대해 부여됩니다.
3. 이 옵션은 바꿀 오브젝트가 이미 있는 경우에만 적용됩니다. 존재하지 않는 경우, 검사는 DEFINE *object* NOREPLACE로 수행됩니다.

### IBM i *IBM i에서 PCF 명령의 권한 부여*

이 권한 부여를 통해 사용자는 관리 명령을 PCF 명령으로 발행할 수 있습니다. 이 메소드는 해당 사용자를 대신하여 실행하기 위해 프로그램이 관리 명령을 큐 관리자에 메시지로 송신하도록 허용합니다.

이 절에서는 각 PCF 명령에 필요한 권한을 요약합니다.

검사 안함은 권한 검사가 수행되지 않는 것을 나타내며 적용할 수 없음은 권한 검사가 이 조작에 연관되지 않음을 나타냅니다.

명령을 제출하는 프로그램이 실행 중인 사용자 ID는 다음 권한도 있어야 합니다.

- 큐 관리자에 대한 MQZAO_CONNECT 권한
- PCF 명령을 수행하기 위한 큐 관리자의 DISPLAY 권한

특수 권한 MQZAO_ALL_ADMIN에는 다음 권한이 포함됩니다.

- MQZAO_CHANGE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY

- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

MQZAO_CREATE는 특정 오브젝트 또는 오브젝트 유형에 특정되지 않기 때문에 포함되지 않습니다.

***object* 변경**

| **오브젝트** | **필수 권한** |
|---|---|
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | MQZAO_CHANGE |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |

***object* 지우기**

| **오브젝트** | **필수 권한** |
|---|---|
| 큐 | MQZAO_CLEAR |
| 토픽 | MQZAO_CLEAR |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | 적용할 수 없음 |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**오브젝트 복사(대체 제외) (169 페이지의 『1』)**

| **오브젝트** | **필수 권한** |
|---|---|
| 큐 | MQZAO_CREATE (**169 페이지의 『2』**) |
| 토픽 | MQZAO_CREATE (**169 페이지의 『2』**) |
| 프로세스 | MQZAO_CREATE (**169 페이지의 『2』**) |
| 큐 관리자 | 적용할 수 없음 |
| NamelistMQZAO_CREATE | MQZAO_CREATE (**169 페이지의 『2』**) |
| 인증 정보 | MQZAO_CREATE (**169 페이지의 『2』**) |
| 채널 | MQZAO_CREATE (**169 페이지의 『2』**) |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 클라이언트 연결 채널 | MQZAO_CREATE (169 페이지의 『2』) |
| 리스너 | MQZAO_CREATE (169 페이지의 『2』) |
| 서비스 | MQZAO_CREATE (169 페이지의 『2』) |

**Copy *object*(바꾸기 포함)( 169 페이지의 『1』, 169 페이지의 『4』 )**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CHANGE |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |

**오브젝트 작성(대체 제외) (169 페이지의 『3』)**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CREATE (169 페이지의 『2』) |
| 토픽 | MQZAO_CREATE (169 페이지의 『2』) |
| 프로세스 | MQZAO_CREATE (169 페이지의 『2』) |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CREATE (169 페이지의 『2』) |
| 인증 정보 | MQZAO_CREATE (169 페이지의 『2』) |
| 채널 | MQZAO_CREATE (169 페이지의 『2』) |
| 클라이언트 연결 채널 | MQZAO_CREATE (169 페이지의 『2』) |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |

**Create *object*(바꾸기 포함)( 169 페이지의 『3』, 169 페이지의 『4』 )**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CHANGE |
| 토픽 | MQZAO_CHANGE |
| 프로세스 | MQZAO_CHANGE |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CHANGE |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 인증 정보 | MQZAO_CHANGE |
| 채널 | MQZAO_CHANGE |
| 클라이언트 연결 채널 | MQZAO_CHANGE |
| 리스너 | MQZAO_CHANGE |
| 서비스 | MQZAO_CHANGE |

***object* 삭제**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DELETE |
| 토픽 | MQZAO_DELETE |
| 프로세스 | MQZAO_DELETE |
| 큐 관리자 | MQZAO_DELETE |
| 이름 목록 | MQZAO_DELETE |
| 인증 정보 | MQZAO_DELETE |
| 채널 | MQZAO_DELETE |
| 클라이언트 연결 채널 | MQZAO_DELETE |
| 리스너 | MQZAO_DELETE |
| 서비스 | MQZAO_DELETE |

***object* 조회**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DISPLAY |
| 토픽 | MQZAO_DISPLAY |
| 프로세스 | MQZAO_DISPLAY |
| 큐 관리자 | MQZAO_DISPLAY |
| 이름 목록 | MQZAO_DISPLAY |
| 인증 정보 | MQZAO_DISPLAY |
| 채널 | MQZAO_DISPLAY |
| 클라이언트 연결 채널 | MQZAO_DISPLAY |
| 리스너 | MQZAO_DISPLAY |
| 서비스 | MQZAO_DISPLAY |

***object* 이름 조회**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 검사 안함 |
| 토픽 | 검사 안함 |
| 프로세스 | 검사 안함 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 관리자 | 검사 안함 |
| 이름 목록 | 검사 안함 |
| 인증 정보 | 검사 안함 |
| 채널 | 검사 안함 |
| 클라이언트 연결 채널 | 검사 안함 |
| 리스너 | 검사 안함 |
| 서비스 | 검사 안함 |

**채널 ping**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**채널 재설정**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL_EXTENDED |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**큐 통계 재설정**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DISPLAY and MQZAO_CHANGE |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | 적용할 수 없음 |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | |
| 서비스 | |

**채널 분석**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL_EXTENDED |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**채널 시작**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**채널 중지**

| 오브젝트 | 필수 권한 |
|---|---|
| 큐 | 적용할 수 없음 |
| 토픽 | 적용할 수 없음 |
| 프로세스 | 적용할 수 없음 |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | 적용할 수 없음 |
| 인증 정보 | 적용할 수 없음 |
| 채널 | MQZAO_CONTROL |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |

**참고:**

1. 복사 명령의 경우, From 오브젝트에 대한 MQZAO_DISPLAY 권한도 필요합니다.
2. MQZAO_CREATE 권한은 특정 오브젝트나 오브젝트 유형에 특정되지 않습니다. 작성 권한은 GRTMQMAUT 명령에 오브젝트 유형 QMGR을 지정하여 지정된 큐 관리자의 모든 오브젝트에 대해 부여됩니다.
3. Create 명령의 경우 적절한 SYSTEM.DEFAULT.* 오브젝트.
4. 이 옵션은 바꿀 오브젝트가 이미 있는 경우에만 적용됩니다. 없는 경우, 검사는 복사 또는 작성(바꾸기 없음)에 대해 수행됩니다.

## IBM i IBM i의 일반 OAM 프로파일

오브젝트 권한 관리자(OAM) 일반 프로파일을 사용하면 각 오브젝트가 작성될 때마다 개별적으로 별도의 **GRTMQMAUT** 명령을 발행하지 않고 사용자가 여러 오브젝트에 가지는 권한을 한 번에 설정할 수 있습니다. **GRTMQMAUT** 명령에서 일반 프로파일을 사용하면 해당 프로파일에 맞게 나중에 작성되는 모든 오브젝트에 대해 일반 권한을 설정할 수 있습니다.

이 절의 나머지에서는 일반 프로파일 사용에 대해 좀 더 자세히 설명합니다.


- 169 페이지의 『와일드카드 문자 사용』
- 170 페이지의 『프로파일 우선순위』


### 와일드카드 문자 사용

프로파일을 일반으로 작성하는 방법은 프로파일 이름에 특수 문자(와일드카드 문자)를 사용하는 것입니다. 예를 들어 물음표(?) 와일드카드 문자는 이름에서 단일 문자를 일치시킵니다. 따라서, ABC.?EF를 지정하면 해당 프로파일에 부여한 권한이 ABC.DEF, ABC.CEF, ABC.BEF 이름 등으로 작성되는 모든 오브젝트에 적용됩니다.

사용 가능한 와일드카드 문자는 다음과 같습니다.

**?**

단일 문자 대신 물음표(?) 사용. 예를 들어, AB.?D는 오브젝트 AB.CD, AB.ED, AB.FD에 적용됩니다.

*****

별표(*)를 다음과 같이 사용하십시오.

- 오브젝트 이름에 있는 규정자와 일치하는 프로파일 이름의 규정자. 규정자는 마침표로 구분되는 오브젝트 이름의 한 부분입니다. 예를 들어, ABC.DEF.GHI에서 규정자는 ABC, DEF 및 GHI입니다.

  예를 들어, ABC.*.JKL은 오브젝트 ABC.DEF.JKL 및 ABC.GHI.JKL에 적용됩니다. (이는 ABC.JKL에 적용되지 **않습니다**. 이 컨텍스트에 사용되는 *는 항상 하나의 규정자를 표시합니다.)

- 오브젝트 이름에 있는 규정자 내에서 0개 또는 그 이상의 문자와 일치하는 프로파일 이름의 규정자 문자.

  예를 들어, ABC.DE*.JKL는 오브젝트 ABC.DE.JKL, ABC.DEF.JKL, ABC.DEGH.JKL에 적용됩니다.

**

다음과 같이 프로파일 이름에서는 이중 별표(**)를 **한 번만** 사용하십시오.

- 모든 오브젝트 이름과 일치하는 전체 프로파일 이름. 예를 들어 OBJTYPE（*PRC）키워드를 사용하여 프로세스를 식별하는 경우, 프로파일 이름으로 **를 사용하면 모든 프로세스의 권한을 변경할 수 있습니다.
- 오브젝트 이름에 있는 0개 또는 그 이상의 규정자와 일치하는 프로파일 이름의 시작, 중간 또는 종료 규정자로서 사용됩니다. 예를 들어, **.ABC는 모든 오브젝트를 최종 규정자 ABC로 식별합니다.

## 프로파일 우선순위

일반 프로파일을 사용할 경우 이해해야 하는 중요한 점은 작성 중인 오브젝트에 적용할 권한을 결정할 때 프로파일에 지정되는 우선순위입니다. 예를 들어, 다음 명령을 발행한 것으로 가정해 보십시오.

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

첫 번째는 이름이 AB. *; 프로파일과 일치하는 프린시펄 FRED 에 대한 모든 큐에 넣기 권한을 부여합니다. 두 번째는 AB.C*

이제 AB.CD 큐를 작성한다고 가정합니다. 와일드카드 적용에 대한 규칙에 따라 GRTMQMAUT 중 하나가 해당 큐에 적용됩니다. 따라서, 넣기 또는 가져오기 권한이 있습니까?

응답을 찾기 위해 여러 프로파일을 오브젝트에 적용할 수 있을 때마다 **가장 특정되게 적용되는** 규칙만 적용합니다. 이 규칙을 적용하는 방법은 프로파일 이름을 왼쪽에서 오른쪽으로 비교하는 것입니다. 다를 경우, 일반 문자가 아닌 문자가 일반 문자보다 더 특정되는 문자입니다. 따라서, 이전 예에서 큐 AB.CD는 **가져오기** 권한을 가집니다(AB.C*는 AB.*보다 더 특정됩니다).

일반 문자를 비교할 때 특이성의 순서는 다음과 같습니다.

1. ?
2. *
3. **

## IBM i IBM i에 설치된 권한 서비스 지정

사용하려는 권한 서비스 컴포넌트를 지정할 수 있습니다.

**GRTMQMAUT** 및 **RVKMQMAUT** 의 **Service Component name** 매개변수를 사용하여 설치된 권한 서비스 컴포넌트의 이름을 지정할 수 있습니다.

초기 패널에서 **F24**를 선택하고 명령 중 하나의 다음 패널에서 **F9=모든 매개변수**를 선택하면 설치된 권한 부여 컴포넌트(*DFT) 또는 큐 관리자 qm.ini 파일의 서비스 스탠자에 지정된 필수 권한 서비스 컴포넌트 이름을 지정할 수 있습니다.

**DSPMQMAUT**도 이 추가 매개변수를 포함합니다. 이 매개변수를 사용하면 설치된 모든 권한 부여 컴포넌트(*DFT) 또는 지정된 권한 서비스 컴포넌트 이름을 지정한 오브젝트 이름, 오브젝트 유형, 사용자에 대해 검색할 수 있습니다.

## IBM i IBM i에서 권한 프로파일을 사용하거나 사용하지 않고 작업

이 정보를 사용하여 권한 프로파일에 대해 작업하는 방법 및 권한 프로파일 없이 작업하는 방법에 대해 학습합니다.

171 페이지의 『권한 프로파일에 대해 작업』 에서 설명하는 대로 권한 프로파일을 사용하여 작업하거나 여기에서 설명하는 대로 권한 프로파일을 사용하지 않고 작업할 수 있습니다.

권한 프로파일을 사용하지 않고 작업하려면 *NONE을 **GRTMQMAUT**에서 Authority 매개변수로 사용하여 권한없이 프로파일을 작성하십시오. 기존의 프로파일은 모두 변경되지 않고 유지됩니다.

**RVKMQMAUT**에서 Authority 매개변수로 *REMOVE를 사용하여 기존 권한 프로파일을 제거하십시오.

## 권한 프로파일에 대해 작업

권한 프로파일에는 두 개의 연관된 명령이 있습니다.

- **WRKMQMAUT**
- **WRKMQMAUTD**

해당 명령에는 명령행에서 직접 또는 WRKMQM 패널에서 다음 방법으로 액세스 가능합니다.

1. 큐 관리자 이름에 입력하고 Enter 키를 눌러서 **WRKMQM** 결과 패널에 액세스하십시오.
2. 이 패널에서 F23=More options 선택하십시오.

옵션 24는 **WRKMQMAUT** 명령 에 대한 결과 패널을 선택하고 옵션 25는 SSL 바인딩 계층과 함께 사용되는 **WRKMQMAUTI** 명령을 선택합니다.

## WRKMQMAUT

이 명령을 사용하여 권한 큐에 보유된 권한 데이터에 대해 작업할 수 있습니다.

**참고:** 이 명령을 실행하려면 큐 관리자에 대해 *connect 및 *admdsp 권한이 있어야 합니다. 그렇지만 프로파일을 작성하거나 삭제하려면 QMQMADM 권한이 필요합니다.

정보를 화면으로 출력하는 경우 권한 프로파일 이름 목록이 해당 유형과 같이 표시됩니다. 출력을 인쇄하는 경우 모든 권한 데이터, 등록된 사용자, 해당 권한에 대한 자세한 목록을 수신합니다.

이 패널에서 오브젝트 또는 프로파일 이름을 입력하고 ENTER를 누르면 **WRKMQMAUT**에 대한 결과 패널이 표시됩니다.

4=Delete를 선택하면 지정하는 일반 권한 프로파일 이름에 등록되는 모든 사용자 이름 삭제를 확인할 수 있는 새 패널로 이동합니다. 이 옵션은 모든 사용자에 대해 *REMOVE 옵션과 같이 **RVKMQMAUT**를 실행하고 일반 프로파일 이름에**만** 적용됩니다.

12=Work with profile을 선택하면 171 페이지의 『WRKMQMAUTD』 에서 설명하는 대로 **WRKMQMAUTD** 명령 결과 패널로 이동합니다.

## WRKMQMAUTD

이 명령을 사용하면 특정 권한 프로파일 이름 및 오브젝트 유형에 등록되는 모든 사용자를 표시할 수 있습니다. 이 명령을 실행하려면 큐 관리자에 대해 *connect 및 *admdsp 권한이 있어야 합니다. 그렇지만 프로파일에 권한 부여, 실행, 작성 또는 삭제하려면 QMQMADM 권한이 필요합니다.

초기 입력 패널에서 F24=More keys 를 선택한 후 F9=All Parameters 옵션을 선택하면 **GRTMQMAUT** 및 **RVKMQMAUT**의 경우와 같이 서비스 컴포넌트 이름이 표시됩니다.

**참고:** F11=Display Object Authorizations 키는 다음 유형의 권한 사이에서 토글됩니다.

- 오브젝트 권한
- 컨텍스트 권한
- MQI 권한

화면의 옵션은 다음과 같습니다.

**2=Grant**
현재 권한을 추가하는 **GRTMQMAUT** 패널로 이동합니다.

**3=Revoke**
일부 현재 정의를 제거하는 **RVKMQMAUT** 패널로 이동합니다.

**4=Delete**
지정한 사용자에 대한 권한 데이터를 삭제할 수 있는 패널로 이동합니다. 이는 *REMOVE 옵션과 같이 **RVKMQMAUT**를 실행합니다.

**5=Display**
기존 **DSPMQMAUT** 명령으로 이동합니다.

**F6=Create**
프로파일 권한 레코드를 작성할 수 있는 **GRTMQMAUT** 패널로 이동합니다.

## IBM i IBM i에서 오브젝트 권한 관리자 지침

오브젝트 권한 관리자(OAM) 사용에 대한 추가 힌트 및 팁

### 민감한 조작에 대한 액세스 제한

일부 조작은 민감하며 권한이 있는 사용자로 제한됩니다. 예:

- 전송 큐와 같은 일부 특수 큐 또는 명령 큐 SYSTEM.ADMIN.COMMAND.QUEUE에 액세스
- 전체 MQI 컨텍스트 옵션을 사용하는 프로그램 실행
- 애플리케이션 큐 작성 및 복사

### 큐 관리자 디렉토리

큐 및 기타 큐 관리자 데이터를 포함하는 디렉토리 및 라이브러리는 제품에 대해 개인용입니다. MQI 자원에 대한 권한을 부여하거나 취소할 때 표준 운영 체제 명령을 사용하지 마십시오.

### 큐

동적 큐의 권한은 도출된 모델 큐를 기반으로 하지만 동일할 필요는 없습니다.

알리어스 큐 또는 리모트 큐의 경우, 권한은 알리어스 큐나 리모트 큐가 해결되는 큐가 아니라 오브젝트 자체에 대한 권한입니다. 사용자 프로파일에 액세스 권한이 없는 로컬 큐로 해결되는 알리어스 큐에 액세스하도록 사용자 프로파일에 권한을 부여할 수 있습니다.

큐 작성 권한을 권한이 있는 사용자로 제한하십시오. 그렇게 하지 않으면 사용자가 알리어스를 작성하여 일반 액세스 제어를 무시할 수 있습니다.

### 대체 사용자 권한

대체 사용자 권한은 한 개의 사용자 프로파일이 IBM MQ 오브젝트 액세스 시에 다른 사용자 프로파일의 권한을 사용할 수 있는지 여부를 제어합니다. 이 기술은 서버가 프로그램의 요청을 수신하고 서버가 프로그램에 요청에 필요한 권한이 있는지 확인하려는 경우에 필수입니다. 서버에는 필수 권한이 있을 수 있지만 프로그램이 요청한 조치에 대한 권한이 있는지 여부를 알아야 합니다.

예를 들면, 다음과 같습니다.

- 사용자 프로파일 PAYSERV에서 실행 중인 서버 프로그램은 사용자 프로파일 USER1으로 큐에 넣은 요청 메시지를 큐에서 검색합니다.
- 서버 프로그램이 요청 메시지를 가져오면 요청을 처리하고 요청 메시지에 지정된 응답 대상 큐로 응답을 다시 넣습니다.
- 응답 대상 큐 열기 권한을 부여하기 위해 자체 사용자 프로파일(PAYSERV)을 사용하는 대신 서버는 다른 사용자 프로파일(이 경우에는 USER1)을 지정할 수 있습니다. 이 예에서 대체 사용자 권한을 사용하여 응답 대상 큐를 열 때 PAYSERV가 대체 사용자 프로파일로 USER1을 지정할 수 있는지 여부를 제어할 수 있습니다.

대체 사용자 프로파일은 오브젝트 디스크립터의 *AlternateUserId* 필드에 지정됩니다.

**참고:** 모든 IBM MQ 오브젝트에서 대체 사용자 프로파일을 사용할 수 있습니다. 대체 사용자 프로파일 사용은 다른 자원 관리자에서 사용되는 사용자 프로파일에는 영향을 주지 않습니다.

### 컨텍스트 권한

컨텍스트는 특정 메시지에 적용되는 정보이며, 메시지의 일부인 메시지 설명자 MQMD에 포함되어 있습니다.

컨텍스트와 관련된 메시지 디스크립터 필드에 대한 설명은 MQMD-메시지 디스크립터를 참조하십시오.

컨텍스트 옵션에 대한 정보는 메시지 컨텍스트를 참조하십시오.

## 원격 보안 고려사항

원격 보안에 대해서는 다음을 고려하십시오.

**넣기 권한**
큐 관리자에서 보안을 위해 채널이 다른 큐 관리자에서 송신된 메시지를 수신할 때 사용되는 넣기 권한을 지정할 수 있습니다.

이 매개변수는 RCVR, RQSTR 또는 CLUSRCVR 채널 유형에 대해서만 유효합니다. 다음과 같인 채널 속성 PUTAUT를 지정하십시오.

**DEF**
기본 사용자 프로파일. 이는 메시지 채널 에이전트가 실행 중인 QMQM 사용자 프로파일입니다.

**CTX**
메시지 컨텍스트의 사용자 프로파일.

**전송 큐**
큐 관리자는 자동으로 원격 메시지를 전송 큐에 넣으며 이를 위해 특수 권한이 필요하지는 않습니다. 그렇지만 메시지를 전송 큐에 직접 넣으려면 특수한 권한이 필요합니다.

**채널 엑시트**
채널 엑시트는 추가된 보안에 사용할 수 있습니다.

**채널 인증 레코드**
채널 레벨에서 연결 시스템에 부여된 액세스 권한에 대한 보다 정밀한 제어를 실행하기 위해 사용합니다.

원격 보안에 대한 자세한 정보는 106 페이지의 『채널 권한 부여』의 내용을 참조하십시오.

## SSL/TLS을 사용하는 채널 보호

TLS(Transport Layer Security) 프로토콜은 도청, 도용, 위장에 대한 보호를 사용하여 채널 보안을 제공합니다. TLS에 대한 IBM MQ 지원을 사용하면 채널 정의에서 특정 채널이 TLS 보안을 사용하도록 지정할 수 있습니다. 또한, 사용하려는 암호화 알고리즘과 같은 필요한 보안 세부사항을 지정할 수도 있습니다.

IBM MQ에서의 TLS 지원은 큐 관리자 인증 정보 오브젝트와 다양한 CL 및 MQSC 명령, 필요한 TLS 지원 세부사항을 자세하게 정의하는 큐 관리자 및 채널 매개변수를 사용합니다.

다음 CL 명령이 TLS를 지원합니다.

**WRKMQMAUTI**
인증 정보 오브젝트 속성에 대해 작업합니다.

**CHGMQMAUTI**
인증 정보 오브젝트 속성을 수정합니다.

**CRTMQMAUTI**
인증 정보 오브젝트를 작성합니다.

**CPYMQMAUTI**
기존 인증 정보 오브젝트를 복사하여 이를 작성합니다.

**DLTMQMAUTI**
인증 정보 오브젝트를 삭제합니다.

**DSPMQMAUTI**
특정 인증 정보 오브젝트 속성을 표시합니다.

TLS를 사용하는 채널 보안 개요는 다음을 참조하십시오.

- TLS로 채널 보호

TLS와 연관된 PCF 명령에 대한 자세한 내용은 다음을 참조하십시오.

- 인증 정보 오브젝트 변경, 복사 및 작성
- 인증 정보 오브젝트 삭제
- 인증 정보 오브젝트 조회

# z/OS Setting up security on z/OS

Security considerations specific to z/OS.

Security in IBM MQ for z/OS is controlled using RACF or an equivalent external security manager (ESM).

The following instructions assume that you are using RACF.

**Related concepts**

Security scenario: two queue managers on z/OS

Security scenario: queue sharing group on z/OS

## z/OS RACF security classes

RACF classes are used to hold the profiles required for IBM MQ security checking. Many of the member classes have equivalent group classes. You must activate the classes and enable them to accept generic profiles.

Each RACF class holds one or more profiles used at some point in the checking sequence, as shown in Table 23 on page 174.

*Table 23. RACF classes used by IBM MQ*

| Member class | Group class | Contents |
|---|---|---|
| MQADMIN | GMQADMIN | Profiles that are used mainly for administrative functions. For example:<br>• Profiles for IBM MQ security switches.<br>• The RESLEVEL security profile.<br>• Profiles for alternate user security.<br>• Profiles for context security.<br>• Profiles for command resource security.<br>This class can hold only uppercase RACF profiles. |
| MXADMIN | GMXADMIN | Profiles that are used mainly for administrative functions. For example:<br>• Profiles for IBM MQ security switches.<br>• The RESLEVEL security profile.<br>• Profiles for alternate user security.<br>• Profiles for context security.<br>• Profiles for command resource security.<br>This class can hold both uppercase and mixed-case RACF profiles. |
| MQCONN | | Profiles used for connection security. |
| MQCMDS | | Profiles used for command security. |
| MQQUEUE | GMQQUEUE | Uppercase profiles used in queue resource security. |
| MXQUEUE | GMXQUEUE | Mixed-case and uppercase profiles used in queue resource security. |
| MQPROC | GMQPROC | Uppercase profiles used in process resource security. |
| MXPROC | GMXPROC | Mixed-case and uppercase profiles used in process resource security. |

| Table 23. RACF classes used by IBM MQ (continued) | | |
|---|---|---|
| **Member class** | **Group class** | **Contents** |
| MQNLIST | GMQNLIST | Uppercase profiles used in namelist resource security. |
| MXNLIST | GMXNLIST | Mixed-case and uppercase profiles used in namelist resource security. |
| MXTOPIC | GMXTOPIC | Mixed-case and uppercase profiles used in topic security. |

Some classes have a related *group class* that enables you to put together groups of resources that have similar access requirements. For details about the difference between the member and group classes and when to use a member or group class, see the z/OS Security Server RACF Security Administrator's Guide.

The classes must be activated before security checks can be made. To activate all the IBM MQ classes, you can use this RACF command:

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                  MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

You should also ensure that you set up the classes so that they can accept generic profiles. You also do this with the RACF command **SETROPTS**, for example:

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                 MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

## z/OS RACF profiles

All RACF profiles used by IBM MQ contain a prefix, which is either the queue manager name or the queue sharing group name. Be careful when you use the percent sign as a wildcard.

All RACF profiles used by IBM MQ contain a prefix. For queue sharing group level security, this is the queue sharing group name. For queue manager level security, the prefix is the queue manager name. If you are using a mixture of queue manager and queue sharing group level security, you will use profiles with both types of prefix. Queue sharing group and queue manager level security are described in Security controls and options in IBM MQ for z/OS.

For example, if you want to protect a queue called QUEUE_FOR_SUBSCRIBER_LIST in queue sharing group QSG1 at queue sharing group level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

If you want to protect a queue called QUEUE_FOR_LOST_CARD_LIST, that belongs to queue manager STCD at queue manager level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

This means that different queue managers and queue sharing groups can share the same RACF database and yet have different security options.

Do not use generic queue manager names in profiles to avoid unanticipated user access.

IBM MQ allows the use of the percent sign (%) in object names. However, RACF uses the % character as a single-character wildcard. This means that when you define an object name with a % character in its name, you must consider this when you define the corresponding profile.

For example, for the queue CREDIT_CARD_%_RATE_INQUIRY, on queue manager CRDP, the profile would be defined to RACF as follows:

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

This queue cannot be protected by a generic profile, such as, CRDP.**.

IBM MQ allows the use of mixed-case characters in object names. You can protect these objects by defining:

1. Mixed-case profiles in the appropriate mixed-case RACF classes, or
2. Generic profiles in the appropriate uppercase RACF classes.

To use mixed-case profiles and mixed-case RACF classes you must follow the steps described in "Migrating a z/OS queue manager to mixed-case security" on page 252.

There are some profiles, or parts of profiles, that remain uppercase only as the values are provided by IBM MQ. These are:

- Switch profiles.
- All high-level qualifiers (HLQ) including subsystem and queue sharing group identifiers.
- Profiles for SYSTEM objects.
- Profiles for Default objects.
- The **MQCMDS** class, so all command profiles are uppercase only.
- The **MQCONN** class, so all connection profiles are uppercase only.
- **RESLEVEL** profiles.
- The `'object'` qualification in command resource profiles; for example, `hlq.QUEUE.queuename`. The resource name only is mixed case.
- Dynamic queue profiles `hlq.CSQOREXX.*`, `hlq.CSQUTIL.*`, and `CSQXCMD.*`.
- The `'CONTEXT'` part of `hlq.CONTEXT.resourcename`.
- The `'ALTERNATE.USER'` part of `hlq.ALTERNATE.USER.userid`.

For example, you can define a profile to grant access to a queue called `PAYROLL.Dept1` on queue manager `QM01` in one of the following ways.

- If you are using mixed-case profiles, you can define a profile in the IBM MQ RACF class `MXQUEUE` using the following command:

  ```
  RDEFINE MXQUEUE MQ01.PAYROLL.Dept1
  ```

- If you are using uppercase profiles, you can define a profile in the IBM MQ RACF class `MQQUEUE` using the following command:

  ```
  RDEFINE MQQUEUE MQ01.PAYROLL.*
  ```

The first example, using mixed-case profiles, gives you more granular control over granting authority to access the resource.

## z/OS Switch profiles

To control the security checking performed by IBM MQ, you use *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to IBM MQ. The access list in switch profiles is not used by IBM MQ.

IBM MQ maintains an internal switch for each switch type shown in tables Switch profiles for subsystem level security, Switch profiles for queue sharing group or queue manager level security,and Switch profiles for resource checking. Switch profiles can be maintained at queue sharing group level, or at queue manager level, or at a combination of both. Using a single set of queue sharing group security switch profiles, you can control security on all the queue managers within a queue sharing group.

When a security switch is set on, the security checks associated with the switch are performed. When a security switch is set off, the security checks associated with the switch are bypassed. The default is that all security switches are set on.

### z/OS *Switches and classes*

When you start a queue manager or refresh security, IBM MQ sets switches according to the state of various RACF classes.

When a queue manager is started (or when the MQADMIN or MXADMIN class is refreshed by the IBM MQ REFRESH SECURITY command), IBM MQ first checks the status of RACF and the appropriate class:

- The MQADMIN class if you are using uppercase profiles
- The MXADMIN class if you are using mixed case profile.

It sets the subsystem security switch off if any of these conditions is true:

- RACF is inactive or not installed.
- The MQADMIN or MXADMIN class is not defined (these classes are always defined for RACF because they are included in the class descriptor table (CDT)).
- The MQADMIN or MXADMIN class has not been activated.

If both RACF and the MQADMIN or MXADMIN class are active, IBM MQ checks the MQADMIN or MXADMIN class to see whether any of the switch profiles have been defined. It first checks the profiles described in "Profiles to control subsystem security" on page 178. If subsystem security is not required, IBM MQ sets the internal subsystem security switch off, and performs no further checks.

The profiles determine whether the corresponding IBM MQ switch is set on or off.

- If the switch is off, that type of security is deactivated.
- If any IBM MQ switch is set on, IBM MQ checks the status of the RACF class associated with the type of security corresponding to the IBM MQ switch. If the class is not installed or not active, the IBM MQ switch is set off. For example, process security checks are not carried out if the MQPROC or MXPROC class has not been activated. The class not being active is equivalent to defining NO.PROCESS.CHECKS profile for every queue manager and queue sharing group that uses this RACF database.

### z/OS *How switches work*

To set off a security switch, define a NO.* switch profile for it. You can override a NO.* profile set at the queue sharing group level by defining a YES.* profile for a queue manager.

To set off a security switch, you need to define a NO.* switch profile for it. The existence of a NO.* profile means that security checks are **not** performed for that type of resource, unless you choose to override a queue sharing group level setting on a particular queue manager. This is described in "Overriding queue sharing group level settings" on page 178.

If your queue manager is not a member of a queue sharing group, you do not need to define any queue sharing group level profiles or any override profiles. However, you must remember to define these profiles if the queue manager joins a queue sharing group at a later date.

Each NO.* switch profile that IBM MQ detects turns off the checking for that type of resource. Switch profiles are activated during startup of the queue manager. If you change the switch profiles while any affected queue managers are running, you can get IBM MQ to recognize the changes by issuing the IBM MQ REFRESH SECURITY command.

The switch profiles must always be defined in the MQADMIN or MXADMIN class. Do not define them in the GMQADMIN or GMXADMIN class. Tables Switch profiles for subsystem level security and Switch profiles for resource checking show the valid switch profiles and the security type they control.

## Overriding queue sharing group level settings

You can override queue sharing group level security settings for a particular queue manager that is a member of that group. If you want to perform queue manager checks on an individual queue manager that are not performed on other queue managers in the group, use the (qmgr-name.YES.*) switch profiles.

Conversely, if you do not want to perform a certain check on one particular queue manager within a queue sharing group, define a (qmgr-name.NO.*) profile for that particular resource type on the queue manager, and do not define a profile for the queue sharing group. ( IBM MQ only checks for a queue sharing group level profile if it does not find a queue manager level profile.)

### z/OS *Profiles to control subsystem security*

IBM MQ checks whether subsystem security checks are required for the subsystem, for the queue manager, and for the queue sharing group.

The first security check made by IBM MQ is used to determine whether security checks are required for the whole IBM MQ subsystem. If you specify that you do not want subsystem security, no further checks are made.

The following switch profiles are checked to determine whether subsystem security is required. Figure 14 on page 178 shows the order in which they are checked.

*Table 24. Switch profiles for subsystem level security*

| Switch profile name | Type of resource or checking that is controlled |
|---|---|
| qmgr-name.NO.SUBSYS.SECURITY | Subsystem security for this queue manager |
| qsg-name.NO.SUBSYS.SECURITY | Subsystem security for this queue sharing group |
| qmgr-name.YES.SUBSYS.SECURITY | Subsystem security override for this queue manager |

If your queue manager is not a member of a queue sharing group, IBM MQ checks for the qmgr-name.NO.SUBSYS.SECURITY switch profile only.

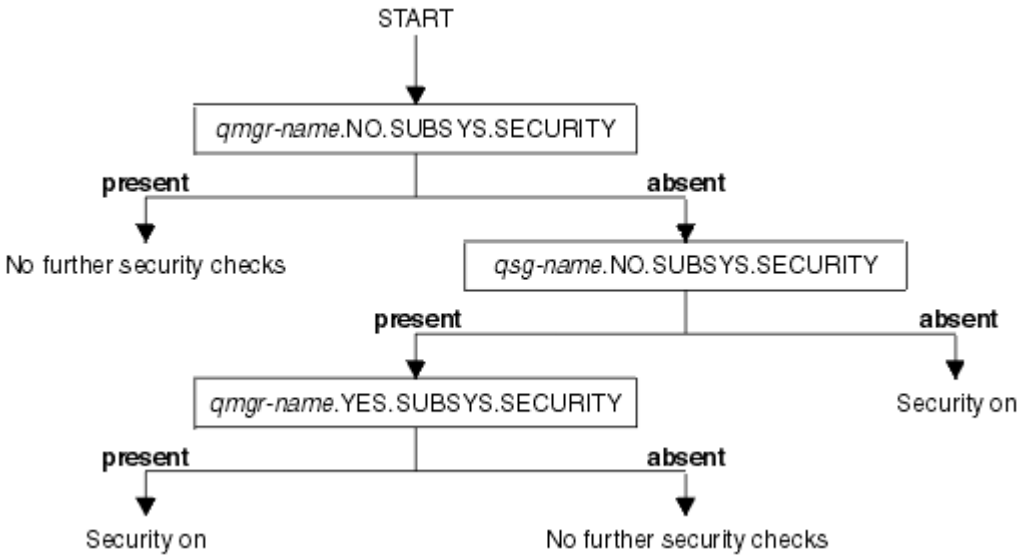



*Figure 14. Checking for subsystem security*

### z/OS *Profiles to control queue sharing group or queue manager level security*

If subsystem security checking is required, IBM MQ checks whether security checking is required at queue sharing group or queue manager level.

When IBM MQ has determined that security checking is required, it then determines whether checking is required at queue sharing group or queue manager level, or both. These checks are not performed if your queue manager is not a member of a queue sharing group.

The following switch profiles are checked to determine the level required. Figure 15 on page 179 and Figure 16 on page 180 show the order in which they are checked.

*Table 25. Switch profiles for queue sharing group or queue manager level security*

| **Switch profile name** | **Type of resource or checking that is controlled** |
|---|---|
| qmgr-name.NO.QMGR.CHECKS | No queue manager level checks for this queue manager |
| qsg-name.NO.QMGR.CHECKS | No queue manager level checks for this queue sharing group |
| qmgr-name.YES.QMGR.CHECKS | Queue manager level checks override for this queue manager |
| qmgr-name.NO.QSG.CHECKS | No queue sharing group level checks for this queue manager |
| qsg-name.NO.QSG.CHECKS | No queue sharing group level checks for this queue sharing group |
| qmgr-name.YES.QSG.CHECKS | Queue sharing group level checks override for this queue manager |

If subsystem security is active, you cannot switch off both queue sharing group and queue manager level security. If you try to do so, IBM MQ sets security checking on at both levels.

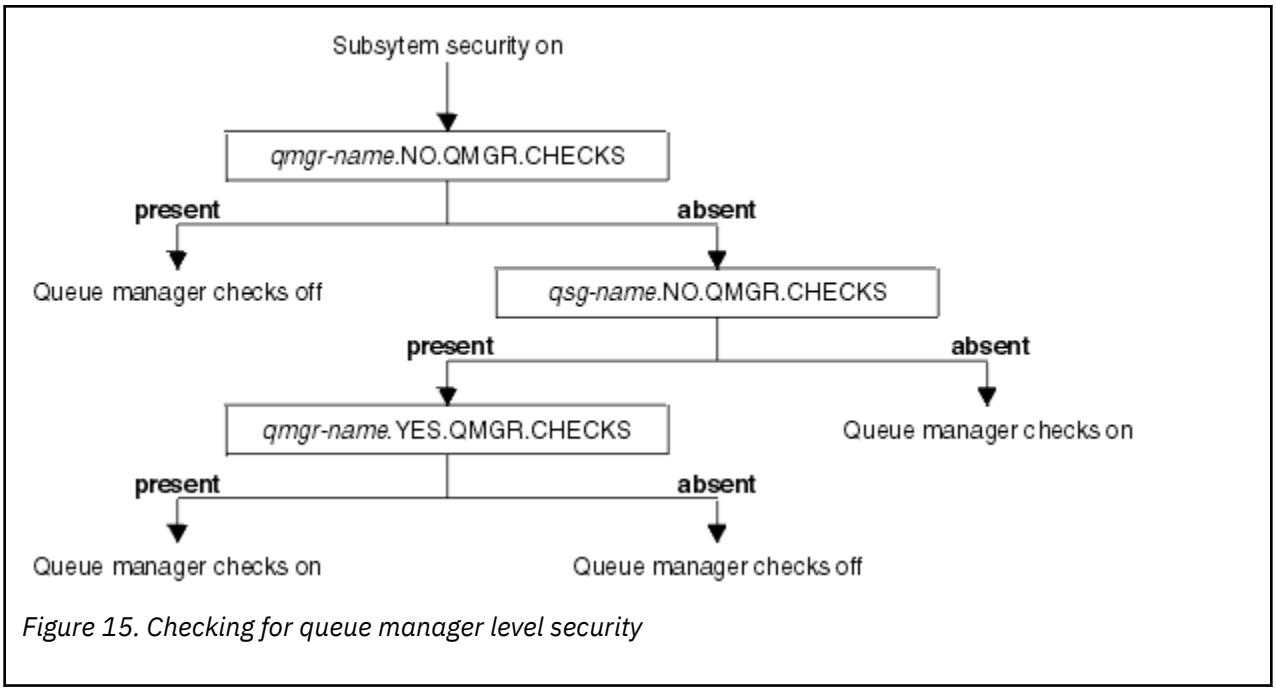



*Figure 15. Checking for queue manager level security*

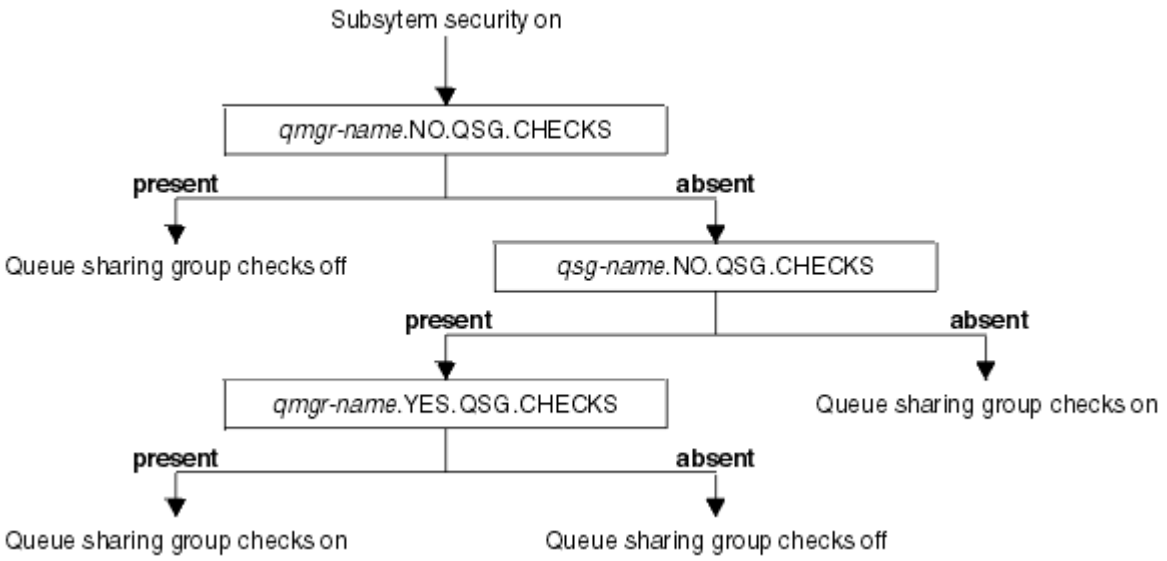



*Figure 16. Checking for queue sharing group level security*

z/OS *Valid combinations of security switches*

Only certain combinations of switches are valid. If you use a combination of switch settings that is not valid, message CSQH026I is issued and security checking is set on at both queue sharing group and queue manager level.

Table 26 on page 180, Table 27 on page 180, Table 28 on page 181, and Table 29 on page 181 show the sets of combinations of switch settings that are valid for each type of security level.

*Table 26. Valid security switch combinations for queue manager level security*

| **Combinations** |
|---|
| qmgr-name.NO.QSG.CHECKS |
| qsg-name.NO.QSG.CHECKS |
| qmgr-name.NO.QSG.CHECKS<br>qsg-name.NO.QMGR.CHECKS<br>qmgr-name.YES.QMGR.CHECKS |
| qsg-name.NO.QSG.CHECKS<br>qsg-name.NO.QMGR.CHECKS<br>qmgr-name.YES.QMGR.CHECKS |

*Table 27. Valid security switch combinations for queue sharing group level security*

| **Combinations** |
|---|
| qmgr-name.NO.QMGR.CHECKS |
| qsg-name.NO.QMGR.CHECKS |
| qmgr-name.NO.QMGR.CHECKS<br>qsg-name.NO.QSG.CHECKS<br>qmgr-name.YES.QSG.CHECKS |

| *Table 27. Valid security switch combinations for queue sharing group level security (continued)* |
|---|
| **Combinations** |
| qsg-name.NO.QMGR.CHECKS<br>qsg-name.NO.QSG.CHECKS<br>qmgr-name.YES.QSG.CHECKS |

| *Table 28. Valid security switch combinations for queue manager and queue sharing group level security* |
|---|
| **Combinations** |
| qsg-name.NO.QMGR.CHECKS<br>qmgr-name.YES.QMGR.CHECKS<br>No QSG.* profiles defined |
| No QMGR.* profiles defined<br>qsg-name.NO.QSG.CHECKS<br>qmgr-name.YES.QSG.CHECKS |
| qsg-name.NO.QMGR.CHECKS<br>qmgr-name.YES.QMGR.CHECKS<br>qsg-name.NO.QSG.CHECKS<br>qmgr-name.YES.QSG.CHECKS |
| No profiles for either switch defined |

| *Table 29. Other valid security switch combinations that switch both levels of checking **on**.* |
|---|
| **Combinations** |
| qmgr-name.NO.QMGR.CHECKS<br>qmgr-name.NO.QSG.CHECKS |
| qsg-name.NO.QMGR.CHECKS<br>qsg-name.NO.QSG.CHECKS |
| qmgr-name.NO.QMGR.CHECKS<br>qsg-name.NO.QSG.CHECKS |
| qsg-name.NO.QMGR.CHECKS<br>qmgr-name.NO.QSG.CHECKS |

### z/OS *Resource level checks*

A number of switch profiles are used to control access to resources. Some stop checking being performed on either a queue manager or a queue sharing group. These can be overridden by profiles that enable checking for specific queue managers.

Table 30 on page 182 shows the switch profiles used to control access to IBM MQ resources.

If your queue manager is part of a queue sharing group and you have both queue manager and queue sharing group security active, you can use a YES.* switch profile to override queue sharing group level profiles and specifically turn on security for a particular queue manager.

Some profiles apply to both queue managers and queue sharing groups. These are prefixed by the string *hlq* and you should substitute the name of your queue sharing group or queue manager, as applicable.

Profile names shown prefixed by *qmgr-name* are queue manager override profiles; you should substitute the name of your queue manager.

*Table 30. Switch profiles for resource checking*

| Type of resource checking that is controlled | Switch profile name | Override profile for a particular queue manager |
|---|---|---|
| Connection security | hlq.NO.CONNECT.CHECKS | qmgr-name.YES.CONNECT.CHECKS |
| Queue security | hlq.NO.QUEUE.CHECKS | qmgr-name.YES.QUEUE.CHECKS |
| Process security | hlq.NO.PROCESS.CHECKS | qmgr-name.YES.PROCESS.CHECKS |
| Namelist security | hlq.NO.NLIST.CHECKS | qmgr-name.YES.NLIST.CHECKS |
| Context security | hlq.NO.CONTEXT.CHECKS | qmgr-name.YES.CONTEXT.CHECKS |
| Alternate user security | hlq.NO.ALTERNATE.USER.CHECKS | qmgr-name.YES.ALTERNATE.USER.CHECKS |
| Command security | hlq.NO.CMD.CHECKS | qmgr-name.YES.CMD.CHECKS |
| Command resource security | hlq.NO.CMD.RESC.CHECKS | qmgr-name.YES.CMD.RESC.CHECKS |
| Topic security | hlq.NO.TOPIC.CHECKS | qmgr-name.YES.TOPIC.CHECKS |
| **Note:** Generic switch profiles such as hlq.NO.** are ignored by IBM MQ | | |

For example, if you want to perform process security checks on queue manager QM01, which is a member of queue sharing group QSG3 but you do not want to perform process security checks on any of the other queue managers in the group, define the following switch profiles:

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

If you want to have queue security checks performed on all the queue managers in the queue sharing group, except QM02, define the following switch profile:

```
QM02.NO.QUEUE.CHECKS
```

(There is no need to define a profile for the queue sharing group because the checks are automatically enabled if there is no profile defined.)

### z/OS *An example of defining switches*

Different IBM MQ subsystems have different security requirements, which can be implemented using different switch profiles.

Four IBM MQ subsystems have been defined:

- MQP1 (a production system)
- MQP2 (a production system)
- MQD1 (a development system)
- MQT1 (a test system)

All four queue managers are members of queue sharing group QS01. All IBM MQ RACF classes have been defined and activated.

These subsystems have different security requirements:

- The production systems require full IBM MQ security checking to be active at queue sharing group level on both systems.

  This is done by specifying the following profile:

  ```
  RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
  ```

  This sets queue sharing group level checking for all the queue managers in the queue sharing group. You do not need to define any other switch profiles for the production queue managers because you want to check everything for these systems.
- Test queue manager MQT1 also requires full security checking. However, because you might want to change this later, security can be defined at queue manager level so that you can change the security settings for this queue manager without affecting the other members of the queue sharing group.

  This is done by defining the NO.QSG.CHECKS profile for MQT1 as follows:

  ```
  RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
  ```
- Development queue manager MQD1 has different security requirements from the rest of the queue sharing group. It requires only connection and queue security to be active.

  This is done by defining a `MQD1.YES.QMGR.CHECKS` profile for this queue manager, and then defining the following profiles to switch off security checking for the resources that do not need to be checked:

  ```
  RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
  RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
  RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
  RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
  RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
  RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
  ```

When the queue manager is active, you can display the current security settings by issuing the DISPLAY SECURITY MQSC command.

You can also change the switch settings when the queue manager is running by defining or deleting the appropriate switch profile in the MQADMIN class. To make the changes to the switch settings active, you must issue the REFRESH SECURITY command for the MQADMIN class.

See “Refreshing queue manager security on z/OS” on page 234 for more details about using the DISPLAY SECURITY and REFRESH SECURITY commands.

## z/OS Profiles used to control access to IBM MQ resources

You must define RACF profiles to control access to IBM MQ resources, in addition to the switch profiles that might have been defined. This collection of topics contains information about the RACF profiles for the different types of IBM MQ resource.

If you do not have a resource profile defined for a particular security check, and a user issues a request that would involve making that check, IBM MQ denies access. You do not have to define profiles for security types relating to any security switches that you have deactivated.

### z/OS *Profiles for connection security*

If connection security is active, you must define profiles in the MQCONN class and permit the necessary groups or user IDs access to those profiles, so that they can connect to IBM MQ.

To enable a connection to be made, you must grant users RACF READ access to the appropriate profile. (If no queue manager level profile exists, and your queue manager is a member of a queue sharing group, checks might be made against queue sharing group level profiles, if the security is set up to do this.)

A connection profile qualified with a queue manager name controls access to a specific queue manager and users given access to this profile can connect to that queue manager. A connection profile qualified with queue sharing group name controls access to all queue managers within the queue sharing group for

that connection type. For example, a user with access to QS01.BATCH can use a batch connection to any queue manager in queue sharing group QS01 that has not got a queue manager level profile defined.

**Note:**

1. For information about the user IDs checked for different security requests, see "User IDs for security checking on z/OS" on page 224.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see "RESLEVEL 보안 프로파일" on page 218.

IBM MQ security recognizes the following different types of connection:

- Batch (and batch-type) connections, these include:
  - z/OS batch jobs
  - TSO applications
  - z/OS UNIX System Services sign-ons
  - Db2 stored procedures
- CICS connections
- IMS connections from control and application processing regions
- The IBM MQ channel initiator

z/OS *Connection security profiles for batch connections*

Profiles for checking batch-type connections are composed of the queue manager or queue sharing group name followed by the word *BATCH*. Give the user ID associated with the connecting address space READ access to the connection profile.

Profiles for checking batch and batch-type connections take the form:

```
hlq.BATCH
```

where `hlq` can be either the `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails.

For batch or batch-type connection requests, you must permit the user ID associated with the connecting address space to access the connection profile. For example, the following RACF command allows users in the CONNTQM1 group to connect to the queue manager TQM1; these user IDs will be permitted to use any batch or batch-type connection.

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

z/OS *Using **CHCKLOCL** on locally bound applications*

**CHCKLOCL** only applies to connections that are made through BATCH connections and does not apply to connections made from CICS or IMS. Connections made through the channel initiator are controlled by **CHCKCLNT**.

## Overview

If you want to configure your z/OS queue manager to mandate user ID and password checking for some, but not all, of your locally bound applications, you need to do some additional configuration.

The reason for this is that once **CHCKLOCL** (*REQUIRED*) is configured, legacy batch applications that use the MQCONN API call can no longer connect to the queue manager.

For z/OS only, a more granular mechanism based on the connection security of an address space can be used to downgrade the global CHCKLOCL(REQUIRED) configuration to CHCKLOCL(OPTIONAL) for specifically defined user IDs. The mechanism used, is described in the following text, together with an example.

In order to allow more granularity on **CHCKLOCL** ( *REQUIRED*) than just EVERYONE, you modify **CHCKLOCL** in the same manner as you modify the access level of the user ID associated with the connecting address space to the `hlq.batch` connection profiles in the MQCONN class.

If the address space user ID only has READ access, which is the minimum you require to be able to connect at all, the **CHCKLOCL** configuration applies as written.

If the address space user ID has UPDATE access (or above) then the **CHCKLOCL** configuration operates in *OPTIONAL* mode. That is, you do not have to provide a user ID and password, but if you do, the user ID and password must be a valid pair.

#### Connection security already configured for your z/OS queue manager

If you have connection security configured for your z/OS queue manager and you want **CHCKLOCL** (*REQUIRED*) to apply to WAS locally bound applications, and no others, carry out the following steps:

1. Start with **CHCKLOCL** (*OPTIONAL*) as your configuration. This means that any user ID and passwords that are supplied are checked for validity, but not mandated.
2. List all the users that have access to the connection security profiles by issuing the command:

   ```
   RLIST MQCONN MQ23.BATCH AUTHUSER
   ```

   This command displays, for example:

   ```
   CLASS    NAME
   -----    ----
   MQCONN   MQ23.BATCH

   USER     ACCESS  ACCESS COUNT
   ----     ------  ------ -----
   JOHNDOE  READ    000009
   JDOE1    READ    000003
   WASUSER  READ    000000
   ```

3. For each user ID listed as having READ access, change the access to

   ```
   UPDATE:- PERMIT MQ23.BATCH CLASS(MQCONN) ID(JOHNDOE) ACCESS(UPDATE)
   ```

4. Update the IBM MQ configuration to **CHCKLOCL** (*REQUIRED*).

   The combination of UPDATE access to `MQ23.BATCH` and the current setting means that you are using **CHCKLOCL** (*OPTIONAL*).

5. Now, apply the **CHCKLOCL** (*REQUIRED*) behavior to one specific user ID, for example WASUSER, so that all the connections coming from that region must provide a user ID and password.

   Do this by reversing the change you made previously, by issuing the command:

   ```
   PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
   ```

#### Connection security is not configured for your z/OS queue manager

In this situation, you must:

1. Create connection profiles for `hlq.BATCH` in the MQCONN class, by issuing the command:

```
RDEFINE MQCONN MQ23.BATCH UACC(NONE)
```

2. Authorize all user IDs that create batch connections to the queue manager, so that they have UPDATE access to this profile. Doing this bypasses the **CHCKLOCL** ( *REQUIRED*) requirement for the user ID and password at the time of connection.

   Do this by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN)ID(JOHNDOE) ACCESS(UPDATE)
```

   These include user IDs:

   a. Used for CSQUTIL, ISPF panels, and other locally bound tools.

   b. Associated with batch like connections to the queue manager. Consider for example, Advanced Message Security, IBM Integration Bus, Db2 stored procedures, z/OS UNIX System Services and TSO users, and Java applications

3. Delete the switch profile for the queue manager by issuing the command:

```
hlq.NO.CONNECT.CHECKS
```

4. Now, apply the **CHCKLOCL** (*REQUIRED*) behavior to one specific user ID, for example WASUSER, so that all the connections coming from that region must provide a user ID and password.

   Do this by reversing the change you made previously, by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

z/OS *Connection security profiles for CICS connections*

Profiles for checking CICS connections are composed of the queue manager or queue sharing group name followed by the word *CICS* . Give the user ID associated with the CICS address space READ access to the connection profile.

Profiles for checking connections from CICS take the form:

```
hlq.CICS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by CICS, you need only permit the CICS address space user ID access to the connection profile.

For example, the following RACF commands allow the CICS address space user ID KCBCICS to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

z/OS *Connection security profiles for IMS connections*

Profiles for checking IMS connections are composed of the queue manager or queue sharing group name followed by the word *IMS* . Give the IMS control and dependent region user IDs READ access to the connection profile.

Profiles for checking connections from IMS take the form:

```
hlq.IMS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by IMS, permit access to the connection profile for the IMS control and dependent region user IDs.

For example, the following RACF commands allow:

- The IMS region user ID, IMSREG, to connect to the queue manager TQM1.
- Users in group BMPGRP to submit BMP jobs.

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

z/OS *Connection security profiles for the channel initiator*

Profiles for checking connections from the channel initiator are composed of the queue manager or queue sharing group name followed by the word *CHIN*. Give the user ID used by the channel initiator started task address space READ access to the connection profile.

Profiles for checking connections from the channel initiator take the form:

```
hlq.CHIN
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by the channel initiator, define access to the connection profile for the user ID used by the channel initiator started task address space.

For example, the following RACF commands allow the channel initiator address space running with user ID DQCTRL to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

### z/OS *Profiles for queue security*

If queue security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to these profiles. Queue security profiles are named after the queue manager or queue sharing group, and the queue to be opened.

If queue security is active, you must:

- Define profiles in the **MQQUEUE** or **GMQQUEUE** classes if using uppercase profiles.
- Define profiles in the **MXQUEUE** or **GMXQUEUE** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use queues.

Profiles for queue security take the form:

```
hlq.queuename
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `queuename` is the name of the queue being opened, as specified in the object descriptor on the `MQOPEN` or `MQPUT1` call.

A profile prefixed by the queue manager name controls access to a single queue on that queue manager. A profile prefixed by the queue sharing group name controls access to access to one or more queues with that queue name on all queue managers within the queue sharing group, or access to a shared queue by any queue manager within the group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that queue on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

If you are using shared queues, you are recommended to use queue sharing group level security.

For details of how queue security operates when the queue name is that of an alias or a model queue, see “Considerations for alias queues” on page 190 and “Considerations for model queues” on page 191 .

The RACF access required to open a queue depends on the `MQOPEN` or `MQPUT1` options specified. If more than one of the MQOO_* and MQPMO_* options is coded, the queue security check is performed for the highest RACF authority required.

*Table 31. Access levels for queue security using the MQOPEN or MQPUT1 calls*

| **MQOPEN or MQPUT1 option** | **RACF access level required to hlq.queuename** |
|---|---|
| MQOO_BROWSE | READ |
| MQOO_INQUIRE | READ |
| MQOO_BIND_* | UPDATE |
| MQOO_INPUT_* | UPDATE |
| MQOO_OUTPUT or MQPUT1 | UPDATE |
| MQOO_PASS_ALL_CONTEXT<br>MQPMO_PASS_ALL_CONTEXT | UPDATE |
| MQOO_PASS_IDENTITY_CONTEXT<br>MQPMO_PASS_IDENTITY_CONTEXT | UPDATE |
| MQOO_SAVE_ALL_CONTEXT | UPDATE |
| MQOO_SET_IDENTITY_CONTEXT<br>MQPMO_SET_IDENTITY_CONTEXT | UPDATE |

| *Table 31. Access levels for queue security using the MQOPEN or MQPUT1 calls (continued)* | |
|---|---|
| **MQOPEN or MQPUT1 option** | **RACF access level required to hlq.queuename** |
| MQOO_SET_ALL_CONTEXT<br>MQPMO_SET_ALL_CONTEXT | UPDATE |
| MQOO_SET | ALTER |

For example, on IBM MQ queue manager QM77, all user IDs in the RACF group PAYGRP are to be given access to get messages from or put messages to all queues with names beginning with 'PAY.'. You can do this using these RACF commands:

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Also, all user IDs in the PAYGRP group must have access to put messages on queues that do not follow the PAY naming convention. For example:

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

You can do this by defining profiles for these queues in the GMQQUEUE class and giving access to that class as follows:

```
RDEFINE GMQQUEUE PAYROLL.EXTRAS UACC(NONE)
        ADDMEM(QM77.REQUEST_QUEUE_FOR_PAYROLL,
               QM77.SALARY.INCREASE.SERVER,
               QM77.REPLIES.FROM.SALARY.MODEL)
PERMIT PAYROLL.EXTRAS CLASS(GMQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

**Note:**

1. If the RACF access level that an application has to a queue security profile is changed, the changes only take effect for any new object handles obtained (that is, new MQOPEN s) for that queue. Those handles already in existence at the time of the change retain their existing access to the queue. If an application is required to use its changed access level to the queue rather than its existing access level, it must close and reopen the queue for each object handle that requires the change.
2. In the example, the queue manager name QM77 could also be the name of a queue sharing group.

Other types of security checks might also occur at the time the queue is opened depending on the open options specified and the types of security that are active. See also "Profiles for context security" on page 203 and "Profiles for alternate user security" on page 202. For a summary table showing the open options and the security authorization needed when queue, context, and alternate user security are all active, see Table 36 on page 195.

If you are using publish/subscribe you must consider the following. When an MQSUB request is processed a security check is performed to ensure that the user ID making the request has the required access to put messages to the target IBM MQ queue as well as the required access to subscribe to the IBM MQ topic.

| *Table 32. Access levels for queue security using the MQSUB call* | |
|---|---|
| **MQSUB option** | **RACF access level required to hlq.queuename** |
| MQSO_ALTER, MQSO_CREATE, and MQSO_RESUME | UPDATE |

**Note:**

1. The `hlq.queuename` is the destination queue for publications. When this is a managed queue, you need access to the appropriate model queue to be used for the managed queue and the dynamic queue that are created.
2. You can use a technique like this for the destination queue you provide on an MQSUB API call if you want to distinguish between the users making the subscriptions, and the users retrieving the publications from the destination queue.

z/OS *Considerations for alias queues*

When you issue an MQOPEN or MQPUT1 call for an alias queue, IBM MQ makes a resource check against the queue name specified in the object descriptor (MQOD) on the call. It does not check if the user is allowed access to the target queue name.

For example, an alias queue called PAYROLL.REQUEST resolves to a target queue of PAY.REQUEST. If queue security is active, you need only be authorized to access the queue PAYROLL.REQUEST. No check is made to see if you are authorized to access the queue PAY.REQUEST.

z/OS *Using alias queues to distinguish between MQGET and MQPUT requests*

The range of MQI calls available in one access level can cause a problem if you want to restrict access to a queue to allow only the **MQPUT** call or only the **MQGET** call. A queue can be protected by defining two aliases that resolve to that queue: one that enables applications to get messages from the queue, and one that enable applications to put messages on the queue.

The following text gives you an example of how you can define your queues to IBM MQ:

```
DEFINE QLOCAL(MUST_USE_ALIAS_TO_ACCESS) GET(ENABLED)
       PUT(ENABLED)

DEFINE QALIAS(USE_THIS_ONE_FOR_GETS) GET(ENABLED)
       PUT(DISABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)

DEFINE QALIAS(USE_THIS_ONE_FOR_PUTS) GET(DISABLED)
       PUT(ENABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)
```

You must also make the following RACF definitions:

```
RDEFINE MQQUEUE hlq.MUST_USE_ALIAS_TO_ACCESS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_GETS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_PUTS UACC(NONE)
```

Then you ensure that no users have access to the queue hlq.MUST_USE_ALIAS_TO_ACCESS, and give the appropriate users or groups access to the alias. You can do this using the following RACF commands:

```
PERMIT hlq.USE_THIS_ONE_FOR_GETS CLASS(MQQUEUE)
       ID(GETUSER,GETGRP) ACCESS(UPDATE)
PERMIT hlq.USE_THIS_ONE_FOR_PUTS CLASS(MQQUEUE)
       ID(PUTUSER,PUTGRP) ACCESS(UPDATE)
```

This means user ID GETUSER and user IDs in the group GETGRP are only allowed to get messages on MUST_USE_ALIAS_TO_ACCESS through the alias queue USE_THIS_ONE_FOR_GETS; and user ID PUTUSER and user IDs in the group PUTGRP are only allowed to put messages through the alias queue USE_THIS_ONE_FOR_PUTS.

**Note:**

1. If you want to use a technique like this, you must inform your application developers, so that they can design their programs appropriately.
2. You can use a technique like this for the destination queue you provide on an MQSUB API request if you want to distinguish between the users making the subscriptions and the users 'getting' the publications from the destination queue.

z/OS *Considerations for model queues*

To open a model queue, you must be able to open both the model queue itself and the dynamic queue to which it resolves. Define generic RACF profiles for dynamic queues, including dynamic queues used by IBM MQ utilities.

When you open a model queue, IBM MQ security makes two queue security checks:

1. Are you authorized to access the model queue?
2. Are you authorized to access the dynamic queue to which the model queue resolves?

If the dynamic queue name contains a trailing asterisk (*) character, this * is replaced by a character string generated by IBM MQ, to create a dynamic queue with a unique name. However, because the whole name, including this generated string, is used for checking authority, you should define generic profiles for these queues.

For example, an MQOPEN call uses a model queue name of CREDIT.CHECK.REPLY.MODEL and a dynamic queue name of CREDIT.REPLY.* on queue manager (or queue sharing group) MQSP.

To do this, you must issue the following RACF commands to define the necessary queue profiles:

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

You must also issue the corresponding RACF PERMIT commands to allow the user access to these profiles.

A typical dynamic queue name created by an MQOPEN is something like CREDIT.REPLY.A346EF00367849A0. The precise value of the last qualifier is unpredictable; this is why you should use generic profiles for such queue names.

A number of IBM MQ utilities put messages on dynamic queues. You should define profiles for the following dynamic queue names, and provide RACF UPDATE access to the relevant user IDs (see "User IDs for security checking on z/OS" on page 224 for the correct user IDs):

```
SYSTEM.CSQUTIL.*  (used by CSQUTIL)
SYSTEM.CSQOREXX.* (used by the operations and control panels)
SYSTEM.CSQXCMD.*  (used by the channel initiator when processing CSQINPX)
CSQ4SAMP.*        (used by the IBM MQ supplied samples)
```

You might also consider defining a profile to control use of the dynamic queue name used by default in the application programming copy members. The IBM MQ-supplied copybooks contain a default *DynamicQName*, which is CSQ.*. This enables an appropriate RACF profile to be established.

**Note:** Do not allow application programmers to specify a single * for the dynamic queue name. If you do, you must define an hlq.** profile in the MQQUEUE class, and you would have to give it wide-ranging access. This means that this profile could also be used for other non-dynamic queues that do not have a more specific RACF profile. Your users could, therefore, gain access to queues you do not want them to access.

z/OS *Close options on permanent dynamic queues*

If an application opens a permanent dynamic queue that was created by another application and then attempts to delete that queue with an MQCLOSE option, some extra security checks are applied when the attempt is made.

*Table 33. Access levels for close options on permanent dynamic queues*

| **MQCLOSE option** | **RACF access level required to hlq.queuename** |
|---|---|
| MQCO_DELETE | ALTER |
| MQCO_DELETE_PURGE | ALTER |

z/OS *Security and remote queues*

When a message is put on a remote queue, the queue security that is implemented by the local queue manager depends on how the remote queue is specified when it is opened.

The following rules are applied:

1. If the remote queue has been defined on the local queue manager through the IBM MQ DEFINE QREMOTE command, the queue that is checked is the name of the remote queue. For example, if a remote queue is defined on queue manager MQS1 as follows:

   ```
   DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
          RNAME(CREDIT.SCORING.REQUEST)
          RQMNAME(BNK7)
          XMITQ(BANK1.TO.BANK7)
   ```

   In this case, a profile for BANK7.CREDIT.REFERENCE must be defined in the MQQUEUE class.
2. If the *ObjectQMgrName* for the request does not resolve to the local queue manager, a security check is carried out against the resolved (remote) queue manager name except in the case of a cluster queue where the check is made against the cluster queue name.

   For example, the transmission queue BANK1.TO.BANK7 is defined on queue manager MQS1. An MQPUT1 request is then issued on MQS1 specifying *ObjectName* as BANK1.INTERBANK.TRANSFERS and an *ObjectQMgrName* of BANK1.TO.BANK7. In this case, the user performing the request must have access to BANK1.TO.BANK7.
3. If you make an MQPUT request to a queue and specify *ObjectQMgrName* as the name of an alias of the local queue manager, only the queue name is checked for security, not that of the queue manager.

When the message gets to the remote queue manager it might be subject to additional security processing. For more information, see "원격 메시징의 보안" on page 94.

z/OS *Dead-letter queue security*

Special considerations apply to the dead-letter queue, because many users must be able to put messages on it, but access to retrieve messages must be tightly restricted. You can achieve this by applying different RACF authorities to the dead-letter queue and an alias queue.

Undelivered messages can be put on a special queue called the dead-letter queue. If you have sensitive data that could possibly end up on this queue, you must consider the security implications of this because you do not want unauthorized users to retrieve this data.

Each of the following must be allowed to put messages onto the dead-letter queue:

- Application programs.
- The channel initiator address space and any MCA user IDs. (If the RESLEVEL profile is not present, or is defined so that channel user IDs are checked, the channel user ID also needs authority to put messages on the dead-letter queue.)
- CKTI, the CICS-supplied CICS task initiator.
- CSQQTRMN, the IBM MQ-supplied IMS trigger monitor.

The only application that can retrieve messages from the dead-letter queue should be a 'special' application that processes these messages. However, a problem arises if you give applications RACF UPDATE authority to the dead-letter queue for MQPUT s because they can then automatically retrieve messages from the queue using MQGET calls. You cannot disable the dead-letter queue for get operations because, if you do, not even the 'special' applications could retrieve the messages.

One solution to this problem is set up a two-level access to the dead-letter queue. CKTI, message channel agent transactions or the channel initiator address space, and 'special' applications have direct access; other applications can only access the dead-letter queue through an alias queue. This alias is defined to allow applications to put messages on the dead-letter queue, but not to get messages from it.

This is how it might work:

1. Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED), as shown in the sample thlqual.SCSQPROC(CSQ4INYG).
2. Give RACF UPDATE authority for the dead-letter queue to the following user IDs:
   - User IDs that the CKTI and the MCAs or channel initiator address space run under.
   - The user IDs associated with the 'special' dead-letter queue processing application.
3. Define an alias queue that resolves to the real dead-letter queue, but give the alias queue these attributes: PUT(ENABLED) and GET(DISABLED). Give the alias queue a name with the same stem as the dead-letter queue name but append the characters ".PUT" to this stem. For example, if the dead-letter queue name is hlq.DEAD.QUEUE, the alias queue name would be hlq.DEAD.QUEUE.PUT.
4. To put a message on the dead-letter queue, an application uses the alias queue. This is what your application must do:
   - Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using MQOPEN and then issues an MQINQ to get the dead-letter queue name.
   - Build the name of the alias queue by appending the characters '.PUT' to this name, in this case, hlq.DEAD.QUEUE.PUT.
   - Open the alias queue, hlq.DEAD.QUEUE.PUT.
   - Put the message on the real dead-letter queue by issuing an MQPUT against the alias queue.
5. Give the user ID associated with the application RACF UPDATE authority to the alias, but no access (authority NONE) to the real dead-letter queue. This means that:
   - The application can put messages onto the dead-letter queue using the alias queue.
   - The application cannot get messages from the dead-letter queue using the alias queue because the alias queue is disabled for get operations.

   The application cannot get any messages from the real dead-letter queue either because it does have the correct RACF authority.

Table 34 on page 193 summarizes the RACF authority required for the various participants in this solution.

*Table 34. RACF authority to the dead-letter queue and its alias*

| **Associated user IDs** | **Real dead-letter queue (hlq.DEAD.QUEUE)** | **Alias dead-letter queue (hlq.DEAD.QUEUE.PUT)** |
|---|---|---|
| MCA or channel initiator address space and CKTI | UPDATE | NONE |
| 'Special' application (for dead-letter queue processing) | UPDATE | NONE |
| User-written application user IDs | NONE | UPDATE |

If you use this method, the application cannot determine the maximum message length (MAXMSGL) of the dead-letter queue. This is because the MAXMSGL attribute cannot be retrieved from an alias queue.

Therefore, your application should assume that the maximum message length is 100 MB, the maximum size IBM MQ for z/OS supports. The real dead-letter queue should also be defined with a MAXMSGL attribute of 100 MB.

**Note:** User-written application programs do not normally use alternate user authority to put messages on the dead-letter queue. This reduces the number of user IDs that have access to the dead-letter queue.

z/OS *System queue security*

You must set up RACF access to allow certain user IDs access to particular system queues.

Many of the system queues are accessed by the ancillary parts of IBM MQ:

- The CSQUTIL utility
- The message security policy utility (CSQ0UTIL)
- The operations and control panels
- The channel initiator address space (including the Queued Pub/Sub Daemon)
- The mqweb server, used by the IBM MQ Console and REST API.

The user IDs under which these run must be given RACF access to these queues, as shown in Table 35 on page 194.

*Table 35. Access required to the SYSTEM queues by IBM MQ*

| SYSTEM queue | CSQUTIL | CSQ0UTIL | mqweb server | Operations and control panels | Channel initiator for distributed queuing |
|---|---|---|---|---|---|
| SYSTEM.ADMIN.CHANNEL.EVENT | - | - | - | - | UPDATE |
| SYSTEM.ADMIN.COMMAND.QUEUE | - | - | UPDATE | - | - |
| SYSTEM.BROKER.ADMIN.STREAM | - | - | - | - | ALTER |
| SYSTEM.BROKER.CONTROL.QUEUE | - | - | - | - | ALTER |
| SYSTEM.BROKER.DEFAULT.STREAM | - | - | - | - | ALTER |
| SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS | - | - | - | - | UPDATE |
| SYSTEM.CHANNEL.INITQ | - | - | - | - | UPDATE |
| SYSTEM.CHANNEL.SYNCQ | - | - | - | - | UPDATE |
| SYSTEM.CLUSTER.COMMAND.QUEUE | - | - | - | - | ALTER |
| SYSTEM.CLUSTER.REPOSITORY.QUEUE | - | - | - | - | UPDATE |
| SYSTEM.CLUSTER.TRANSMIT.QUEUE | - | - | - | - | ALTER |
| SYSTEM.COMMAND.INPUT | UPDATE | - | - | UPDATE | UPDATE |
| SYSTEM.COMMAND.REPLY.* | - | - | - | - | UPDATE |
| SYSTEM.COMMAND.REPLY.MODEL | UPDATE | - | - | UPDATE | UPDATE |
| SYSTEM.CSQOREXX.* | - | - | - | UPDATE | - |
| SYSTEM.CSQUTIL.* | UPDATE | - | - | - | - |
| SYSTEM.CSQXCMD.* | - | - | - | - | UPDATE |
| SYSTEM.HIERARCHY.STATE | - | - | - | - | UPDATE |
| SYSTEM.INTER.QMGR.CONTROL | - | - | - | - | UPDATE |

| *Table 35. Access required to the SYSTEM queues by IBM MQ (continued)* | | | | | |
|---|---|---|---|---|---|
| **SYSTEM queue** | **CSQUTIL** | **CSQ0UTIL** | **mqweb server** | **Operations and control panels** | **Channel initiator for distributed queuing** |
| SYSTEM.INTER.QMGR.PUBS | - | - | - | - | UPDATE |
| SYSTEM.INTER.QMGR.FANREQ | - | - | - | - | UPDATE |
| SYSTEM.PROTECTION.ERROR.QUEUE | - | - | - | - | UPDATE |
| SYSTEM.PROTECTION.POLICY.QUEUE | - | UPDATE "1" on page 195 | - | - | READ |
| SYSTEM.QSG.CHANNEL.SYNCQ | - | - | - | - | UPDATE |
| SYSTEM.QSG.TRANSMIT.QUEUE | - | - | - | - | UPDATE |
| SYSTEM.REST.REPLY.QUEUE | - | - | UPDATE | - | - |
| SYSTEM.BLUEMIX.REGISTRATION.QUEUE | - | - | - | - | UPDATE |

**Notes:**

1. The Advanced Message Security address space user also requires READ access to this queue.

z/OS *API-resource security access quick reference*

A summary of the **MQOPEN**, **MQPUT1**, **MQSUB**, and **MQCLOSE** options and the access required by the different resource security types.

| *Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required.* Callouts shown like this **(1)** refer to the notes following this table. | | | | |
|---|---|---|---|---|
| | **Minimum RACF access level required** | | | |
| **RACF class:** | **MXTOPIC** | **MQQUEUE or MXQUEUE ( 1 )** | **MQADMIN or MXADMIN** | **MQADMIN or MXADMIN** |
| **RACF profile:** | **( 15 or 16 )** | **( 2 )** | **( 3 )** | **( 4 )** |
| MQOPEN option | | | | |
| MQOO_INQUIRE | | READ **( 5 )** | No check | No check |
| MQOO_BROWSE | | READ | No check | No check |
| MQOO_INPUT_* | | UPDATE | No check | No check |
| MQOO_SAVE_ALL_CONTEXT **( 6 )** | | UPDATE | No check | No check |
| MQOO_OUTPUT (USAGE=NORMAL) **( 7 )** | | UPDATE | No check | No check |
| MQOO_PASS_IDENTITY_CONTEXT **( 8 )** | | UPDATE | READ | No check |
| MQOO_PASS_ALL_CONTEXT **( 8 ) ( 9 )** | | UPDATE | READ | No check |
| MQOO_SET_IDENTITY_CONTEXT **( 8 ) ( 9 )** | | UPDATE | UPDATE | No check |
| MQOO_SET_ALL_CONTEXT **( 8 ) ( 10 )** | | UPDATE | CONTROL | No check |
| MQOO_OUTPUT (USAGE (XMITQ) **( 11 )** | | UPDATE | CONTROL | No check |

*Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required.* Callouts shown like this **(1)** refer to the notes following this table. *(continued)*

| | **Minimum RACF access level required** | | | |
|---|---|---|---|---|
| **RACF class:** | **MXTOPIC** | **MQQUEUE or MXQUEUE ( 1 )** | **MQADMIN or MXADMIN** | **MQADMIN or MXADMIN** |
| **RACF profile:** | **( 15 or 16 )** | **( 2 )** | **( 3 )** | **( 4 )** |
| MQOO_OUTPUT (topic object) | UPDATE **( 16 )** | | | |
| MQOO_OUTPUT (alias queue to topic object) | UPDATE **( 16 )** | UPDATE | | |
| MQOO_SET | | ALTER | No check | No check |
| MQOO_ALTERNATE_USER_AUTHORITY | | **( 12 )** | **( 12 )** | UPDATE |
| MQPUT1 option | | | | |
| Put on a normal queue **( 7 )** | | UPDATE | No check | No check |
| MQPMO_PASS_IDENTITY_CONTEXT | | UPDATE | READ | No check |
| MQPMO_PASS_ALL_CONTEXT | | UPDATE | READ | No check |
| MQPMO_SET_IDENTITY_CONTEXT | | UPDATE | UPDATE | No check |
| MQPMO_SET_ALL_CONTEXT | | UPDATE | CONTROL | No check |
| MQOO_OUTPUT<br><br>Put on a transmission queue **( 11 )** | | UPDATE | CONTROL | No check |
| MQOO_OUTPUT (topic object) | UPDATE **( 16 )** | | | |
| MQOO_OUTPUT (alias queue to topic object) | UPDATE **( 16 )** | UPDATE | | |
| MQPMO_ALTERNATE_USER_AUTHORITY | | **( 13 )** | **( 13 )** | UPDATE |
| MQCLOSE option | | | | |
| MQCO_DELETE **( 14 )** | | ALTER | No check | No check |
| MQCO_DELETE_PURGE **( 14 )** | | ALTER | No check | No check |
| MQCO_REMOVE_SUB | ALTER **( 15 )** | | | |
| MQSUB option | | | | |
| MQSO_CREATE | ALTER **( 15 )** | **( 17 )** | **( 18 )** | |
| MQSO_ALTER | ALTER **( 15 )** | **( 17 )** | **( 18 )** | |
| MQSO_RESUME | READ **( 15 )** | **( 17 )** | No check | |
| MQSO_ALTERNATE_USER_AUTHORITY | | | | UPDATE |
| MQSO_SET_IDENTITY_CONTEXT | | | **( 18 )** | |

**Note:**

1. This option is not restricted to queues. Use the MQNLIST or MXNLIST class for namelists, and the MQPROC or MXPROC class for processes.
2. Use RACF profile: hlq.resourcename
3. Use RACF profile: hlq.CONTEXT.queuename
4. Use RACF profile: hlq.ALTERNATE.USER. `alternateuserid`

`alternateuserid` is the user identifier that is specified in the *AlternateUserId* field of the object descriptor. Note that up to 12 characters of the *AlternateUserId* field are used for this check, unlike other checks where only the first 8 characters of a user identifier are used.

5. No check is made when opening the queue manager for inquiries.
6. MQOO_INPUT_* must be specified as well. This is valid for a local, model or alias queue.
7. This check is done for a local or model queue that has a **Usage** queue attribute of MQUS_NORMAL, and also for an alias or remote queue (that is defined to the connected queue manager.) If the queue is a remote queue that is opened specifying an *ObjectQMgrName* (not the name of the connected queue manager) explicitly, the check is carried out against the queue with the same name as *ObjectQMgrName* (which must be a local queue with a **Usage** queue attribute of MQUS_TRANSMISSION).
8. MQOO_OUTPUT must be specified as well.
9. MQOO_PASS_IDENTITY_CONTEXT is implied as well by this option.
10. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT and MQOO_SET_IDENTITY_CONTEXT are implied as well by this option.
11. This check is done for a local or model queue that has a **Usage** queue attribute of MQUS_TRANSMISSION, and is being opened directly for output. It does not apply if a remote queue is being opened.
12. At least one of MQOO_INQUIRE, MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT or MQOO_SET must be specified as well. The check carried out is the same as that for the other options specified.
13. The check carried out is the same as that for the other options specified.
14. This applies only for permanent dynamic queues that have been opened directly, that is, not opened through a model queue. No security is required to delete a temporary dynamic queue.
15. Use RACF profile hlq.SUBSCRIBE.topicname.
16. Use RACF profile hlq.PUBLISH.topicname.
17. If on the MQSUB request you specified a destination queue for the publications to be sent to, then a security check is carried out against that queue to ensure that you have put authority to that queue.
18. If on the MQSUB request, with MQSO_CREATE or MQSO_ALTER options specified, you want to set any of the identity context fields in the MQSD structure, you also need to specify the MQSO_SET_IDENTITY_CONTEXT option and you also need the appropriate authority to the context profile for the destination queue.

### z/OS *Profiles for topic security*

If topic security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

The concept of topic security within a topic tree is described in Publish/subscribe security.

If topic security is active, you must perform the following actions:

- Define profiles in the **MXTOPIC** or **GMXTOPIC** classes.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use topics.

Profiles for topic security take the form:

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

where

- `hlq` is either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).
- `topicname` is the name of the topic administration node in the topic tree, associated either with the topic being subscribed to through an MQSUB call, or being published to through an MQOPEN call.

A profile prefixed by the queue manager name controls access to a single topic on that queue manager. A profile prefixed by the queue sharing group name controls access to one or more topics with that topic name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that topic on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

## Subscribe

To subscribe to a topic, you need access to both the topic you are trying to subscribe to, and the destination queue for the publications.

When you issue an MQSUB request, the following security checks take place:

- Whether you have the appropriate level of access to subscribe to that topic, and also that the destination queue (if specified) is opened for output
- Whether you have the appropriate level of access to that destination queue.

*Table 37. Access level required for topic security to subscribe*

| **MQSUB option** | **RACF access required to `hlq.SUBSCRIBE.topicname` profile in MXTOPIC class** |
|---|---|
| MQSO_CREATE and MQSO_ALTER | ALTER |
| MQSO_RESUME | READ |

*Table 38. Additional authority required to subscribe using a non-managed destination queue*

| **MQSUB option** | **RACF access required to `hlq.CONTEXT.queuename` profile in MQADMIN or MXADMIN class** |
|---|---|
| MQSO_CREATE, MQSO_ALTER, and MQSO_RESUME | UPDATE |
| | **RACF access required to `hlq.queuename` profile in MQQUEUE or MXQUEUE class** |
| MQSO_CREATE and MQSO_ALTER | UPDATE |
| | **RACF access required to `hlq.ALTERNATE.USER.alternateuserid` profile in MQADMIN or MXADMIN class** |
| MQSO_ALTERNATE_USER_AUTHORITY | UPDATE |

## Considerations for managed queues for subscriptions

A security check is carried out to see if you are allowed to subscribe to the topic. However, no security checks are carried out when the managed queue is created, or to determine if you have access to put messages to this destination queue.

You cannot close delete a managed queue.

The model queues used are: `SYSTEM.DURABLE.MODEL.QUEUE` and `SYSTEM.NDURABLE.MODEL.QUEUE`.

The managed queues created from these model queues are of the form `SYSTEM.MANAGED.DURABLE.A346EF00367849A0` and `SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0` where the last qualifier is unpredictable.

Do not give any user access to these queues. The queues can be protected using generic profiles of the form SYSTEM.MANAGED.DURABLE.* and SYSTEM.MANAGED.NDURABLE.* with no authorities granted.

Messages can be retrieved from these queues using the handle returned on the MQSUB request.

If you explicitly issue an MQCLOSE call for a subscription with the MQCO_REMOVE_SUB option specified, and you did not create the subscription you are closing under this handle, a security check is performed at the time of closure to ensure that you have the correct authority to perform the operation.

*Table 39. Access level required to profiles for topic security for closure of a subscribe operation*

| **MQCLOSE option** | **RACF access required to hlq.SUBSCRIBE.topicname profile in MXTOPIC class** |
|---|---|
| MQCO_REMOVE_SUB | ALTER |

## Publish

To publish on a topic you need access to the topic and, if you are using alias queues, to the alias queue as well.

*Table 40. Access level required for topic security to publish*

| **MQOPEN or MQPUT1 option** | **RACF access required to hlq.PUBLISH.topicname profile in MXTOPIC class** |
|---|---|
| MQOO_OUTPUT or MQPUT1 | UPDATE |

*Table 41. Access level required to open an alias queue that resolves to a topic*

| **MQOPEN or MQPUT1 option** | **RACF access required to hlq.queuename profile in MQQUEUE or MXQUEUE class for the alias queue** |
|---|---|
| MQOO_OUTPUT or MQPUT1 | UPDATE |

For details of how topic security operates when an alias queue that resolves to a topic name is opened for publish, see "Considerations for alias queues that resolve to topics for a publish operation" on page 199.

When you consider alias queues used for destination queues for PUT or GET restrictions, see "Considerations for alias queues" on page 190.

If the RACF access level that an application has to a topic security profile is changed, the changes take effect only for any new object handles obtained (that is, a new MQSUB or MQOPEN) for that topic. Those handles already in existence at the time of the change retain their existing access to the topic. Also, existing subscribers retain their access to any subscriptions that they have already made.

## Considerations for alias queues that resolve to topics for a publish operation

When you issue an MQOPEN or MQPUT1 call for an alias queue that resolves to a topic, IBM MQ makes two resource checks:

- The first one against the alias queue name specified in the object descriptor (MQOD) on the MQOPEN or MQPUT1 call.
- The second against the topic to which the alias queue resolves

You must be aware that this behavior is different from the behavior you get when alias queues resolve to other queues. You need the correct access to both profiles in order for the publish action to proceed.

## System topic security

The following system topics are accessed by the channel initiator address space.

The user IDs under which this runs must be given RACF access to these queues, as shown in Table 42 on page 200.

*Table 42. Access required to the SYSTEM topics*

| SYSTEM topic | Profile | Channel initiator for distributed queuing |
|---|---|---|
| SYSTEM.BROKER.ADMIN.STREAM | hlq.PUBLISH.topicname | UPDATE |
| SYSTEM.BROKER.ADMIN.STREAM | hlq.SUBSCRIBE.topicname | ALTER |

### z/OS *Profiles for processes*

If process security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

If process security is active, you must:

- Define profiles in the **MQPROC** or **GMQPROC** classes if using uppercase profiles.
- Define profiles in the **MXPROC** or **GMXPROC** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use processes.

Profiles for processes take the form:

```
hlq.processname
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `processname` is the name of the process being opened.

A profile prefixed by the queue manager name controls access to a single process definition on that queue manager. A profile prefixed by the queue sharing group name controls access to one or more process definitions with that name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that process definition on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access required for opening a process.

*Table 43. Access levels for process security*

| MQOPEN option | RACF access level required to hlq.processname |
|---|---|
| MQOO_INQUIRE | READ |

For example, on queue manager MQS9, the RACF group INQVPRC must be able to inquire ( `MQINQ` ) on all processes starting with the letter V. The RACF definitions for this would be:

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

Alternate user security might also be active, depending on the open options specified when a process definition object is opened.

### z/OS *Profiles for namelists*

If namelist security is active, you define profiles in the appropriate classes and give the necessary groups or user IDs access to these profiles.

If namelist security is active, you must:

- Define profiles in the **MQNLIST** or **GMQNLIST** classes if using uppercase profiles.
- Define profiles in the **MXNLIST** or **GMXNLIST** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles.

Profiles for namelists take the form:

```
hlq.namelistname
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `namelistname` is the name of the namelist being opened.

A profile prefixed by the queue manager name controls access to a single namelist on that queue manager. A profile prefixed by the queue sharing group name controls access to access to one or more namelists with that name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that namelist on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access required for opening a namelist.

*Table 44. Access levels for namelist security*

| **MQOPEN option** | **RACF access level required to hlq.namelistname** |
|---|---|
| MQOO_INQUIRE | READ |

For example, on queue manager (or queue sharing group) PQM3, the RACF group DEPT571 must be able to inquire ( `MQINQ` ) on these namelists:

- All namelists starting with "DEPT571".
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

The RACF definitions to do this are:

```
RDEFINE MQNLIST PQM3.DEPT571.** UACC(NONE)
PERMIT PQM3.DEPT571.** CLASS(MQNLIST) ID(DEPT571) ACCESS(READ)

RDEFINE GMQNLIST NLISTS.FOR.DEPT571 UACC(NONE)
        ADDMEM(PQM3.PRINTER/DESTINATIONS/DEPT571,
               PQM3.AGENCY/REQUEST/QUEUES,
               PQM3.WAREHOUSE.BROADCAST)
PERMIT NLISTS.FOR.DEPT571 CLASS(GMQNLIST) ID(DEPT571) ACCESS(READ)
```

Alternate user security might be active, depending on the options specified when a namelist object is opened.

## System namelist security

Many of the system namelists are accessed by the ancillary parts of IBM MQ:

- The CSQUTIL utility
- The operations and control panels
- The channel initiator address space (including the Queued Publish/Subscribe Daemon)

The user IDs under which these run must be given RACF access to these namelists, as shown in Table 45 on page 202.

*Table 45. Access required to the SYSTEM namelists by IBM MQ*

| SYSTEM namelist | CSQUTIL | Operations and control panels | Channel initiator for distributed queuing |
|---|---|---|---|
| SYSTEM.QPUBSUB.QUEUE.NAMELIST | - | - | READ |
| SYSTEM.QPUBSUB.SUBPOINT.NAMELIST | - | - | READ |

### z/OS *Profiles for alternate user security*

If alternate user security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

For more information about *AlternateUserId*, see AlternateUserID (MQCHAR12).

If alternate user security is active, you must:

- Define profiles in the MQADMIN or GMQADMIN classes if you are using uppercase profiles.
- Define profiles in the MXADMIN or GMXADMIN classes if you are using mixed case profiles.

Permit the necessary groups or user IDs access to these profiles, so that they can use the ALTERNATE_USER_AUTHORITY options when the object is opened.

Profiles for alternate user security can be specified at subsystem level or at queue sharing group level and take the following form:

```
hlq.ALTERNATE.USER.alternateuserid
```

Where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `alternateuserid` is the value of the *AlternateUserId* field in the object descriptor.

A profile prefixed by the queue manager name controls use of an alternative user ID on that queue manager. A profile prefixed by the queue sharing group name controls use of an alternative user ID on all queue managers within the queue sharing group. This alternative user ID can be used on any queue manager within the queue sharing group by a user that has the correct access. This access can be overridden on an individual queue manager by defining a queue manager level profile for that alternative user ID on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access when specifying an alternative user option.

*Table 46. Access levels for alternate user security*

| MQOPEN, MQSUB, or MQPUT1 option | RACF access level required |
|---|---|
| MQOO_ALTERNATE_USER_AUTHORITY<br>MQSO_ALTERNATE_USER_AUTHORITY<br>MQPMO_ALTERNATE_USER_AUTHORITY | UPDATE |

In addition to alternate user security checks, other security checks for queue, process, namelist, and context security can also be made. The alternative user ID, if provided, is only used for security checks on

queue, process definition, or namelist resources. For alternate user and context security checks, the user ID requesting that the check is used. For details about how user IDs are handled, see "User IDs for security checking on z/OS" on page 224. For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 195.

An alternative user profile gives the requesting user ID access to resources associated with the user ID specified in the alternative user ID. For example, the payroll server running under user ID PAYSERV on queue manager QMPY processes requests from personnel user IDs, all of which start with PS. To cause the work performed by the payroll server to be carried out under the user ID of the requesting user, alternative user authority is used. The payroll server knows which user ID to specify as the alternative user ID because the requesting programs generate messages using the MQPMO_DEFAULT_CONTEXT put message option. See "User IDs for security checking on z/OS" on page 224 for more details about from where alternative user IDs are obtained.

The following example RACF definitions enable the server program to specify alternative user IDs starting with the characters PS:

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

**Note:**

1. The *AlternateUserId* fields in the object descriptor and subscription descriptor are 12 bytes long. All 12 bytes are used in the profile checks, but only the first 8 bytes are used as the user ID by IBM MQ. If this user ID truncation is not desirable, application programs making the request must translate any alternative user ID over 8 bytes into something more appropriate.
2. If you specify MQOO_ALTERNATE_USER_AUTHORITY, MQSO_ALTERNATE_USER_AUTHORITY, or MQPMO_ALTERNATE_USER_AUTHORITY and you do not specify an *AlternateUserId* field in the object descriptor, a user ID of blanks is used. For the purposes of the alternate user security check the user ID used for the *AlternateUserId* qualifier is -BLANK-. For example RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-.

   If the user is allowed to access this profile, all further checks are made with a user ID of blanks. For details of blank user IDs, see "Blank user IDs and UACC levels" on page 232.

The administration of alternative user IDs is easier if you have a naming convention for user IDs that enables you to use generic alternative user profiles. If they do not, you can use the RACF RACVAR feature. For details about using RACVAR, see the z/OS Security Server RACF documentation..

When a message is put to a queue that has been opened with alternative user authority and the context of the message has been generated by the queue manager, the MQMD_USER_IDENTIFIER field is set to the alternative user ID.

### z/OS *Profiles for context security*

If context security is active, to control access to the message context information you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles. The message context is contained within the message descriptor (MQMD).

## Using profiles for context security

If context security is active, to permit users to access context information for messages on a particular queue, or when publishing to a particular topic, you must define a profile in one of the following classes:

- The MQADMIN class if using uppercase profiles.
- The MXADMIN class if using mixed-case profiles.

Profiles for context security can be specified at subsystem level or at queue sharing group level and take the following form:

```
hlq.CONTEXT.queuename
hlq.CONTEXT.topicname
```

where *hlq* can be either the queue manager name or the queue sharing group name, and *queuename* and *topicname* can be either the full or generic name of the queue or topic you want to define the context profile for.

A profile prefixed by the queue manager name, and with ** specified as the queue or topic name, allows control for context security on all queues and topics belonging to that queue manager. This can be overridden on an individual queue or topic by defining a specific profile for context on that queue or topic.

A profile prefixed by the queue sharing group name, and with ** specified as the queue or topic name, allows control for context on all queues and topics belonging to the queue managers within the queue sharing group. This can be overridden on an individual queue manager by defining a queue manager level profile for context on that queue manager, by specifying a profile prefixed by the queue manager name. It can also be overridden on an individual queue or topic by specifying a profile suffixed with the queue or topic name.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

You must permit the necessary groups or user IDs access to this profile. The following table shows the access level required, depending on the specification of the context options when the queue is opened.

*Table 47. Access levels for context security*

| **MQOPEN or MQPUT1 option** | **RACF access level required to hlq.CONTEXT.queuename or hlq.CONTEXT.topicname** |
|---|---|
| MQPMO_NO_CONTEXT | No context security check |
| MQPMO_DEFAULT_CONTEXT | No context security check |
| MQOO_SAVE_ALL_CONTEXT | No context security check |
| MQOO_PASS_IDENTITY_CONTEXT<br>MQPMO_PASS_IDENTITY_CONTEXT | READ |
| MQOO_PASS_ALL_CONTEXT<br>MQPMO_PASS_ALL_CONTEXT | READ |
| MQOO_SET_IDENTITY_CONTEXT<br>MQPMO_SET_IDENTITY_CONTEXT | UPDATE |
| MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT | CONTROL |
| MQOO_OUTPUT or MQPUT1 (USAGE(XMITQ)) | CONTROL |
| **MQSUB option** | |
| MQSO_SET_IDENTITY_CONTEXT **( Note 2 )** | UPDATE |

**Note:**

1. The user IDs used for distributed queuing require CONTROL access to `hlq.CONTEXT.queuename` to put messages on the destination queue. See "User IDs used by the channel initiator" on page 227 for information about the user IDs used.
2. If on the MQSUB request, with MQSO_CREATE or MQSO_ALTER options specified, you want to set any of the identity context fields in the MQSD structure, you need to specify the MQSO_SET_IDENTITY_CONTEXT option. You require also, the appropriate authority to the context profile for the destination queue.

If you put commands on the system-command input queue, use the default context put message option to associate the correct user ID with the command.

For example, the IBM MQ-supplied utility program CSQUTIL can be used to offload and reload messages in queues. When offloaded messages are restored to a queue, the CSQUTIL utility uses the MQOO_SET_ALL_CONTEXT option to return the messages to their original state. In addition to the queue security required by this open option, context authority is also required. For example, if this authority is required by the group BACKGRP on queue manager MQS1, this would be defined by:

```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

Depending on the options specified, and the types of security performed, other types of security checks might also occur when the queue is opened. These include queue security (see "Profiles for queue security" on page 188 ), and alternate user security (see "Profiles for alternate user security" on page 202 ). For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 195.

## System queue context security

Many of the system queues are accessed by the ancillary parts of IBM MQ, for example the channel initiator address space, and the mqweb server used by the IBM MQ Console and REST API.

The user IDs under which these run under must be given RACF access to these queues, as shown in Table 48 on page 205.

*Table 48. Access required to the SYSTEM queues for context operations*

| **SYSTEM queue** | **Channel initiator for distributed queuing** | **mqweb server** |
|---|---|---|
| SYSTEM.ADMIN.COMMAND.QUEUE | - | CONTROL |
| SYSTEM.BROKER.CONTROL.QUEUE | CONTROL | - |
| SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS | CONTROL | - |
| SYSTEM.CHANNEL.SYNCQ | CONTROL | - |
| SYSTEM.CLUSTER.COMMAND.QUEUE | CONTROL | - |
| SYSTEM.CLUSTER.TRANSMIT.QUEUE | CONTROL | - |

### z/OS *Profiles for command security*

To enable security checking for commands, add profiles to the MQCMDS class. The profile names are based on the MQSC commands but control both MQSC and PCF commands. Profiles can apply to a queue manager or a queue sharing group.

If you want security checking for commands (so you have not defined the command security switch profile hlq.NO.CMD.CHECKS) you must add profiles to the MQCMDS class.

The same security profiles control both MQSC and PCF commands. The names of the RACF profiles for command security checking are based on the MQSC command names themselves. These profiles take the form:

```
hlq.verb.pkw
```

Where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), `verb` is the verb part of the command name, for example ALTER, and `pkw` is the object type, for example QLOCAL for a local queue.

Thus, the profile name for the ALTER QLOCAL command in subsystem CSQ1 is:

```
CSQ1.ALTER.QLOCAL
```

You can use generic profiles to protect sets of commands so that you have fewer profiles to maintain and, therefore, fewer access lists. Consider creating a generic profile that applies to all commands not protected by a more specific profile. Define this profile with UACC(NONE) and grant ALTER access only to the RACF groups containing administrators. You might then create a generic profile applicable to all DISPLAY commands and grant widespread access to it. Between these extremes, you might identify groups of users needing access to certain sets of commands, in which case you can create profiles for those sets and grant access to RACF groups representing those classes of user. Avoid giving users access to commands they do not require: Apply the principle of least privilege, so that users only have access to the commands that are required for their jobs.

A profile prefixed by the queue manager name controls the use of the command on that queue manager. A profile prefixed by the queue sharing group name controls the use of the command on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

By setting up command profiles at queue manager level, a user can be restricted from issuing commands on a particular queue manager. Alternatively, you can define one profile for a queue sharing group for each command verb, and all security checks take place against that profile instead of individual queue managers.

If both subsystem security and queue sharing group security are active and a local profile is not found, a command security check is performed to see if the user has access to a queue sharing group profile.

If you use the CMDSCOPE attribute to route a command to other queue managers in a queue sharing group, security is checked on each queue manager where the command is run, but not necessarily on the queue manager where the command is entered.

Table 49 on page 206 shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 50 on page 211 shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

*Table 49. MQSC commands, profiles, and their access levels*

| **Command** | **Command profile for MQCMDS** | **Access level for MQCMDS** | **Command resource profile for MQADMIN or MXADMIN** | **Access level for MQADMIN or MXADMIN** |
|---|---|---|---|---|
| ALTER AUTHINFO | hlq.ALTER.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| ALTER BUFFPOOL | hlq.ALTER.BUFFPOOL | ALTER | No check | - |
| ALTER CFSTRUCT | hlq.ALTER.CFSTRUCT | ALTER | No check | - |
| ALTER CHANNEL | hlq.ALTER.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |

*Table 49. MQSC commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| ALTER NAMELIST | hlq.ALTER.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| ALTER PROCESS | hlq.ALTER.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| ALTER PSID | hlq.ALTER.PSID | ALTER | No check | - |
| ALTER QALIAS | hlq.ALTER.QALIAS | ALTER | hlq.QUEUE.queue | ALTER |
| ALTER QLOCAL"5" on page 211 | hlq.ALTER.QLOCAL | ALTER | hlq.QUEUE.queue | ALTER |
| ALTER QMGR | hlq.ALTER.QMGR | ALTER | No check | - |
| ALTER QMODEL"5" on page 211 | hlq.ALTER.QMODEL | ALTER | hlq.QUEUE.queue | ALTER |
| ALTER QREMOTE | hlq.ALTER.QREMOTE | ALTER | hlq.QUEUE.queue | ALTER |
| ALTER SECURITY | hlq.ALTER.SECURITY | ALTER | No check | - |
| ALTER SMDS | hlq.ALTER.SMDS | ALTER | No check | - |
| ALTER STGCLASS | hlq.ALTER.STGCLASS | ALTER | No check | - |
| ALTER SUB | hlq.ALTER.SUB | ALTER | No check | - |
| ALTER TOPIC | hlq.ALTER.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| ALTER TRACE | hlq.ALTER.TRACE | ALTER | No check | - |
| ARCHIVE LOG | hlq.ARCHIVE.LOG | CONTROL | No check | - |
| BACKUP CFSTRUCT | hlq.BACKUP.CFSTRUCT | CONTROL | No check | - |
| CLEAR QLOCAL | hlq.CLEAR.QLOCAL | ALTER | hlq.QUEUE.queue | ALTER |
| CLEAR TOPICSTR "3" on page 211 | hlq.CLEAR.TOPICSTR | ALTER | hlq.TOPIC.topic | ALTER |
| DEFINE AUTHINFO | hlq.DEFINE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| DEFINE BUFFPOOL | hlq.DEFINE.BUFFPOOL | ALTER | No check | - |
| DEFINE CFSTRUCT | hlq.DEFINE.CFSTRUCT | ALTER | No check | - |
| DEFINE CHANNEL | hlq.DEFINE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| DEFINE LOG | hlq.DEFINE.LOG | ALTER | No check | - |
| DEFINE MAXSMSGS | hlq.DEFINE.MAXSMSGS | ALTER | No check | - |
| DEFINE NAMELIST | hlq.DEFINE.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| DEFINE PROCESS | hlq.DEFINE.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| DEFINE PSID | hlq.DEFINE.PSID | ALTER | No check | - |
| DEFINE QALIAS | hlq.DEFINE.QALIAS | ALTER | hlq.QUEUE.queue | ALTER |

*Table 49. MQSC commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| DEFINE QLOCAL"5" on page 211 | hlq.DEFINE.QLOCAL | ALTER | hlq.QUEUE.queue | ALTER |
| DEFINE QMODEL"5" on page 211 | hlq.DEFINE.QMODEL | ALTER | hlq.QUEUE.queue | ALTER |
| DEFINE QREMOTE | hlq.DEFINE.QREMOTE | ALTER | hlq.QUEUE.queue | ALTER |
| DEFINE STGCLASS | hlq.DEFINE.STGCLASS | ALTER | No check | - |
| DEFINE SUB | hlq.DEFINE.SUB | ALTER | No check | - |
| DEFINE TOPIC | hlq.DEFINE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| DELETE AUTHINFO | hlq.DELETE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| DELETE BUFFPOOL | hlq.DELETE.BUFFPOOL | ALTER | No check | - |
| DELETE CFSTRUCT | hlq.DELETE.CFSTRUCT | ALTER | No check | - |
| DELETE CHANNEL | hlq.DELETE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| DELETE NAMELIST | hlq.DELETE.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| DELETE PROCESS | hlq.DELETE.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| DELETE PSID | hlq.DELETE.PSID | ALTER | No check | - |
| DELETE QALIAS | hlq.DELETE.QALIAS | ALTER | hlq.QUEUE.queue | ALTER |
| DELETE QLOCAL | hlq.DELETE.QLOCAL | ALTER | hlq.QUEUE.queue | ALTER |
| DELETE QMODEL | hlq.DELETE.QMODEL | ALTER | hlq.QUEUE.queue | ALTER |
| DELETE QREMOTE | hlq.DELETE.QREMOTE | ALTER | hlq.QUEUE.queue | ALTER |
| DELETE STGCLASS | hlq.DELETE.STGCLASS | ALTER | No check | - |
| DELETE SUB | hlq.DELETE.SUB | ALTER | No check | - |
| DELETE TOPIC | hlq.DELETE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| DISPLAY ARCHIVE "1" on page 211 | hlq.DISPLAY.ARCHIVE | READ | No check | - |
| DISPLAY AUTHINFO | hlq.DISPLAY.AUTHINFO | READ | No check | - |
| DISPLAY CFSTATUS | hlq.DISPLAY.CFSTATUS | READ | No check | - |
| DISPLAY CFSTRUCT | hlq.DISPLAY.CFSTRUCT | READ | No check | - |
| DISPLAY CHANNEL | hlq.DISPLAY.CHANNEL | READ | No check | - |
| DISPLAY CHINIT | hlq.DISPLAY.CHINIT | READ | No check | - |
| DISPLAY CHLAUTH | hlq.DISPLAY.CHLAUTH | READ | No check | - |
| DISPLAY CHSTATUS | hlq.DISPLAY.CHSTATUS | READ | No check | - |
| DISPLAY CLUSQMGR | hlq.DISPLAY.CLUSQMGR | READ | No check | - |

*Table 49. MQSC commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| DISPLAY CMDSERV | hlq.DISPLAY.CMDSERV | READ | No check | - |
| DISPLAY CONN "1" on page 211 | hlq.DISPLAY.CONN | READ | No check | - |
| DISPLAY GROUP | hlq.DISPLAY.GROUP | READ | No check | - |
| DISPLAY LOG "1" on page 211 | hlq.DISPLAY.LOG | READ | No check | - |
| DISPLAY MAXSMSGS | hlq.DISPLAY.MAXSMSGS | READ | No check | - |
| DISPLAY NAMELIST | hlq.DISPLAY.NAMELIST | READ | No check | - |
| DISPLAY PROCESS | hlq.DISPLAY.PROCESS | READ | No check | - |
| DISPLAY PUBSUB | hlq.DISPLAY.PUBSUB | READ | No check | - |
| DISPLAY QALIAS | hlq.DISPLAY.QALIAS | READ | No check | - |
| DISPLAY QCLUSTER | hlq.DISPLAY.QCLUSTER | READ | No check | - |
| DISPLAY QLOCAL | hlq.DISPLAY.QLOCAL | READ | No check | - |
| DISPLAY QMGR | hlq.DISPLAY.QMGR | READ | No check | - |
| DISPLAY QMODEL | hlq.DISPLAY.QMODEL | READ | No check | - |
| DISPLAY QREMOTE | hlq.DISPLAY.QREMOTE | READ | No check | - |
| DISPLAY QSTATUS | hlq.DISPLAY.QSTATUS | READ | No check | - |
| DISPLAY QUEUE | hlq.DISPLAY.QUEUE | READ | No check | - |
| DISPLAY SBSTATUS | hlq.DISPLAY.SBSTATUS | READ | No check | - |
| DISPLAY SMDS | hlq.DISPLAY.SMDS | READ | No check | - |
| DISPLAY SMDSCONN | hlq.DISPLAY.SMDSCONN | READ | No check | - |
| DISPLAY SUB | hlq.DISPLAY.SUB | READ | No check | - |
| DISPLAY SECURITY | hlq.DISPLAY.SECURITY | READ | No check | - |
| DISPLAY STGCLASS | hlq.DISPLAY.STGCLASS | READ | No check | - |
| DISPLAY SYSTEM "1" on page 211 | hlq.DISPLAY.SYSTEM | READ | No check | - |
| DISPLAY THREAD | hlq.DISPLAY.THREAD | READ | No check | - |
| DISPLAY TPSTATUS | hlq.DISPLAY.TPSTATUS | READ | No check | - |
| DISPLAY TOPIC | hlq.DISPLAY.TOPIC | READ | No check | - |
| DISPLAY TPSTATUS | hlq.DISPLAY.TPSTATUS | READ | No check | - |
| DISPLAY TRACE | hlq.DISPLAY.TRACE | READ | No check | - |
| DISPLAY USAGE "1" on page 211 | hlq.DISPLAY.USAGE | READ | No check | - |

*Table 49. MQSC commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| MOVE QLOCAL | hlq.MOVE.QLOCAL | ALTER | hlq.QUEUE.from-queue<br>hlq.QUEUE.to-queue | ALTER |
| PING CHANNEL | hlq.PING.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| RECOVER BSDS | hlq.RECOVER.BSDS | CONTROL | No check | - |
| RECOVER CFSTRUCT | hlq.RECOVER.CFSTRUCT | CONTROL | No check | - |
| REFRESH CLUSTER | hlq.REFRESH.CLUSTER | ALTER | No check | - |
| REFRESH QMGR | hlq.REFRESH.QMGR | ALTER | No check | - |
| REFRESH SECURITY | hlq.REFRESH.SECURITY | ALTER | No check | - |
| RESET CFSTRUCT | hlq.RESET.CFSTRUCT | CONTROL | No check | - |
| RESET CHANNEL | hlq.RESET.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| RESET CLUSTER | hlq.RESET.CLUSTER | CONTROL | No check | - |
| RESET QMGR | hlq.RESET.QMGR | CONTROL | No check | - |
| RESET QSTATS | hlq.RESET.QSTATS | CONTROL | hlq.QUEUE.queue | CONTROL |
| RESET SMDS | hlq.RESET.SMDS | CONTROL | No check | - |
| RESET TPIPE | hlq.RESET.TPIPE | CONTROL | No check | - |
| RESOLVE CHANNEL | hlq.RESOLVE.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| RESOLVE INDOUBT | hlq.RESOLVE.INDOUBT | CONTROL | No check | - |
| RESUME QMGR | hlq.RESUME.QMGR | CONTROL | No check | - |
| RVERIFY SECURITY | hlq.RVERIFY.SECURITY | ALTER | No check | - |
| SET ARCHIVE | hlq.SET.ARCHIVE | CONTROL | No check | - |
| SET CHLAUTH | hlq.SET.CHLAUTH | CONTROL | No check | - |
| SET LOG | hlq.SET.LOG | CONTROL | No check | - |
| SET SYSTEM | hlq.SET.SYSTEM | CONTROL | No check | - |
| START CHANNEL | hlq.START.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| START CHINIT “4” on page 211 | hlq.START.CHINIT | CONTROL | No check | - |
| START CMDSERV | hlq.START.CMDSERV | CONTROL | No check | - |
| START LISTENER | hlq.START.LISTENER | CONTROL | No check | - |
| START QMGR | None “2” on page 211 | - | - | - |
| START SMDSCONN | hlq.START.SMDSCONN | CONTROL | No check | - |
| START TRACE | hlq.START.TRACE | CONTROL | No check | - |
| STOP CHANNEL | hlq.STOP.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |

| *Table 49. MQSC commands, profiles, and their access levels (continued)* | | | | |
|---|---|---|---|---|
| **Command** | **Command profile for MQCMDS** | **Access level for MQCMDS** | **Command resource profile for MQADMIN or MXADMIN** | **Access level for MQADMIN or MXADMIN** |
| STOP CHINIT | hlq.STOP.CHINIT | CONTROL | No check | - |
| STOP CMDSERV | hlq.STOP.CMDSERV | CONTROL | No check | - |
| STOP LISTENER | hlq.STOP.LISTENER | CONTROL | No check | - |
| STOP QMGR | hlq.STOP.QMGR | CONTROL | No check | - |
| STOP SMDSCONN | hlq.STOP.SMDSCONN | CONTROL | No check | - |
| STOP TRACE | hlq.STOP.TRACE | CONTROL | No check | - |
| SUSPEND QMGR | hlq.SUSPEND.QMGR | CONTROL | No check | - |

**Notes:**

1. These commands might be issued internally by the queue manager; no authority is checked in these cases.
2. IBM MQ does not check the authority of the user who issues the START QMGR command. However, you can use RACF, or your alternative security facilities to control access to the START xxxxMSTR command that is issued as a result of the START QMGR command.

   This is done by controlling access to the MVS.START.STC.xxxxMSTR profile in the RACF operator commands (OPERCMDS) class. For details of this procedure, see Granting the user access to the RACF OPERCMDS class in *z/OS MVS Planning: Operations*. If you use this technique, and an unauthorized user tries to start the queue manager, it terminates with a reason code of 00F30216.
3. The **hlq.TOPIC.topic** resource refers to the Topic object derived from the TOPICSTR. For more details, see "발행/구독 보안" on page 452
4. In IBM MQ for z/OS, the resource name MVS.START.STC.CSQ1CHIN has an additional JOBNAME qualifier appended. This can cause problems when starting the channel initiator.

   To resolve the problem replace MVS.START.STC. *ssid* CHIN with a profile for a resource named MVS.START.STC. *ssid* CHIN **.*** or MVS.START.STC. *ssid* CHIN. *ssid* CHIN where *ssid* is the subsystem ID for the queue manager. This requires RACF UPDATE authority. For more details, see MVS™ Commands, RACF Access Authorities, and Resource Names in *z/OS MVS Planning: Operations*.

   The START for *ssid* MSTR does not include the JOBNAME= parameter. For consistency, you might want to update the profile for MVS.START.STC.ssidMSTR to MVS.START.STC.ssidMSTR.*.
5. Setting the queue attribute STREAMQ to a non blank value also requires ALTER access level to MQADMIN or MXADMIN for `hlq.ALTER.streamQ`.

| *Table 50. PCF commands, profiles, and their access levels* | | | | |
|---|---|---|---|---|
| **Command** | **Command profile for MQCMDS** | **Access level for MQCMDS** | **Command resource profile for MQADMIN or MXADMIN** | **Access level for MQADMIN or MXADMIN** |
| Backup CF Structure | hlq.BACKUP.CFSTRUCT | CONTROL | No check | - |
| Change Authentication Information Object | hlq.ALTER.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| Change CF Structure | hlq.ALTER.CFSTRUCT | ALTER | No check | - |

*Table 50. PCF commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| Change Channel | hlq.ALTER.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Change Namelist | hlq.ALTER.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| Change Process | hlq.ALTER.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| Change Queue"2" on page 215 | hlq.ALTER.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |
| Change Queue Manager | hlq.ALTER.QMGR | ALTER | No check | - |
| Change Security | hlq.ALTER.SECURITY | ALTER | No check | - |
| Change SMDS | hlq.ALTER.SMDS | ALTER | No check | - |
| Change Storage Class | hlq.ALTER.STGCLASS | ALTER | No check | - |
| Change Subscription | hlq.ALTER.SUB | ALTER | No check | - |
| Change Topic | hlq.ALTER.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Clear Queue | hlq.CLEAR.QLOCAL | ALTER | hlq.QUEUE.queue | ALTER |
| Clear Topic String "1" on page 215 | hlq.CLEAR.TOPICSTR | ALTER | hlq.TOPIC.topic | ALTER |
| Copy Authentication Information Object | hlq.DEFINE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| Copy CF Structure | hlq.DEFINE.CFSTRUCT | ALTER | No check | - |
| Copy Channel | hlq.DEFINE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Copy Namelist | hlq.DEFINE.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| Copy Process | hlq.DEFINE.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| Copy Queue | hlq.DEFINE.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |
| Copy Subscription | hlq.DEFINE.SUB | ALTER | No check | - |
| Copy Storage Class | hlq.DEFINE.STGCLASS | ALTER | No check | - |
| Copy Topic | hlq.DEFINE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Create Authentication Information Object | hlq.DEFINE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| Create CF Structure | hlq.DEFINE.CFSTRUCT | ALTER | No check | - |
| Create Channel | hlq.DEFINE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Create Namelist | hlq.DEFINE.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| Create Process | hlq.DEFINE.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| Create Queue"2" on page 215 | hlq.DEFINE.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |
| Create Storage Class | hlq.DEFINE.STGCLASS | ALTER | No check | - |
| Create Subscription | hlq.DEFINE.SUB | ALTER | No check | - |
| Create Topic | hlq.DEFINE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Delete Authentication Information Object | hlq.DELETE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |

*Table 50. PCF commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| Delete CF Structure | hlq.DELETE.CFSTRUCT | ALTER | No check | - |
| Delete Channel | hlq.DELETE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Delete Namelist | hlq.DELETE.NAMELIST | ALTER | hlq.NAMELIST.namelist | ALTER |
| Delete Process | hlq.DELETE.PROCESS | ALTER | hlq.PROCESS.process | ALTER |
| Delete Queue | hlq.DELETE.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |
| Delete Storage Class | hlq.DELETE.STGCLASS | ALTER | No check | - |
| Delete Subscription | hlq.DELETE.SUB | ALTER | No check | - |
| Delete Topic | hlq.DELETE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Inquire Archive | hlq.DISPLAY.ARCHIVE | READ | No check | - |
| Inquire Authentication Information Object | hlq.DISPLAY.AUTHINFO | READ | No check | - |
| Inquire Authentication Information Object Names | hlq.DISPLAY.AUTHINFO | READ | No check | - |
| Inquire CF Structure | hlq.DISPLAY.CFSTRUCT | READ | No check | - |
| Inquire CF Structure Names | hlq.DISPLAY.CFSTRUCT | READ | No check | - |
| Inquire CF Structure Status | hlq.DISPLAY.CFSTATUS | READ | No check | - |
| Inquire Channel | hlq.DISPLAY.CHANNEL | READ | No check | - |
| Inquire Channel Authentication Records | hlq.DISPLAY.CHLAUTH | READ | No check | - |
| Inquire Channel Initiator | hlq.DISPLAY.CHINIT | READ | No check | - |
| Inquire Channel Names | hlq.DISPLAY.CHANNEL | READ | No check | - |
| Inquire Channel Status | hlq.DISPLAY.CHSTATUS | READ | No check | - |
| Inquire Cluster Queue Manager | hlq.DISPLAY.CLUSQMGR | READ | No check | - |
| Inquire Connection | hlq.DISPLAY.CONNPCF | READ | No check | - |
| Inquire Group | hlq.DISPLAY.GROUP | READ | No check | - |
| Inquire Log | hlq.DISPLAY.LOG | READ | No check | - |
| Inquire Namelist | hlq.DISPLAY.NAMELIST | READ | No check | - |
| Inquire Namelist Names | hlq.DISPLAY.NAMELIST | READ | No check | - |
| Inquire Process | hlq.DISPLAY.PROCESS | READ | No check | - |
| Inquire Process Names | hlq.DISPLAY.PROCESS | READ | No check | - |
| Inquire Pub/Sub Status | hlq.DISPLAY.PUBSUB | READ | No check | - |
| Inquire Queue | hlq.DISPLAY.QUEUE | READ | No check | - |
| Inquire Queue Manager | hlq.DISPLAY.QMGR | READ | No check | - |

*Table 50. PCF commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| Inquire Queue Names | hlq.DISPLAY.QUEUE | READ | No check | - |
| Inquire Queue Status | hlq.DISPLAY.QSTATUS | READ | No check | - |
| Inquire Security | hlq.DISPLAY.SECURITY | READ | No check | - |
| Inquire SMDS | hlq.DISPLAY.SMDS | READ | No check | - |
| Inquire SMDSCONN | hlq.DISPLAY.SMDSCONN | READ | No check | - |
| Inquire Storage Class | hlq.DISPLAY.STGCLASS | READ | No check | - |
| Inquire Storage Class Names | hlq.DISPLAY.STGCLASS | READ | No check | - |
| Inquire Subscription | hlq.INQUIRE.SUB | READ | No check | - |
| Inquire Subscription Status | hlq.INQUIRE.SBSTATUS | READ | No check | - |
| Inquire System | hlq.DISPLAY.SYSTEM | READ | No check | - |
| Inquire Topic | hlq.DISPLAY.TOPIC | READ | No check | - |
| Inquire Topic Names | hlq.DISPLAY.TOPIC | READ | No check | - |
| Inquire Topic Status | hlq.DISPLAY.TPSTATUS | READ | No check | - |
| Inquire Usage | hlq.DISPLAY.USAGE | READ | No check | - |
| Move Queue | hlq.MOVE.QLOCAL | ALTER | hlq.QUEUE.from-queue<br>hlq.QUEUE.to-queue | ALTER |
| Ping Channel | hlq.PING.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Recover CF Structure | hlq.RECOVER.CFSTRUCT | CONTROL | No check | - |
| Refresh Cluster | hlq.REFRESH.CLUSTER | ALTER | No check | - |
| Refresh Queue Manager | hlq.REFRESH.QMGR | ALTER | No check | - |
| Refresh Security | hlq.REFRESH.SECURITY | ALTER | No check | - |
| Reset CF Structure | hlq.RESET.CFSTRUCT | CONTROL | No check | - |
| Reset Channel | hlq.RESET.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Reset Cluster | hlq.RESET.CLUSTER | CONTROL | No check | - |
| Reset Queue Manager | hlq.RESET.QMGR | CONTROL | No check | - |
| Reset Queue Statistics | hlq.RESET.QSTATS | CONTROL | hlq.QUEUE.queue | CONTROL |
| Reset SMDS | hlq.RESET.SMDS | CONTROL | No check | - |
| Resolve Channel | hlq.RESOLVE.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Resume Queue Manager | hlq.RESUME.QMGR | CONTROL | No check | - |
| Resume Queue Manager Cluster | hlq.RESUME.QMGR | CONTROL | No check | - |
| Reverify Security | hlq.RVERIFY.SECURITY | ALTER | No check | - |
| Set Archive | hlq.SET.ARCHIVE | CONTROL | No check | - |

| Table 50. PCF commands, profiles, and their access levels (continued) | | | | |
|---|---|---|---|---|
| **Command** | **Command profile for MQCMDS** | **Access level for MQCMDS** | **Command resource profile for MQADMIN or MXADMIN** | **Access level for MQADMIN or MXADMIN** |
| Set Channel Authentication Record | hlq.SET.CHLAUTH | CONTROL | No check | - |
| Set Log | hlq.SET.LOG | CONTROL | No check | - |
| Set System | hlq.SET.SYSTEM | CONTROL | No check | - |
| Start Channel | hlq.START.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Start Channel Initiator | hlq.START.CHINIT | CONTROL | No check | - |
| Start Channel Listener | hlq.START.LISTENER | CONTROL | No check | - |
| Start SMDS Connection | hlq.START.SMDSCONN | CONTROL | No check | - |
| Stop Channel | hlq.STOP.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Stop Channel Initiator | hlq.STOP.CHINIT | CONTROL | No check | - |
| Stop Channel Listener | hlq.STOP.LISTENER | CONTROL | No check | - |
| Stop SMDS Connection | hlq.STOP.SMDSCONN | CONTROL | No check | - |
| Suspend Queue Manager | hlq.SUSPEND.QMGR | CONTROL | No check | - |
| Suspend Queue Manager Cluster | hlq.SUSPEND.QMGR | CONTROL | No check | - |

**Notes:**

1. The **hlq.TOPIC.topic** resource refers to the Topic object derived from the TOPICSTR. For more details, see "발행/구독 보안" on page 452
2. Setting the queue attribute STREAMQ to a non blank value also requires ALTER access level to MQADMIN or MXADMIN for `hlq.ALTER.streamQ`.

See "IBM MQ Console - required command security profiles" on page 215 for details of the IBM MQ PCF profiles required, when using the IBM MQ Console.

z/OS *IBM MQ Console - required command security profiles*

Operations performed in the IBM MQ Console by a user in the `MQWebAdmin`, or `MQWebAdminRO`, role take place under the security context of the mqweb server started task user ID. If you want to use the IBM MQ Console, the mqweb server started task user ID needs authorization to issue certain PCF commands.

Table 51 on page 215 shows, for each IBM MQ PCF command, the command security profiles required, and the corresponding access level for each profile in the MQCMDS class needed by the IBM MQ Console.

| Table 51. IBM MQ Console PCF commands, profiles, and their access levels | | | | |
|---|---|---|---|---|
| **Command** | **Command profile for MQCMDS** | **Access level for MQCMDS** | **Command resource profile for MQADMIN or MXADMIN** | **Access level for MQADMIN or MXADMIN** |
| Change Authentication Information Object | hlq.ALTER.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| Change Channel | hlq.ALTER.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Change Queue | hlq.ALTER.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |

*Table 51. IBM MQ Console PCF commands, profiles, and their access levels (continued)*

| Command | Command profile for MQCMDS | Access level for MQCMDS | Command resource profile for MQADMIN or MXADMIN | Access level for MQADMIN or MXADMIN |
|---|---|---|---|---|
| Change Queue Manager | hlq.ALTER.QMGR | ALTER | No check | - |
| Change Topic | hlq.ALTER.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Clear Queue | hlq.CLEAR.QLOCAL | ALTER | hlq.QUEUE.queue | ALTER |
| Create Authentication Information Object | hlq.DEFINE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| Create Channel | hlq.DEFINE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Create Queue | hlq.DEFINE.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |
| Create Subscription | hlq.DEFINE.SUB | ALTER | No check | - |
| Create Topic | hlq.DEFINE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Delete Authentication Information Object | hlq.DELETE.AUTHINFO | ALTER | hlq.AUTHINFO.resourcename | ALTER |
| Delete Channel | hlq.DELETE.CHANNEL | ALTER | hlq.CHANNEL.channel | ALTER |
| Delete Queue | hlq.DELETE.QUEUE | ALTER | hlq.QUEUE.queue | ALTER |
| Delete Subscription | hlq.DELETE.SUB | ALTER | No check | - |
| Delete Topic | hlq.DELETE.TOPIC | ALTER | hlq.TOPIC.topic | ALTER |
| Inquire Authentication Information Object | hlq.DISPLAY.AUTHINFO | READ | No check | - |
| Inquire Authentication Information Object Names | hlq.DISPLAY.AUTHINFO | READ | No check | - |
| Inquire Channel | hlq.DISPLAY.CHANNEL | READ | No check | - |
| Inquire Channel Authentication Records | hlq.DISPLAY.CHLAUTH | READ | No check | - |
| Inquire Channel Initiator | hlq.DISPLAY.CHINIT | READ | No check | - |
| Inquire Channel Names | hlq.DISPLAY.CHANNEL | READ | No check | - |
| Inquire Channel Status | hlq.DISPLAY.CHSTATUS | READ | No check | - |
| Inquire Queue | hlq.DISPLAY.QUEUE | READ | No check | - |
| Inquire Queue Manager | hlq.DISPLAY.QMGR | READ | No check | - |
| Inquire Queue Names | hlq.DISPLAY.QUEUE | READ | No check | - |
| Inquire Queue Status | hlq.DISPLAY.QSTATUS | READ | No check | - |
| Inquire Subscription | hlq.INQUIRE.SUB | READ | No check | - |
| Inquire Subscription Status | hlq.INQUIRE.SBSTATUS | READ | No check | - |
| Inquire Topic | hlq.DISPLAY.TOPIC | READ | No check | - |
| Inquire Topic Names | hlq.DISPLAY.TOPIC | READ | No check | - |
| Inquire Topic Status | hlq.DISPLAY.TPSTATUS | READ | No check | - |
| Ping Channel | hlq.PING.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |

| Table 51. IBM MQ Console PCF commands, profiles, and their access levels (continued) | | | | |
|---|---|---|---|---|
| **Command** | **Command profile for MQCMDS** | **Access level for MQCMDS** | **Command resource profile for MQADMIN or MXADMIN** | **Access level for MQADMIN or MXADMIN** |
| Refresh Cluster | hlq.REFRESH.CLUSTER | ALTER | No check | - |
| Refresh Security | hlq.REFRESH.SECURITY | ALTER | No check | - |
| Reset Channel | hlq.RESET.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Resolve Channel | hlq.RESOLVE.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Set Channel Authentication Record | hlq.SET.CHLAUTH | CONTROL | No check | - |
| Start Channel | hlq.START.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |
| Stop Channel | hlq.STOP.CHANNEL | CONTROL | hlq.CHANNEL.channel | CONTROL |

### z/OS *Profiles for command resource security*

If you have not defined the command resource security switch profile, because you want security checking for resources associated with commands, you must add resource profiles for each resource to the appropriate class. The same security profiles control both MQSC and PCF commands.

If you have not defined the command resource security switch profile, `hlq.NO.CMD.RESC.CHECKS`, because you want security checking for resources associated with commands, you must:

- Add a resource profile in the **MQADMIN** class, if using uppercase profiles, for each resource.
- Add a resource profile in the **MXADMIN** class, if using mixed case profiles, for each resource.

The same security profiles control both MQSC and PCF commands.

Profiles for command resource security checking take the form:

```
hlq.type.resourcename
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).

A profile prefixed by the queue manager name controls access to the resources associated with commands on that queue manager. A profile prefixed by the queue sharing group name controls access to the resources associated with commands on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command resource on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

For example, the RACF profile name for command resource security checking against the model queue CREDIT.WORTHY in subsystem CSQ1 is:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

Because the profiles for all types of command resource are held in the MQADMIN class, the "type" part of the profile name is needed in the profile to distinguish between resources of different types that have the same name. The "type" part of the profile name can be CHANNEL, QUEUE, TOPIC, PROCESS, or NAMELIST. For example, a user might be authorized to define hlq.QUEUE.PAYROLL.ONE, but not authorized to define hlq.PROCESS.PAYROLL.ONE

If the resource type is a queue, and the profile is a queue sharing group level profile, it controls access to one or more local queues within the queue sharing group, or access to a single shared queue from any queue manager in the queue sharing group.

MQSC commands, profiles, and their access levels shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

PCF commands, profiles, and their access levels shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

z/OS *Command resource security checking for alias queues and remote queues*

Alias queue and remote queues both provide indirection to another queue. Additional points apply when you consider security checking for these queues.

### Alias queues

When you define an alias queue, command resource security checks are only performed against the name of the alias queue, not against the name of the target queue to which the alias resolves.

Alias queues can resolve to both local and remote queues. If you do not want to permit users access to certain local or remote queues, you must do both of the following:

1. Do not allow the users access to these local and remote queues.
2. Restrict the users from being able to define aliases for these queues. That is, prevent them from being able to issue DEFINE QALIAS and ALTER QALIAS commands.

### Remote queues

When you define a remote queue, command resource security checks are performed only against the name of the remote queue. No checks are performed against the names of the queues specified in the RNAME or XMITQ attributes in the remote queue object definition.

## z/OS RESLEVEL 보안 프로파일

API 자원 보안에 대해 검사되는 사용자 ID 수를 제어하기 위해 MQADMIN 또는 MXADMIN 클래스에 특수 프로파일을 정의할 수 있습니다. 이 프로파일을 RESLEVEL 프로파일이라고 합니다. 이 프로파일이 API 자원 보안에 영향을 주는 방법은 IBM MQ에 대한 액세스 방법에 따라 다릅니다.

애플리케이션이 IBM MQ 연결을 시도하면 IBM MQ는 연결에 연관된 사용자 ID가 호출된 MQADMIN 또는 MXADMIN 클래스에 프로파일에 대해 가지는 액세스를 검사합니다.

```
hlq.RESLEVEL
```

여기서 `hlq`는 `ssid`(서브시스템 ID) 또는 `qsg`(큐 공유 그룹 ID)입니다.

각 연결 유형에 연관된 사용자 ID는 다음과 같습니다.

- 배치 연결에 대한 연결 작업의 사용자 ID
- CICS 연결을 위한 CICS 주소 공간 사용자 ID
- IMS 연결을 위한 IMS 지역 주소 공간 사용자 ID
- 채널 시작기 연결을 위한 채널 시작기 주소 공간 사용자 ID

**주의:** RESLEVEL은 매우 강력한 옵션입니다. 이로 인해 특정 연결에 대한 모든 자원 보안 검사가 생략될 수 있습니다.

정의된 RESLEVEL 프로파일이 없는 경우 MQADMIN 클래스의 다른 프로파일이 hlq.RESLEVEL에 일치하지 않도록 주의해야 합니다. 예를 들어, MQADMIN에 hlq. * * 라는 프로파일이 있는 경우 hlq.RESLEVEL 프로파일이 없는 경우 hlq. * *의 결과를 주의하십시오. RESLEVEL 검사에 사용되기 때문입니다.

RESLEVEL 프로파일이 전혀 없는 것보다는 hlq.RESLEVEL 프로파일을 정의하고 UACC를 NONE으로 설정하십시오. 가능하면 액세스 목록에 있는 사용자나 그룹 수를 작게 하십시오. RESLEVEL 액세스 감사 방법에 대한 자세한 내용은 243 페이지의 『Auditing considerations on z/OS』의 내용을 참조하십시오.

큐 관리자 레벨 보안만 사용 중인 경우 IBM MQ는 `qmgr-name.RESLEVEL` 프로파일에 대해 RESLEVEL 검사를 수행합니다. 큐 공유 그룹 레벨 보안만 사용 중인 경우, IBM MQ는 `qsg-name.RESLEVEL` 프로파일에 대해 RESLEVEL 검사를 수행합니다. 큐 관리자와 큐 공유 그룹 레벨 보안 결합을 사용 중인 경우, IBM MQ는 우선 큐 관리자 레벨에서 RESLEVEL 프로파일의 존재 여부를 검사합니다. 찾을 수 없는 경우에는 큐 공유 그룹 레벨에서 RESLEVEL 프로파일을 검사합니다.

RESLEVEL 프로파일을 찾을 수 없는 경우, IBM MQ는 CICS 또는 IMS 연결에 대해 작업 및 태스크(또는 대체 사용자) ID 모두에 대한 검사를 사용합니다. 배치 연결의 경우, IBM MQ는 작업(또는 대체) 사용자 ID 검사를 사용합니다. 채널 시작기의 경우, IBM MQ는 채널 사용자 ID 및 MCA(또는 대체) 사용자 ID에 대해 검사를 사용합니다.

RESLEVEL 프로파일이 있는 경우 검사 레벨은 환경 및 프로파일의 액세스 레벨에 따라 다릅니다.

큐 관리자가 큐 공유 그룹의 구성원이고 큐 관리자 레벨에서 이 프로파일을 정의하지 않는 경우, 검사 레벨에 영향을 주는 큐 공유 그룹 레벨에서 정의된 프로파일이 있을 수 있습니다. 두 사용자 ID의 검사를 활성화하려면 UACC (NONE) 를 사용하여 RESLEVEL 프로파일 (큐 공유 그룹 이름의 큐 관리자 이름이 접두부로 추가됨) 을 정의하고 관련 사용자에게 이 프로파일에 대한 액세스 권한이 부여되지 않았는지 확인하십시오.

채널 시작기 사용자 ID가 RESLEVEL에 가지는 액세스를 고려할 때 채널 시작기로 설정된 연결은 채널에서도 사용되는 연결임에 주의하십시오. 채널 시작기 사용자 ID에 대한 모든 자원 보안 검사 생략을 초래하는 설정은 효율적으로 모든 채널에 대한 보안 검사도 생략합니다. RESLEVEL에 대한 채널 시작기 사용자 ID 액세스가 NONE이 아닌 경우, 하나의 사용자 ID(READ 또는 UPDATE의 액세스 레벨에 대해)만 액세스에 대해 검사되거나 사용자 ID(CONTROL 또는 ALTER 액세스 레벨에 대해)가 액세스에 대해 검사되지 않습니다. RESLEVEL에 대해 NONE 이외의 액세스 레벨을 채널 시작기 사용자 ID에 부여하는 경우, 채널에 대해 수행되는 보안 검사에서 이 설정의 영향에 대해 이해하고 있어야 합니다.

RESLEVEL 프로파일 사용은 정상 보안 감사 레코드가 사용되지 않는 것을 의미합니다. 예를 들어, 사용자에 UAUDIT를 넣는 경우, MQADMIN에서 hlq.RESLEVEL 프로파일에 대한 액세스는 감사되지 않습니다.

RACF WARNING 옵션을 hlq.RESLEVEL 프로파일에서 사용하는 경우 RESLEVEL 클래스의 프로파일에 대해 RACF 경고 메시지가 생성되지 않습니다.

COD와 같은 보고서 메시지에 대한 보안 검사는 원래 애플리케이션에 연관된 RESLEVEL 프로파일로 제어됩니다. 예를 들어, 배치 작업 사용자 ID가 RESLEVEL 프로파일에 대해 CONTROL 또는 ALTER 권한을 가지는 경우, 배치 작업으로 수행되는 모든 자원 검사가 생략되며 여기에는 보고서 메시지 보안 검사도 포함됩니다.

RESLEVEL 프로파일을 변경하는 경우, 변경이 적용되기 전에 사용자는 연결을 끊은 후 다시 연결해야 합니다. (여기에는 분산 큐잉 주소 공간 사용자 ID가 RESLEVEL 프로파일에 대해 가지는 액세스가 변경되는 경우 채널 시작기 중지 및 재시작도 포함됩니다.)

RESLEVEL 감사를 끄려면 RESAUDIT 시스템 매개변수를 사용하십시오.

### z/OS *RESLEVEL and batch connections*

By default, when an IBM MQ resource is being accessed through batch and batch-type connections, the user must be authorized to access that resource for the particular operation. You can bypass the security check by setting up an appropriate RESLEVEL definition.

Whether the user is checked or not is based on the user ID used at connect time, the same user ID used for the connection check.

For example, you can set up RESLEVEL so that when a user you trust accesses certain resources through a batch connection, no API-resource security checks are done; but when a user you do not trust tries to access the same resources, security checks are carried out as normal. You should set up RESLEVEL checking to bypass API-resource security checks only when you sufficiently trust the user and the programs run by that user.

The following table shows the checks made for batch connections.

| *Table 52. Checks made at different RACF access levels for batch connections* | |
|---|---|
| **RACF access level** | **Level of checking** |
| NONE | Resource checks performed |
| READ | Resource checks performed |
| UPDATE | Resource checks performed |
| CONTROL | No check. |
| ALTER | No check. |

### z/OS *RESLEVEL and system functions*

The application of RESLEVEL to the operation and control panels, and to CSQUTIL.

The operation and control panels and the CSQUTIL utility are batch-type applications that make requests to the queue manager's command server, and so they are subject to the considerations described in “RESLEVEL and batch connections” on page 219. You can use RESLEVEL to bypass security checking for the SYSTEM.COMMAND.INPUT and SYSTEM.COMMAND.REPLY.MODEL queues that they use, but not for the dynamic queues SYSTEM.CSQXCMD.*, SYSTEM.CSQOREXX.*, and SYSTEM.CSQUTIL.*.

The command server is an integral part of the queue manager and so does not have connection or RESLEVEL checking associated with it. To maintain security, therefore, the command server must confirm that the user ID of the requesting application has authority to open the queue being used for replies. For the operations and control panels, this is SYSTEM.CSQOREXX.*. For CSQUTIL, it is SYSTEM.CSQUTIL.*. Users must be authorized to use these queues, as described in “System queue security” on page 194, in addition to any RESLEVEL authorization they are given.

For other applications using the command server, it is the queue they name as their reply-to queue. Such other applications might deceive the command server into placing messages on unauthorized queues by passing (in the message context) a more trusted user ID than its own to the command server. To prevent this, use a CONTEXT profile to protect the identity context of messages placed on SYSTEM.COMMAND.INPUT.

### z/OS *RESLEVEL and CICS connections*

By default, when an API-resource security check is made on a CICS connection, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

The first user ID checked is that of the CICS address space. This is the user ID on the job card of the CICS job, or the user ID assigned to the CICS started task by the z/OS STARTED class or the started procedures table. (It is not the CICS DFLTUSER.)

The second user ID checked is the user ID associated with the CICS transaction.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED. Both the CICS address space user ID and the user ID of the person running the CICS transaction must have access to the resource at the correct level.

## How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. See Table 53 on page 221 for more information.

The user IDs checked depend on the user ID used at connection time, that is, the CICS address space user ID. This control enables you to bypass API-resource security checking for IBM MQ requests coming from one system (for example, a test system, TESTCICS,) but to implement them for another (for example, a production system, PRODCICS).

**Note:** If you set up your CICS address space user ID with the "trusted" attribute in the STARTED class or the RACF started procedures table ICHRIN03, this overrides any user ID checks for the CICS address

space established by the RESLEVEL profile for your queue manager (that is, the queue manager does not perform the security checks for the CICS address space). For more information, see Securing CICS.

The following table shows the checks made for CICS connections.

*Table 53. Checks made at different RACF access levels for CICS connections*

| RACF access level | Level of checking |
|---|---|
| NONE | IBM MQ checks the CICS address space user ID and the transaction user ID. |
| READ | IBM MQ checks the CICS address space user ID only. |
| UPDATE | If the transaction is defined to CICS with RESSEC(YES), IBM MQ checks the CICS address space user ID and the transaction user ID. |
| UPDATE | If the transaction is defined to CICS with RESSEC(NO), IBM MQ checks the CICS address space user ID only. |
| CONTROL or ALTER | IBM MQ does not check any user IDs. |

### z/OS *RESLEVEL and IMS connections*

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked to see if access is allowed to the resource.

The first user ID checked is that of the address space of the IMS region. This is taken from either the USER field from the job card or the user ID assigned to the region from the z/OS STARTED class or the started procedures table (SPT).

The second user ID checked is associated with the work being done in the dependent region. It is determined according to the type of the dependent region as shown in How the second user ID is determined for the IMS(tm) connection.

If either the first or second IMS user ID does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

The setting of IBM MQ RESLEVEL profiles cannot alter the user ID under which IMS transactions are scheduled from the IBM-supplied MQ-IMS trigger monitor program CSQQTRMN. This user ID is the PSBNAME of that trigger monitor, which by default is CSQQTRMN.

## How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. The possible checks are:

- Check the IMS region address space user ID and the second user ID or alternate user ID.
- Check IMS region address space user ID only.
- Do not check any user IDs.

The following table shows the checks made for IMS connections.

*Table 54. Checks made at different RACF access levels for IMS connections*

| RACF access level | Level of checking |
|---|---|
| NONE | Check the IMS address space user ID and the IMS second user ID or alternate user ID. |
| READ | Check the IMS address space user ID. |
| UPDATE | Check the IMS address space user ID. |

| Table 54. Checks made at different RACF access levels for IMS connections (continued) | |
|---|---|
| **RACF access level** | **Level of checking** |
| CONTROL | No check. |
| ALTER | No check. |

### z/OS *RESLEVEL and the channel initiator connection*

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked to see if access is allowed to the resource.

The user IDs checked can be that specified by the MCAUSER channel attribute, that received from the network, that of the channel initiator address space, or the alternate user ID for the message descriptor. Which user IDs are checked depends on the communication protocol you are using and the setting of the PUTAUT channel attribute. See "User IDs used by the channel initiator" on page 227 for more information.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

## How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested, and how many are checked.

The following table shows the checks made for the channel initiator's connection, and for all channels since they use this connection.

| Table 55. Checks made at different RACF access levels for channel initiator connections | |
|---|---|
| **RACF access level** | **Level of checking** |
| NONE | Check two user IDs. |
| READ | Check one user ID. |
| UPDATE | Check one user ID. |
| CONTROL | No check. |
| ALTER | No check. |
| **Note:** See "User IDs used by the channel initiator" on page 227 for a definition of the user IDs checked | |

### z/OS *RESLEVEL and intra-group queuing*

By default, when an API-resource security check is made by the intra-group queuing agent, two user IDs are checked to see if access is allowed to the resource. You can change which user IDs are checked by setting up an RESLEVEL profile.

The user IDs checked can be the user ID determined by the IGQUSER attribute of the receiving queue manager, the user ID of the queue manager within the queue sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE, or the alternate user ID specified in the *UserIdentifier* field of the message descriptor of the message. See "User IDs used by the intra-group queuing agent" on page 231 for more information.

Because the intra-group queuing agent is an internal queue manager task, it does not issue an explicit connect request and runs under the user ID of the queue manager. The intra-group queuing agent starts at queue manager initialization. During the initialization of the intra-group queuing agent, IBM MQ checks

the access that the user ID associated with the queue manager has to a profile in the MQADMIN class called:

```
hlq.RESLEVEL
```

This check is always performed unless the hlq.NO.SUBSYS.SECURITY switch has been set.

If there is no RESLEVEL profile, IBM MQ enables checking for two user IDs. If there is a RESLEVEL profile, the level of checking depends on the access level granted to the user ID of the queue manager for the profile. Checks made at different RACF(r) access levels for the intra-group queuing agent shows the checks made for the intra-group queuing agent.

*Table 56. Checks made at different RACF access levels for the intra-group queuing agent*

| **RACF access level** | **Level of checking** |
|---|---|
| NONE | Check two user IDs. |
| READ | Check one user ID. |
| UPDATE | Check one user ID. |
| CONTROL | No check. |
| ALTER | No check. |
| **Note:** See "User IDs used by the intra-group queuing agent" on page 231 for a definition of the user IDs checked | |

If the permissions granted to the RESLEVEL profile for the queue manager's user ID are changed, the intra-group queuing agent must be stopped and restarted to pick up the new permissions. Because there is no way to independently stop and restart the intra-group queuing agent, the queue manager must be stopped and restarted to achieve this.

### z/OS *RESLEVEL and the user IDs checked*

Example of setting a RESLEVEL profile and granting access to it.

User ID checking against profile name for batch connections through User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels show how RESLEVEL affects which user IDs are checked for different MQI requests.

For example, you have a queue manager called QM66 with the following requirements:

- User WS21B is to be exempt from resource security.
- CICS started task WXNCICS running under address space user ID CICSWXN is to perform full resource checking only for transactions defined with RESSEC(YES).

To define the appropriate RESLEVEL profile, issue the following RACF command:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

Then give the users access to this profile, using the following commands:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

If you make these changes while the user IDs are connected to queue manager QM66, the users must disconnect and connect again before the change takes place.

If subsystem security is not active when a user connects but, while this user is still connected, subsystem security becomes active, full resource security checking is applied to the user. The user must reconnect to get the correct RESLEVEL processing.

## z/OS User IDs for security checking on z/OS

IBM MQ initiates security checks based on user IDs associated with users, terminals, applications, and other resources. This collection of topics lists which user IDs are used for each type of security check.

### z/OS *User IDs for connection security*

The user ID used for connection security depends on the type of connection.

| Connection type | User ID contents |
|---|---|
| Batch connection | The user ID of the connecting task. For example:<br>• The TSO user ID<br>• The user ID assigned to a batch job by the USER JCL parameter<br>• The user ID assigned to a started task by the STARTED class or the started procedures table |
| CICS connection | The CICS address space user ID. |
| IMS connection | The IMS region address space user ID. |
| Channel initiator connection | The channel initiator address space user ID. |

### z/OS *User IDs for command and command resource security*

The user ID used for command security or command resource security depends on where the command is issued from.

| Issued from... | User ID contents |
|---|---|
| CSQINP1, CSQINP2, or CSQINPT | No check is made. |
| System command input queue | The user ID found in the *UserIdentifier* of the message descriptor of the message that contains the command. If the message does not contain a *UserIdentifier*, a user ID of blanks is passed to the security manager. |
| Console | The user ID signed onto the console. If the console is not signed on, the default user ID set by the CMDUSER system parameter in CSQ6SYSP.<br>To issue commands from a console, the console must have the z/OS SYS AUTHORITY attribute. |
| SDSF/TSO console | TSO or job user ID. |
| Operations and control panels | TSO user ID.<br>If you are going to use the operations and control panels, you must have the appropriate authority to issue the commands corresponding to the actions that you choose. In addition, you must have READ access to all the hlq.DISPLAY. *object* profiles in the MQCMDS class because the panels use the various DISPLAY commands to gather the information that they present. |
| MGCRE | If MGCRE is used with UTOKEN, the user ID in the UTOKEN.<br>If MGCRE is issued without the UTOKEN, the TSO or job user ID is used. |
| CSQ0UTIL | Job user ID. |
| CSQUTIL | Job user ID. |
| CSQINPX | User ID of the channel initiator address space. |

### z/OS *User IDs for resource security (MQOPEN, MQSUB, and MQPUT1)*

This information shows the contents of the user IDs for normal and alternate user IDs for each type of connection. The number of checks is defined by the RESLEVEL profile. The user ID checked is that used for **MQOPEN**, **MQSUB**, or **MQPUT1** calls.

**Note:** All user ID fields are checked exactly as they are received. No conversions take place, and, for example, three user ID fields containing "Bob", "BOB", and "bob" are not equivalent.

#### z/OS *User IDs checked for batch connections*

The user ID checked for a batch connection depends on how the task is run and whether an alternate user ID has been specified.

*Table 57. User ID checking against profile name for batch connections*

| Alternate user ID specified on open? | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|
| ***No*** | - | JOB | JOB |
| ***Yes*** | JOB | JOB | ALT |

Key:

**ALT**
Alternate user ID.

**JOB**

- The user ID of a TSO or z/OS UNIX System Services sign-on.
- The user ID assigned to a batch job.
- The user ID assigned to a started task by the STARTED class or the started procedures table.
- The user ID associated with the executing Db2 stored procedure

A Batch job is performing an MQPUT1 to a queue called Q1 with RESLEVEL set to READ and alternate user ID checking turned off.

Checks made at different RACF(r) access levels for batch connections and User ID checking against profile name for batch connections show that the job user ID is checked against profile hlq.Q1.

#### z/OS *User IDs checked for CICS connections*

The user IDs checked for CICS connections depend on whether one or two checks are to be carried out, and whether an alternate user ID is specified.

*Table 58. User ID checking against profile name for CICS-type user IDs*

| Alternate user ID specified on open? | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|
| ***No, 1 check*** | - | ADS | ADS |
| ***No, 2 checks*** | - | ADS+TXN | ADS+TXN |
| ***Yes, 1 check*** | ADS | ADS | ADS |
| ***Yes, 2 checks*** | ADS+TXN | ADS+TXN | ADS+ALT |

Key:

**ALT**
Alternate user ID

**ADS**
The user ID associated with the CICS batch job or, if CICS is running as a started task, through the STARTED class or the started procedures table.

**TXN**
The user ID associated with the CICS transaction. This is normally the user ID of the terminal user who started the transaction. It can be the CICS DFLTUSER, a PRESET security terminal, or a manually signed-on user.

Determine the user IDs checked for the following conditions:

- The RACF access level to the RESLEVEL profile, for a CICS address space user ID, is set to NONE.
- An MQOPEN call is made against a queue with MQOO_OUTPUT and MQOO_PASS_IDENTITY_CONTEXT.

First, see how many CICS user IDs are checked based on the CICS address space user ID access to the RESLEVEL profile. From Table 53 on page 221 in topic "RESLEVEL and CICS connections" on page 220, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 225 on, these checks are carried out:

- The hlq.ALTERNATE.USER.userid profile is not checked.
- The hlq.CONTEXT.queuename profile is checked with both the CICS address space user ID and the CICS transaction user ID.
- The hlq.resourcename profile is checked with both the CICS address space user ID and the CICS transaction user ID.

This means that four security checks are made for this MQOPEN call.

z/OS *User IDs checked for IMS connections*

The user IDs checked for IMS connections depend on whether one or two checks are to be performed, and whether an alternate user ID is specified. If a second user ID is checked, it depends on the type of dependent region and on which user IDs are available.

*Table 59. User ID checking against profile name for IMS-type user IDs*

| Alternate user ID specified on open? | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|
| ***No, 1 check*** | - | REG | REG |
| ***No, 2 checks*** | - | REG+SEC | REG+SEC |
| ***Yes, 1 check*** | REG | REG | REG |
| ***Yes, 2 checks*** | REG+SEC | REG+SEC | REG+ALT |

Key:

**ALT**
Alternate user ID.

**REG**
The user ID is normally set through the STARTED class or the started procedures table or, if IMS is running, from a submitted job, by the USER JCL parameter.

**SEC**
The second user ID is associated with the work being done in a dependent region. It is determined according to Table 60 on page 227.

| Table 60. How the second user ID is determined for the IMS connection | |
|---|---|
| **Types of dependent region** | **Hierarchy for determining the second user ID** |
| • BMP message driven and successful GET UNIQUE issued.<br>• IFP and GET UNIQUE issued.<br>• MPP. | User ID associated with the IMS transaction if the user is signed on.<br>LTERM name if available.<br>PSBNAME. |
| • BMP message driven and successful GET UNIQUE not issued.<br>• BMP not message driven.<br>• IFP and GET UNIQUE not issued. | User ID associated with the IMS dependent region address space if this is not all blanks or all zeros.<br>PSBNAME. |

z/OS *User IDs used by the channel initiator*

This collection of topics describes the user IDs used and checked for receiving channels and for client MQI requests issued over server-connection channels. Information is provided for TCP/IP and for LU6.2

You can use the PUTAUT parameter of the receiving channel definition to determine the type of security checking used. To get consistent security checking throughout your IBM MQ network, you can use the ONLYMCA and ALTMCA options.

You can use the DISPLAY CHSTATUS command to determine the user identifier used by the MCA.

z/OS *Receiving channels using TCP/IP*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

| Table 61. User IDs checked against profile name for TCP/IP channels | | | |
|---|---|---|---|
| **PUTAUT option specified on receiver or requester channel** | **hlq.ALTERNATE.USER.userid profile** | **hlq.CONTEXT.queuename profile** | **hlq.resourcename profile** |
| ***DEF, 1 check*** | - | CHL | CHL |
| ***DEF, 2 checks*** | - | CHL + MCA | CHL + MCA |
| ***CTX, 1 check*** | CHL | CHL | CHL |
| ***CTX, 2 checks*** | CHL + MCA | CHL + MCA | CHL + ALT |
| ***ONLYMCA, 1 check*** | - | MCA | MCA |
| ***ONLYMCA, 2 checks*** | - | MCA | MCA |
| ***ALTMCA, 1 check*** | MCA | MCA | MCA |
| ***ALTMCA, 2 checks*** | MCA | MCA | MCA + ALT |

Key:

**MCA (MCA user ID)**
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

**CHL (Channel user ID)**
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by

using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space of the receiver or requester end is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

**Note:** The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the receiver or requester is used. This also applies to TCP/IP channels using TLS.

**ALT (Alternate user ID)**
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an **MQOPEN** or **MQPUT1** call is issued for the target destination queue.

z/OS *Receiving channels using LU 6.2*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

*Table 62. User IDs checked against profile name for LU 6.2 channels*

| PUTAUT option specified on receiver or requester channel | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|
| ***DEF, 1 check*** | - | CHL | CHL |
| ***DEF, 2 checks*** | - | CHL + MCA | CHL + MCA |
| ***CTX, 1 check*** | CHL | CHL | CHL |
| ***CTX, 2 checks*** | CHL + MCA | CHL + MCA | CHL + ALT |
| ***ONLYMCA, 1 check*** | - | MCA | MCA |
| ***ONLYMCA, 2 checks*** | - | MCA | MCA |
| ***ALTMCA, 1 check*** | MCA | MCA | MCA |
| ***ALTMCA, 2 checks*** | MCA | MCA | MCA + ALT |

Key:

**MCA (MCA user ID)**
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

**CHL (Channel user ID)**

**Requester-server channels**
If the channel is started from the requester, there is no opportunity to receive a network user ID (the channel user ID).

If the PUTAUT parameter is set to DEF or CTX on the requester channel, the channel user ID is that of the channel initiator address space of the requester because no user ID is received from the network.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA, the channel user ID is ignored and the MCA user ID of the requester is used.

**Other channel types**

If the PUTAUT parameter is set to DEF or CTX on the receiver or requester channel, the channel user ID is the user ID received from the communications system when the channel is initiated.

- If the sending channel is on z/OS, the channel user ID received is the channel initiator address space user ID of the sender.
- If the sending channel is on a different platform (for example, AIX), the channel user ID received is typically provided by the USERID parameter of the channel definition.

If the user ID received is blank, or no user ID is received, a channel user ID of blanks is used.

**ALT (Alternate user ID)**

The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an MQOPEN or MQPUT1 call is issued for the target destination queue.

z/OS *Client MQI requests*

Various user IDs can be used, depending on which user IDs and environment variables have been set. These user IDs are checked against various profiles, depending on the PUTAUT option used and whether an alternate user ID is specified.

This section describes the user IDs checked for client MQI requests issued over server-connection channels for TCP/IP and LU 6.2. The MCA user ID and channel user ID are as for the TCP/IP and LU 6.2 channels described in the previous sections.

For server-connection channels, the user ID received from the client is used if the MCAUSER attribute is blank.

See "클라이언트의 액세스 제어" on page 96 for more information.

For client **MQOPEN**, **MQSUB**, and **MQPUT1** requests, use the following rules to determine the profile that is checked:

- If the request specifies alternate-user authority, a check is made against the *hlq*.ALTERNATE.USER. *userid* profile.
- If the request specifies context authority, a check is made against the *hlq*.CONTEXT. *queuename* profile.
- For all **MQOPEN**, **MQSUB**, and **MQPUT1** requests, a check is made against the *hlq.resourcename* profile.

When you have determined which profiles are checked, use the following table to determine which user IDs are checked against these profiles.

*Table 63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels*

| PUTAUT option specified on server-connection channel | Alternate user ID specified on open? | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|---|
| ***DEF, 1 check*** | No | - | CHL | CHL |
| ***DEF, 1 check*** | Yes | CHL | CHL | CHL |
| ***DEF, 2 checks*** | No | - | CHL + MCA | CHL + MCA |
| ***DEF, 2 checks*** | Yes | CHL + MCA | CHL + MCA | CHL + ALT |

*Table 63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels (continued)*

| PUTAUT option specified on server-connection channel | Alternate user ID specified on open? | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|---|
| ***ONLYMCA, 1 check*** | No | - | MCA | MCA |
| ***ONLYMCA, 1 check*** | Yes | MCA | MCA | MCA |
| ***ONLYMCA, 2 checks*** | No | - | MCA | MCA |
| ***ONLYMCA, 2 checks*** | Yes | MCA | MCA | MCA + ALT |

Key:

**MCA (MCA user ID)**
The user ID specified for the MCAUSER channel attribute at the server-connection; if blank, the channel initiator address space user ID is used.

**CHL (Channel user ID)**
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

**Note:** The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the server-connection channel is used. This also applies to TCP/IP channels using TLS.

**ALT (Alternate user ID)**
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object or subscription descriptor before an **MQOPEN**, **MQSUB** or **MQPUT1** call is issued on behalf of the client application.

z/OS *Channel initiator example*

An example of how user IDs are checked against RACF profiles.

A user performs an **MQPUT1** operation to a queue on queue manager QM01 that resolves to a queue called QB on queue manager QM02. The message is sent on a TCP/IP channel called QM01.TO.QM02. RESLEVEL is set to NONE, and the open is performed with alternate user ID and context checking. The receiver channel definition has PUTAUT(CTX) and the MCA user ID is set. Which user IDs are used on the receiving channel to put the message to queue QB?

***Answer:*** Table 55 on page 222 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 227 shows that, with PUTAUT set to CTX and 2 checks, the following user IDs are checked:

- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.ALTERNATE.USER.userid profile.
- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.CONTEXT.queuename profile.
- The channel initiator user ID and the alternate user ID specified in the message descriptor (MQMD) are checked against the hlq.Q2 profile.

z/OS *User IDs used by the intra-group queuing agent*

The user IDs that are checked when the intra-group queuing agent opens destination queues are determined by the values of the **IGQAUT** and **IGQUSER** queue manager attributes.

The possible user IDs are:

**Intra-group queuing user ID (IGQ)**
The user ID determined by the **IGQUSER** attribute of the receiving queue manager. If this is set to blanks, the user ID of the receiving queue manager is used. However, because the receiving queue manager has authority to access all queues defined to it, security checks are not performed for the receiving queue manager's user ID. In this case:

- If only one user ID is to be checked and the user ID is that of the receiving queue manager, no security checks take place. This can occur when **IGQAUT** is set to ONLYIGQ or ALTIGQ.
- If two user IDs are to be checked and one of the user IDs is that of the receiving queue manager, security checks take place for the other user ID only. This can occur when **IGQAUT** is set to DEF, CTX, or ALTIGQ.
- If two user IDs are to be checked and both user IDs are that of the receiving queue manager, no security checks take place. This can occur when **IGQAUT** is set to ONLYIGQ.

**Sending queue manager user ID (SND)**
The user ID of the queue manager within the queue sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE.

**Alternate user ID (ALT)**
The user ID specified in the *UserIdentifier* field in the message descriptor of the message.

*Table 64. User IDs checked against profile name for intra-group queuing*

| IGQAUT option specified on receiving queue manager | hlq.ALTERNATE.USER.userid profile | hlq.CONTEXT.queuename profile | hlq.resourcename profile |
|---|---|---|---|
| ***DEF, 1 check*** | - | SND | SND |
| ***DEF, 2 checks*** | - | SND +IGQ | SND +IGQ |
| ***CTX, 1 check*** | SND | SND | SND |
| ***CTX, 2 checks*** | SND + IGQ | SND +IGQ | SND + ALT |
| ***ONLYIGQ, 1 check*** | - | IGQ | IGQ |
| ***ONLYIGQ, 2 checks*** | - | IGQ | IGQ |
| ***ALTIGQ, 1 check*** | - | IGQ | IGQ |
| ***ALTIGQ, 2 checks*** | IGQ | IGQ | IGQ + ALT |

Key:

**ALT**
Alternate user ID.

**IGQ**
IGQ user ID.

**SND**
Sending queue manager user ID.

### z/OS *Blank user IDs and UACC levels*

If a blank user ID occurs, a RACF undefined user is signed on. Do not grant wide-ranging access to the undefined user.

Blank user IDs can exist when a user is manipulating messages using context or alternate-user security, or when IBM MQ is passed a blank user ID. For example, a blank user ID is used when a message is written to the system-command input queue without context.

**Note:** A user ID of " * " (that is, an asterisk character followed by seven spaces) is treated as an undefined user ID.

IBM MQ passes the blank user ID to RACF and a RACF undefined user is signed on. All security checks then use the universal access (UACC) for the relevant profile. Depending on how you have set your access levels, the UACC might give the undefined user a wide-ranging access.

For example, if you issue this RACF command from TSO:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

you define a profile that enables both z/OS-defined user IDs (that have not been put in the access list) and the RACF undefined user ID to put messages on, and get messages from, that queue.

To protect against blank user IDs you must plan your access levels carefully, and limit the number of people who can use context and alternate-user security. You must prevent people using the RACF undefined user ID from getting access to resources that they must not access. However, at the same time, you must allow access to people with defined user IDs. To do this, you can specify a user ID of asterisk (*) in a RACF command PERMIT, giving access to resources for all defined user IDs. Therefore all undefined user IDs (such as " * ") are denied access. For example, these RACF commands prevent the RACF undefined user ID from gaining access to the queue to put or get messages:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

## z/OS z/OS user IDs and Multi-Factor Authentication (MFA)

IBM Multi-Factor Authentication for z/OS allows z/OS security administrators to enhance SAF authentication, by requiring identified users to use multiple authentication factors (for example, both a password and a cryptographic token) to sign on to a z/OS system. IBM MFA also provides support for time-based one time password generation technologies such as RSA SecureId.

For the most part, IBM MQ is unaware of how users have "logged on" to the CICS or batch systems that are driving IBM MQ work, the signed on user ID credential is associated with the z/OS task or address space and IBM MQ uses this for checking authorization to resources. User IDs enabled for MFA can be used for authorization to IBM MQ resources and authentication through pass tickets used with the CICS and IMS bridges.

**Important:** Special considerations apply however, when using applications, such as the IBM MQ Explorer, which pass a user ID and password credentials on an MQCONNX API call with the *MQCSP_AUTH_USER_ID_AND_PWD* option. IBM MQ has no facility to pass an additional credential on this API request.

Limitations and potential workarounds are described in the following text.

### IBM MQ Explorer

The IBM MQ Explorer cannot be used to log on to a z/OS system with a userid for which MFA is enabled because there is no facility for passing a second authentication factor from the IBM MQ Explorer to z/OS.

Additionally, there are two different mechanisms used by the IBM MQ Explorer to re-use a user ID and password credential, that need special attention when one time use passwords are in effect:

1. IBM MQ Explorer has the capability to store passwords in an obfuscated format on the local machine for login at a later time. This capability must be disabled by having explorer prompt for a password each time a connection is made to the z/OS queue manager.

   To do this, use the following procedure:

   a. Select **Queue Managers**.

   b. From the list displayed, choose the queue manager you require and right click that queue manager.

   c. Select **Connection Details** from the menu list that appears.

   d. Select **Properties** from the next menu list and choose the **Userid** tab.

   Ensure that you select the **prompt for password** radio button.
2. Various operations in the IBM MQ Explorer, such as browsing messages on queues, testing subscriptions, and so on, start a new thread which authenticates to IBM MQ using the credential first used at logon. Since the password credential cannot be re-used, you cannot use these operations.

There are two possible workarounds at the MFA configuration level for these issues:

- Use the application ID exclusion of MFA to exclude the IBM MQ tasks from MFA processing altogether.

  To do this, issue the following commands:

  1. ```
     RDEFINE MFADEF MFABYPASS.USERID.chinuser
     ```

     where *chinuser* is the channel initiator address space level user Id (associated with the channel initiator through the STC class)
  2. ```
     PERMIT MFABYPASS.USERID.chinuser CLASS MFADEF ACCESS(READ) ID(explorer user)
     ```

  For more information on this approach, see Bypassing IBM MFA for applications.
- Use Out-of-band support on MFA, which was introduced with IBM MFA 1.2. With this approach, you pre-authenticate to the IBM MFA web server, and in addition to your user ID and password, specify additional authentication as determined through the policy. IBM MFA server generates a cache token credential that you then specify on the IBM MQ Explorer authentication dialogue. The security administrator can allow this credential to be replayed for a reasonable period of time, so enabling normal IBM MQ Explorer use.

For more information on this approach see Introduction to IBM MFA.

## z/OS IBM MQ for z/OS security management

IBM MQ uses an in-storage table to hold information relating to each user and the access requests made by each user. To manage this table efficiently and to reduce the number of requests made from IBM MQ to the external security manager (ESM), a number of controls are available.

These controls are available through both the operations and control panels and IBM MQ commands.

### z/OS *User ID reverification*

If the RACF definition of a user who is using IBM MQ resources has been changed, for example by connecting the user to a new group, you can tell the queue manager to sign this user on again the next time it tries to access an IBM MQ resource. You can do this by using the IBM MQ command RVERIFY SECURITY.

- User HX0804 is getting and putting messages to the PAYROLL queues on queue manager PRD1. However HX0804 now requires access to some of the PENSION queues on the same queue manager (PRD1).
- The data security administrator connects user HX0804 to the RACF group that allows access to the PENSION queues.
- So that HX0804 can access the PENSION queues immediately (that is, without shutting down queue manager PRD1 or waiting for HX0804 to time out) you must use the IBM MQ command:

```
RVERIFY SECURITY(HX0804)
```

**Note:** If you turn off user ID timeout for long periods of time (days or even weeks) while the queue manager is running, you must remember to run the RVERIFY SECURITY command for any users that have been revoked or deleted in that time.

### z/OS *User ID timeouts*

You can make IBM MQ sign a user off a queue manager after a period of inactivity.

When a user accesses an IBM MQ resource, the queue manager tries to sign this user on to the queue manager (if subsystem security is active). This means that the user is authenticated to the ESM. This user remains signed on to IBM MQ until either the queue manager is shut down, or until the user ID is *timed out* (the authentication lapses) or reverified (reauthenticated).

When a user is timed out, the user ID is *signed off* within the queue manager and any security-related information retained for this user is discarded. The signing on and off of the user within the queue manager is not apparent to the application program or to the user.

Users are eligible for timeout when they have not used any IBM MQ resources for a predetermined amount of time. This time period is set by the MQSC ALTER SECURITY command.

Two values can be specified in the ALTER SECURITY command:

**TIMEOUT**
The time period in minutes that an unused user ID and its associated resources can remain within the IBM MQ queue manager.

**INTERVAL**
The time period in minutes between checks for user IDs and their associated resources, to determine whether the *TIMEOUT* has expired.

For example, if the *TIMEOUT* value is 30 and the *INTERVAL* value is 10, every 10 minutes IBM MQ checks user IDs and their associated resources to determine whether any have not been used for 30 minutes. If a timed-out user ID is found, that user ID is signed off within the queue manager. If any timed-out resource information associated with non-timed-out user IDs is found, that resource information is discarded. If you do not want to time out user IDs, set the *INTERVAL* value to zero. However, if the *INTERVAL* value is zero, storage occupied by user IDs and their associated resources is not freed until you issue a **REFRESH SECURITY** or **RVERIFY SECURITY** command.

Tuning this value can be important if you have many one-off users. If you set small interval and timeout values, resources that are no longer required are freed.

**Note:** If you use values for *INTERVAL* or *TIMEOUT* other than the defaults, you must reenter the command at every queue manager startup. You can do this automatically by putting the **ALTER SECURITY** command in the CSQINP1 data set for that queue manager.

### z/OS *Refreshing queue manager security on z/OS*

IBM MQ for z/OS caches RACF data to improve performance. When you change certain security classes, you must refresh this cached information. Refresh security infrequently, for performance reasons. You can also choose to refresh only TLS security information.

When a queue is opened for the first time (or for the first time since a security refresh) IBM MQ performs a RACF check to obtain the user's access rights and places this information in the cache. The cached data

includes user IDs and resources on which security checking has been performed. If the queue is opened again by the same user, the presence of the cached data means that IBM MQ does not have to issue RACF checks, which improves performance. The action of a security refresh is to discard any cached security information and so force IBM MQ to make a new check against RACF. Whenever you add, change or delete a RACF resource profile that is held in the MQADMIN, MXADMIN, MQPROC, MXPROC, MQQUEUE, MXQUEUE, MQNLIST, MXNLIST, or MXTOPIC class, you must tell the queue managers that use this class to refresh the security information that they hold. To do this, issue the following commands:

- The RACF SETROPTS RACLIST(classname) REFRESH command to refresh at the RACF level.
- The IBM MQ REFRESH SECURITY command to refresh the security information held by the queue manager. This command needs to be issued by each queue manager that accesses the profiles that have changed. If you have a queue sharing group, you can use the command scope attribute to direct the command to all the queue managers in the group.

  **Note:** If you have connected a new user to an existing group, you need to run the IBM MQ RVERIFY SECURITY(userid) command. The REFRESH SECURITY(*) command does not let the queue manager sign this user on again, the next time it tries to access an IBM MQ resource.

If you are using generic profiles in any of the IBM MQ classes, you must also issue normal RACF refresh commands if you change, add, or delete any generic profiles. For example, SETROPTS GENERIC(classname) REFRESH.

However, if a RACF resource profile is added, changed or deleted, and the resource to which it applies has not yet been accessed (so no information is cached), IBM MQ uses the new RACF information without a REFRESH SECURITY command being issued.

If RACF auditing is turned on, (for example, by using the RACF RALTER AUDIT(access-attempt (audit_access_level)) command), no caching takes place, and therefore IBM MQ refers directly to the RACF dataspace for every check. Changes are therefore picked up immediately and REFRESH SECURITY is not necessary to access the changes. You can confirm whether RACF auditing is on by using the RACF RLIST command. For example, you could issue the command

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

and receive the results

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

This indicates that auditing is set on. For more information, see the *z/OS Security Server RACF Auditor's Guide* and the *z/OS Security Server RACF Command Language Reference*.

Figure 17 on page 236 summarizes the situations in which security information is cached and in which cached information is used.

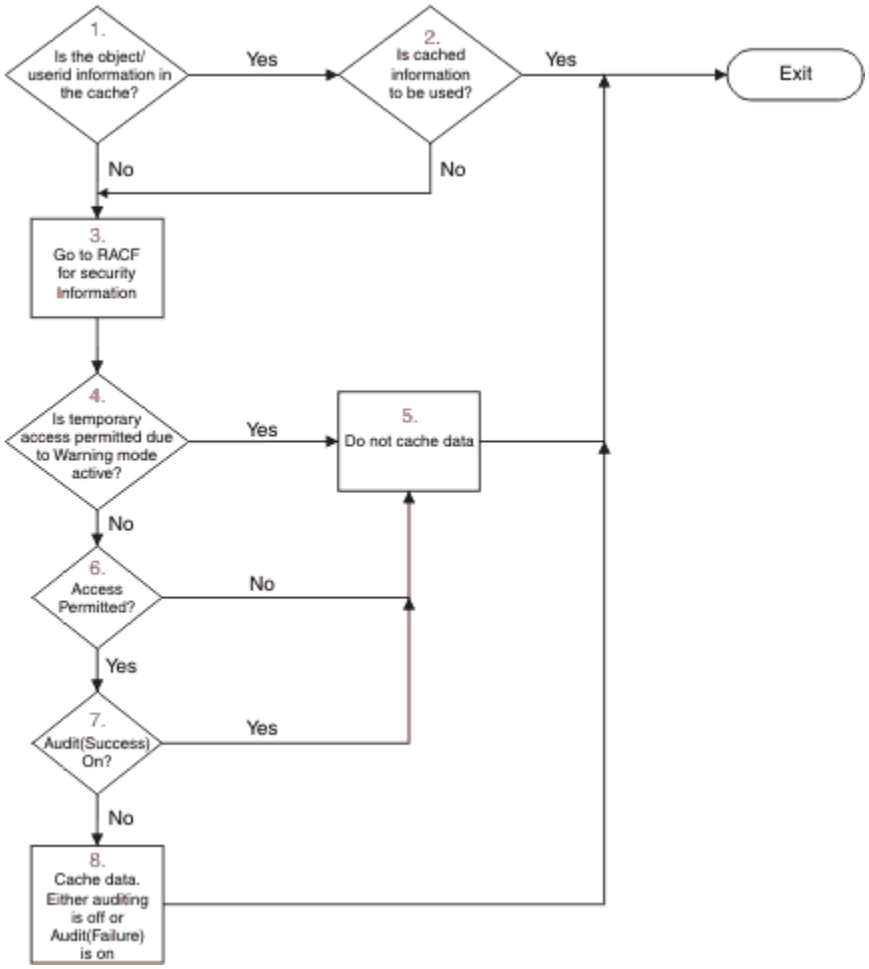



*Figure 17. Logic flow for IBM MQ security caching*

If you change your security settings by adding or deleting switch profiles in the MQADMIN or MXADMIN classes, use one of these commands to pick up these changes dynamically:

```
REFRESH SECURITY(*)
REFRESH SECURITY(MQADMIN)
REFRESH SECURITY(MXADMIN)
```

This means you can activate new security types, or deactivate them without having to restart the queue manager.

For performance reasons, these are the only classes affected by the REFRESH SECURITY command. You do not need to use REFRESH SECURITY if you change a profile in either the MQCONN or MQCMDS classes.

**Note:** A refresh of the MQADMIN or MXADMIN class is not required if you change a RESLEVEL security profile.

For performance reasons, use REFRESH SECURITY as infrequently as possible, ideally at off-peak times. You can minimize the number of security refreshes by connecting users to RACF groups that are already in the access list for IBM MQ profiles, rather than putting individual users in the access lists. In this way, you change the user rather than the resource profile. You can also RVERIFY SECURITY the appropriate user instead of refreshing security.

As an example of REFRESH SECURITY, suppose you define the new profiles to protect access to queues starting with INSURANCE.LIFE on queue manager PRMQ. You use these RACF commands:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

You must issue the following command to tell RACF to refresh the security information that it holds, for example:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

Because these profiles are generic, you must tell RACF to refresh the generic profiles for MQQUEUE. For example:

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

Then you must use this command to tell queue manager PRMQ that the queue profiles have changed:

```
REFRESH SECURITY(MQQUEUE)
```

### Refreshing SSL/TLS security

To refresh the cached view of the TLS Key Repository, issue the REFRESH SECURITY command with the option TYPE(SSL). This enables you to update some of your TLS settings without having to restart your channel initiator.

### z/OS *Displaying security status*

To display the status of the security switches, and other security controls, issue the MQSC DISPLAY SECURITY command.

The following figure shows typical output of the DISPLAY SECURITY ALL command.

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMELIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

*Figure 18. Typical output from the DISPLAY SECURITY command*

The example shows that the queue manager that replied to the command has subsystem, command, alternate user, process, namelist, and queue security active at queue manager level but not at queue sharing group level. Connection, command resource, and context security are not active. It also shows that user ID timeouts are active, and that every 12 minutes the queue manager checks for user IDs that have not been used in this queue manager for 54 minutes and removes them.

**Note:** This command shows the current security status. It does not necessarily reflect the current status of the switch profiles defined to RACF, or the status of the RACF classes. For example, the switch profiles might have been changed since the last restart of this queue manager or REFRESH SECURITY command.

## z/OS Security installation tasks for z/OS

After installing and customizing IBM MQ, authorize started task procedures to RACF, authorize access to various resources, and set up RACF definitions. Optionally, configure your system for TLS.

When IBM MQ is first installed and customized, you must perform these security-related tasks:

1. Set up IBM MQ data set and system security by:
   - Authorizing the queue manager started-task procedure xxxxMSTR and the distributed queuing started-task procedure xxxxCHIN to run under RACF.
   - Authorizing access to queue manager data sets.
   - Authorizing access to resources for those user IDs that will use the queue manager and utility programs.
   - Authorizing access for those queue managers that will use the coupling facility list structures.
   - Authorizing access for those queue managers that will use Db2.
2. Set up RACF definitions for IBM MQ security.
3. If you want to use Transport Layer Security (TLS), prepare your system to use certificates and keys.

## z/OS *Setting up IBM MQ for z/OS data set security*

There are many types of IBM MQ user. Use RACF to control their access to system data sets.

The possible users of IBM MQ data sets include the following entities:

- The queue manager itself.
- The channel initiator
- IBM MQ administrators, who need to create IBM MQ data sets, run utility programs, and similar tasks.
- Application programmers who need to use the IBM MQ-supplied copybooks, include data sets, macros, and similar resources.
- Applications involving one or more of:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.*

For all these potential users, protect the IBM MQ data sets with RACF.

You must also control access to all your 'CSQINP' data sets.

### z/OS *RACF authorization of started-task procedures*

Some IBM MQ data sets are for the exclusive use of the queue manager. If you protect your IBM MQ data sets using RACF, you must also authorize the queue manager started-task procedure xxxxMSTR, and the distributed queuing started-task procedure xxxxCHIN, using RACF. To do this, use the STARTED class. Alternatively, you can use the started procedures table (ICHRIN03), but then you must perform an IPL of your z/OS system before the changes take effect.

For more information, see the *z/OS Security Server RACF System Programmer's Guide*.

The RACF user ID identified must have the required access to the data sets in the started-task procedure. For example, if you associate a queue manager started task procedure called CSQ1MSTR with the RACF user ID QMGRCSQ1, the user ID QMGRCSQ1 must have access to the z/OS resources accessed by the CSQ1 queue manager.

Also, the content of the GROUP field in the user ID of the queue manager must be the same as the content of the GROUP field in the STARTED profile for that queue manager. If the content in each GROUP field does not match then the appropriate user ID is prevented from entering the system. This situation causes IBM MQ to run with an undefined user ID and consequently close due to a security violation.

The RACF user IDs associated with the queue manager and channel initiator started task procedures must not have the TRUSTED attribute set.

z/OS *Authorizing access to data sets*

The IBM MQ data sets should be protected so that no unauthorized user can run a queue manager instance, or gain access to any queue manager data. To do this, use normal z/OS RACF data set protection.

Table 65 on page 239 summarizes the RACF access that the queue manager started task procedure must have to the different data sets.

*Table 65. RACF access to data sets associated with a queue manager*

| RACF access | Data sets |
|---|---|
| READ | • `thlqual.SCSQAUTH` and `thlqual.SCSQANLx` (where x is the language letter for your national language).<br>• The data sets referred to by CSQINP1, CSQINP2 and CSQXLIB in the queue manager's started task procedure.<br>• SMDS data sets owned by other queue managers in the group.<br>• Log, BSDS and archive log data sets for other queue managers in the group. |
| UPDATE | • All page sets and log and BSDS data sets.<br>• SMDS data sets owned by a queue manager<br>• SMDS data sets owned by other queue managers in the group, for the structures that the queue manager performs the RECOVER CFSTRUCT command. |
| ALTER | • All archive log data sets. |

Table 66 on page 239 summarizes the RACF access that the started task procedure for distributed queuing must have to the different data sets.

*Table 66. RACF access to data sets associated with distributed queuing*

| RACF access | Data sets |
|---|---|
| READ | • thlqual.SCSQAUTH, thlqual.SCSQANLx (where x is the language letter for your national language), and thlqual.SCSQMVR1.<br>• LE library data sets.<br>• The data sets referred to by CSQXLIB and CSQINPX in the channel initiator started task procedure. |
| UPDATE | • Data sets CSQOUTX and CSQSNAP |

For more information, see the *z/OS Security Server RACF Security Administrator's Guide*.

z/OS *Encrypting data sets*

The IBM MQ data sets can be encrypted with z/OS data set encryption, so that the data is protected, or for regulatory reasons.

You can protect all page sets, active log, archive log, and bootstrap (BSDS) data sets with z/OS data set encryption.

**Attention:** You cannot protect shared message data sets (SMDS) with z/OS data set encryption by IBM MQ for z/OS 9.1.4 or earlier.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

## z/OS *Setting up IBM MQ for z/OS resource security*

There are many types of IBM MQ user. Use RACF to control their access to IBM MQ resources.

The possible users of IBM MQ resources, such as queues and channels include the following entities:

- The queue manager itself.
- The channel initiator
- IBM MQ administrators, who need to create IBM MQ data sets, run utility programs, and similar tasks
- Application programmers who need to use the IBM MQ-supplied copybooks, include data sets, macros, and similar resources.
- Applications involving one or more of:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.*

For all these potential users, protect the IBM MQ resources with RACF. In particular, note that the channel initiator needs access to various resources, as described in "Security considerations for the channel initiator on z/OS" on page 246, and so the user ID under which it runs must be authorized to access these resources.

If you are using a queue sharing group, the queue manager might issue various commands internally, so the user ID it uses must be authorized to issue such commands. The commands are:

- DEFINE, ALTER, and DELETE for every object that has QSGDISP(GROUP)
- START and STOP CHANNEL for every channel used with CHLDISP(SHARED)

## z/OS *Configuring your z/OS system to use TLS*

Use this topic as example of how to configure IBM MQ for z/OS with Transport Layer Security (TLS) using RACF commands.

If you want to use TLS for channel security, there are a number of tasks you need to perform on your system. (For details on using RACF commands for certificates and key repositories (key rings), see Working with TLS on z/OS .)

1. Create a key ring in RACF to hold all the keys and certificates for your system, using the RACF RACDCERT command. For example:

   ```
   RACDCERT ID(CHINUSER) ADDRING(QM1RING)
   ```

   The ID must be either the channel initiator address space user ID or the user ID you want to own the key ring if it is to be a shared key ring.

2. Create a digital certificate for each queue manager, using the RACF RACDCERT command.

   The label of the certificate must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details. In this example it is `ibmWebSphereMQQM1`.

   For example:

```
RACDCERT ID(USERID) GENCERT
SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
WITHLABEL('ibmWebSphereMQQM1')
```

3. Connect the certificate in RACF to the key ring, using the RACF RACDCERT command. For example:

```
RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
CONNECT ID(CHINUSER)
```

   You also need to connect any relevant signer certificates (from a certificate authority) to the key ring. That is, all certificate authorities for the TLS certificate of this queue manager and all certificate authorities for all TLS certificates that this queue manager communicates with. For example:

```
RACDCERT ID(CHINUSER)
CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
```

4. On each of your queue managers, use the IBM MQ ALTER QMGR command to specify the key repository that the queue manager needs to point to. For example, if the key ring is owned by the channel initiator address space:

```
ALTER QMGR SSLKEYR(QM1RING)
```

   or if you are using a shared key ring:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

   where *userid* is the user ID that owns the shared key ring.

5. Certificate Revocation Lists (CRLs) allow the certificate authorities to revoke certificates that can no longer be trusted. CRLs are stored in LDAP servers. To access this list on the LDAP server, you first need to create an AUTHINFO object of AUTHTYPE CRLLDAP, using the IBM MQ DEFINE AUTHINFO command. For example:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

   In this example, the certificate revocation list is stored in a public area of the LDAP server, so the LDAPUSER and LDAPPWD fields are not necessary.

   Next, put your AUTHINFO object into a namelist, using the IBM MQ DEFINE NAMELIST command. For example:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

Finally, associate the namelist with each queue manager, using the IBM MQ ALTER QMGR command. For example:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. Set up your queue manager to run TLS calls, using the IBM MQ ALTER QMGR command. This defines server subtasks that handle SSL calls only, which leaves the normal dispatchers to continue processing as normal without being affected by any SSL calls. You must have at least two of these subtasks. For example:

```
ALTER QMGR SSLTASKS(8)
```

   This change only takes effect when the channel initiator is restarted.

7. Specify the cipher specification to be used for each channel, using the IBM MQ DEFINE CHANNEL or ALTER CHANNEL command. For example:

```
ALTER CHANNEL(LDAPCHL)
CHLTYPE(SDR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

   Both ends of the channel must specify the same cipher specification.

## z/OS Managing channel authentication records in a QSG

Channel authentication records apply to the queue manager that they are created on, they are not shared throughout the queue sharing group (QSG). Therefore if all the queue managers in the queue sharing group are required to have the same rules, some management needs to be carried out to keep all the rules the consistent.

1. Always add the `CMDSCOPE(*)` option to all `SET CHLAUTH` commands. This will send the command to all running queue managers in the queue sharing group
2. Use the `DISPLAY CHLAUTH` command with the `CMDSCOPE(*)` option and then analyze the responses to see if the records are the same from all queue managers. When an inconsistency is found a `SET CHLAUTH` command can be issued containing the same rule with `CMDSCOPE(*)` or `CMDSCOPE(`*qmgr-name*`)`.
3. Add a member to the queue manager's CSQINP2 concatenation (see Initialization commands for details) that has the full set of rules. These will be read as part of the queue manager's initialization process. If the `SET CHLAUTH` command uses `ACTION(ADD)` the rule will only be added if it didn't exist. Using `ACTION(REPLACE)` will replace an existing rule if it already exists or add it if it does not. The same member could then be placed in the CSQINP2 concatenation of all queue managers in the queue sharing group.
4. Use the CSQUTIL utility (see Issuing commands to IBM MQ (COMMAND) for details) to extract the rules from one queue manager using either the `MAKEDEF` or `MAKEREP` option. Then replay the output using CSQUTIL into the target queue manager.

**Related concepts**

Channel authentication records

채널 레벨에서 연결 시스템에 부여된 액세스 권한에 대한 보다 정밀한 제어를 실행하려면 채널 인증 레코드를 사용하십시오.

## z/OS Auditing considerations on z/OS

The normal RACF auditing controls are available for conducting a security audit of a queue manager. IBM MQ does not gather any security statistics of its own. The only statistics are those that can be created by auditing.

RACF auditing can be based upon:

- User IDs
- Resource classes
- Profiles

For more details, see the *z/OS Security Server RACF Auditor's Guide*.

**Note:** Auditing degrades performance; the more auditing you implement, the more performance is degraded. This is also a consideration for the use of the RACF WARNING option.

### z/OS *Auditing RESLEVEL*

Use the RESAUDIT system parameter to control the production of RESLEVEL audit records. RACF GENERAL audit records are produced.

Produce RESLEVEL audit records by setting the RESAUDIT system parameter to YES. If the RESAUDIT parameter is set to NO, audit records are not produced. For more details about setting this parameter, see Using CSQ6SYSP.

If RESAUDIT is set to YES, no normal RACF audit records are taken when the RESLEVEL check is made to see what access an address space user ID has to the hlq.RESLEVEL profile. Instead, IBM MQ requests that RACF create a GENERAL audit record (event number 27). These checks are only carried out at connect time, so the performance cost is minimal.

**Attention:** RACFRW is no longer the suggested utility for processing RACF audit records. You should use the RACF SMF data unload utility as this is the preferred reporting method.

You can report the IBM MQ general audit records using the RACF report writer (RACFRW). You could use the following RACFRW commands to report the RESLEVEL access:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

A sample report from RACFRW, excluding the *Date*, *Time*, and *SYSID* fields, is shown in Figure 19 on page 243.

```
       RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE   4
                                E
                                V  Q
                                E  U
*JOB/USER *STEP/  --TERMINAL--  N  A
   NAME    GROUP     ID    LVL  T  L

 WS21B    MQMGRP IGJZM000   0   27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
    TRUSTED USER                      AUTH=(NONE),REASON=(NONE)
                                SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST
PROFILE(QM66.RESLEVEL),
                                CLASS(MQADMIN), ACCESS EQUATES TO
(CONTROL)',RESULT=SUCCESS,MQADMIN
```

*Figure 19. Sample output from RACFRW showing RESLEVEL general audit records*

From checking the LOGSTR data in this sample output, you can see that TSO user WS21B has CONTROL access to QM66.RESLEVEL. This means that all resource security checks are bypassed when user WS21B access QM66 resources.

For more information about using RACFRW, see The RACF report writer in the *z/OS Security Server RACF Auditor's Guide*.

## z/OS Customizing security

If you want to change the way IBM MQ security operates, you must do this through the SAF exit (ICHRFR00), or exits in your external security manager.

To find out more about RACF exits, see the *z/OS Security Server RACROUTE Macro Reference* documentation.

**Note:** Because IBM MQ optimizes calls to the ESM, RACROUTE requests might not be made on, for example, every open for a particular queue by a particular user.

## z/OS Security violation messages on z/OS

A security violation is indicated by the return code MQRC_NOT_AUTHORIZED in an application program or by a message in the job log.

A return code of MQRC_NOT_AUTHORIZED can be returned to an application program for the following reasons:

- A user is not allowed to connect to the queue manager. In this case, you get an ICH408I message in the Batch/TSO, CICS, or IMS job log.
- A user sign-on to the queue manager has failed because, for example, the job user ID is not valid or appropriate, or the task user ID or alternate user ID is not valid. One or more of these user IDs might not be valid because they have been revoked or deleted. In this case, you get an ICHxxxx message and possibly an IRRxxxx message in the queue manager job log giving the reason for the sign-on failure. For example:

```
ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
  LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
```

- An alternate user has been requested, but the job or task user ID does not have access to the alternate user ID. For this failure, you get a violation message in the job log of the relevant queue manager.
- A context option has been used or is implied by opening a transmission queue for output, but the job user ID or, where applicable, the task or alternate user ID does not have access to the context option. In this case, a violation message is put in the job log of the relevant queue manager.
- An unauthorized user has attempted to access a secured queue manager object, for example, a queue. In this case, an ICH408I message for the violation is put in the job log of the relevant queue manager. This violation might be due to the job or, when applicable, the task or alternate user ID.

Violation messages for command security and command resource security can also be found in the job log of the queue manager.

If the ICH408I violation message shows the queue manager jobname rather than a user ID, this is normally the result of a blank alternate user ID being specified. For example:

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

You can find out who is allowed to use blank alternate user IDs by checking the access list of the MQADMIN profile hlq.ALTERNATE.USER.-BLANK-.

An ICH408I violation message can also be generated by:

- A command being sent to the system-command input queue without context. User-written programs that write to the system-command input queue should always use a context option. For more information, see “Profiles for context security” on page 203.
- When the job accessing the IBM MQ resource does not have a user ID associated with it, or when an IBM MQ adapter cannot extract the user ID from the adapter environment.

Violation messages might also be issued if you are using both queue sharing group and queue manager level security. You might get messages indicating that no profile has been found at queue manager level, but still be granted access because of a queue sharing group level profile.

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

See the z/OS for Security Server RACF Messages and Codes documentation for more information on ICH408I messages.

## z/OS What to do if access is allowed or disallowed incorrectly

In addition to the information detailed in the z/OS documentation, use this checklist if access to a resource appears to be incorrectly controlled.

See the *z/OS Security Server RACF Security Administrator's Guide* for the detailed steps if access is allowed or disallowed.

- Are the switch profiles correctly set?
  - Is RACF active?
  - Are the IBM MQ RACF classes installed and active?

    Use the RACF command, SETROPTS LIST, to check this.
  - Use the IBM MQ DISPLAY SECURITY command to display the current switch status from the queue manager.
  - Check the switch profiles in the MQADMIN class.

    Use the RACF commands, SEARCH and RLIST, for this.
  - Recheck the RACF switch profiles by issuing the IBM MQ REFRESH SECURITY(MQADMIN) command.
- Has the RACF resource profile changed? For example, has universal access on the profile changed or has the access list of the profile changed?
  - Is the profile generic?

    If it is, issue the RACF command, SETROPTS GENERIC(classname) REFRESH.
  - Have you refreshed the security on this queue manager?

    If required, issue the RACF command SETROPTS RACLIST(classname) REFRESH.

    If required, issue the IBM MQ REFRESH SECURITY(*) command.
- Has the RACF definition of the user changed? For example, has the user been connected to a new group or has the user access authority been revoked?
  - Have you reverified the user by issuing the IBM MQ RVERIFY SECURITY(userid) command?
- Are security checks being bypassed due to RESLEVEL?
  - Check the connecting user ID's access to the RESLEVEL profile. Use the RACF audit records to determine what the RESLEVEL is set to.

- For channels, remember that the access level that the channel initiator's userid has to RESLEVEL is inherited by all channels, so an access level, such as ALTER, that causes all checks to be bypassed causes security checks to be bypassed for all channels.
- If you are running from CICS, check the transaction's RESSEC setting.
- If RESLEVEL has been changed while a user is connected, they must disconnect and reconnect before the new RESLEVEL setting takes effect.

- Are you using queue sharing groups?
  - If you are using both queue sharing group and queue manager level security, check that you have defined all the correct profiles. If queue manager profile is not defined, a message is sent to the log stating that the profile was not found.
  - Have you used a combination of switch settings that is not valid so that full security checking has been set on?
  - Do you need to define security switches to override some of the queue sharing group settings for your queue manager?
  - Is a queue manager level profile taking precedence over a queue sharing group level profile?

## z/OS Security considerations for the channel initiator on z/OS

If you are using resource security in a distributed queuing environment, the Channel initiator address space needs appropriate access to various IBM MQ resources. You can use the Integrated Cryptographic Support Facility (ICSF) to seed the password protection algorithm.

See the z/OS Cryptographic Services documentation for more information on ICSF.

### Using resource security

If you are using resource security, consider the following points if you are using distributed queuing:

**System queues**
The channel initiator address space needs RACF UPDATE access to the system queues listed at "System queue security" on page 194, and to all the user destination queues and the dead-letter queue (but see "Dead-letter queue security" on page 192 ).

**Transmission queues**
The channel initiator address space needs ALTER access to all the user transmission queues.

**Context security**
The channel user ID (and the MCA user ID if one has been specified) need RACF CONTROL access to the hlq.CONTEXT.queuename profiles in the MQADMIN class. Depending on the RESLEVEL profile, the channel user ID might also need CONTROL access to these profiles.

All channels need CONTROL access to the MQADMIN hlq.CONTEXT. dead-letter-queue profile. All channels (whether initiating or responding) can generate reports, and consequently they need CONTROL access to the hlq.CONTEXT.reply-q profile.

SENDER, CLUSSDR, and SERVER channels need CONTROL access to the hlq.CONTEXT.xmit-queue-name profiles since messages can be put onto the transmission queue to wake up the channel to end gracefully.

**Note:** If the channel user ID, or a RACF group to which the channel user ID is connected, has CONTROL or ALTER access to the hlq.RESLEVEL, then there are no resource checks for the channel initiator or any of its channels.

See "Profiles for context security" on page 203 "RESLEVEL and the channel initiator connection" on page 222 and "User IDs for security checking on z/OS" on page 224 for more information.

**CSQINPX**
If you are using the CSQINPX input data set, the channel initiator also needs READ access to CSQINPX, and UPDATE access to data set CSQOUTX and dynamic queues SYSTEM.CSQXCMD.*.

**Connection security**
The channel initiator address space connection requests use a connection type of CHIN, for which appropriate access security must be set, see "Connection security profiles for the channel initiator" on page 187.

**Data sets**
The channel initiator address space needs appropriate access to queue manager data sets, see "Authorizing access to data sets" on page 239.

**Commands**
The distributed queuing commands (for example, DEFINE CHANNEL, START CHINIT, START LISTENER, and other channel commands) must have appropriate command security set, see Table 49 on page 206.

If you are using a queue sharing group, the channel initiator might issue various commands internally, so the user ID it uses must be authorized to issue such commands. These commands are START and STOP CHANNEL for every channel used with CHLDISP(SHARED).

If the PSMODE of the queue manager is not DISABLED, the channel initiator must have READ access to the DISPLAY PUBSUB command.

**Channel security**
Channels, particularly receivers and server-connections, need appropriate security to be set up; see "User IDs for security checking on z/OS" on page 224 for more information.

You can also use the Transport Layer Security (TLS) protocol to provide security on channels. See "IBM MQ의 TLS 보안 프로토콜" on page 22 for more information about using TLS with IBM MQ.

See also "클라이언트의 액세스 제어" on page 96 for information about server-connection security.

**User IDs**
The user IDs described in "User IDs used by the channel initiator" on page 227 and "User IDs used by the intra-group queuing agent" on page 231 need the following access:

- RACF UPDATE access to the appropriate destination queues and the dead-letter queue
- RACF CONTROL access to the `hlq.CONTEXT.queuename` profile if context checking is performed at the receiver
- Appropriate access to the hlq.ALTERNATE.USER.userid profiles they might need to use.
- For clients, the appropriate RACF access to the resources to be used.

**APPC security**
Set appropriate APPC security if you are using the LU 6.2 transmission protocol. (Use the APPCLU RACF class for example.) For information about setting up security for APPC, see the following documentation:

- *z/OS MVS Planning: APPC Management*
- *z/OS MVS Programming: Writing Servers for APPC/MVS*

Outbound transmissions use the "SECURITY(SAME)" APPC option. As a result, the user ID of the channel initiator address space and its default profile ( RACF GROUP) are flowed across the network to the receiver with an indicator that the user ID has already been verified (ALREADYV).

If the receiving side is also z/OS, the user ID and profile are verified by APPC and the user ID is presented to the receiver channel and used as the channel user ID.

In an environment where the queue manager is using APPC to communicate with another queue manager on the same or another z/OS system, you need to ensure that either:

- The VTAM definition for the communicating LU specifies SETACPT(ALREADYV)
- There is a RACF APPCLU profile for the connection between LUs that specifies CONVSEC(ALREADYV)

**Changing security settings**
If the RACF access level that either the channel user ID or MCA user ID has to a destination queue is changed, this change takes effect only for new object handles (that is, new MQOPEN s) for the destination queue. The times when MCAs open and close queues is variable; if a channel is already running when such an access change is made, the MCA can continue to put messages on the destination queue using the existing security access of the user IDs rather than the updated security access. Stopping and restarting the channels to enforce the updated access level avoids this scenario.

**Automatic restart**
If you are using the z/OS Automatic Restart Manager (ARM) to restart the channel initiator, the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ START CHINIT command.

## Using the Integrated Cryptographic Service Facility (ICSF)

The channel initiator can use ICSF to generate a random number when seeding the password protection algorithm to obfuscate passwords flowing over client channels if TLS is not being used. The process of generating a random number is called *entropy*.

If you have the z/OS feature installed but have not started ICSF, you see message CSQX213E and the channel initiator uses STCK for entropy.

Message CSQX213E warns you that the password protection algorithm is not as secure as it could be. However, you can continue your process; there is no other impact on runtime.

If you do not have the z/OS feature installed, the channel initiator automatically uses STCK.

**Notes:**

1. Using ICSF for entropy generates more random sequences than using STCK.
2. If you start ICSF you must restart the channel initiator.
3. ICSF is required for certain CipherSpecs. If you attempt to use one of these CipherSpecs and you do not have ICSF installed, you receive message CSQX629E.

## z/OS Security in queue manager clusters on z/OS

Security considerations for clusters are the same for queue managers and channels that are not clustered. The channel initiator needs access to some additional system queues, and some additional commands need appropriate security set.

You can use the MCA user ID, channel authentication records, TLS, and security exits to authenticate cluster channels (as with conventional channels). The channel authentication records or security exit relating to the cluster-receiver channel must check that the remote queue manager is permitted access to the server queue manager's cluster queues. You can start to use IBM MQ cluster support without changing your existing queue access security. You must, however, allow other queue managers in the cluster to write to the SYSTEM.CLUSTER.COMMAND.QUEUE if they are to join the cluster.

IBM MQ cluster support does not provide a mechanism to limit a member of a cluster to the client role only. As a result, you must be sure that you trust any queue managers that you allow into the cluster. If any queue manager in the cluster creates a queue with a particular name, it can receive messages for that queue, regardless of whether the application putting messages to that queue intended this or not.

To restrict the membership of a cluster, take the same action that you would take to prevent queue managers connecting to receiver channels. You restrict the membership of a cluster by using channel authentication records or by writing a security exit program on the receiver channel. You can also write an exit program to prevent unauthorized queue managers from writing to the SYSTEM.CLUSTER.COMMAND.QUEUE.

**Note:** It is not advisable to permit applications to open the SYSTEM.CLUSTER.TRANSMIT.QUEUE directly. It is also not advisable to permit an application to open any other transmission queue directly.

If you are using resource security, consider the following points in addition to the considerations contained in “Security considerations for the channel initiator on z/OS” on page 246:

**System queues**
The channel initiator needs RACF ALTER access to the following system queues:

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

and UPDATE access to SYSTEM.CLUSTER.REPOSITORY.QUEUE

It also needs READ access to any namelists used for clustering.

**Commands**
Set appropriate command security (as described in Table 49 on page 206 ) for the cluster support commands (REFRESH and RESET CLUSTER, SUSPEND, and RESUME QMGR.

## z/OS Security considerations for using IBM MQ with CICS

All the CICS versions supported by IBM MQ 9.0.0, and later, use the CICS supplied version of the adapter and bridge.

For details of security considerations, see:

- Security for the CICS-MQ adapter.
- Security for the CICS-MQ bridge.

## z/OS Security considerations for using IBM MQ with IMS

Use this topic to plan your security requirements when you use IBM MQ with IMS.

### Using the OPERCMDS class

If you are using RACF to protect resources in the OPERCMDS class, ensure that the userid associated with your IBM MQ queue manager address space has authority to issue the MODIFY command to any IMS system to which it can connect.

### Security considerations for the IMS bridge

There are four aspects that you must consider when deciding your security requirements for the IMS bridge, these are:

- What security authorization is needed to connect IBM MQ to IMS
- How much security checking is performed on applications using the bridge to access IMS
- Which IMS resources these applications are allowed to use
- What authority is to be used for messages that are put and got by the bridge

When you define your security requirements for the IMS bridge you must consider the following:

- Messages passing across the bridge might have originated from applications on platforms that do not offer strong security features
- Messages passing across the bridge might have originated from applications that are not controlled by the same enterprise or organization

#### z/OS *Security considerations for connecting to IMS*

Grant the user ID of the IBM MQ queue manager address space access to the OTMA group.

The IMS bridge is an OTMA client. The connection to IMS operates under the user ID of the IBM MQ queue manager address space. This is normally defined as a member of the started task group. This user ID must be granted access to the OTMA group (unless the /SECURE OTMA setting is NONE).

To do this, define the following profile in the FACILITY class:

```
IMSXCF.xcfgname.mqxcfmname
```

Where `xcfgname` is the XCF group name and `mqxcfmname` is the XCF member name of IBM MQ.

You must give your IBM MQ queue manager user ID read access to this profile.

**Note:**

1. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
2. If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, no user ID is passed to IMS and the connection fails unless the /SECURE OTMA setting is NONE.

### z/OS *Application access control for the IMS bridge*

Define a RACF profile in the FACILITY class for each IMS system. Grant an appropriate level of access to the IBM MQ queue manager user ID.

For each IMS system that the IMS bridge connects to, you can define the following RACF profile in the FACILITY class to determine how much security checking is performed for each message passed to the IMS system.

```
IMSXCF.xcfgname.imsxcfmname
```

Where `xcfgname` is the XCF group name and `imsxcfmname` is the XCF member name for IMS. (You need to define a separate profile for each IMS system.)

The access level you allow for the IBM MQ queue manager user ID in this profile is returned to IBM MQ when the IMS bridge connects to IMS, and indicates the level of security that is required on subsequent transactions. For subsequent transactions, IBM MQ requests the appropriate services from RACF and, where the user ID is authorized, passes the message to IMS.

OTMA does not support the IMS /SIGN command; however, IBM MQ allows you to set the access checking for each message to enable implementation of the necessary level of control.

The following access level information can be returned:

**NONE or NO PROFILE FOUND**

These values indicate that maximum security is required, that is, authentication is required for every transaction. A check is made to verify that the user ID specified in the *UserIdentifier* field of the MQMD structure, and the password or PassTicket in the *Authenticator* field of the MQIIH structure are known to RACF, and are a valid combination. A UTOKEN is created with a password or PassTicket, and passed to IMS ; the UTOKEN is not cached.

**Note:** If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, this level of security overrides whatever is defined in the profile.

**READ**

This value indicates that the same authentication is to be performed as for `NONE` under the following circumstances:

- The first time that a specific user ID is encountered
- When the user ID has been encountered before but the cached UTOKEN was not created with a password or PassTicket

IBM MQ requests a UTOKEN if required, and passes it to IMS.

**Note:** If a request to reverify security has been acted on, all cached information is lost and a UTOKEN is requested the first time each user ID is later encountered.

**UPDATE**
A check is made that the user ID in the *UserIdentifier* field of the MQMD structure is known to RACF.

A UTOKEN is built and passed to IMS ; the UTOKEN is cached.

**CONTROL/ALTER**
These values indicate that no security UTOKENs need to be provided for any user IDs for this IMS system. (You would probably only use this option for development and test systems.)

**Attention:** Note that the user ID contained in the *UserIdentifier* field of the MQMD structure is still passed for **CONTROL/ALTER**.

**Note:**

1. This access is defined when IBM MQ connects to IMS, and lasts for the duration of the connection. To change the security level, the access to the security profile must be changed and then the bridge stopped and restarted (for example, by stopping and restarting OTMA).
2. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
3. You can use a password or a PassTicket, but you must remember that the IMS bridge does not encrypt data. For information about using PassTickets, see "Using RACF PassTickets in the IMS header" on page 252.
4. Some of these results might be affected by security settings in IMS, using the /SECURE OTMA command.
5. Cached UTOKEN information is held for the duration defined by the INTERVAL and TIMEOUT parameters of the IBM MQ ALTER SECURITY command.
6. The RACF WARNING option has no effect on the IMSXCF.xcfgname.imsxcfmname profile. Its use does not affect the level of access granted, and no RACF WARNING messages are produced.

## z/OS *Security checking on IMS*

Messages that pass across the bridge contain security information. The security checks made depend on the setting of the IMS command /SECURE OTMA.

Each IBM MQ message that passes across the bridge contains the following security information:

- A user ID contained in the *UserIdentifier* field of the MQMD structure
- The security scope contained in the *SecurityScope* field of the MQIIH structure (if the MQIIH structure is present)
- A UTOKEN (unless the IBM MQ sub system has CONTROL or ALTER access to the relevant `IMSXCF.xcfgname.imsxcfmname` profile)

The security checks made depend on the setting of the IMS command /SECURE OTMA, as follows:

**/SECURE OTMA NONE**
No security checks are made for the transaction.

**/SECURE OTMA CHECK**
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE (Accessor Environment Element) is built in the IMS control region.

**/SECURE OTMA FULL**
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE is built in the IMS dependent region as well as the IMS control region.

**/SECURE OTMA PROFILE**
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking

The *SecurityScope* field in the MQIIH structure is used to determine whether to build an ACEE in the IMS dependent region as well as the control region.

**Note:**

1. If you change the authorities in the TIMS or CIMS class, or the associated group classes GIMS or DIMS, you must issue the following IMS commands to activate the changes:
   - /MODIFY PREPARE RACF
   - /MODIFY COMMIT
2. If you do not use /SECURE OTMA PROFILE, any value specified in the **SecurityScope** field of the MQIIH structure is ignored.

### z/OS *Security checking done by the IMS bridge*

Different authorities are used depending on the action being performed.

When the bridge puts or gets a message, the following authorities are used:

**Getting a message from the bridge queue**
No security checks are performed.

**Putting an exception, or COA report message**
Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure.

**Putting a reply message**
Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure of the original message

**Putting a message to the dead-letter queue**
No security checks are performed.

**Note:**

1. If you change the IBM MQ class profiles, you must issue the IBM MQ REFRESH SECURITY(*) command to activate the changes.
2. If you change the authority of a user, you must issue the MQSC RVERIFY SECURITY command to activate the change.

### z/OS *Using RACF PassTickets in the IMS header*

You can use a PassTicket in place of a password in the IMS header.

If you want to use a PassTicket instead of a password in the IMS header (MQIIH), specify the application name against which the PassTicket is validated in the PASSTKTA attribute of the STGCLASS definition of the IMS bridge queue to which the message is to be routed.

If the PASSTKTA value is left blank, you must arrange to have a PassTicket generated. The application name in this case must be of the form MVSxxxx, where xxxx is the SMFID of the z/OS system on which the target queue manager runs.

A PassTicket is built from a user ID, the target application name, and a secret key. It is an 8-byte value containing uppercase alphabetic and numeric characters. It can be used only once, and is valid for a 20 minute period. If a PassTicket is generated by a local RACF system, RACF only checks that the profile exists and not that the user has authority against the profile. If the PassTicket was generated on a remote system, RACF validates the access of the user ID to the profile. For full information about PassTickets, see the *z/OS Security Server RACF Security Administrator's Guide*.

PassTickets in IMS headers are given to RACF by IBM MQ, not IMS.

## z/OS Migrating a z/OS queue manager to mixed-case security

Follow these steps to migrate a queue manager to mixed-case security. You review the level of security product you are using and activate the new IBM MQ external security manager classes. Run the **REFRESH SECURITY** command to activate the mixed-case profiles.

### Before you begin

1. Ensure all IBM MQ external security manager classes are activated.
2. Ensure your queue manager is started.

### About this task

Follow these steps to convert a queue manager to mixed-case security.

### Procedure

1. Copy all your existing profiles and access levels from the uppercase classes to the equivalent mixed-case external security manager class.
   a) MQADMIN to MXADMIN.
   b) MQPROC to MXPROC.
   c) MQNLIST to MXNLIST.
   d) MQQUEUE to MXQUEUE.
2. Change the value of the SCYCASE queue manager attribute to MIXED by issuing the following command.

```
ALTER QMGR SCYCASE(MIXED)
```

3. Activate the security profiles by issuing the following command.

```
REFRESH SECURITY(*) TYPE(CLASSES)
```

4. Test that your security profiles are working correctly.

### What to do next

Review your object definitions and create new mixed-case profiles as appropriate, using the **REFRESH SECURITY** command as required to activate the profiles.

## IBM MQ MQI client 보안 설정

클라이언트 애플리케이션이 서버에서 자원에 대한 무제한 액세스를 갖지 않도록 IBM MQ MQI client 보안을 고려해야 합니다.

클라이언트 애플리케이션을 실행할 때 필요한 것보다 많은 액세스 권한을 가지고 있는 사용자 ID를 사용하여 애플리케이션을 실행하지 마십시오. 예를 들어, mqm 그룹의 사용자 또는 mqm 사용자 자체입니다.

너무 많은 액세스 권한을 가진 사용자로서 애플리케이션을 실행하면 애플리케이션이 실수로 또는 악의적으로 큐 관리자의 일부에 액세스하고 변경하는 위험이 있을 수 있습니다.

클라이언트 애플리케이션과 해당 큐 관리자 서버 간에는 인증 및 액세스 제어라는 두 가지 측면의 보안이 있습니다.

- 인증은 특정 사용자로서 실행 중인 클라이언트 애플리케이션이 본인이 주장하는 사용자가 맞는지를 확인하는 데 사용할 수 있습니다. 인증을 사용하면 공격자가 애플리케이션 중 하나를 위장하여 사용자의 큐 관리자에 대한 액세스를 얻는 것을 방지할 수 있습니다.

  인증은 다음 두 옵션 중 하나로 제공됩니다.

  - 연결 인증 기능.

    연결 인증에 대한 자세한 정보는 66 페이지의 『연결 인증』의 내용을 참조하십시오.
  - TLS 내에서 상호 인증 사용.

    TLS에 대한 자세한 정보는 260 페이지의 『SSL/TLS에 대한 작업』의 내용을 참조하십시오.

- 특정 사용자 또는 사용자 그룹의 액세스 권한을 제공하거나 제거하기 위해 액세스 제어를 사용할 수 있습니다. 특별히 작성된 사용자(또는 특정 그룹에 있는 사용자)로 클라이언트 애플리케이션을 실행하면 애플리케이션이 액세스할 수 없어야 하는 큐 관리자의 일부에 액세스할 수 없도록 하기 위해 액세스 제어를 사용할 수 있습니다.

  액세스 제어를 설정할 때에는 채널 인증 규칙 및 채널의 MCAUSER 필드를 고려해야 합니다. 이러한 두 기능에는 모두 액세스 제어 권한을 확인하는 데 사용되는 사용자 ID를 변경하는 기능이 있습니다.

  액세스 제어에 대한 자세한 정보는 329 페이지의 『오브젝트에 액세스 권한 부여』의 내용을 참조하십시오.

제한된 ID로 특정 채널에 액세스하기 위해 클라이언트 애플리케이션을 설정했지만 채널에는 해당 MCAUSER 필드에 설정된 관리자 ID가 있는 경우에는, 클라이언트 애플리케이션이 성공적으로 연결하는 경우 관리자 ID는 액세스 제어 검사에 사용됩니다. 그러므로 클라이언트 애플리케이션은 큐 관리자에 대한 전체 액세스 권한을 가지고 있습니다.

MCAUSER 속성에 대한 자세한 정보는 362 페이지의 『MCAUSER 사용자 ID에 클라이언트 사용자 ID 맵핑』의 내용을 참조하십시오.

채널 인증 규칙은 승인될 연결을 위해 특정 규칙 및 기준을 설정하여 큐 관리자에 대한 액세스 제어를 위한 메소드로서 사용될 수도 있습니다.

채널 인증 규칙에 대한 자세한 정보는 47 페이지의 『채널 인증 레코드』의 내용을 참조하십시오.

## MQI 클라이언트에서 런타임 시 FIPS 인증 CipherSpec만 사용하도록 지정

FIPS 준수 소프트웨어를 사용하여 키 저장소를 작성한 다음 채널이 FIPS-인증 CipherSpec을 사용해야 함을 지정하십시오.

**참고:** AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

런타임 시 FIPS를 준수하려면 `-fips` 옵션과 함께 **runmqakm** 와 같은 FIPS 준수 소프트웨어만 사용하여 키 저장소를 작성하고 관리해야 합니다.

TLS 채널은 우선순위 순서로 나열된 세 가지 방법으로 FIPS-인증 CipherSpec만을 사용해야 함을 지정할 수 있습니다.

1. MQSCO 구조의 FipsRequired 필드를 MQSSL_FIPS_YES로 설정하십시오.
2. 환경 변수 **MQSSLFIPS** 를 YES로 설정하십시오.
3. 클라이언트 구성 파일의 SSL 스탠자에서 **SSLFipsRequired** 속성을 YES로 설정하십시오.

기본적으로 FIPS-인증 CipherSpec은 필요하지 않습니다.

이러한 값은 **ALTER QMGR SSLFIPS** 의 동등한 매개변수 값과 동일한 의미를 갖습니다 ( **ALTER QMGR** (큐 관리자 설정 대체)참조). 현재 클라이언트 프로세스에 활성 TLS 연결이 없고, FipsRequired 값이 SSL MQCONNX에 올바르게 지정된 경우에는 이 프로세스와 연관된 모든 후속 TLS 연결은 이 값과 연관된 CipherSpec만을 사용해야 합니다. 이는 이 연결 및 다른 모든 TLS 연결이 중지될 때까지 적용되며, 이 단계에서 후속 MQCONNX가 FipsRequired에 새 값을 제공할 수 있습니다.

암호화 하드웨어가 존재하면 IBM MQ에 의해 사용되는 암호화 모듈은 하드웨어 제품이 제공하는 모듈로 구성될 수 있고, 이들은 특정 레벨로 FIPS-인증일 수도 있습니다. 구성 가능한 모듈 및 이들이 FIPS-인증인지 여부는 사용 중인 하드웨어 제품에 따라 다릅니다.

가능한 경우, FIPS 전용 CipherSpecs 가 구성되면 MQI 클라이언트는 MQRC_SSL_INITIALIZATION_ERROR 와 함께 비FIPS CipherSpec 을 지정하는 연결을 거부합니다. IBM MQ는 이러한 모든 연결을 거부한다고 보장하지 않으며 IBM MQ 구성이 FIPS 준수인지 판별하는 것은 사용자의 책임입니다.

**관련 개념**

32 페이지의 『AIX, Linux, and Windows용 FIPS(Federal Information Processing Standard)』

AIX, Linux, and Windows 시스템의 SSL/TLS 채널에서 암호화가 필요한 경우 IBM MQ 는 IBM Crypto for C (ICC)라는 암호화 패키지를 사용합니다. AIX, Linux, and Windows 플랫폼에서 ICC 소프트웨어는 미국 국립 표준 기술 연구소 (US National Institute of Standards and Technology) 의 FIPS (Federal Information Processing Standards) Cryptomodule Validation Program 레벨 140-2를 통과했습니다.

## AIX AIX 에서 GSKit 8.0 의 다중 설치를 사용하여 TLS 클라이언트 애플리케이션 실행

AIX 의 TLS 클라이언트 애플리케이션은 다중 IBM Global Security Kit (GSKit) 8.0 설치가 있는 AIX 시스템에서 실행할 때 MQRC_CHANNEL_CONFIG_ERROR 및 오류 AMQ6175 가 발생할 수 있습니다.

다중 GSKit 8.0 설치가 있는 AIX 시스템에서 클라이언트 애플리케이션을 실행할 때 클라이언트 연결 호출은 TLS 를 사용할 때 MQRC_CHANNEL_CONFIG_ERROR 를 리턴할 수 있습니다. /var/mqm/errors는 실패한 클라이언트 애플리케이션에 대해 AMQ6175 및 AMQ9220 레코드 오류를 로그합니다. 예를 들어, 다음과 같습니다.

```
09/08/11 11:16:13 - Process(24412.1) User(user) Program(example)
Host(machine.example.ibm.com) Installation(Installation1)
VRMF(7.1.0.0)
AMQ6175: The system could not dynamically load the shared library
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so'. The system returned
error number '8' and error message 'Symbol resolution failed
for /usr/mqm/gskit8/lib64/libgsk8ssl_64.so because:
Symbol VALUE_EC_NamedCurve_secp256r1__9GSKASNOID (number 16) is not
exported from dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_NamedCurve_secp384r1__9GSKASNOID (number 17) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_NamedCurve_secp521r1__9GSKASNOID (number 18) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecPublicKey__9GSKASNOID (number 19) is not exported from dependent
module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecdsa_with_SHA1__9GSKASNOID (number 20) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecdsa__9GSKASNOID (number 21) is not exported from dependent
module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.'.

EXPLANATION:
This message applies to AIX systems. The shared library
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so' failed
to load correctly due to a problem with the library.
ACTION:
Check the file access permissions and that the file has not been corrupted.
----- amqxufnx.c : 1284 ---------------------------------------------------------
09/08/11 11:16:13 - Process(24412.1) User(user) Program(example)
Host(machine.example.ibm.com) Installation(Installation1)
VRMF(7.1.0.0)
AMQ9220: The GSKit communications program could not be loaded.

EXPLANATION:
The attempt to load the GSKit library or procedure
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so' failed with error code
536895861.
ACTION:
Either the library must be installed on the system or the environment changed
to allow the program to locate it.
----- amqcgska.c : 836 ----------------------------------------------------------
```

이 오류의 일반적인 원인은 LIBPATH 또는 LD_LIBRARY_PATH 환경 변수의 설정으로 인해 IBM MQ 클라이언트가 두 개의 다른 GSKit 8.0 설치에서 혼합 라이브러리 세트를 로드했기 때문입니다. Db2 환경에서 IBM MQ 클라이언트 애플리케이션을 실행하면 이 오류가 발생할 수 있습니다.

이 오류를 피하려면 IBM MQ 라이브러리가 우선할 수 있도록 라이브러리 경로 앞에 IBM MQ 라이브러리 디렉토리를 포함하십시오. 이는 **setmqenv** 명령을 **-k** 매개변수와 함께 사용하여 달성할 수 있습니다. 예를 들어 다음과 같습니다.

```
. /usr/mqm/bin/setmqenv -s -k
```

**setmqenv** 명령의 사용에 대한 자세한 정보는 setmqenv(IBM MQ 환경 설정)를 참조하십시오.

# MQSC를 사용하여 TLS 채널 구성

TLS 채널을 구성하려면 **runmqsc** 및 ALTER CHANNEL 명령을 사용하십시오. 제공된 값과 일치하는 소유자의 식별 이름으로 된 속성이 있는 인증서만을 승인하도록 채널을 선택적으로 구성할 수 있습니다. 시작하는 쪽에서 자체 개인 인증서를 송신하지 않으면 큐 관리자가 연결을 거부할 수 있도록 선택적으로 큐 관리자 채널을 구성할 수도 있습니다.

## 이 태스크 정보

IBM MQ Explorer에서 채널을 구성하려면 IBM MQ Explorer를 사용하여 TLS 채널 구성을 참조하십시오.

**runmqsc**를 사용하여 채널을 구성하려면 다음 단계를 완료하십시오.

## 프로시저

1. 대상 큐 관리자에 연결하는 **runmqsc** 명령을 호출하십시오.
2. TLS에 대해 사용으로 설정할 채널을 식별하십시오.
   채널 이름과 채널 유형을 모두 기록해 두십시오.
3. ALTER CHANNEL 명령을 사용하여 IBM MQ 채널의 다양한 특성을 대체할 수 있습니다.
   명령 외에 채널 이름 및 채널 유형을 제공합니다. 예를 들어, MQ.TEST 는 다음 명령을 실행합니다.

   ```
   ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR)
   ```

   IBM MQ 채널 정의에서 조정할 수 있는 TLS와 관련된 다양한 채널 속성이 있습니다.

## 다음에 수행할 작업

메시지 보안 설정

TLS 사용 가능 메시징은 메시지 보안을 보장하는 두 가지 방법을 제공합니다.

- 암호화는 메시지를 가로챈 경우 읽을 수 없게 합니다.
- 해시 기능은 메시지가 변경된 경우, 이를 감지하도록 합니다.

이러한 방법의 조합은 암호 스펙 또는 CipherSpec이라고 합니다. 동일한 CipherSpec이 채널의 양측에 설정되어야 합니다. 그렇지 않으면 TLS 사용 가능 메시징이 실패합니다. 자세한 정보는 7 페이지의 『IBM MQ 보호』의 내용을 참조하십시오.

IBM MQ 채널 사용 TLS를 대체하려면 SSLCIPH 속성에 값을 지정하십시오. 이 속성은 392 페이지의 『CipherSpec 사용 가능』 목록에서 큐 관리자의 큐 플랫폼에 대해 올바른 CipherSpec 으로 설정되어야 합니다.

TLS를 사용 안함으로 설정하도록 IBM MQ 채널을 대체하려면 SSLCIPH를 공백 값으로 설정하십시오. 예를 들면, 다음과 같습니다.

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

**참고:** 문자 대소문자를 유지하려면 채널 이름을 작은따옴표로 묶어야 합니다. 작은따옴표 없이 IBM MQ 는 문자열을 모두 대문자로 변환합니다.

고유 이름으로 인증서 필터링

인증서에는 인증서 소유자의 식별 이름이 포함되어 있습니다. 제공된 값과 일치하는 소유자의 식별 이름으로 된 속성이 있는 인증서만을 승인하도록 채널을 선택적으로 구성할 수 있습니다.

IBM MQ가 필터링할 수 있는 속성 이름은 다음 표에 나열되어 있습니다.

| 속성 이름 | 의미 |
|---|---|
| SERIALNUMBER | 인증서 일련 번호 |
| 메일 | 이메일 주소 |

| 속성 이름 | 의미 |
| --- | --- |
| Deprecated E | 이메일 주소(MAIL보다 우선적으로는 더 이상 사용되지 않음) |
| UID 또는 USERID | 사용자 ID |
| CN | 공용 이름 |
| T | 제목 |
| OU | 조직 단위 이름 |
| DC | 도메인 컴포넌트 |
| O | 조직 이름 |
| STREET | 상세 주소/주소 두 번째 줄 |
| L | 지역 이름 |
| ST(또는 SP 또는 S) | 시 또는 도 이름 |
| PC | 우편번호 |
| C | 국가 |
| UNSTRUCTUREDNAME | 호스트 이름 |
| UNSTRUCTUREDADDRESS | IP 주소 |
| DNQ | 식별 이름 규정자 |

임의의 수의 문자 대신 속성 값의 시작 또는 끝에 와일드카드 문자 (*) 를 사용할 수 있습니다. 예를 들어, GB의 IBM에서 일하며 이름이 Smith로 끝나는 사용자로부터만 인증서를 승인하려는 경우, 다음을 입력하십시오.

```
CN=*Smith, O=IBM, C=GB
```

예를 들면, 다음과 같습니다.

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

**참고:** 문자 대소문자가 유지되도록 하려면 SSLPEER 문자열을 작은따옴표로 묶어야 합니다. 작은따옴표 없이 IBM MQ 는 문자열을 모두 대문자로 변환합니다.

큐 관리자에 대한 연결을 시작하는 관계자 인증

다른 당사자가 큐 관리자에 대한 TLS 사용 가능 연결을 시작하면 큐 관리자가 ID를 증명하기 위해 시작하는 당사자에게 해당 개인 인증서를 송신해야 합니다. 시작하는 쪽에서 자체 개인 인증서를 송신하지 않으면, 큐 관리자가 연결을 거부할 수 있도록 선택적으로 큐 관리자 채널을 구성할 수도 있습니다.

이를 수행하려면 SSLCAUTH 속성을 설정하십시오. 이 속성은 부울 속성이며 OPTIONAL 또는 REQUIRED값을 가질 수 있습니다.

- OPTIONAL은 연결 클라이언트의 인증서가 제공되었지만 클라이언트가 인증서를 전송할 필요가 없는 경우 연결 클라이언트의 인증서를 인증합니다. 클라이언트가 유효하지 않은 인증서를 보내면 거부됩니다.
- 필수는 올바른 TLS 인증서를 제공하지 않는 모든 연결 클라이언트를 거부합니다.

예를 들면, 다음과 같습니다.

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

## IBM i IBM i에서 SSL 또는 TLS에 대한 통신 설정

SSL 또는 TLS 암호화 보안 프로토콜을 사용하는 보안 통신은 통신 채널 설정 및 인증하는 데 사용할 디지털 인증서 관리를 수반합니다.

SSL 또는 TLS 설치를 설정하려면 SSL 또는 TLS를 사용하도록 채널을 정의해야 합니다. 디지털 인증서도 작성하고 관리해야 합니다. 일부 운영 체제에서는 자체 서명 인증서로 테스트를 수행할 수 있습니다. 하지만 IBM i에서는 로컬 CA가 서명한 개인 인증서를 사용해야 합니다.

인증서 작성 및 관리에 대한 전체 정보는 260 페이지의 『IBM i에서 SSL/TLS에 대한 작업』의 내용을 참조하십시오.

이 주제 모음에서는 SSL 또는 TLS 통신 설정과 관련된 태스크 중 일부를 소개하고 이러한 태스크를 완료하는 것과 관련된 단계별 지침을 제공합니다.

SSL 및 TLS 프로토콜의 선택적 부분인 SSL 또는 TLS 클라이언트 인증도 테스트하려고 할 수 있습니다. SSL 또는 TLS 데이터 교환 동안에 SSL 또는 TLS 클라이언트는 항상 서버로부터 디지털 인증서를 확보하고 유효성 검증합니다. IBM MQ 구현을 사용하는 경우 SSL 또는 TLS 서버는 항상 클라이언트로부터 인증서를 요청합니다.

IBM i의 경우 SSL 또는 TLS 클라이언트는 올바른 IBM MQ 형식으로 레이블 지정된 인증서가 있는 경우에만 인증서를 전송합니다.

- 큐 관리자의 경우는 소문자로 변경된 큐 관리자 이름이 뒤따라오는 `ibmwebspheremq`입니다. 예를 들어, `QM1`의 경우 `ibmwebspheremqqm1`입니다.
- IBM i용 IBM MQ C 클라이언트의 경우, `ibmwebspheremq` 뒤에 로그온 사용자 ID가 소문자로 변경됩니다(예: `ibmwebspheremqmyuserid`).

IBM MQ는 레이블에 `ibmwebspheremq` 접두부를 사용하여 다른 제품에 대한 인증서와의 혼동을 피합니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). SSL 또는 TLS 클라이언트가 인증서를 전송하지 않으면 SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 SSLCAUTH 매개변수가 REQUIRED로 설정되거나 SSLPEER 매개변수 값이 설정된 상태로 정의된 경우에만 인증이 실패합니다. 자세한 정보는 SSL 또는 TLS를 사용하여 두 개의 큐 관리자 연결을 참조하십시오.

## ALW AIX, Linux, and Windows에서 SSL 또는 TLS에 대한 통신 설정

SSL 또는 TLS 암호화 보안 프로토콜을 사용하는 보안 통신은 통신 채널 설정 및 인증하는 데 사용할 디지털 인증서 관리를 수반합니다.

SSL 또는 TLS 설치를 설정하려면 SSL 또는 TLS를 사용하도록 채널을 정의해야 합니다. 디지털 인증서도 작성하고 관리해야 합니다. AIX, Linux, and Windows 시스템에서 자체 서명 인증서로 테스트를 수행할 수 있습니다.

**주의:** TLS 지원 채널을 사용하여 함께 조인할 큐 관리자에서 Elliptic Curve 서명 인증서와 RSA 서명 인증서를 혼합해서 사용하는 것은 불가능합니다.

TLS 지원 채널을 사용하는 큐 관리자는 모두 RSA 서명 인증서를 사용하거나 모두 EC 서명 인증서를 사용해야 합니다. 이 둘을 혼합해서 사용할 수는 없습니다.

자세한 정보는 43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』의 내용을 참조하십시오.

자체 서명 인증서는 폐기할 수 없으므로 개인 키가 손상된 후 공격자가 ID를 모방할 수 있습니다. CA는 손상된 인증서를 폐기할 수 있으며 폐기된 후에는 더 이상 사용할 수 없습니다. 따라서 CA 서명 인증서는 프로덕션 환경에서 사용하기에 더 안전하며 자체 서명 인증서는 테스트 시스템에서 더 편리합니다.

인증서 작성 및 관리에 대한 전체 정보는 276 페이지의 『AIX, Linux, and Windows에서 SSL/TLS에 대한 작업』의 내용을 참조하십시오.

이 주제 모음에서는 SSL 통신 설정과 관련된 태스크 중 일부를 소개하고 이러한 태스크를 완료하는 것과 관련된 단계별 지침을 제공합니다.

프로토콜의 선택적 부분인 SSL 또는 TLS 클라이언트 인증도 테스트하려고 할 수 있습니다. SSL 또는 TLS 데이터 교환 동안에 SSL 또는 TLS 클라이언트는 항상 서버로부터 디지털 인증서를 확보하고 유효성 검증합니다. IBM MQ 구현을 사용하는 경우 SSL 또는 TLS 서버는 항상 클라이언트로부터 인증서를 요청합니다.

AIX, Linux, and Windows의 경우 SSL 또는 TLS 클라이언트는 올바른 IBM MQ 형식으로 레이블 지정된 인증서가 있는 경우에만 인증서를 전송합니다.

- 큐 관리자의 경우 형식은 소문자로 변경된 큐 관리자 이름이 뒤따라오는 `ibmwebspheremq`입니다. 예를 들어, `QM1`의 경우 `ibmwebspheremqqm1`입니다.
- IBM MQ 클라이언트의 경우 소문자로 변경된 로그온 사용자 ID가 뒤따라오는 `ibmwebspheremq`입니다(예: `ibmwebspheremqmyuserid`).

IBM MQ는 레이블에 `ibmwebspheremq` 접두부를 사용하여 다른 제품에 대한 인증서와의 혼동을 피합니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). 클라이언트가 인증서를 전송하지 않으면 SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 SSLCAUTH 매개변수가 REQUIRED로 설정되거나 SSLPEER 매개변수 값이 설정된 상태로 정의된 경우에만 인증이 실패합니다. 자세한 정보는 SSL 또는 TLS를 사용하여 두 개의 큐 관리자 연결을 참조하십시오.

## z/OS Setting up communications for SSL or TLS on z/OS

Secure communications that use the SSL or TLS cryptographic security protocols involve setting up the communication channels and managing the digital certificates that you will use for authentication.

To set up your SSL or TLS installation you must define your channels to use SSL or TLS. You must also create and manage your digital certificates. On z/OS you can perform the tests with self-signed certificates, or with personal certificates signed by a local certificate authority (CA).

Self-signed certificates cannot be revoked, which could allow an attacker to spoof an identity after a private key has been compromised. CAs can revoke a compromised certificate, which prevents its further use. CA-signed certificates are therefore safer to use in a production environment, though self-signed certificates are more convenient for a test system.

For full information about creating and managing certificates, see "Working with SSL/TLS on z/OS" on page 288.

See the CERTLABL and CERTQSGL parameters of the ALTER QMGR command and the CERLABL parameter of the DEFINE CHANNEL command for more information.

The order of precedence is:

- Channel CERTLABL parameter
- QMGR CERTQSGL parameter if the channel is shared.

  For a sender channel, that means the transmission queue (XMITQ) is shared. For a receiver channel, that means the channel started through the shared listener, that is the listener with INDISP(GROUP).
- QMGR CERTLABL
- The default label of `ibmWebSphereMQ` followed by the name of the queue sharing group for shared channels, or the name of the queue manager.

This collection of topics introduces some of the tasks involved in setting up SSL or TLS communications, and provides step-by-step guidance on completing those tasks.

You might also want to test SSL or TLS client authentication, which are an optional part of the protocols. During the SSL or TLS handshake, the SSL or TLS client always obtains and validates a digital certificate from the server. With the IBM MQ implementation, the SSL or TLS server always requests a certificate from the client.

If the channel is shared, the channel first tries to find a certificate for the queue sharing group. If it does not find a certificate for a queue sharing group, it tries to find a certificate for the queue manager.

On z/OS, IBM MQ uses the `ibmWebSphereMQ` prefix on a label to avoid confusion with certificates for other products.

The SSL or TLS server always validates the client certificate if one is sent. If the SSL or TLS client does not send a certificate, authentication fails only if the end of the channel acting as the SSL or TLS server is

defined with either the SSLCAUTH parameter set to REQUIRED or an SSLPEER parameter value set. For more information, see Connecting two queue managers using SSL or TLS.

# SSL/TLS에 대한 작업

이 주제에서는 IBM MQ을 사용하여 TLS 사용과 관련된 단일 태스크를 수행하기 위한 지시사항을 제공합니다.

이들 대부분은 다음 절에 설명된 고급 수준 태스크에서 단계로서 사용됩니다.

- 300 페이지의 『사용자 식별 및 인증』
- 329 페이지의 『오브젝트에 액세스 권한 부여』
- 391 페이지의 『메시지의 기밀성』
- 444 페이지의 『메시지의 데이터 무결성』
- 445 페이지의 『클러스터 보안 유지』

## IBM i IBM i에서 SSL/TLS에 대한 작업

이 주제 콜렉션에서는 IBM MQ for IBM i에서 TLS(Transport Layer Security) 관련 작업을 수행하는 개별 태스크에 대한 지시사항을 제공합니다.

IBM i의 경우, TLS 지원은 운영 체제에 통합되어 있습니다. IBM i의 하드웨어 및 소프트웨어 요구사항에 나열된 필수조건을 설치했는지 확인하십시오.

IBM i에서 디지털 인증 관리자(DCM) 도구를 사용하여 키 및 디지털 인증을 관리합니다.

### *DCM* 액세스

DCM 인터페이스에 액세스하려면 다음 지시사항을 따르십시오.

### 이 태스크 정보

프레임을 지원하는 웹 브라우저에서 다음 단계를 수행하십시오.

### 프로시저

1. `http://machine.domain:2001` 또는 `https://machine.domain:2010`로 이동하십시오. 여기서 *machine* 은 컴퓨터의 이름입니다.
2. 요청되면 올바른 사용자 프로파일 및 비밀번호를 입력하십시오.

   사용자 프로파일에 새 인증서 저장소를 작성할 수 있는 *ALLOBJ 및 *SECADM 특수 권한이 있는지 확인하십시오. 특수 권한이 없는 경우에는 개인 인증서를 관리하거나 권한이 있는 오브젝트의 오브젝트 서명을 보기만 할 수 있습니다. 오브젝트 서명 애플리케이션을 사용할 권한이 있으면 DCM으로부터 오브젝트에 서명할 수도 있습니다.
3. 인터넷 구성 페이지에서 **디지털 인증 관리자**를 클릭하십시오.

   디지털 인증 관리자 페이지가 표시됩니다.

### *IBM i*에서 큐 관리자에 인증서 지정

큐 관리자에 인증서를 지정하려면 DCM을 사용하십시오.

인증서를 큐 관리자에 지정하려면 기존 IBM i 디지털 인증서 관리를 사용하십시오. 즉 큐 관리자가 시스템 인증서 저장소를 사용하고, 큐 관리자가 디지털 인증 관리자와 함께 애플리케이션으로서 사용하도록 등록되도록 지정할 수 있음을 의미합니다. 이를 수행하려면 큐 관리자 **SSLKEYR** 속성의 값을 *SYSTEM으로 변경하십시오.

**SSLKEYR** 매개변수가 *SYSTEM으로 변경되면 IBM MQ 는 큐 관리자를 고유 애플리케이션 레이블이 QIBM_WEBSPHERE_MQ_QMGRNAME이고 설명이 Qmgrname (WMQ) 인 레이블이 있는 서버 애플리케이션으로 등록합니다. *SYSTEM 인증서 저장소를 사용하는 경우에는 채널 **CERTLABL** 속성이 사용되지 않음을 유의하십시오. 그러면, 큐 관리자가 디지털 서명 관리자(DCM)의 서버 애플리케이션으로 표시되고 시스템 저장소에 있는 임의의 서버 또는 클라이언트 인증서를 이 애플리케이션에 지정할 수 있습니다.

큐 관리자는 애플리케이션으로 등록되므로 CA 신뢰 목록 정의 등과 같은 DCM의 고급 기능을 실행할 수 있습니다.

**SSLKEYR** 매개변수가 *SYSTEM이외의 값으로 변경되면 IBM MQ 는 큐 관리자를 디지털 Certificate Manager를 사용하는 애플리케이션으로 등록 해제합니다. 큐 관리자가 삭제되면 이는 또한 DCM에서 등록 취소됩니다. 충분한 *SECADM 권한이 있는 사용자는 또한 DCM에서 애플리케이션을 수동으로 추가하거나 제거할 수도 있습니다.

## *IBM i*에서 키 저장소 설정

키 저장소는 연결의 양쪽 끝에 설정되어야 합니다. 기본 인증서 저장소를 사용하거나 새로 고유의 것을 새로 만들 수 있습니다.

TLS 연결의 경우 연결의 각 끝에 키 저장소가 있어야 합니다. 각 큐 관리자 및 IBM MQ MQI client에는 키 저장소에 대한 액세스가 있어야 합니다. 파일 이름 및 비밀번호를 사용하여(즉, *SYSTEM 옵션을 사용하지 않고) 키 저장소에 액세스하려면 QMQM 사용자 프로파일에 다음 권한이 있는지 확인하십시오.

- 키 저장소를 포함하는 디렉토리의 권한을 실행하십시오.
- 키 저장소를 포함하는 파일의 권한을 읽으십시오.

자세한 정보는 23 페이지의 『SSL/TLS 키 저장소』의 내용을 참조하십시오. *SYSTEM 인증서 저장소를 사용하는 경우에는 채널 **CERTLABL** 속성이 사용되지 않습니다.

IBM i에서, 디지털 인증서는 DCM으로 관리되는 인증서 저장소에 저장됩니다. 이러한 디지털 인증서에는 레이블이 있고, 이는 인증서를 큐 관리자 또는 IBM MQ MQI client와 연관시킵니다. TLS가 인증 목적으로 인증서를 사용합니다.

레이블은 설정된 경우, **CERTLABL** 속성의 값이거나, 큐 관리자의 이름이 있는 기본 `ibmwebspheremq` 또는 사용자 로그온 ID가 추가된 IBM MQ MQI client(모두 소문자)입니다. 자세한 내용은 디지털 인증서 레이블을 참조하십시오.

큐 관리자 또는 IBM MQ MQI client인증서 저장소 이름은 경로 및 스템 이름으로 구성됩니다. 기본 경로는 `/QIBM/UserData/ICSS/Cert/Server/`이고 기본 스템 이름은 `Default`입니다. IBM i에서 기본 인증서 저장소인 `/QIBM/UserData/ICSS/Cert/Server/Default.kdb`는 `*SYSTEM`라고도 합니다. 또는 사용자 고유의 경로와 스템 이름을 정의할 수도 있습니다.

사용자 고유의 경로나 파일 이름을 정의하는 경우에는 이에 대한 액세스를 엄격하게 제어하기 위해 파일에 권한을 설정하십시오.

264 페이지의 『IBM i에서 큐 관리자의 키 저장소 위치 변경』에서는 인증서 저장소 이름 지정에 대해서 알려줍니다. 인증서 저장소를 작성하기 전이나 후에 인증서 저장소 이름을 지정할 수 있습니다.

**참고:** DCM으로 수행할 수 있는 조작은 사용자 프로파일의 권한에 의해 제한될 수도 있습니다. 예를 들어, CA 인증서를 작성하려면 *ALLOBJ 및 *SECADM 권한이 필요합니다.

### IBM i *IBM i* 에서 키 저장소 비밀번호 암호화

여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

다음 IBM MQ 컴포넌트 및 기능은 키 저장소 비밀번호를 저장하기 위한 두 가지 다른 방법을 지원합니다.

- 큐 관리자 TLS키 저장소입니다.
- TLS를 사용하는 IBM MQ MQI clients .

이러한 구성요소에서 사용할 키 저장소 비밀번호는 IBM MQ 비밀번호 보호 시스템을 사용하여 보호됩니다. 비밀번호를 제공하고 암호화하는 메커니즘은 컴포넌트에 따라 약간 다릅니다.

#### 큐 관리자 TLS키 저장소

비밀번호는 **SSLKEYRPWD** 큐 관리자 속성이 CHGMQM (메시지 큐 관리자 변경) 명령을 사용하여 설정될 때 암호화됩니다.

비밀번호는 AES-128 알고리즘으로 암호화됩니다. 이 알고리즘의 세부사항은 공개적으로 알려져 있으며 안전한 것으로 간주됩니다.

비밀번호는 키 저장소에 액세스할 수 있는 다른 소프트웨어에서 이해할 수 없는 독점 형식으로 스태쉬 파일에 저장됩니다.

하나의 IBM MQ 구성요소에 의해 암호화되는 비밀번호는 다른 IBM MQ 구성요소에 의해 사용될 수 없습니다.

키 저장소 비밀번호가 암호화될 때 고유한 암호화 키를 제공할 수 있습니다. 고유한 암호화 키는 암호화 키에 대한 액세스 권한이 없는 사용자가 비밀번호를 복호화할 수 없도록 합니다. 암호화할 비밀번호를 제공하기 전에 설정해야 하는 **INITKEY** 큐 관리자 속성을 통해 이 키를 제공합니다.

IBM MQ 비밀번호 보호 시스템에 대한 자세한 정보는 527 페이지의 『IBM MQ 컴포넌트 구성 파일에서 비밀번호 보호』의 내용을 참조하십시오.

**TLS를 사용하는 IBM MQ MQI clients**

274 페이지의 『IBM i용 IBM MQ SSL 클라이언트 유틸리티(amqrsslc)』 는 키 저장소 비밀번호를 스태쉬 파일에 저장할 수 있습니다. IBM i에서 MQSC 명령을 사용하여 관리도 참조하십시오.

비밀번호는 AES-128 알고리즘으로 암호화됩니다. 이 알고리즘의 세부사항은 공개적으로 알려져 있으며 안전한 것으로 간주됩니다.

비밀번호는 키 저장소에 액세스할 수 있는 다른 소프트웨어에서 이해할 수 없는 독점 형식으로 스태쉬 파일에 저장됩니다.

키 저장소 비밀번호가 암호화될 때 고유한 암호화 키를 제공할 수 있습니다. 고유한 암호화 키는 암호화 키에 대한 액세스 권한이 없는 사용자가 비밀번호를 복호화할 수 없도록 합니다. **-sf** 매개변수를 통해 이 키를 제공합니다.

암호화된 비밀번호는 키 저장소 파일과 동일한 디렉토리의 스태쉬 파일에 저장됩니다.

IBM MQ MQI clients 는 다른 메커니즘을 통해 제공되는 비밀번호도 지원합니다. 265 페이지의 『IBM i 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』을 참조하십시오.

키 저장소 비밀번호를 암호화하기 위해 선택하는 방법에 관계없이 저장된 비밀번호를 암호화하는 데 대한 제한사항을 알고 있는지 확인하십시오. 533 페이지의 『비밀번호 암호화를 통한 보호 한계』의 내용을 참조하십시오.

**관련 개념**

264 페이지의 『IBM i 에서 큐 관리자의 키 저장소 비밀번호 제공』
키 저장소에는 중요한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

265 페이지의 『IBM i 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』
키 저장소에는 중요한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

260 페이지의 『IBM i에서 SSL/TLS에 대한 작업』
이 주제 콜렉션에서는 IBM MQ for IBM i에서 TLS(Transport Layer Security) 관련 작업을 수행하는 개별 태스크에 대한 지시사항을 제공합니다.

*IBM i*에서 인증서 저장소 작성
기본 인증서 저장소를 사용하지 않으려면 이 프로시저를 따라 사용자 고유의 것을 작성하십시오.

## 이 태스크 정보

IBM i 기본 인증서 저장소를 사용하지 않으려는 경우에만 새 인증서 저장소를 작성하십시오.

IBM i 시스템 인증서 저장소가 사용되도록 지정하려면 큐 관리자의 SSLKEYR 속성 값을 *SYSTEM으로 변경하십시오. 이 값은 큐 관리자가 시스템 인증서 저장소를 사용하고 큐 관리자가 디지털 인증 관리자(DCM)을 사용하여 애플리케이션으로서 사용하도록 등록되어 있음을 나타냅니다.

## 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널에서 **새 인증서 저장소 작성**을 클릭하십시오.
   새 인증서 저장소 작성 페이지가 태스크 프레임에 표시됩니다.
3. 태스크 프레임에서 **다른 시스템 인증서 저장소**를 선택하고 **계속**을 클릭하십시오.

새 인증서 저장소 페이지에 인증서 작성이 태스크 프레임에 표시됩니다.

4. **아니오 - 인증서 저장소에 인증서를 작성하지 않음**을 선택하고 **계속**을 클릭하십시오.
   인증서 저장소 이름 및 비밀번호 페이지가 태스크 프레임에 표시됩니다.
5. **인증서 저장소 경로 및 파일 이름** 필드에 IFS 경로 및 파일 이름을 입력하십시오(예: `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`)
6. **비밀번호** 필드에 비밀번호를 입력하고 **비밀번호 확인** 필드에 다시 입력하십시오. **계속**을 클릭하십시오.
   저장소 키를 감출 때 필요하므로 비밀번호(대소문자 구분)를 기록해 두십시오.
7. DCM을 종료하려면 브라우저 창을 닫으십시오.

## 다음에 수행할 작업

DCM을 사용하여 인증서 저장소를 작성한 경우에는 263 페이지의 『IBM i 시스템에 인증서 저장소 비밀번호 숨기기』에 설명된 대로 비밀번호를 숨기십시오.

**관련 태스크**

272 페이지의 『IBM i에서 인증서를 키 저장소로 가져오기』
인증서를 가져오려면 이 프로시저를 따르십시오.

### *IBM i* 시스템에 인증서 저장소 비밀번호 숨기기

CL 명령을 사용하여 인증서 저장소 비밀번호를 숨기십시오.

다음 지시사항은 큐 관리자를 위해 IBM i에 인증서 저장소 비밀번호를 숨기는 데 적용됩니다. 또는 IBM MQ MQI client의 경우 *SYSTEM 인증서 저장소를 사용하지 않는 경우 (즉, MQSSLKEYR 환경이 *SYSTEM 이외의 값으로 설정됨) 274 페이지의 『IBM i용 IBM MQ SSL 클라이언트 유틸리티(amqrsslc)』의 275 페이지의 『인증서 저장소 비밀번호를 숨기십시오.』 절에 설명된 프로시저를 따르십시오.

*SYSTEM 인증서 저장소가 사용됨을 지정한 경우에는(큐 관리자의 SSLKEYR 속성을 *SYSTEM으로 변경하여) 다음 단계를 따르지 않아야 합니다.

DCM을 사용하여 인증서 저장소를 작성한 경우에는 다음 명령을 사용하여 비밀번호를 숨기십시오.

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

비밀번호는 대소문자를 구분합니다. 262 페이지의 『IBM i에서 인증서 저장소 작성』의 6단계에서 입력한 대로 정확하게 비밀번호를 작은따옴표 안에 입력해야 합니다.

**참고:** 기본 시스템 인증서 저장소를 사용 중이 아니고 비밀번호를 숨기지 않은 경우에는 인증서 저장소에 액세스하는 데 필요한 비밀번호를 확보할 수 없으므로 TLS 채널 시작에 실패합니다.

### 비밀번호 보호

키 저장소 비밀번호가 지정되면 IBM MQ 는 IBM MQ 비밀번호 보호 시스템을 사용하여 비밀번호를 암호화합니다. 비밀번호를 암호화하기 위해 초기 키가 사용됩니다. 이 키가 큐 관리자에 제공되지 않으면 대신 기본 키가 사용됩니다.

키 저장소 비밀번호를 제공하기 전에 큐 관리자에 대한 고유 초기 키를 설정해야 합니다. **ALTER QMGR** MQSC 명령의 **INITKEY** 속성을 사용하여 이를 수행할 수 있습니다.

```
ALTER QMGR INITKEY('value')
```

### *IBM i*에서 키 저장소에서 큐 관리자 찾기

이 프로시저를 사용하여 큐 관리자의 인증서 저장소의 위치를 확보하십시오.

### 프로시저

1. 다음 명령을 사용하여 큐 관리자의 속성을 표시하십시오.

```
DSPMQM MQMNAME('queue manager name')
```

2. 인증서 저장소의 경로 및 스템 이름의 명령 출력을 검사하십시오.
   예를 들어, /QIBM/UserData/ICSS/Cert/Server/Default입니다. 여기서 /QIBM/UserData/ICSS/Cert/Server은(는) 경로이고 Default은(는) 스템 이름입니다.

### *IBM i*에서 큐 관리자의 키 저장소 위치 변경

CHGMQM 또는 ALTER QMGR을 사용하여 큐 관리자의 인증서 저장소의 위치를 변경하십시오.

### 프로시저

CHGMQM 명령 또는 ALTER QMGR MQSC 명령을 사용하여 큐 관리자의 키 저장소 속성을 설정하십시오.

a) CHGMQM 사용: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) ALTER QMGR 사용: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

두 경우 모두 인증서 저장소에는 완전한 파일 이름(/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb)이 있습니다.

### 다음에 수행할 작업

큐 관리자의 인증서 저장소의 위치를 변경할 때 인증서는 이전 위치에서 전송되지 않습니다. 인증서 저장소를 작성할 때 사전 설치된 CA 인증서가 충분하지 않으면 272 페이지의 『IBM i에서 인증서를 키 저장소로 가져오기』에 설명된 대로 새 인증서 저장소에 인증서를 채워야 합니다. 또한 263 페이지의 『IBM i 시스템에 인증서 저장소 비밀번호 숨기기』에 설명된 대로 새 위치의 비밀번호를 숨겨야 합니다.

### IBM i *IBM i* 에서 큐 관리자의 키 저장소 비밀번호 제공

키 저장소에는 중요한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

IBM MQ 는 큐 관리자에 키 저장소 비밀번호를 제공하는 메커니즘을 제공합니다.

- **CHGMQM** 명령의 **SSLKEYRPWD** 매개변수

키 저장소 비밀번호는 IBM MQ 비밀번호 보호 시스템을 사용하여 암호화됩니다. 키 저장소 비밀번호를 보호하는 방법에 대한 자세한 정보는 261 페이지의 『IBM i 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

IBM i에서 MQSC 명령을 사용하여 관리도 참조하십시오.

### SSLKEYRPWD 속성

큐 관리자에 직접 키 저장소 비밀번호를 제공하려면 다음 **CHGMQM** 명령을 실행하여 *queue_manager* 를 큐 관리자 이름으로 대체하고 *password* 를 키 저장소 비밀번호로 대체하십시오.

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

**주의:** 큐 관리자 이름 및 비밀번호를 작은따옴표로 묶어야 합니다. 그렇지 않으면 IBM MQ 가 문자를 대문자로 변환합니다.

이 방법을 사용하여 키 저장소 비밀번호를 지정하면 비밀번호가 저장되기 전에 IBM MQ 비밀번호 보호 시스템을 사용하여 암호화됩니다.

초기 키라고 하는 암호화 키는 비밀번호를 암호화하는 데 사용됩니다. 고유한 초기 키를 사용하여 비밀번호를 안전하게 보호하도록 큐 관리자를 설정하십시오. 초기 키를 제공하지 않으면 기본 키가 사용됩니다.

키 저장소 비밀번호를 설정하기 전에 큐 관리자가 고유한 초기 키로 구성되었는지 확인하십시오. **ALTER QMGR** 명령에서 **INITKEY** 속성을 사용하여 초기 키를 수정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
ALTER QMGR INITKEY('mykey')
```

**경고:** 키 저장소 비밀번호를 설정한 후 초기 키를 수정하는 경우 키 저장소 비밀번호는 새 초기 키로 암호화되지 않습니다. 초기 키를 변경하는 경우 키 저장소 비밀번호도 재설정해야 합니다. 그렇지 않으면 IBM MQ 가 키 저장소 비밀번호를 복호화할 수 없으므로 키 저장소에 액세스할 수 없습니다.

**SSLKEYRPWD** 속성에 대한 자세한 정보는 **CHGMQM** 명령의 **SSLKEYRPWD** 매개변수를 참조하십시오.

**관련 개념**

261 페이지의 『IBM i 에서 키 저장소 비밀번호 암호화』
여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

265 페이지의 『IBM i 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』
키 저장소에는 중요한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

## IBM i *IBM i 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공*

키 저장소에는 중요한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

IBM MQ 에서는 IBM MQ MQI client에 키 저장소 비밀번호를 제공하기 위한 네 가지 메커니즘을 제공합니다.

- 265 페이지의 『MQSCO의 KeyRepoPassword 필드』
- 266 페이지의 『MQKEYRPWD 환경 변수』
- 266 페이지의 『클라이언트 구성 파일의 SSLKeyRepositoryPassword 속성』
- 266 페이지의 『키 저장소 스태쉬 파일』

키 저장소 스태쉬 파일을 사용하지 않는 경우 키 저장소 비밀번호를 일반 텍스트 문자열 또는 IBM MQ 비밀번호 보호 시스템을 사용하여 암호화되는 문자열로 제공할 수 있습니다. 키 저장소 비밀번호를 보호하는 방법에 대한 자세한 정보는 261 페이지의 『IBM i 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

## MQSCO의 KeyRepoPassword 필드

MQSCO 구조를 사용하여 키 저장소 비밀번호를 제공하려면 다음 세 가지 변수 문자열 필드의 조합을 사용해야 합니다.

**KeyRepoPasswordLength**
비밀번호의 길이입니다.

**KeyRepoPasswordPtr**
비밀번호를 포함하는 메모리의 위치에 대한 포인터입니다.

**KeyRepoPasswordOffset**
메모리에서 비밀번호의 위치이며 MQSCO 구조의 시작부터 바이트 수로 표시됩니다.

**참고: KeyRepoPasswordPtr** 또는 **KeyRepoPasswordOffset**중 하나만 제공할 수 있습니다.

예를 들면, 다음과 같습니다.

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

**주의:** 이 방법을 사용하여 비밀번호를 제공하는 경우 IBM MQ client 애플리케이션에 제공되기 전에 비밀번호를 암호화하십시오. 자세한 정보는 266 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

MQCSO 구조에 대한 자세한 정보는 MQSCO-SSL/TLS 구성 옵션을 참조하십시오.

### *MQKEYRPWD* 환경 변수

MQSCO 구조를 사용하여 키 저장소 비밀번호가 클라이언트에 제공되지 않는 경우 *MQKEYRPWD* 환경 변수를 사용하여 키 저장소 비밀번호를 지정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
export MQKEYRPWD=passw0rd
```

또는

```
set MQKEYRPWD=passw0rd
```

여기서 *passw0rd* 는 암호입니다.

**주의:** 이 방법을 사용하여 비밀번호를 제공하는 경우 환경 변수의 값을 설정하기 전에 비밀번호를 암호화하십시오. 자세한 정보는 266 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

### 클라이언트 구성 파일의 SSLKeyRepositoryPassword 속성

다른 방법 중 하나를 사용하여 키 저장소 비밀번호가 클라이언트에 제공되지 않는 경우, 클라이언트 구성 파일의 **SSL** 스탠자에서 **SSLKeyRepositoryPassword** 속성을 사용하여 키 저장소 비밀번호를 지정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

**주의:** 이 방법을 사용하여 비밀번호를 제공하는 경우 **SSLKeyRepositoryPassword** 속성의 값을 설정하기 전에 비밀번호를 암호화하십시오. 자세한 정보는 266 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

클라이언트 구성 파일의 SSL 스탠자에 대한 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오.

### 키 저장소 스태쉬 파일

키 저장소 비밀번호가 다른 방법 중 하나를 사용하여 클라이언트에 제공되지 않으면 IBM MQ 는 키 저장소와 동일한 디렉토리에 스태쉬 파일이 있다고 가정합니다. 스태쉬 파일에는 키 저장소와 동일한 스템 이름이 있지만 확장자는 `.sth` 입니다.

키 저장소 스태쉬 파일은 **amqrsslc** 명령행 도구를 사용하여 작성됩니다. 스태쉬 파일을 작성하려면 다음 명령을 실행하십시오.

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

이 명령은 암호화할 비밀번호를 입력하도록 프롬프트를 표시합니다. 비밀번호는 **-sf** 매개변수를 사용하여 제공되지 않는 경우 기본 암호화 키를 사용하여 IBM MQ 비밀번호 보호 시스템에 의해 암호화됩니다.

자세한 정보는 274 페이지의 『IBM i용 IBM MQ SSL 클라이언트 유틸리티(amqrsslc)』 및 266 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

### 키 저장소 비밀번호 암호화

스태쉬 파일 이외의 방법을 사용하여 키 저장소 비밀번호를 제공하는 경우 IBM MQ 비밀번호 보호 시스템을 사용하여 비밀번호를 암호화하십시오. 비밀번호를 암호화하려면 **runmqicred** 명령을 실행하십시오. 프롬프트가 표시되면 키 저장소 비밀번호를 입력하십시오. 이 명령은 암호화된 비밀번호를 출력합니다. 설명된 방법을 사용하여 일반 텍스트 비밀번호 대신 암호화된 비밀번호를 IBM MQ MQI client 에 제공할 수 있습니다.

초기 키라고 하는 암호화 키는 비밀번호를 암호화하는 데 사용됩니다. 비밀번호를 암호화할 때 고유한 초기 키를 사용하여 비밀번호를 안전하게 보호하십시오. 사용자 고유의 초기 키를 제공하려면 **-sf** 매개변수를 **runmqicred** 명령에 사용하십시오. 초기 키를 제공하지 않으면 기본 키가 사용됩니다.

자세한 정보는 runmqicred ( IBM MQ 클라이언트 비밀번호 보호)를 참조하십시오.

키 저장소 비밀번호가 암호화될 때 사용자 고유의 초기 키를 제공하고 암호화된 비밀번호를 IBM MQ MQI client에 제공하는 경우에도 IBM MQ MQI client에 동일한 초기 키를 제공해야 합니다. IBM MQ MQI client에 초기 키를 제공하는 방법에 대한 자세한 정보는 267 페이지의 『IBM i 의 IBM MQ MQI client 에 대한 초기 키 제공』의 내용을 참조하십시오.

**관련 개념**

261 페이지의 『IBM i 에서 키 저장소 비밀번호 암호화』
여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

264 페이지의 『IBM i 에서 큐 관리자의 키 저장소 비밀번호 제공』
키 저장소에는 중요한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

IBM i *IBM i 의 IBM MQ MQI client 에 대한 초기 키 제공*
IBM MQ 비밀번호 보호 시스템을 사용하여 암호화된 IBM MQ MQI client 에 변수를 제공하는 경우 값을 암호화하는 데 사용된 해당 초기 키를 제공해야 할 수 있습니다.

값을 암호화할 때 초기 키를 지정하지 않은 경우에는 IBM MQ client에 초기 키 값을 제공할 필요가 없습니다. 그러나 고유한 초기 키를 사용한 경우 다음 방법을 사용하여 IBM MQ client 에 초기 키를 제공할 수 있습니다.

- 267 페이지의 『MQCSP 구조를 사용하여 초기 키 제공』
- 267 페이지의 『MQS_MQI_KEYFILE 환경 변수를 사용하여 초기 키 제공』
- 268 페이지의 『클라이언트 구성 파일을 사용하여 초기 키 제공』

## MQCSP 구조를 사용하여 초기 키 제공

MQCSP 구조를 사용하여 초기 키를 제공하려면 다음 세 변수 문자열 필드의 조합을 사용해야 합니다.

**InitialKeyLength**
초기 키의 길이

**InitialKeyPtr**
초기 키를 포함하는 메모리의 위치에 대한 포인터

**InitialKeyOffset**
MQCSP 구조의 시작부터 바이트 수로 표시되는 메모리의 초기 키 위치입니다.

**참고: InitialKeyPtr** 또는 **InitialKeyOffset**중 하나만 제공할 수 있습니다.

예를 들면, 다음과 같습니다.

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

## MQS_MQI_KEYFILE 환경 변수를 사용하여 초기 키 제공

MQCSP 구조를 사용하여 클라이언트에 초기 키가 제공되지 않는 경우 IBM MQ 는 *MQS_MQI_KEYFILE* 환경 변수를 검사합니다. 이 환경 변수를 사용할 초기 키로 구성된 단일 텍스트 행을 포함하는 파일의 위치로 설정해야 합니다.

예를 들어, `mykey.key` 파일이 루트 디렉토리에 있고 초기 키를 포함하는 경우 환경 변수를 다음과 같이 설정해야 합니다.

```
export MQS_MQI_KEYFILE=/mykey.key
```

또는

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

## 클라이언트 구성 파일을 사용하여 초기 키 제공

초기 키가 이전 메커니즘을 사용하여 클라이언트에 제공되지 않는 경우 IBM MQ 는 `mqclient.ini` 파일의 보안 스탠자의 **MQIInitialKeyFile** 속성을 확인합니다. 이 속성을 사용할 초기 키로 구성된 단일 텍스트 행을 포함하는 파일의 위치로 설정해야 합니다.

예를 들어, `mykey.key` 라는 파일이 루트 디렉토리에 있고 초기 키를 포함하는 경우 클라이언트 구성 파일은 다음을 포함해야 합니다.

```
Security:
    MQIInitialKeyFile=/mykey.key
```

**관련 개념**

261 페이지의 『IBM i 에서 키 저장소 비밀번호 암호화』
여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

260 페이지의 『IBM i에서 SSL/TLS에 대한 작업』
이 주제 콜렉션에서는 IBM MQ for IBM i에서 TLS(Transport Layer Security) 관련 작업을 수행하는 개별 태스크에 대한 지시사항을 제공합니다.

## *IBM i* 테스트를 위한 인증 기관 및 인증서 작성

이 프로시저를 사용하여 인증서 요청을 서명하기 위해 로컬 CA 인증서를 작성하고 CA 인증서를 작성 및 설치하십시오.

### 시작하기 전에

이 주제의 지시사항은 로컬 인증 기관(CA)이 존재하지 않는다고 가정합니다. 로컬 CA가 존재하는 경우에는 269 페이지의 『IBM i에서 서버 인증서 요청』으로 이동하십시오.

### 이 태스크 정보

TLS를 설치할 때 제공된 CA 인증서는 CA를 발행하여 서명됩니다. IBM i에서는 시스템에서 TLS 통신을 테스트하기 위해 서버 인증서에 서명할 수 있는 로컬 인증 기관을 생성할 수 있습니다. 로컬 CA 인증서를 작성하려면 웹 브라우저에서 다음 단계를 따르십시오.

### 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널에서 **인증 기관 작성**을 클릭하십시오.
   인증 기관 작성 페이지가 태스크 프레임에 표시됩니다.
3. **인증서 저장소 비밀번호** 필드에 비밀번호를 입력하고 **비밀번호 확인** 필드에 다시 입력하십시오.
4. **인증 기관(CA) 이름** 필드에 이름을 입력하십시오(예: `TLS Test Certificate Authority`).
5. **공통 이름** 및 **조직** 필드에 적합한 값을 입력하고 국가를 선택하십시오. 나머지 선택 필드에는 필요한 값을 입력하십시오.
6. **검증 기간** 필드에 로컬 CA의 검증 기관을 입력하십시오.
   기본값은 1095일입니다.
7. **계속**을 클릭하십시오.
   CA가 작성되고 DCM은 로컬 CA를 위해 인증서 저장소 및 CA 인증서를 작성합니다.
8. **인증서 설치**를 클릭하십시오.

다운로드 관리자 대화 상자가 표시됩니다.

9. CA 인증서를 저장하려는 임시 파일의 전체 경로 이름을 입력하고 **저장**을 클릭하십시오.
10. 다운로드가 완료되면 **열기**를 클릭하십시오.
    인증서 창이 표시됩니다.
11. **인증서 설치**를 클릭하십시오.
    인증서 가져오기 마법사가 표시됩니다.
12. **다음** 을 클릭하십시오.
13. **인증서 유형을 기반으로 인증서 저장소 자동 선택**을 선택하고 **다음**을 클릭하십시오.
14. **완료**를 누르십시오.
    확인 창이 표시됩니다.
15. **확인**을 클릭하십시오.
16. 인증서 창에서 **확인**을 클릭하십시오.
17. **계속**을 클릭하십시오.
    인증 기관 정책 페이지가 태스크 프레임에 표시됩니다.
18. **사용자 인증서 작성 허용** 필드에서 **예**를 선택하십시오.
19. **검증 기간** 필드에 로컬 CA가 발행한 인증서의 검증 기간을 입력하십시오.
    기본값은 365일입니다.
20. **계속**을 클릭하십시오.
    새 인증서 저장소 페이지에 인증서 작성이 태스크 프레임에 표시됩니다.
21. 애플리케이션의 아무 것도 선택되지 않았는지 확인하십시오.
22. 로컬 CA 설정을 완료하려면 **계속**을 클릭하십시오.

## 다음에 수행할 작업

기존 인증서를 갱신해야 하는 경우 IBM i 문서에서 기존 인증서 갱신 을 참조하십시오.

## *IBM i*에서 서버 인증서 요청

디지털 인증서는 위장으로부터 보호하고 공개 키가 지정된 엔티티에 속하는지 확인합니다. 새 서버 인증서는 디지털 인증 관리자(DCM)를 사용하여 인증 기관으로부터 요청될 수 있습니다.

## 이 태스크 정보

웹 브라우저에서 다음 단계를 수행하십시오.

## 프로시저

1. 260 페이지의 『DCM 액세스』 에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널에서 **인증서 저장소 선택**을 클릭하십시오.
   인증서 저장소 선택 페이지가 태스크 프레임에 표시됩니다.
3. 사용하려는 인증서 저장소를 선택하고 **계속**을 클릭하십시오.
4. 옵션: 3단계에서 ***SYSTEM**을 선택한 경우에는 시스템 저장소 비밀번호를 입력하고 **계속**을 클릭하십시오.
5. 옵션: 3단계에서 **다른 시스템 인증서 저장소**를 선택한 경우에는 **인증서 저장소 경로 및 파일 이름** 필드에 인증서 저장소를 작성할 때 설정한 IFS 경로 및 파일 이름을 입력하십시오. 또한 **인증서 저장소 비밀번호** 필드에 비밀번호를 입력하십시오. 그런 다음 **계속**을 클릭하십시오.
6. 탐색 패널에서 **인증서 작성**을 클릭하십시오.
7. 태스크 프레임에서 **서버 또는 클라이언트 인증서** 단일 선택 단추를 선택하고 **계속**을 클릭하십시오.
   인증 기관(CA) 선택 페이지가 태스크 프레임에 표시됩니다.
8. 워크스테이션에 로컬 CA가 있는 경우 로컬 CA 또는 상업 CA를 선택하여 인증서에 서명하십시오. 원하는 CA의 단일 선택 단추를 선택하고 **계속**을 클릭하십시오.
   인증서 작성 페이지가 태스크 프레임에 표시됩니다.

9. 옵션: 큐 관리자의 경우 **인증서 레이블** 필드에 인증서 레이블을 입력하십시오.
   레이블은 설정된 경우 **CERTLABL** 속성의 값이거나 큐 관리자의 이름이 추가된 ibmwebspheremq(모두 소문자)입니다. 자세한 내용은 디지털 인증서 레이블을 참조하십시오.
   예를 들어, 큐 관리자 QM1의 경우 기본값을 사용하려면 ibmwebspheremqqm1을 입력하십시오.
10. 옵션: IBM MQ MQI client의 경우 **인증서 레이블** 필드에 소문자로 접힌 로그온 사용자 ID가 뒤따르는 ibmwebspheremq를 입력하십시오.
    예를 들어 ibmwebspheremqmyuserID를 입력하십시오.
11. **공통 이름** 및 **조직** 필드에 적합한 값을 입력하고 국가를 선택하십시오. 나머지 선택 필드에는 필요한 값을 입력하십시오.

## 결과

인증서에 서명하기 위해 상업용 CA를 선택한 경우에는 DCM은 인증서 요청을 PEM(Privacy-Enhanced Mail) 형식으로 작성합니다. 요청을 선택한 CA로 전달하십시오.

인증서에 서명하기 위해 로컬 CA를 선택한 경우에는 DCM은 인증서가 인증서 저장소에 작성되었고 사용할 수 있음을 사용자에게 알립니다.

## 원격 시스템에 대한 서버 인증서 요청*IBM i*

로컬 인증 기관(CA)에서 서명한 인증서를 생성하거나 다른 플랫폼의 키 저장소로 가져오기 위해 상업용 CA에서 서명한 서버 인증서를 신청하려면 이 절차를 따르세요.

## 이 태스크 정보

사용자 인증서는 디지털 인증 관리자(DCM)가 다중 플랫폼에서 IBM MQ을 위한 인증서 관리자의 역할을 할 때 사용되어야 합니다. 다른 플랫폼에 배포되고 키 저장소로 가져오는 개인 인증서의 경우 웹 브라우저에서 다음 단계를 수행하십시오.

## 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. **탐색** 분할창에서 **인증서 작성**을 클릭하십시오.
   **인증서 작성** 페이지가 태스크 프레임에 표시됩니다.
3. **인증서 작성** 패널에서 **사용자 인증서** 단일 선택 단추를 선택하고 **계속**을 클릭하십시오.
   **사용자 인증서 작성** 페이지가 표시됩니다.
4. **사용자 인증서 작성** 패널에서 **조직 이름**, **시/도** 또는 **구/군/시**, **국가** 또는 **지역**의 인증서 정보 아래에서 필수 필드를 완료하십시오. 또는 **조직 단위** 및 **로컬성** 또는 **구/군/시** 필드에 값을 입력하십시오. **계속**을 클릭하십시오.
   **공통 이름**은 iSeries 시스템에 로그온할 때 사용하는 사용자 ID로 자동으로 설정됩니다.
5. 다음 **사용자 인증서 작성** 패널에서 **인증서 설치**를 클릭하고 **계속**을 클릭하십시오.
   다음과 같은 메시지가 표시됩니다. 개인 인증서가 설치되었습니다. 이 인증서의 백업 사본을 보관해야 합니다.
6. **확인**을 클릭하십시오.
7. DCM에 액세스하는 데 사용한 웹 브라우저에 따라 다음 단계 중 하나를 완료하세요.
   - Microsoft Edge의 경우 **도구> 인터넷 옵션> 컨텐츠 탭> 인증서 단추> 개인 탭**을 선택하십시오. 인증서를 선택하고 **내보내기**를 클릭하십시오.
   - Mozilla Firefox의 경우 **도구> 옵션> 고급> 암호화 탭> 인증서 보기 단추> 사용자 인증서 탭**을 선택하십시오. 인증서를 선택하고 **백업**을 클릭하십시오. 경로 및 파일 이름을 선택하고 **확인**을 클릭하십시오.
8. FTP를 사용하여 내보낸 인증서를 2진 형식으로 원격 시스템으로 전송하십시오.
9. 단계에서 내보낸 인증서를 가져옵니다.270 페이지의 『7』 원격 시스템의 키 저장소로 이동합니다.
   - 인증서가 다음을 사용하여 저장된 경우Microsoft Edge에 설명된 지침을 따르세요.517 페이지의 『Microsoft.pfx 파일에서 개인 인증서 가져오기』 파일.

- 인증서가 Mozilla Firefox를 사용하여 저장된 경우에는 개인 인증서를 키 저장소로 가져오기에 설명된 지시사항을 사용하십시오.

가져오기 중에 개인 인증서 및 서명자 인증서의 레이블 이름이 다음 값으로 변경되었는지 확인하십시오.IBM MQ 기대합니다. 라벨은 다음 중 하나의 값이어야 합니다.IBM MQ 큐 관리자**CERTLABL** 속성(설정된 경우) 또는 기본값`ibmwebspheremq` 큐 관리자 이름이 추가되며 모두 소문자입니다. 자세한 내용은 다음을 참조하세요. 디지털 인증서 라벨 .

## *IBM i*에서 키 저장소에 서버 인증서 추가

요청된 인증서를 키 저장소에 추가하려면 이 프로시저를 따르십시오.

### 이 태스크 정보

CA가 새 서버 인증서를 송신한 후에는 이를 사용자가 요청을 생성한 인증서 저장소에 추가할 수 있습니다. CA가 이메일 메시지의 일부로서 인증서를 송신하면 인증서를 별도의 파일로 복사하십시오.

**참고:**

- 서버 인증서가 로컬 CA에 의해 서명된 경우에는 이 프로시저를 수행할 필요가 없습니다.
- 서버 인증서를 PKCS #12 형식으로 DCM에 가져오기 전에 먼저 해당하는 CA 인증서를 가져와야 합니다.

서버 인증서를 큐 관리자 인증서 저장소에서 수신하려면 다음 프로시저를 사용하십시오.

### 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널의 **인증서 관리** 태스크 범주에서 **인증서 가져오기**를 클릭하십시오.
   인증서 가져오기 페이지가 태스크 프레임에 표시됩니다.
3. 인증서 유형에 맞는 단일 선택 단추를 선택하고 **계속**을 클릭하십시오.
   서버 또는 클라이언트 인증서 가져오기 페이지 또는 인증 기관(CA) 인증서 가져오기 페이지가 태스크 프레임에 표시됩니다.
4. **파일 가져오기** 필드에 가져오려는 인증서의 파일 이름을 입력하고 **계속**을 클릭하십시오.
   DCM은 자동으로 파일의 형식을 판별합니다.
5. 인증서가 **서버 또는 클라이언트** 인증서이면 태스크 프레임에 비밀번호를 입력하고 **계속**을 클릭하십시오.
   DCM은 인증서가 가져오기되었음을 알려줍니다.

## *IBM i*에서 키 저장소로부터 인증서 내보내기

인증서를 내보내면 공개 키와 개인 키가 모두 내보내집니다. 개인 키가 누출되면 보안이 심각하게 위협받으므로 이 조치는 각별히 주의하여 실행해야 합니다.

### 시작하기 전에

사용자의 인증서를 다른 사용자와 공유할 때 공개 키를 교환합니다. 이 프로세스는 **태스크 5. Sharing Certificates** in the Sharing Certificates section of 571 페이지의 『AIX and Linux 의 AMS 용 빠른 시작 안내서』. 여기에 설명된 것과 같이 인증서를 내보내면 개인 키와 공개 키가 모두 내보내집니다. 개인 키가 누출되면 보안이 심각하게 위협받으므로 이 조치는 각별히 주의하여 실행해야 합니다.

### 이 태스크 정보

인증서를 내보내려는 컴퓨터에서 다음 단계를 수행하십시오.

### 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널에서 **인증서 저장소 선택**을 클릭하십시오.
   인증서 저장소 선택 페이지가 태스크 프레임에 표시됩니다.
3. 사용하려는 인증서 저장소를 선택하고 **계속**을 클릭하십시오.

4. 옵션: 3단계에서 ***SYSTEM**을 선택한 경우에는 시스템 저장소 비밀번호를 입력하고 **계속**을 클릭하십시오.
5. 옵션: 3단계에서 **다른 시스템 인증서 저장소**를 선택한 경우에는 **인증서 저장소 경로 및 파일 이름** 필드에 인증서 저장소를 작성할 때 설정한 IFS 경로 및 파일 이름을 입력하고 **인증서 저장소 비밀번호** 필드에 비밀번호를 입력하십시오. 그런 다음 **계속**을 클릭하십시오.
6. 탐색 패널의 **인증서 관리** 태스크 범주에서 **인증서 내보내기**를 클릭하십시오.

   인증서 내보내기 페이지가 태스크 프레임에 표시됩니다.
7. 인증서 유형에 맞는 단일 선택 단추를 선택하고 **계속**을 클릭하십시오.

   서버 또는 클라이언트 인증서 내보내기 페이지 또는 인증 기관(CA) 인증서 내보내기 페이지가 태스크 프레임에 표시됩니다.
8. 내보내려는 인증서를 선택하십시오.
9. 인증서를 파일에 또는 다른 인증서 저장소에 직접 내보내려는지 여부를 지정하려면 단일 선택 단추를 선택하십시오.
10. 서버나 클라이언트 인증서를 파일에 내보내기로 선택한 경우에는 다음 정보를 제공하십시오.
    - 내보낸 인증서를 저장하려는 위치의 경로 및 파일 이름.
    - 개인 인증서의 경우 내보낸 인증서 및 대상 릴리스를 암호화하는 데 사용되는 비밀번호. CA 인증서의 경우 비밀번호를 지정할 필요가 없습니다.
11. 인증서를 다른 인증서 저장소로 직접 내보내기로 선택한 경우에는 대상 인증서 저장소 및 해당 비밀번호를 지정하십시오.
12. **계속**을 클릭하십시오.

## *IBM i*에서 인증서를 키 저장소로 가져오기

인증서를 가져오려면 이 프로시저를 따르십시오.

### 시작하기 전에

PKCS #12 형식으로 개인 인증서를 DCM으로 가져오기 전에 먼저 해당하는 CA 인증서를 가져와야 합니다.

### 이 태스크 정보

인증서를 가져오려는 시스템에서 다음 단계를 수행하십시오.

### 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널에서 **인증서 저장소 선택**을 클릭하십시오.

   인증서 저장소 선택 페이지가 태스크 프레임에 표시됩니다.
3. 사용하려는 인증서 저장소를 선택하고 **계속**을 클릭하십시오.
4. 옵션: 3단계에서 ***SYSTEM**을 선택한 경우에는 시스템 저장소 비밀번호를 입력하고 **계속**을 클릭하십시오.
5. 옵션: 3단계에서 **다른 시스템 인증서 저장소**를 선택한 경우에는 **인증서 저장소 경로 및 파일 이름** 필드에 인증서 저장소를 작성할 때 설정한 IFS 경로 및 파일 이름을 입력하고 **인증서 저장소 비밀번호** 필드에 비밀번호를 입력하십시오. 그런 다음 **계속**을 클릭하십시오.
6. 탐색 패널의 **인증서 관리** 태스크 범주에서 **인증서 가져오기**를 클릭하십시오.

   인증서 가져오기 페이지가 태스크 프레임에 표시됩니다.
7. 인증서 유형에 맞는 단일 선택 단추를 선택하고 **계속**을 클릭하십시오.

   서버 또는 클라이언트 인증서 가져오기 페이지 또는 인증 기관(CA) 인증서 가져오기 페이지가 태스크 프레임에 표시됩니다.
8. **파일 가져오기** 필드에 가져오려는 인증서의 파일 이름을 입력하고 **계속**을 클릭하십시오.

   DCM은 자동으로 파일의 형식을 판별합니다.
9. 인증서가 **서버 또는 클라이언트** 인증서이면 태스크 프레임에 비밀번호를 입력하고 **계속**을 클릭하십시오. DCM은 인증서가 가져오기되었음을 알려줍니다.

## *IBM i*에서 인증서 제거

개인 인증서를 제거하려면 이 프로시저를 사용하십시오.

### 프로시저

1. 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 탐색 패널에서 **인증서 저장소 선택**을 클릭하십시오.
   인증서 저장소 선택 페이지가 태스크 프레임에 표시됩니다.
3. **다른 시스템 인증서 저장소** 선택란을 선택한 후 **계속**을 클릭하십시오.
   인증서 저장소 및 비밀번호 페이지가 표시됩니다.
4. **인증서 저장소 경로 및 파일 이름** 필드에 인증서 저장소를 작성할 때 설정한 IFS 경로 및 파일 이름을 입력하십시오.
5. **인증서 저장소 비밀번호** 필드에 비밀번호를 입력하십시오. **계속**을 클릭하십시오.
   현재 인증서 저장소 페이지가 태스크 프레임에 표시됩니다.
6. 탐색 패널의 **인증서 관리** 태스크 범주에서 **인증서 삭제**를 클릭하십시오.
   인증서 삭제 확인 페이지가 태스크 프레임에 표시됩니다.
7. 삭제하려는 인증서를 선택하십시오. **삭제**를 클릭하십시오.
8. 인증서를 삭제하고 싶음을 확인하려면 **예**를 클릭하십시오. 그렇지 않으면 **아니오**를 클릭하십시오.
   DCM은 인증서가 삭제되었음을 알립니다.

## *IBM i* 에서 단방향 인증을 위해 **SYSTEM* 인증서 저장소 사용

단방향 인증을 설정하려면 이러한 지시사항을 따르십시오.

### 시작하기 전에

- 큐 관리자, 채널 및 전송 큐를 작성하십시오.
- 서버 큐 관리자에서 서버 또는 클라이언트 인증서를 작성하십시오.
- CA 인증서를 클라이언트 큐 관리자로 전송하고 키 저장소로 가져오기했습니다.
- 서버 및 클라이언트 큐 관리자에서 리스너를 시작하십시오.

### 이 태스크 정보

IBM i 를 TLS 서버로 실행하는 컴퓨터를 사용하여 단방향 인증을 사용하려면 SSL키 저장소 (SSLKEYR) 매개변수를 *SYSTEM으로 설정하십시오. 이 설정은 IBM MQ 큐 관리자를 애플리케이션으로서 등록합니다. 그런 다음 단방향 인증을 사용으로 설정하기 위해 큐 관리자에 인증서를 지정할 수 있습니다.

개인 키 저장소를 사용하여 키 저장소에 클라이언트 큐 관리자를 위한 더미 인증서를 작성하여 단방향 인증을 구현할 수도 있습니다.

### 프로시저

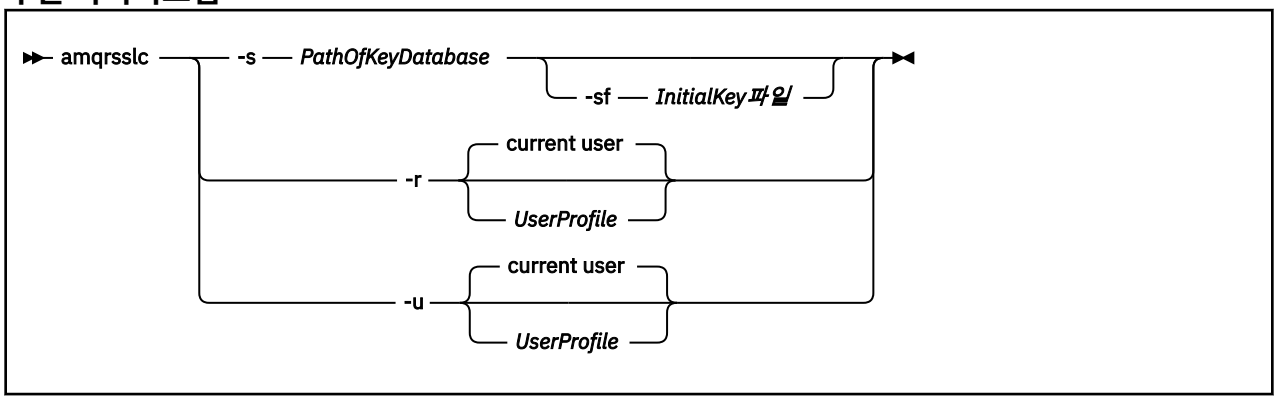
1. 서버 및 클라이언트 큐 관리자에서 다음 단계를 수행하십시오.
   a) `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)` 명령을 실행하여 SSLKEYR 매개변수를 설정하려면 큐 관리자를 변경하십시오.
   b) `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')` 명령을 실행하여 기본 키 저장소의 비밀번호를 숨기십시오.
   비밀번호는 작은따옴표에 있어야 합니다.
   c) SSLCIPHER 매개변수에서 올바른 CipherSpec을 가지도록 채널을 변경하십시오.
   d) `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)` 명령을 실행하여 TLS 보안을 새로 고치십시오.
2. 다음과 같이 DCM을 사용하여 서버 큐 관리자에 인증서를 지정하십시오.
   a) 260 페이지의 『DCM 액세스』에 설명된 대로 DCM 인터페이스에 액세스하십시오.

b) 탐색 패널에서 **인증서 저장소 선택**을 클릭하십시오.
   인증서 저장소 선택 페이지가 태스크 프레임에 표시됩니다.
c) *SYSTEM 인증서 저장소를 선택하고 **계속**을 클릭하십시오.
d) 왼쪽 패널에서 **애플리케이션 관리**를 펼치십시오.
e) 큐 관리자가 애플리케이션으로 등록되었는지 확인하려면 **애플리케이션 보기** 정의를 선택하십시오.
   SSL(WMQ)이 테이블에 나열됩니다.
f) **인증서 지정 업데이트**를 선택하십시오.
g) **서버**를 선택하고 **계속**을 클릭하십시오.
h) QMGRNAME(WMQ)을 선택하고 **인증서 지정 업데이트**를 클릭하십시오.
i) 인증서를 선택하고 **새 인증서 지정**을 클릭하십시오. 인증서가 애플리케이션에 지정되었음을 알리는 창이 열립니다.

## *IBM i용 IBM MQ SSL 클라이언트 유틸리티(amqrsslc)*

IBM i 용 IBM MQ SSL 클라이언트 유틸리티 (amqrsslc) 는 IBM i 시스템의 IBM MQ MQI client 에서 클라이언트 사용자 프로파일을 등록 또는 등록 취소하거나 인증서 저장소 비밀번호를 숨기는 데 사용됩니다. 유틸리티는 *ALLOBJ 특수 권한이 있는 프로파일이 있는 사용자 또는 디지털 인증 관리자(DCM)에서 애플리케이션 등록을 작성하거나 삭제할 옵션이 있는 QMQMADM의 구성원에 의해서만 실행될 수 있습니다.

## 구문 다이어그램




## 클라이언트 사용자 프로파일 등록

IBM MQ MQI client 가 *SYSTEM 인증서 저장소를 사용하는 경우 디지털 Certificate Manager (DCM)를 사용하여 애플리케이션으로 사용할 클라이언트 사용자 프로파일 (로그온 사용자) 을 등록해야 합니다.

클라이언트 사용자 프로파일을 등록하려면 *UserProfile*을 사용하여 `-r` 옵션과 함께 **amqrsslc** 프로그램을 실행하십시오. **amqrsslc**를 호출할 때 사용되는 사용자 프로파일에는 *USE 권한이 있어야 합니다. `-r` 옵션과 함께 *UserProfile* 을 제공하면 *UserProfile* 이 고유 애플리케이션 레이블이 QIBM_WEBSPHERE_MQ_*UserProfile* 이고 설명이 *UserProfile* (WMQ) 인 레이블이 있는 서버 애플리케이션으로 등록됩니다. 그런 다음 이 서버 애플리케이션은 DCM에 표시되고 시스템 저장소에 있는 모든 서버나 클라이언트 인증서를 이 애플리케이션에 지정할 수 있습니다.

**참고:** 사용자 프로파일이 `-r` 옵션으로 지정되지 않은 경우에는 **amqrsslc** 도구를 실행 중인 사용자의 사용자 프로파일이 등록됩니다.

다음 코드는 **amqrsslc**를 사용하여 사용자 프로파일을 등록합니다. 첫 번째 예에서 지정된 사용자 프로파일이 등록됩니다. 두 번째에서는 이는 로그인한 사용자의 프로파일입니다.

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

## 클라이언트 사용자 프로파일 등록 취소

클라이언트 프로파일을 등록 취소하려면 *UserProfile* 프로파일을 -u 옵션과 함께 사용하여 **amqrsslc** 프로그램을 실행하십시오. **amqrsslc**를 호출할 때 사용되는 사용자 프로파일에는 *USE 권한이 있어야 합니다. *UserProfile* 에 -u 옵션을 제공하면 *UserProfile* 이 DCM에서 레이블 QIBM_WEBSPHERE_MQ_*UserProfile* 로 등록 해제됩니다.

**참고:** 사용자 프로파일이 -u 옵션으로 지정되지 않은 경우에는 **amqrsslc** 도구를 실행 중인 사용자의 사용자 프로파일이 등록됩니다.

다음 코드는 **amqrsslc**를 사용하여 사용자 프로파일을 등록 취소합니다. 첫 번째 예에서 지정된 사용자 프로파일이 등록 취소됩니다. 두 번째에서는 이는 로그인한 사용자의 프로파일입니다.

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

## 인증서 저장소 비밀번호를 숨기십시오.

IBM MQ MQI client 가 *SYSTEM 인증서 저장소를 사용하지 않고 다른 인증서 저장소를 사용하는 경우 (즉, MQSSLKEYR이 *SYSTEM 이외의 값으로 설정된 경우), 키 데이터베이스의 비밀번호는 클라이언트 애플리케이션이 실행될 때 지정할 필요가 없도록 스태시될 수 있습니다.

키 데이터베이스의 비밀번호를 숨기려면 -s 옵션을 사용하십시오. 키 데이터베이스의 전체 경로 및 이름을 지정하십시오. 파일 확장자를 제공하지 않으면 .kdb로 가정합니다.

다음 코드에서 인증서 저장소의 완전한 파일 이름은 /Path/Of/KeyDatabase/MyKey.kdb입니다.

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

이 코드를 실행하면 이 키 데이터베이스의 비밀번호가 요청됩니다. 이 비밀번호는 확장자가 .sth 인 키 데이터베이스와 이름이 동일한 파일에 보관됩니다.

또한 비밀번호를 암호화하기 위한 초기 키를 지정할 수 있습니다. 초기 키는 단일 텍스트 행으로 파일에 저장되어야 하며, 그런 다음 해당 파일의 위치는 **-sf** 플래그를 통해 프로그램에 제공됩니다. 초기 키 파일이 제공되지 않으면 기본 키를 사용하여 비밀번호를 암호화합니다.

스태쉬 파일은 키 데이터베이스와 동일한 경로에 저장됩니다. 이 코드 예에서는 스태쉬 파일 /Path/Of/KeyDatabase/MyKey.sth를 생성합니다.

QMQM은 사용자 소유자이고 QMQMADM은 이 파일의 그룹 소유자입니다. QMQM 및 QMQMADM에는 읽기, 쓰기 권한이 있고 다른 프로파일에는 읽기 권한만 있습니다.

## 인증서 또는 인증서 저장소의 변경사항이 *IBM i*에서 적용되는 시기

인증서 저장소에서 인증서 또는 인증서 저장소의 위치를 변경할 때 변경사항은 채널 유형 및 채널이 실행 중인 방법에 따라 적용됩니다.

인증서 저장소의 인증서 및 키 저장소 속성의 변경사항은 다음 상황에서 적용됩니다.

- 새 아웃바운드 단일 프로세스가 처음으로 TLS 채널을 실행할 때.
- 새 인바운드 TCP/IP 단일 채널 프로세스가 처음으로 TLS 채널을 시작하라는 요청을 수신할 때.
- MQSC 명령 REFRESH SECURITY TYPE(SSL)이 IBM MQ TLS 환경을 새로 고치기 위해 실행될 때.
- 클라이언트 애플리케이션 프로세스의 경우, 프로세스에서 마지막 TLS 연결이 닫힐 때. 다음 TLS 연결에는 인증서 변경사항이 적용됩니다.
- 프로세스 풀링 프로세스(amqrmppa)의 스레드로서 실행되는 채널의 경우 프로세스 풀링 프로세스가 시작되거나 재시작되고 TLS 채널을 처음 시작할 때. 프로세스 풀링 프로세스가 이미 TLS 채널을 실행했고 변경사항을 즉시 적용하고 싶은 경우, MQSC 명령 REFRESH SECURITY TYPE(SSL)을 실행하십시오.
- 채널 시작기의 스레드로서 실행되는 채널의 경우 채널 시작기가 시작되거나 재시작되고 TLS 채널을 처음 실행할 때. 채널 시작기 프로세스가 이미 TLS 채널을 실행했고 변경사항을 즉시 적용하고 싶은 경우에는 MQSC 명령 REFRESH SECURITY TYPE(SSL)을 실행하십시오.

- TCP/IP 리스너의 스레드로서 실행되는 채널의 경우 리스너가 시작되거나 재시작되고 TLS 채널을 시작하라는 요청을 처음으로 수신할 때. 리스너가 이미 TLS 채널을 실행했고 변경사항을 즉시 적용하고 싶으면 MQSC 명령 REFRESH SECURITY TYPE(SSL)을 실행하십시오.

### *IBM i*에서 암호화 하드웨어 구성

이 프로시저를 사용하여 IBM i에서 암호화 코프로세서를 구성하십시오.

#### 시작하기 전에

사용자 프로파일에 코프로세서 하드웨어를 구성할 수 있게 해주는 *ALLOBJ 및 *SECADM 특수 권한이 있는지 확인하십시오.

#### 프로시저

1. `http://machine.domain:2001` 또는 `https://machine.domain:2010`로 이동하십시오. 여기서 *machine* 은 컴퓨터의 이름입니다.

   사용자 이름 및 비밀번호를 요청하는 대화 상자가 표시됩니다.
2. 올바른 IBM i 사용자 프로파일 및 비밀번호를 입력하십시오.
3. 자세한 정보를 보려면 암호화로 이동하여 해당 링크를 클릭하십시오.

#### 다음에 수행할 작업

4767 암호화 코프로세서 구성에 대한 자세한 정보는 4767 암호화 코프로세서를 참조하십시오.

## ALW AIX, Linux, and Windows에서 SSL/TLS에 대한 작업

AIX, Linux, and Windows 시스템에서 TLS(Transport Layer Security) 지원이 IBM MQ와 함께 설치됩니다.

**참고:** Deprecated V 9.4.0 IBM MQ 9.4.0부터 IBM MQ Java 애플리케이션과 함께 CMS 키 저장소 및 스태쉬 파일을 사용하는 것은 더 이상 사용되지 않습니다. PKCS #12 키 저장소 사용으로 마이그레이션하고 IBM MQ 비밀번호 보호 시스템을 사용하여 키 저장소 비밀번호를 보호하십시오.

**중요사항:** V 9.4.0 V 9.4.0 IBM MQ 9.4.0부터 CMS 키 저장소 및 스태쉬 파일은 SSL/TLS를 사용하는 AMQP및 MQTT 채널에서 지원되지 않습니다. PKCS #12 키 저장소를 사용하고 대신 IBM MQ 비밀번호 보호 시스템을 사용하여 키 저장소 비밀번호를 보호하십시오.

인증서 유효성 검증 정책에 대한 보다 자세한 정보는 인증서 유효성 검증 및 신뢰 정책 설계의 내용을 참조하십시오.

AIX, Linux, and Windows에서 키 저장소 및 인증서를 관리하는 데 사용되는 명령에 대한 자세한 정보는 505 페이지의 『AIX, Linux, and Windows 의 runmqakm 및 runmqktool 명령』 의 내용을 참조하십시오.

### ALW *AIX, Linux, and Windows*에서 키 저장소 설정

새 키 저장소를 작성하려면 이 프로시저를 따르십시오.

#### 시작하기 전에

키 저장소는 중요한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소를 작성하기 전에 키 저장소 비밀번호를 안전하게 저장하기 위해 IBM MQ 에서 제공하는 옵션을 검토하십시오. 자세한 정보는 279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』 의 내용을 참조하십시오.

**참고:** Deprecated V 9.4.0 IBM MQ 9.4.0부터 IBM MQ Java 애플리케이션과 함께 CMS 키 저장소 및 스태쉬 파일을 사용하는 것은 더 이상 사용되지 않습니다. PKCS #12 키 저장소 사용으로 마이그레이션하고 IBM MQ 비밀번호 보호 시스템을 사용하여 키 저장소 비밀번호를 보호하십시오.

**중요사항:** V 9.4.0 V 9.4.0 IBM MQ 9.4.0부터 CMS 키 저장소 및 스태쉬 파일은 SSL/TLS를 사용하는 AMQP및 MQTT 채널에서 지원되지 않습니다. PKCS #12 키 저장소를 사용하고 대신 IBM MQ 비밀번호 보호

시스템을 사용하여 키 저장소 비밀번호를 보호하십시오. 다음 명령을 사용하여 PKCS #12 키 저장소를 작성할 수 있습니다.

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

이 명령은 지정된 비밀번호로 보안되는 *filename*.p12 라는 PKCS #12 키 저장소 파일을 작성합니다.

## 이 태스크 정보

TLS 연결의 경우 연결의 각 끝에 키 저장소가 있어야 합니다. 각 IBM MQ 큐 관리자 및 IBM MQ MQI client에는 키 저장소에 대한 액세스가 있어야 합니다. 자세한 정보는 23 페이지의 『SSL/TLS 키 저장소』의 내용을 참조하십시오.

디지털 인증서는 키 저장소에 저장됩니다. 이러한 디지털 인증서에는 레이블이 있습니다. 인증서 레이블은 개인 인증서를 특정 큐 관리자 또는 IBM MQ MQI client와 연관시킵니다. TLS가 인증 목적으로 해당 인증서를 사용합니다. AIX, Linux, and Windows 시스템에서 IBM MQ 는 인증서 레이블에 대해 다음 값 중 하나를 사용합니다.

- **CERTLABL** 큐 관리자 또는 채널 속성 (설정된 경우) 의 값입니다.
- 큐 관리자의 이름 또는 IBM MQ MQI client 사용자 로그온 ID가 추가된 ibmwebspheremq의 기본값은 모두 소문자입니다.

자세한 정보는 디지털 인증서 레이블을 참조하십시오.

키 저장소 파일 이름은 경로 및 스템 이름으로 구성됩니다.

- AIX and Linux 시스템에서 큐 관리자의 기본 경로(큐 관리자를 작성할 때 설정됨)는 /var/mqm/qmgrs/*queue_manager_name*/ssl입니다.

  Windows 시스템에서 기본 경로는 *MQ_DATA_PATH*\qmgrs*queue_manager_name*\ssl입니다. 여기서 *MQ_DATA_PATH* 는 IBM MQ설치 중에 선택되는 데이터 경로입니다. 예를 들어, C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl입니다.

  기본 파일 이름은 key.kdb입니다. 또는 사용자 고유의 경로 및 파일 이름을 사용할 수 있습니다.

  사용자 고유의 경로나 파일 이름을 선택하는 경우에는 이에 대한 액세스를 엄격하게 제어하기 위해 파일에 권한을 설정하십시오.
- IBM MQ 클라이언트의 경우 기본 경로 또는 파일 이름이 없습니다. 이 파일에 대한 액세스를 엄격하게 제어하십시오.

파일 레벨 잠금을 지원하지 않는 파일 시스템에서 키 저장소를 작성하지 마십시오(예: Linux 시스템에서 NFS 버전 2).

키 데이터베이스 파일 이름 검사 및 지정에 대한 자세한 정보는 282 페이지의 『AIX, Linux, and Windows에서 큐 관리자의 키 저장소 위치 변경』의 내용을 참조하십시오. 키 저장소가 작성되기 전이나 후에 키 데이터베이스 파일 이름을 지정할 수 있습니다.

**runmqakm** (GSKCapiCmd) 또는 V 9.4.0 V 9.4.0 **runmqktool** (keytool) 명령을 사용하여 IBM MQ에서 사용되는 키 저장소를 관리할 수 있습니다. 자세한 정보는 505 페이지의 『AIX, Linux, and Windows의 runmqakm 및 runmqktool 명령』의 내용을 참조하십시오.

키 저장소를 관리하기 위해 명령을 실행하는 사용자 ID에는 키 저장소 파일이 작성되거나 업데이트되는 디렉토리에 대한 쓰기 권한이 있어야 합니다. 기본 ssl 디렉토리를 사용하는 큐 관리자의 경우 **runmqakm** 또는 **runmqktool** 명령을 실행하는 사용자 ID는 mqm 그룹의 구성원이어야 합니다. IBM MQ MQI client의 경우, 클라이언트를 실행하는 사용자 ID와 다른 사용자 ID에서 **runmqakm** 또는 **runmqktool** 를 실행하는 경우 IBM MQ MQI client 가 키 저장소에 액세스할 수 있도록 파일 권한을 변경해야 합니다. 자세한 정보는 280 페이지의 『Windows에서 키 데이터베이스 파일 액세스 및 보안』 또는 280 페이지의 『AIX and Linux 시스템에서 키 데이터베이스 파일 액세스 및 보안』의 내용을 참조하십시오.

**runmqakm** 명령을 사용하여 비어 있는 새 키 저장소를 작성할 수 있습니다. V 9.4.0 V 9.4.0 **runmqktool** 명령을 대신 사용하는 경우 인증서를 작성하거나 가져오기 위해 명령을 실행할 때 키 저장소가 작성됩니다.

**참고:** FIPS를 준수하는 방법으로 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

## 프로시저

1. 다음 명령을 실행하여 **runmqakm** 명령으로 키 저장소를 작성하십시오.

```
runmqakm -keydb -create -db filename -pw password -type type
         -stash -fips -strong
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-type *type***
V 9.4.0 V 9.4.0 키 저장소의 유형을 지정합니다. IBM MQ에서 사용되는 키 저장소의 경우 가능한 값은 다음과 같습니다.

- pkcs12
- Deprecated cms

  **참고:** IBM MQ 9.4.0부터 CMS 키 저장소 및 스태쉬 파일의 사용은 IBM MQ Java 애플리케이션에 대해 더 이상 사용되지 않으며 SSL/TLS를 사용하는 AMQP및 MQTT 채널에 대해 지원되지 않습니다.

**-stash**
선택사항. 키 저장소 비밀번호를 스태쉬 파일에 저장하려면 이 옵션을 지정하십시오. 대신 IBM MQ 비밀번호 보호 시스템을 사용하여 비밀번호를 암호화하는 경우 스태쉬 파일에 비밀번호를 저장할 필요가 없습니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

**-strong**
입력한 비밀번호가 비밀번호 강도에 대한 최소 요구사항을 충족하는지 검사합니다. 비밀번호의 최소 요구사항은 다음과 같습니다.

- 비밀번호는 최소 14자여야 합니다.
- 비밀번호에는 최소 하나의 소문자, 하나의 대문자 및 하나의 숫자 또는 특수 문자를 포함해야 합니다. 특수 문자에는 별표(*), 달러 부호($), 숫자 부호(#) 및 퍼센트 부호(%)가 포함됩니다. 공백은 특수 문자로 분류됩니다.
- 각 문자는 비밀번호에서 최대 3회 사용될 수 있습니다.
- 비밀번호에서 최대 두 개의 연속하는 문자가 동일할 수 있습니다.
- 모든 문자는 0x20 - 0x7E 범위 내의 표준 ASCII 인쇄 가능한 문자 세트에 있습니다.

2. 280 페이지의 『Windows에서 키 데이터베이스 파일 액세스 및 보안』 또는 280 페이지의 『AIX and Linux 시스템에서 키 데이터베이스 파일 액세스 및 보안』에 설명된 대로 키 저장소 파일에 대한 액세스 권한을 설정하십시오.

Windows에서는 기본적으로 키 저장소를 작성하기 위해 명령을 실행한 사용자 ID에만 스태쉬 (.sth) 파일을 읽을 수 있는 액세스 권한이 부여됩니다. **runmqakm** 명령을 사용하여 스태쉬 파일을 작성한 후 파일 권한을 확인하고 큐 관리자를 실행하는 서비스 계정 또는 로컬 mqm와 같은 그룹에 권한을 부여하십시오.

3. 스태쉬 파일을 사용하지 않는 경우 282 페이지의 『AIX, Linux, and Windows 에서 큐 관리자의 키 저장소 비밀번호 제공』 또는 284 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』의 지시사항에 따라 큐 관리자 또는 클라이언트 애플리케이션에 키 저장소 비밀번호를 제공하십시오.

## 다음에 수행할 작업

필요한 경우 비어 있는 키 저장소에 기본 인증 기관 (CA) 인증서를 추가하십시오. 자세한 정보는 281 페이지의 『AIX, Linux, and Windows 에서 비어 있는 키 저장소에 기본 CA 인증서 추가』의 내용을 참조하십시오.

ALW *AIX, Linux, and Windows*에서 키 저장소 보호를 위한 강력한 비밀번호 생성

**runmqakm**(GSKCapiCmd) 명령을 사용하여 키 저장소 보호를 위해 강력한 비밀번호를 생성할 수 있습니다.

강력한 비밀번호를 생성하기 위해 다음 매개변수와 함께 **runmqakm** 명령을 사용할 수 있습니다.

```
runmqakm -random -create -length password_length -strong -fips
```

여기서 *password_length* 는 생성할 비밀번호의 길이입니다. 지정할 수 있는 최소 비밀번호 길이는 14입니다.

후속 인증서 관리 명령의 **-pw** 매개변수에서 생성된 비밀번호를 사용할 때 항상 비밀번호 주변에 큰따옴표를 두십시오. AIX and Linux 시스템에서 다음 문자가 비밀번호 문자열에 나타나는 경우 백슬래시 문자를 사용하여 이스케이프 처리해야 합니다.

```
!  \  "  '
```

**runmqakm** 또는 V 9.4.0 V 9.4.0 **runmqktool** 명령의 프롬프트에 대한 응답으로 키 저장소 비밀번호를 입력할 때 운영 체제 쉘이 이러한 경우 데이터 입력에 영향을 주지 않으므로 비밀번호를 따옴표로 묶거나 이스케이프할 필요가 없습니다.

ALW *AIX, Linux, and Windows* 에서 키 저장소 비밀번호 암호화

여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

다음 IBM MQ 컴포넌트 및 기능은 키 저장소 비밀번호를 저장하기 위한 두 가지 다른 방법을 지원합니다.

- 큐 관리자 TLS키 저장소입니다.
- TLS를 사용하는 IBM MQ MQI clients .
- V 9.4.0 `qm.ini` 파일의 **NativeHALocalInstance** 스탠자에 있는 고유 HA 구성입니다.
- V 9.4.0 `qm.ini` 파일의 **AuthToken** 스탠자에 있는 토큰 인증 구성입니다.

이러한 구성요소에서 사용할 키 저장소 비밀번호는 다음 방법 중 하나를 사용하여 암호화하고 저장할 수 있습니다.

**IBM MQ 비밀번호 보호 시스템입니다.**

각 IBM MQ 컴포넌트는 키 저장소 비밀번호를 암호화하는 명령을 제공합니다. 명령이 출력하는 암호화된 명령은 파일에 저장됩니다.

큐 관리자 TLS키 저장소의 경우, 비밀번호는 **SSLKEYRPWD** 큐 관리자 속성이 설정될 때 암호화됩니다.

비밀번호는 AES-128 알고리즘으로 암호화됩니다. 이 알고리즘의 세부사항은 공개적으로 알려져 있으며 안전한 것으로 간주됩니다.

비밀번호는 키 저장소에 액세스할 수 있는 다른 소프트웨어에서 이해하지 못하는 독점 형식으로 저장됩니다.

하나의 IBM MQ 구성요소에 의해 암호화되는 비밀번호는 다른 IBM MQ 구성요소에 의해 사용될 수 없습니다.

키 저장소 비밀번호가 암호화될 때 고유한 암호화 키를 제공할 수 있습니다. 고유한 암호화 키는 암호화 키에 대한 액세스 권한이 없는 사용자가 비밀번호를 복호화할 수 없도록 합니다.

키 저장소에 있는 인증서를 관리하려면 일반 텍스트 키 저장소 비밀번호가 필요합니다. IBM MQ 비밀번호 보호 시스템을 사용하여 키 저장소 비밀번호를 암호화하는 것 외에도 키 저장소 비밀번호를 이 용도로 액세스할 수 있는 안전한 위치에 저장해야 합니다.

IBM MQ 비밀번호 보호 시스템에 대한 자세한 정보는 527 페이지의 『IBM MQ 컴포넌트 구성 파일에서 비밀번호 보호』의 내용을 참조하십시오.

**키 저장소 스태쉬 파일입니다.**

**runmqakm** 명령은 키 저장소 비밀번호를 스태쉬 파일에 저장할 수 있습니다.

비밀번호는 IBM MQ의 암호화 제공자인 IBM Global Security Kit (GSKit)에 특정한 독점 메소드를 사용하여 암호화됩니다.

고유한 암호화 키를 제공할 수 없습니다.

암호화된 비밀번호는 키 저장소 파일과 동일한 디렉토리의 스태쉬 파일에 저장됩니다.

키 저장소 및 스태쉬 파일 모두에 대한 읽기 액세스 권한이 있는 사용자는 누구나 키 저장소의 컨텐츠에 액세스하고 관리할 수 있습니다.

**참고:** Deprecated V 9.4.0 IBM MQ 9.4.0부터 IBM MQ Java 애플리케이션과 함께 스태쉬 파일을 사용하는 것은 더 이상 사용되지 않습니다.

**중요사항:** V 9.4.0 V 9.4.0 IBM MQ 9.4.0부터 스태쉬 파일은 TLS를 사용하는 AMQP및 MQTT 채널에서 지원되지 않습니다.

키 저장소 비밀번호를 암호화하기 위해 선택하는 방법에 관계없이 저장된 비밀번호를 암호화하는 데 대한 제한사항을 알고 있는지 확인하십시오. 자세한 정보는 533 페이지의 『비밀번호 암호화를 통한 보호 한계』의 내용을 참조하십시오.

**관련 개념**

282 페이지의 『AIX, Linux, and Windows 에서 큐 관리자의 키 저장소 비밀번호 제공』
키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

284 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』
키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

276 페이지의 『AIX, Linux, and Windows에서 SSL/TLS에 대한 작업』
AIX, Linux, and Windows 시스템에서 TLS(Transport Layer Security) 지원이 IBM MQ와 함께 설치됩니다.

Windows *Windows*에서 키 데이터베이스 파일 액세스 및 보안
키 데이터베이스 파일에는 적합한 액세스 권한이 없을 수도 있습니다. 이러한 파일에 대한 적합한 액세스를 설정해야 합니다.

*key*.p12, *key*.kdb, *key*.sth, *key*.crl및 *key*.rdb파일에 대한 액세스 제어를 설정하십시오. 여기서 *key*는 키 데이터베이스의 스템 이름이며 제한된 사용자 세트에 권한을 부여합니다.

.p12 또는 .kdb이외의 다른 키 저장소 확장을 사용한 경우 이 파일의 권한이 설정되어 있는지도 확인해야 합니다.

다음과 같은 액세스 부여를 고려하십시오.

**전체 권한**
BUILTIN\Administrators, NT AUTHORITY\SYSTEM 및 데이터베이스 파일을 작성한 사용자입니다.

**읽기 권한**
큐 관리자의 경우 로컬 mqm 그룹만 해당됩니다. 이는 MCA가 mqm 그룹에 있는 사용자 ID 하에서 실행 중이라고 가정합니다.

클라이언트의 경우 클라이언트 프로세스를 실행하고 있는 사용자 ID입니다.

Linux AIX *AIX and Linux* 시스템에서 키 데이터베이스 파일 액세스 및 보안
키 데이터베이스 파일에는 적합한 액세스 권한이 없을 수도 있습니다. 이러한 파일에 대한 적합한 액세스를 설정해야 합니다.

큐 관리자의 경우 큐 관리자 및 채널 프로세스가 필요할 때 읽을 수 있고 다른 사용자는 읽을 수 없도록 키 데이터베이스 파일에 대한 권한을 설정하십시오. 일반적으로 mqm 사용자는 읽기 권한이 필요합니다. mqm 사용자로

서 로그인하여 키 데이터베이스 파일을 작성한 경우에는 권한이 충분할 것입니다. mqm 사용자가 아니지만 mqm 그룹의 다른 사용자인 경우에는 mqm 그룹의 다른 사용자에게 읽기 권한을 부여해야 할 수도 있습니다.

마찬가지로 클라이언트의 경우 클라이언트 애플리케이션 프로세스가 필요할 때 읽을 수 있지만 다른 사용자는 읽거나 수정할 수 없도록 키 데이터베이스 파일에 대한 권한을 설정하십시오. 일반적으로 클라이언트 프로세스를 실행하는 사용자는 읽기 권한이 필요합니다. 해당 사용자로서 로그인하여 키 데이터베이스 파일을 작성한 경우에는 권한이 충분할 것입니다. 클라이언트 프로세스 사용자가 아니라 해당 그룹의 다른 사용자라면 그룹의 다른 사용자에게 읽기 권한을 부여해야 할 수도 있습니다.

*key*.p12, *key*.kdb, *key*.sth, *key*.crl및 *key*.rdb파일에 대한 권한을 설정하십시오. 여기서 *key* 는 키 데이터베이스의 스템 이름이고, 파일 소유자의 경우 read 및 write , mqm 또는 클라이언트 사용자 그룹 (-rw - r -----) 의 경우 read 로 설정하십시오.

.p12 또는 .kdb이외의 다른 키 저장소 확장을 사용한 경우 이 파일의 권한이 설정되어 있는지도 확인해야 합니다.

ALW *AIX, Linux, and Windows* 에서 비어 있는 키 저장소에 기본 *CA* 인증서 추가

하나 이상의 기본 인증 기관 (CA) 인증서를 비어 있는 키 저장소에 추가하려면 이 프로시저를 따르십시오.

새 키 저장소를 작성할 때 비어 있습니다. **runmqakm** 명령을 사용하여 키 저장소에 기본 CA 인증서를 추가할 수 있습니다.

### runmqakm 사용

**runmqakm** 명령을 사용하여 키 저장소에 기본 CA 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -populate -db filename -pw password
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**참고:** IBM MQ 는 키 저장소의 CA 인증서에 의해 서명된 모든 인증서를 신뢰합니다. 신뢰할 인증 기관을 신중하게 고려하고 클라이언트 및 큐 관리자를 인증하는 데 필요한 CA 인증서만 추가하십시오. 기본 CA 인증서의 전체 세트를 키 저장소에 추가하는 것은 권장되지 않습니다.

## ALW *AIX, Linux, and Windows*에서 키 저장소에서 큐 관리자 찾기

큐 관리자의 키 데이터베이스 파일 위치를 확보하려면 이 프로시저를 사용하십시오.

### 프로시저

1. 다음 MQSC 명령을 사용하여 큐 관리자의 속성을 표시하십시오.

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

   IBM MQ Explorer 또는 PCF 명령을 사용하여 큐 관리자의 속성을 표시할 수도 있습니다.
2. 키 데이터베이스 파일의 경로 및 스템 이름의 명령 출력을 검사하십시오.
   예:

   a. AIX and Linux의 경우: /var/mqm/qmgrs/QM1/ssl/key. 여기서 /var/mqm/qmgrs/QM1/ssl은 경로이고 key는 스템 이름입니다.

   b. Windows의 경우: *MQ_INSTALLATION_PATH*\qmgrs\QM1\ssl\key. 여기서 *MQ_INSTALLATION_PATH*\qmgrs\QM1\ssl 는 경로이고 key 은 스템 이름입니다. *MQ_INSTALLATION_PATH*는 IBM MQ가 설치되어 있는 상위 레벨 디렉토리를 나타냅니다.

**참고:** IBM MQ 9.3.0 부터 SSLKEYR 필드는 전체 파일 이름 (확장자 포함) 및 스템 이름 (확장자 없음) 을 모두 지원합니다. 스템 이름이 설정되면 IBM MQ 는 자동으로 .kdb 를 추가하고 해당 키 저장소를 사용합니다.

### ALW *AIX, Linux, and Windows*에서 큐 관리자의 키 저장소 위치 변경

MQSC 명령 ALTER QMGR을 포함한 다양한 수단을 사용하여 큐 관리자의 키 데이터베이스 파일의 위치를 변경할 수 있습니다.

큐 관리자의 키 저장소 속성을 설정하려면 MQSC 명령 ALTER QMGR을 사용하여 큐 관리자의 키 데이터베이스 파일 위치를 변경할 수 있습니다. 예를 들어 AIX and Linux에서는 다음과 같습니다.

```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey.kdb')
```

Windows의 경우:

```
ALTER QMGR SSLKEYR('C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb')
```

**주의:** Windows 및 Linux에서 TLS AMQP 채널이 사용되는 경우 키 저장소 파일의 접미부는 다음 중 하나여야 합니다.

- .kdb, CMS 키 저장소의 경우
- .p12 또는 .pkcs12(PKCS #12 키 저장소의 경우).

IBM MQ 탐색기 또는 PCF 명령을 사용하여 큐 관리자의 속성을 변경할 수도 있습니다.

큐 관리자의 키 데이터베이스 파일의 위치를 변경할 때 인증서는 이전 위치에서 전송되지 않습니다. 현재 액세스 중인 키 데이터베이스 파일이 새 키 데이터베이스 파일인 경우 516 페이지의 『AIX, Linux, and Windows에서 개인 인증서를 키 저장소로 가져오기』에 설명된 대로 필요한 CA 및 개인용 인증서로 채워야 합니다.

### *AIX, Linux, and Windows* 에서 큐 관리자의 키 저장소 비밀번호 제공

키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

IBM MQ 는 큐 관리자에 키 저장소 비밀번호를 제공하는 두 가지 메커니즘을 제공합니다.

- 282 페이지의 『KEYRPWD 속성』
- 283 페이지의 『키 저장소 스태쉬 파일』

키 저장소 스태쉬 파일을 사용하지 않는 경우 키 저장소 비밀번호는 IBM MQ 비밀번호 보호 시스템을 사용하여 암호화됩니다. 키 저장소 비밀번호를 보호하는 방법에 대한 자세한 정보는 279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

### KEYRPWD 속성

큐 관리자에 직접 키 저장소 비밀번호를 제공하려면 다음 MQSC 명령을 실행하여 *password* 를 키 저장소 비밀번호로 대체하십시오.

```
ALTER QMGR KEYRPWD('password')
```

**주의:** 비밀번호를 작은따옴표로 묶어야 합니다. 그렇지 않으면 IBM MQ 가 문자를 대문자로 변환합니다.

이 방법을 사용하여 키 저장소 비밀번호를 지정하면 비밀번호가 저장되기 전에 IBM MQ 비밀번호 보호 시스템을 사용하여 암호화됩니다.

초기 키라고 하는 암호화 키는 비밀번호를 암호화하는 데 사용됩니다. 고유한 초기 키를 사용하여 비밀번호를 안전하게 보호하도록 큐 관리자를 설정하십시오. 초기 키를 제공하지 않으면 기본 키가 사용됩니다.

키 저장소 비밀번호를 설정하기 전에 큐 관리자가 고유한 초기 키로 구성되었는지 확인하십시오. **ALTER QMGR** 명령에서 **INITKEY** 속성을 사용하여 초기 키를 수정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
ALTER QMGR INITKEY('mykey')
```

**경고:** 키 저장소 비밀번호를 설정한 후 초기 키를 수정해도 키 저장소 비밀번호가 새 초기 키로 암호화되지 않습니다. 키 저장소 비밀번호를 재설정하지 않고 초기 키를 변경하면 IBM MQ 가 키 저장소 비밀번호를 복호화할 수 없으므로 키 저장소에 액세스할 수 없습니다.

**KEYRPWD** 속성에 대한 자세한 정보는 KEYRPWD를 참조하십시오.

## 키 저장소 스태쉬 파일

**KEYRPWD** 속성을 사용하여 키 저장소 비밀번호가 큐 관리자에 제공되지 않은 경우, IBM MQ 는 키 저장소와 동일한 디렉토리에 스태쉬 파일이 있다고 가정합니다. 스태쉬 파일에는 키 저장소와 동일한 스템 이름이 있지만 확장자는 `.sth` 입니다.

키 저장소 스태쉬 파일은 키 저장소와 동시에 또는 이후에 별도의 **runmqakm** 명령으로 작성됩니다.

**주의:** 스태쉬 파일의 형식은 IBM MQ 암호화 제공자 IBM Global Security Kit (GSKit)에 특정하며 다른 암호화 제공자를 사용하는 플랫폼에서는 사용할 수 없습니다.

키 저장소가 작성될 때 스태쉬 파일을 작성하려면 **-stash** 매개변수를 지정하십시오. 예를 들면, 다음과 같습니다.

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

여기서 *passw0rd* 는 키 저장소 비밀번호입니다.

나중에 스태쉬 파일을 작성하려면 다음 명령을 실행하십시오.

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

여기서 *passw0rd* 는 키 저장소 비밀번호입니다.

**관련 개념**

279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』
여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

284 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』
키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

## ALW *AIX, Linux, and Windows 에서 IBM MQ MQI client 의 키 저장소 찾기*

키 저장소의 위치는 MQSSLKEYR 변수에 의해 제공되거나 MQCONNX 호출에 지정됩니다.

IBM MQ MQI client의 키 데이터베이스 파일의 위치를 찾으려면 MQSSLKEYR 환경 변수를 검사하십시오. 예를 들면, 다음과 같습니다.

```
echo $MQSSLKEYR
```

키 데이터베이스 파일 이름은 283 페이지의 『AIX, Linux, and Windows 에서 IBM MQ MQI client 의 키 저장소 위치 지정』에 설명된 대로 MQCONNX 호출에 설정될 수도 있으므로 애플리케이션도 검사하십시오. MQCONNX 호출에 설정한 값은 MQSSLKEYR의 값을 대체합니다.

## ALW *AIX, Linux, and Windows 에서 IBM MQ MQI client 의 키 저장소 위치 지정*

IBM MQ MQI client의 기본 키 저장소가 없습니다. 위치를 다음 두 방법 중 하나로 지정할 수 있습니다. 키 데이터베이스 파일이 다른 시스템으로 무단 복사되는 것을 막기 위해 의도한 사용자 또는 관리자만이 액세스할 수 있는지 확인하십시오.

IBM MQ MQI client의 키 데이터베이스 파일의 위치를 두 가지 방법으로 지정할 수 있습니다.

- MQSSLKEYR 환경 변수 설정. 예를 들어 AIX and Linux에서는 다음과 같습니다.

```
export MQSSLKEYR=/var/mqm/ssl/key.kdb
```

  Windows의 경우:

```
set MQSSLKEYR=C:\Program Files\IBM\MQ\ssl\key.kdb
```

- 애플리케이션이 MQCONNX 호출을 할 때 MQSCO 구조의 *KeyRepository* 필드에 키 데이터베이스 파일의 경로 및 스템 이름 제공. MQCONNX에서 MQSCO 구조의 사용에 대한 자세한 정보는 MQSCO 개요의 내용을 참조하십시오.

### *AIX, Linux, and Windows* 의 *IBM MQ MQI client* 에 대한 키 저장소 비밀번호 제공

키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

IBM MQ 에서는 IBM MQ MQI client에 키 저장소 비밀번호를 제공하기 위한 네 가지 메커니즘을 제공합니다.

- 284 페이지의 『MQSCO의 KeyRepoPassword 필드』
- 285 페이지의 『MQKEYRPWD 환경 변수』
- 285 페이지의 『클라이언트 구성 파일의 SSLKeyRepositoryPassword 속성』
- 285 페이지의 『키 저장소 스태쉬 파일』

키 저장소 스태쉬 파일을 사용하지 않는 경우 키 저장소 비밀번호를 일반 텍스트 문자열 또는 IBM MQ 비밀번호 보호 시스템을 사용하여 암호화되는 문자열로 제공할 수 있습니다. 키 저장소 비밀번호를 보호하는 방법에 대한 자세한 정보는 279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

### MQSCO의 KeyRepoPassword 필드

MQSCO 구조를 사용하여 키 저장소 비밀번호를 제공하려면 다음 세 가지 변수 문자열 필드의 조합을 사용해야 합니다.

**KeyRepoPasswordLength**
비밀번호의 길이입니다.

**KeyRepoPasswordPtr**
비밀번호를 포함하는 메모리의 위치에 대한 포인터입니다.

**KeyRepoPasswordOffset**
메모리에서 비밀번호의 위치이며 MQSCO 구조의 시작부터 바이트 수로 표시됩니다.

**참고: KeyRepoPasswordPtr** 또는 **KeyRepoPasswordOffset**중 하나만 제공할 수 있습니다.

예를 들면, 다음과 같습니다.

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

**주의:** 이 방법을 사용하여 비밀번호를 제공하는 경우 IBM MQ client 애플리케이션에 제공되기 전에 비밀번호를 암호화하십시오. 자세한 정보는 285 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

MQCSO 구조에 대한 자세한 정보는 MQSCO-SSL/TLS 구성 옵션을 참조하십시오.

### *MQKEYRPWD* 환경 변수

MQSCO 구조를 사용하여 키 저장소 비밀번호가 클라이언트에 제공되지 않은 경우 *MQKEYRPWD* 환경 변수를 사용하여 키 저장소 비밀번호를 지정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
export MQKEYRPWD=passw0rd
```

또는

```
set MQKEYRPWD=passw0rd
```

여기서 passw0rd 는 비밀번호입니다.

**주의:** 이 방법을 사용하여 비밀번호를 제공하는 경우 환경 변수의 값을 설정하기 전에 비밀번호를 암호화하십시오. 자세한 정보는 285 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

### 클라이언트 구성 파일의 SSLKeyRepositoryPassword 속성

키 저장소 비밀번호가 다른 방법 중 하나를 사용하여 클라이언트에 제공되지 않는 경우, 클라이언트 구성 파일의 **SSL** 스탠자에서 **SSLKeyRepositoryPassword** 속성을 사용하여 키 저장소 비밀번호를 지정할 수 있습니다. 예를 들면, 다음과 같습니다.

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

**주의:** 이 방법을 사용하여 비밀번호를 제공하는 경우 **SSLKeyRepositoryPassword** 속성의 값을 설정하기 전에 비밀번호를 암호화하십시오. 자세한 정보는 285 페이지의 『키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

클라이언트 구성 파일의 SSL 스탠자에 대한 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오.

### 키 저장소 스태쉬 파일

키 저장소 비밀번호가 다른 방법 중 하나를 사용하여 클라이언트에 제공되지 않으면 IBM MQ 는 키 저장소와 동일한 디렉토리에 스태쉬 파일이 있다고 가정합니다. 스태쉬 파일에는 키 저장소와 동일한 스템 이름이 있지만 확장자는 .sth 입니다.

키 저장소 스태쉬 파일은 키 저장소와 동시에 작성되거나 별도의 **runmqakm** 명령을 사용하여 나중에 작성됩니다.

**주의:** 스태쉬 파일의 형식은 IBM MQ 암호화 제공자 IBM Global Security Kit (GSKit)에 특정하며 다른 암호화 제공자를 사용하는 플랫폼에서는 사용할 수 없습니다.

키 저장소가 작성될 때 스태쉬 파일을 작성하려면 **-stash** 매개변수를 지정하십시오. 예를 들면, 다음과 같습니다.

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

여기서 *passw0rd* 는 키 저장소 비밀번호입니다.

나중에 스태쉬 파일을 작성하려면 다음 명령을 실행하십시오.

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

여기서 *passw0rd* 는 키 저장소 비밀번호입니다.

### 키 저장소 비밀번호 암호화

스태쉬 파일 이외의 방법을 사용하여 키 저장소 비밀번호를 제공하는 경우 IBM MQ 비밀번호 보호 시스템을 사용하여 비밀번호를 암호화하십시오. 비밀번호를 암호화하려면 **runmqicred** 명령을 실행하십시오. 프롬프트가 표시되면 키 저장소 비밀번호를 입력하십시오. 이 명령은 암호화된 비밀번호를 출력합니다. 설명된 방법을 사용하여 일반 텍스트 비밀번호 대신 암호화된 비밀번호를 IBM MQ MQI client 에 제공할 수 있습니다.

초기 키라고 하는 암호화 키는 비밀번호를 암호화하는 데 사용됩니다. 비밀번호를 암호화할 때 고유한 초기 키를 사용하여 비밀번호를 안전하게 보호하십시오. 사용자 고유의 초기 키를 제공하려면 **-sf** 매개변수를 **runmqicred** 명령에 사용하십시오. 초기 키를 제공하지 않으면 기본 키가 사용됩니다.

자세한 정보는 runmqicred ( IBM MQ 클라이언트 비밀번호 보호)를 참조하십시오.

키 저장소 비밀번호가 암호화될 때 사용자 고유의 초기 키를 제공하고 암호화된 비밀번호를 IBM MQ MQI client에 제공하는 경우에도 IBM MQ MQI client에 동일한 초기 키를 제공해야 합니다. IBM MQ MQI client에 초기 키를 제공하는 방법에 대한 자세한 정보는 286 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 초기 키 제공』의 내용을 참조하십시오.

**관련 개념**

279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』
여러 IBM MQ 구성요소가 디지털 인증서 또는 대칭 키를 포함하는 키 저장소에 액세스해야 합니다. 키 저장소는 민감한 정보를 포함하므로 비밀번호로 보호됩니다. 키 저장소 비밀번호는 키 저장소에 액세스할 때 IBM MQ 가 읽을 수 있는 위치에 저장되어야 합니다. 또한 키 저장소에 대한 권한이 없는 액세스 가능성을 줄이기 위해 비밀번호를 암호화해야 합니다.

282 페이지의 『AIX, Linux, and Windows 에서 큐 관리자의 키 저장소 비밀번호 제공』
키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

ALW *AIX, Linux, and Windows* 의 *IBM MQ MQI client* 에 대한 초기 키 제공

IBM MQ 비밀번호 보호 시스템을 사용하여 암호화된 IBM MQ MQI client 에 변수를 제공하는 경우 값을 암호화하는 데 사용된 해당 초기 키를 제공해야 합니다.

값을 암호화할 때 초기 키를 지정하지 않은 경우에는 IBM MQ client에 초기 키 값을 제공할 필요가 없습니다. 그러나 고유한 초기 키를 사용한 경우 다음 방법을 사용하여 IBM MQ client 에 초기 키를 제공할 수 있습니다.

- 286 페이지의 『MQCSP 구조를 사용하여 초기 키 제공』
- 286 페이지의 『MQS_MQI_KEYFILE 환경 변수를 사용하여 초기 키 제공』
- 287 페이지의 『클라이언트 구성 파일을 사용하여 초기 키 제공』

## MQCSP 구조를 사용하여 초기 키 제공

MQCSP 구조를 사용하여 초기 키를 제공하려면 다음 세 변수 문자열 필드의 조합을 사용해야 합니다.

**InitialKeyLength**
초기 키의 길이

**InitialKeyPtr**
초기 키를 포함하는 메모리의 위치에 대한 포인터

**InitialKeyOffset**
MQCSP 구조의 시작부터 바이트 수로 표시되는 메모리의 초기 키 위치입니다.

**참고: InitialKeyPtr** 또는 **InitialKeyOffset**중 하나만 제공할 수 있습니다.

예를 들면, 다음과 같습니다.

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

## MQS_MQI_KEYFILE 환경 변수를 사용하여 초기 키 제공

MQCSP 구조를 사용하여 클라이언트에 초기 키가 제공되지 않는 경우 IBM MQ 는 *MQS_MQI_KEYFILE* 환경 변수를 검사합니다. 이 환경 변수를 사용할 초기 키로 구성된 단일 텍스트 행을 포함하는 파일의 위치로 설정해야 합니다.

예를 들어, mykey.key 파일이 루트 디렉토리에 있고 초기 키를 포함하는 경우 환경 변수를 다음과 같이 설정해야 합니다.

```
export MQS_MQI_KEYFILE=/mykey.key
```

또는

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

## 클라이언트 구성 파일을 사용하여 초기 키 제공

초기 키가 이전 메커니즘을 사용하여 클라이언트에 제공되지 않는 경우 IBM MQ 는 mqclient.ini 파일의 보안 스탠자의 **MQIInitialKeyFile** 속성을 확인합니다. 이 속성을 사용할 초기 키로 구성된 단일 텍스트 행을 포함하는 파일의 위치로 설정해야 합니다.

예를 들어, mykey.key 라는 파일이 루트 디렉토리에 있고 초기 키를 포함하는 경우 클라이언트 구성 파일은 다음을 포함해야 합니다.

```
Security:
    MQIInitialKeyFile=/mykey.key
```

**관련 개념**

284 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』
키 저장소에 민감한 정보가 포함되어 있으므로 비밀번호로 보안됩니다. TLS 조작을 수행하기 위해 키 저장소 컨텐츠에 액세스할 수 있으려면 IBM MQ 가 키 저장소 비밀번호를 검색할 수 있어야 합니다.

260 페이지의 『SSL/TLS에 대한 작업』
이 주제에서는 IBM MQ을 사용하여 TLS 사용과 관련된 단일 태스크를 수행하기 위한 지시사항을 제공합니다.

## ALW *AIX, Linux, and Windows*에서 인증서 또는 키 저장소의 변경사항이 적용되는 시기

키 저장소에서 인증서를 변경하거나 키 저장소의 위치를 변경하는 경우 채널의 유형 및 채널이 실행되는 방법에 따라 다른 시간에 변경사항이 적용됩니다.

키 저장소의 인증서 또는 키 저장소의 위치에 대한 변경사항은 다음 상황에서 적용됩니다.

- 새 아웃바운드 단일 프로세스가 처음으로 TLS 채널을 실행할 때.
- 새 인바운드 TCP/IP 단일 채널 프로세스가 처음으로 TLS 채널을 시작하라는 요청을 수신할 때.
- MQSC 명령 **REFRESH SECURITY TYPE(SSL)** 가 TLS 환경을 새로 고치기 위해 실행되는 경우.
- 클라이언트 애플리케이션 프로세스의 경우, 프로세스에서 마지막 TLS 연결이 닫힐 때. 다음 TLS 연결에는 인증서 변경사항이 적용됩니다.
- 프로세스 풀링 프로세스(amqrmppa)의 스레드로서 실행되는 채널의 경우 프로세스 풀링 프로세스가 시작되거나 재시작되고 TLS 채널을 처음 시작할 때. 프로세스 풀링 프로세스가 TLS 채널을 이미 실행한 경우 변경사항을 즉시 적용하려면 MQSC 명령 **REFRESH SECURITY TYPE(SSL)**을 실행하십시오.
- 채널 시작기의 스레드로서 실행되는 채널의 경우 채널 시작기가 시작되거나 재시작되고 TLS 채널을 처음 실행할 때. 채널 시작기 프로세스가 이미 TLS 채널을 실행했으며 변경사항을 즉시 적용하려면 MQSC 명령 **REFRESH SECURITY TYPE(SSL)**를 실행하십시오.
- TCP/IP 리스너의 스레드로서 실행되는 채널의 경우 리스너가 시작되거나 재시작되고 TLS 채널을 시작하라는 요청을 처음으로 수신할 때. 리스너가 이미 TLS 채널을 실행했으며 변경사항을 즉시 적용하려면 MQSC 명령 **REFRESH SECURITY TYPE(SSL)**를 실행하십시오.

IBM MQ Explorer 또는 PCF 명령을 사용하여 IBM MQ TLS 환경을 새로 고칠 수도 있습니다.

**중요사항:** Advanced Message Security (AMS) MCA 인터셉터 또는 AMS 클라이언트에서 사용하는 키 저장소 구성 파일 또는 키 저장소에 대한 변경사항은 큐 관리자 또는 애플리케이션이 재시작될 때 적용됩니다.

## ALW *AIX, Linux, and Windows*에서 암호화 하드웨어 구성

큐 관리자 또는 클라이언트의 암호화 하드웨어를 여러 방법으로 구성할 수 있습니다.

다음 방법 중 하나를 사용하여 AIX, Linux, and Windows에서 큐 관리자에 대한 암호화 하드웨어를 구성할 수 있습니다.

- ALTER QMGR에 설명된 대로 **ALTER QMGR** MQSC 명령을 SSLCRYP 매개변수와 함께 사용하십시오.
- IBM MQ Explorer 를 사용하여 AIX, Linux, and Windows 시스템에서 암호화 하드웨어를 구성하십시오. 자세한 정보는 온라인 도움말을 참조하십시오.

다음 방법 중 하나를 사용하여 AIX, Linux, and Windows 에서 IBM MQ 클라이언트에 대한 암호화 하드웨어를 구성할 수 있습니다.

- **MQSSLCRYP** 환경 변수를 설정하십시오. **MQSSLCRYP** 에 허용되는 값은 ALTER QMGR에 설명된 대로 **SSLCRYP** 매개변수의 경우와 동일합니다. 이 환경 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.
  - Linux AIX AIX and Linux 시스템:

    ```
    export MQSSLCRYP=string
    ```

  - Windows Windows 시스템:

    ```
    SET MQSSLCRYP=string
    ```

  여기서 *string* 는 시스템에 있는 암호화 하드웨어를 구성하는 데 사용할 매개변수 문자열을 나타냅니다.

  **SSLCRYP** 매개변수의 GSK_PKCS11 버전을 사용하는 경우 PKCS #11 토큰 레이블은 하드웨어를 구성할 때 사용한 레이블과 일치해야 합니다.
- IBM MQ client 구성 파일의 SSL 스탠자에서 **SSLCryptoHardware** 속성을 설정하십시오. 허용되는 값은 **ALTER QMGR**에 설명된 대로 **SSLCRYP** 매개변수의 경우와 동일합니다.

  **SSLCRYP** 매개변수의 GSK_PKCS11 버전을 사용하는 경우 PKCS #11 토큰 레이블은 하드웨어를 구성할 때 사용한 레이블과 일치해야 합니다.
- MQCONNX 호출에서 SSL 구성 옵션 구조의 **CryptoHardware** 필드, MQSCO를 설정하십시오. 자세한 정보는 MQSCO의 개요를 참조하십시오.

**주의:** > **MQSSLCRYP** 환경 변수 또는 **SSLCryptoHardware** 속성을 통해 암호화 하드웨어에 대한 구성을 제공할 때 저장하기 전에 비밀번호를 보호해야 합니다. 자세한 정보는 530 페이지의 『암호화 하드웨어를 사용하는 IBM MQ clients』의 내용을 참조하십시오.

이러한 방법을 사용하여 PKCS #11 인터페이스를 사용하는 암호화 하드웨어를 구성한 경우에는 사용자가 구성한 암호화 토큰의 키 데이터베이스 파일에 채널에 사용할 개인 인증서를 저장해야 합니다. 이 프로그램은 524 페이지의 『PKCS #11 하드웨어에서 인증서 관리』에서 설명합니다.

## MQ Appliance IBM MQ Appliance에서 SSL/TLS에 대한 작업

IBM MQ Appliance에는 TLS(Transport Layer Security) 지원이 있습니다.

IBM MQ Appliance에는 인증서를 관리하기 위한 특징적 명령을 가지고 있습니다. 인증서 관리에 대한 자세한 정보는 IBM MQ Appliance 문서, TLS 인증서 관리를 참조하십시오

## z/OS Working with SSL/TLS on z/OS

This information describes how you set up and work with Transport Layer Security (TLS) on z/OS.

Each topic includes examples of performing each task using RACF. You can perform similar tasks using the other external security managers.

On z/OS, you must also set the number of server subtasks that each queue manager uses for processing TLS calls, as described in “Setting the SSLTASKS parameter on z/OS” on page 289.

z/OS TLS support is integral to the operating system, and is known as *System SSL*. System SSL is part of the Cryptographic Services Base element of z/OS. The Cryptographic Services Base members are installed

in the *pdsname*. SIEALNKE partitioned data set (PDS). When you install System SSL, ensure that you choose the appropriate options to provide the CipherSpecs that you require.

If you need to renew a self-signed certificate, see Steps for renewing a self-signed certificate in RACF for more information.

## z/OS *z/OS에서 TLS에 대한 추가 사용자 ID 요구사항*

이 정보에서는 z/OS에서 TLS를 설정하고 작업하기 위해 사용자 ID에 필요한 추가 요구사항을 설명합니다.

시스템에 적합한 HIPER(High Impact or Pervasive) 업데이트가 모두 있는지 확인하십시오.

CHINIT 사용자 ID가 키 저장소를 소유하는 경우 이 사용자 ID에는 IRR.DIGTCERT.LISTRING 프로파일을 사용하고, 그렇지 않으면 액세스를 업데이트하고 IRR.DIGTCERT.LIST 프로파일. 필요에 따라 ACCESS (UPDATE) 또는 ACCESS (READ) 와 함께 PERMIT 명령을 사용하여 액세스 권한을 부여하십시오.

다음 필수조건을 설정했는지 확인하십시오.

- *ssidCHIN* 사용자 ID가 RACF에 올바르게 정의되어 있고 *ssidCHIN* 사용자 ID가 다음 프로파일에 대한 적절한 액세스 권한을 가지고 있습니다.
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  이러한 변수는 RACF FACILITY 클래스에 정의되어 있습니다.
- *ssidCHIN* 사용자 ID는 키 링의 소유자입니다.
- 큐 관리자의 개인 인증서는 RACDCERT 명령에 의해 작성된 경우 *ssidCHIN* 사용자 ID와도 같은 인증서 유형 사용자 ID로 작성됩니다.
- 키 링에 작성한 변경사항을 선택하기 위해 채널 시작기가 재이용되거나 **REFRESH SECURITY TYPE(SSL)** 명령이 실행됩니다.
- IBM MQ 채널 시작기 프로시저에는 링크 목록, LPA 또는 STEPLIB DD문을 통해 시스템 SSL 런타임 라이브러리 *pdsname*.SIEALNKE에 대한 액세스가 있습니다. 이 라이브러리는 APF 권한이 있어야 합니다.
- 채널 시작기가 실행 중인 권한이 있는 사용자 ID는 z/OS UNIX System Services 계획 문서에 설명된 대로 z/OS UNIX System Services (z/OS UNIX) 를 사용하도록 구성되어 있습니다.

  채널 시작기는 특수 권한을 필요로 하지 않고 z/OS UNIX 내에서 superuser로 실행하지 않으므로, 채널 시작기가 게스트/기본 UID 및 OMVS 세그먼트를 사용하여 UNIX를 호출하게 하지 않으려는 사용자는 기본 세그먼트에 기초하여 새 OMVS 세그먼트를 모델링하기만 하면 됩니다.

  채널 시작기에 올바른 액세스를 제공하는 방법에 대한 몇 가지 예는 291 페이지의 『Giving the channel initiator the correct access rights on z/OS』 의 PERMIT 명령을 참조하십시오.

## z/OS *Setting the SSLTASKS parameter on z/OS*

Use the ALTER QMGR command to set the number of server subtasks for processing TLS calls

To use TLS channels, ensure that there are at least two server subtasks by setting the SSLTASKS parameter, using the ALTER QMGR command. For example:

```
ALTER QMGR SSLTASKS(5)
```

To avoid problems with storage allocation, do not set the SSLTASKS attribute to a value greater than eight in an environment where there is no Certificate Revocation List (CRL) checking.

If CRL checking is used, an SSLTASK is held by the channel concerned for the duration of that check. This could be for a significant elapsed time while the relevant LDAP server is contacted, because each SSLTASK is a z/OS task control block.

You must restart the channel initiator if you change the value of the SSLTASKS attribute.

## z/OS *Setting up a key repository on z/OS*

Set up a key repository at both ends of the connection. Associate each key repository with its queue manager.

A TLS connection requires a *key repository* at each end of the connection. Each queue manager must have access to a key repository. Use the SSLKEYR parameter on the ALTER QMGR command to associate a key repository with a queue manager. See "SSL/TLS 키 저장소" on page 23 for more information.

On z/OS, digital certificates are stored in a *key ring* that is managed by your External Security Manager (ESM) . These digital certificates have labels, which associate the certificate with a queue manager. TLS uses these certificates for authentication purposes. All the examples that follow use RACF commands. Equivalent commands exist for other ESM programs.

On z/OS, IBM MQ uses either the value of the **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager appended. See Digital certificate labels for details.

The key repository name for a queue manager is the name of a key ring in your RACF database. You can specify the key ring name either before or after creating the key ring.

Use the following procedure to create a new key ring for a queue manager:

1. Ensure that you have the appropriate authority to issue the RACDCERT command (see Controlling the use of the RACDCERT command for more details).
2. Issue the following command:

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

   where:

   - *userid1* is the user ID of the channel initiator address space, or the user ID that is going to own the key ring (if the key ring is shared).
   - *ring-name* is the name you want to give to your key ring. The length of this name can be up to 237 characters. This name is case-sensitive. Specify *ring-name* in uppercase characters to avoid problems.

## z/OS *Making CA certificates available to a queue manager on z/OS*

After you have created your key ring, connect any relevant CA certificates to it.

If you have the CA certificate in a data set, you must first add the certificate to the RACF database by using the following command:

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

Then to connect a CA certificate for `My  CA` to your key ring, use the following command:

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

where *userid1* is either the channel initiator user ID or the owner of a shared key ring.

For more information about CA certificates, refer to "디지털 인증서" on page 12.

## z/OS *Locating the key repository for a queue manager on z/OS*

Use this procedure to obtain the location of your queue manager's key ring.

1. Display your queue manager's attributes, using either of the following MQSC commands:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

2. Examine the command output for the location of the key ring.

### z/OS *Specifying the key repository location for a queue manager on z/OS*

To specify the location of your queue manager's key ring, use the ALTER QMGR MQSC command to set your queue manager's key repository attribute.

For example:

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

if the key ring is owned by the channel initiator address space, or:

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

if it is a shared key ring, where *userid1* is the user ID that owns the key ring.

### z/OS *Giving the channel initiator the correct access rights on z/OS*

The channel initiator (CHINIT) needs access to the key repository and to certain security profiles.

#### Granting the CHINIT access to read the key repository

If the key repository is owned by the CHINIT user ID, this user ID needs read access to the IRR.DIGTCERT.LISTRING profile in the FACILITY class, and update access otherwise, and read access to the IRR.DIGTCERT.LIST profile. Grant access by using the PERMIT command with ACCESS(UPDATE) or ACCESS(READ) as appropriate:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID( userid ) ACCESS(READ)
```

where *userid* is the user ID of the channel initiator address space.

#### Granting the CHINIT read access to the appropriate CSF* profiles

For hardware support provided through the Integrated Cryptographic Service Facility (ICSF) to be used, ensure your CHINIT user ID has read access to the appropriate CSF* profiles in the CSFSERV class by using the following command:

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

where *csf-resource* is the name of the CSF* profile and *userid* is the user ID of the channel initiator address space.

Repeat this command for each of the following CSF* profiles:

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

Your CHINIT user ID might also need read access to other CSF* profiles. For example, if you are using the ECDHE_RSA_AES_256_GCM_SHA384 Cipher Spec, your CHINIT user ID also needs read access to the following CSF* profiles:

- CSF1DVK
- CSF1GAV
- CSF1GKP

- CSF1SKE
- CSF1TRC
- CSF1TRD

For more information, see RACF CSFSERV resource requirements.

If your certificate keys are stored in ICSF and your installation has established access control over keys stored in ICSF, ensure your CHINIT user ID has read access to the profile in the CSFKEYS class by using the following command:

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

where *userid* is the user ID of the channel initiator address space.

## Using the Integrated Cryptographic Service Facility (ICSF)

The channel initiator can use ICSF to generate a random number when seeding the password protection algorithm to obfuscate passwords flowing over client channels if TLS is not being used.

For further information, see "Using the Integrated Cryptographic Service Facility (ICSF)" on page 248

### z/OS When changes to certificates or the key repository become effective on z/OS

Changes become effective when the channel initiator starts or the repository is refreshed.

Specifically, changes to the certificates in the key ring and to the key repository attribute become effective on either of the following occasions:

- When the channel initiator is started or restarted.
- When the REFRESH SECURITY TYPE(SSL) command is issued to refresh the contents of the key repository.

### z/OS Creating a self-signed personal certificate on z/OS

Use this procedure to create a self-signed personal certificate.

1. Generate a certificate and a public and private key pair using the following command:

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.

  *userid1* and *userid2* can be the same ID.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 290.

- *label-name* must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager appended. See Digital certificate labels for details.

### z/OS *Requesting a personal certificate on z/OS*

Apply for a personal certificate using RACF.

To apply for a personal certificate, use RACF as follows:

1. Create a self-signed personal certificate, as in "Creating a self-signed personal certificate on z/OS" on page 292. This certificate provides the request with the attribute values for the Distinguished Name.
2. Create a PKCS #10 Base64-encoded certificate request written to a data set, using the following command:

   ```
   RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
   ```

   where

   - *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space
   - *label_name* is the label used when creating the self-signed certificate

   See "디지털 인증서 레이블, 요구사항 이해" on page 25 for details.
3. Send the data set to a Certificate Authority (CA) to request a new personal certificate.
4. When the signed certificate is returned to you by the Certificate Authority, add the certificate back into the RACF database, using the original label, as described in "Adding personal certificates to a key repository on z/OS" on page 294.

### z/OS *Creating a RACF signed personal certificate*

RACF can function as a certificate authority and issue its own CA certificate.

This section uses the term *signer certificate* to denote a CA certificate issued by RACF.

The private key for the signer certificate must be in the RACF database before you carry out the following procedure:

1. Use the following command to generate a personal certificate signed by RACF, using the signer certificate contained in your RACF database:

   ```
   RACDCERT ID(userid2) GENCERT
   SUBJECTSDN(CN('common-name')
              T('title')
              OU('organizational-unit')
              O('organization')
              L('locality')
              SP('state-or-province')
              C('country'))
   WITHLABEL('label-name')
   SIGNWITH(CERTAUTH LABEL('signer-label'))
   ```

2. Connect the certificate to your key ring using the following command:

   ```
   RACDCERT ID(userid1)
   CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
   ```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.

*userid1* and *userid2* can be the same ID.

- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 290.
- *label-name* must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details.
- *signer-label* is the label of your own signer certificate.

### z/OS *Adding personal certificates to a key repository on z/OS*

Use this procedure to add or import a personal certificate to a key ring.

After the certificate authority sends you a new personal certificate, add it to the key ring using the following procedure:

1. Add the certificate to the RACF database using the following command:

```
RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID( userid1 )
CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 290.
- *input-data-set-name* is the name of the data set containing the CA signed certificate. The data set must be cataloged and must not be a PDS or a member of a PDS. The record format (RECFM) expected by RACDCERT is VB. RACDCERT dynamically allocates and opens the data set, and reads the certificate from it as binary data.
- *label-name* is the label name that was used when you created the original request. It must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details.

### z/OS *Exporting a personal certificate from a key repository on z/OS*

Export the certificate using the RACDCERT command.

On the system from which you want to export the certificate, use the following command:

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the label of the certificate you want to extract.
- *output-data-set-name* is the data set into which the certificate is placed.
- CERTB64 is a DER encoded X.509 certificate that is in Base64 format. You can choose an alternative format, for example:

  **CERTDER**
  DER encoded X.509 certificate in binary format

  **PKCS12B64**
  PKCS #12 certificate in Base64 format

**PKCS12DER**
PKCS #12 certificate in binary format

### z/OS *Deleting a personal certificate from a key repository on z/OS*

Delete a personal certificate using the RACDCERT command.

Before deleting a personal certificate, you might want to save a copy of it. To copy your personal certificate to a data set before deleting it, follow the procedure in "Exporting a personal certificate from a key repository on z/OS" on page 294. Then use the following command to delete your personal certificate:

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the name of the certificate you want to delete.

### z/OS *Renaming a personal certificate in a key repository on z/OS*

Rename a certificate using the RACDCERT command.

If you do not want a certificate with a specific label to be found, but do not want to delete it, you can rename it temporarily using the following command:

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the name of the certificate you want to rename.
- *new-label-name* is the new name of the certificate.

This can be useful when testing TLS client authentication.

### z/OS *Associating a user ID with a digital certificate on z/OS*

IBM MQ can use a user ID associated with a RACF certificate as a channel user ID. Associate a user ID with a certificate by installing it under that user ID, or using a Certificate Name Filter.

The method described in this topic is an alternative to the platform-independent method for associating a user ID with a digital certificate, which uses channel authentication records. For more information about channel authentication records, see "채널 인증 레코드" on page 47.

When an entity at one end of a TLS channel receives a certificate from a remote connection, the entity asks RACF if there is a user ID associated with that certificate. The entity uses that user ID as the channel user ID. If there is no user ID associated with the certificate, the entity uses the user ID under which the channel initiator is running.

Associate a user ID with a certificate in either of the following ways:

- Install that certificate into the RACF database under the user ID with which you want to associate it, as described in "Adding personal certificates to a key repository on z/OS" on page 294.
- Use a Certificate Name Filter (CNF) to map the Distinguished Name of the subject or issuer of the certificate to the user ID, as described in "Setting up a certificate name filter on z/OS" on page 295.

#### z/OS *Setting up a certificate name filter on z/OS*

Use the RACDCERT command to define a certificate name filter (CNF), which maps a Distinguished Name to a user ID.

Perform the following steps to set up a CNF.

1. Enable CNF functions using the following command. You require update authority on the class DIGTNMAP to do this.

```
SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
```

2. Define the CNF. For example:

```
RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
```

   where USER1 is the user ID to be used when:

   - The DN of the subject has an Organization of `IBM` and a Country of UK.
   - The DN of the issuer has an Organization of `ExampleCA` and a Locality of `Internet`.

3. Refresh the CNF mappings:

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

**Note:**

1. If the actual certificate is stored in the RACF database, the user ID under which it is installed is used in preference to the user ID associated with any CNF. If the certificate is not stored in the RACF database, the user ID associated with the most specific matching CNF is used. Matches of the subject DN are considered more specific than matches of the issuer DN.
2. Changes to CNFs do not apply until you refresh the CNF mappings.
3. A DN matches the DN filter in a CNF only if the DN filter is identical to the *least significant portion* of the DN. The least significant portion of the DN comprises the attributes that are usually listed at the right-most end of the DN, but which appear at the beginning of the certificate.

   For example, consider the SDNFILTER `'O=IBM.C=UK'`. A subject DN of `'CN=QM1.O=IBM.C=UK'` matches that filter, but a subject DN of `'CN=QM1.O=IBM.L=Hursley.C=UK'` does not match that filter.

   The least significant portion of some certificates can contain fields that do not match the DN filter. Consider excluding these certificates by specifying a DN pattern in the SSLPEER pattern on the DEFINE CHANNEL command.
4. If the most specific matching CNF is defined to RACF as NOTRUST, the entity uses the user ID under which the channel initiator is running.
5. RACF uses the `'.'` character as a separator. IBM MQ uses either a comma or a semicolon.

You can define CNFs to ensure that the entity never sets the channel user ID to the default, which is the user ID under which the channel initiator is running. For each CA certificate in the key ring associated with the entity, define a CNF with an IDNFILTER that exactly matches the subject DN of that CA certificate. This ensures that all certificates that the entity might use match at least one of these CNFs. This is because all such certificates must either be connected to the key ring associated with the entity, or must be issued by a CA for which a certificate is connected to the key ring associated with the entity.

Refer to the *z/OS Security Server RACF Security Administrator's Guide* for more information about the commands you use to manipulate CNFs.

## z/OS *Defining a sender channel and transmission queue on QMA on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

### Procedure

On QMA, issue commands like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

### Results

A sender channel, TO.QMB, and a transmission queue, QMB, are created.

### z/OS *Defining a receiver channel on QMB on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

### Procedure

On QMB, issue a command like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

### Results

A receiver channel, TO.QMB, is created.

### z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

### Procedure

1. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
2. Optional: If any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
3. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

### Results

The sender channel is started.

### z/OS *Exchanging self-signed certificates on z/OS*

Exchange the certificates you previously extracted. If you use FTP, use the correct format.

### Procedure

Transfer the CA part of the QM1 certificate to the QM2 system and vice versa, for example, by FTP.

If you transfer the certificates using FTP, you must do so in the correct format.

Transfer the following certificate types in *binary* format:

- DER encoded binary X.509
- PKCS #7 (CA certificates)
- PKCS #12 (personal certificates)

Transfer the following certificate types in ASCII format:

- PEM (privacy-enhanced mail)
- Base64 encoded X.509

### z/OS *Defining a sender channel and transmission queue on QM1 on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

#### Procedure

On QM1, issue commands like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

The CipherSpecs at each end of the channel must be the same.

Only the SSLCIPH parameter is mandatory if you want your channel to use TLS. See "IBM MQ의 CipherSpec 및 CipherSuite" on page 38 for information about the permitted values for the SSLCIPH parameter.

#### Results

A sender channel, QM1.TO.QM2, and a transmission queue, QM2, are created.

### z/OS *Defining a receiver channel on QM2 on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

#### Procedure

On QM2, issue a command like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

The channel must have the same name as the sender channel you defined in "Defining a sender channel and transmission queue on QM1 on z/OS" on page 298, and use the same CipherSpec.

### z/OS *Starting the sender channel on QM1 on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

#### Procedure

1. Optional: If you have not already done so, start a listener program on QM2.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener
2. Optional: If any SSL/TLS channels have run previously, issue the command REFRESH SECURITY TYPE(SSL).

   This ensures that all the changes made to the key repository are available.
3. On QM1, start the channel, using the command `START CHANNEL(QM1.TO.QM2)`.

#### Results

The sender channel is started.

### z/OS *Refreshing the SSL or TLS environment on z/OS*

Refresh the TLS environment on queue manager QMA using the **REFRESH SECURITY** command.

#### Procedure

On QMA, enter the following command:

```
REFRESH SECURITY TYPE(SSL)
```

This ensures that all the changes made to the key repository are available.

### z/OS *Allowing anonymous connections on a receiver channel on z/OS*

Use the **ALTER CHANNEL** command to make SSL or TLS client authentication optional.

#### Procedure

On QMB, enter the following command:

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

### z/OS *Starting the sender channel on QM1 on z/OS*

If necessary, start the channel initiator, start a listener program, and refresh security. Then start the channel using the **START CHANNEL** command.

#### Procedure

1. Optional: if you have not already done so, start the channel initiator.
2. Optional: If you have not already done so, start a listener program on QM2.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener
3. Optional: If the channel initiator was already running or any SSL/TLS channels have run previously, issue the command REFRESH SECURITY TYPE(SSL).

   This ensures that all the changes made to the key repository are available.
4. On QM1, start the channel, using the command `START CHANNEL(QM1.TO.QM2)`.

#### Results

The sender channel is started.

### z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start the channel initiator, start a listener program, and refresh security. Then start the channel using the **START CHANNEL** command.

#### Procedure

1. Optional: If you have not already done so, start the channel initiator.
2. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
3. Optional: If the channel initiator was already running or if any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
4. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

### Results
The sender channel is started.

#### z/OS *Modifying elliptic curve key length on z/OS*
How you modify the GSK_CLIENT_ECURVE_LIST environment variable, to set the list of elliptic curves or supported groups that are specified by the client, as a string consisting of one or more 4-character values in order of preference for use.

**Important:** You must apply the fix in z/OS APAR OA61783 to permit certain elliptic curves to be made effective by the operating system, when using TLS 1.0, TLS 1.1 and/or TLS 1.2 negotiated connections.

You can set this TLS environment variable in the channel initiator startup JCL, using the CEEOPTS DD statement:

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

In the dataset referenced above, specify the list that you want to use, for example:

```
ENVAR("GSK_CLIENT_ECURVE_LIST=002300240025")
```

**Important:** Do not use this CEEOPTS statement with in-stream data, as this prevents the environment variable from being set for all TLS tasks using that statement.

Ensure you reference a sequential dataset, or partitioned dataset member, to allow this to work when using an SSLTASKS value greater than one.

You can also use the server analogue equivalent of GSK_CLIENT_ECURVE_LIST, which is GSK_SERVER_ALLOWED_KEX_ECURVES. See Limiting key exchange elliptic curves for more information.

In addition, see Table 5 in Cipher suite definitions for a list of valid 4-character elliptic curve and supported groups specifications.

The default specification is `00210023002400250019`. If TLS V1.3 is enabled, `0029` (x25519) is appended to the end of the default list.

## 사용자 식별 및 인증

X.509 인증서, MQCSP 구조 또는 여러 유형의 사용자 엑시트 프로그램을 사용하여 사용자를 식별하고 인증할 수 있습니다.

### X.509 인증서 사용

**SET CHLAUTH** 명령 및 **SSLPEER** 매개변수와 함께 X.509 인증서를 사용하여 사용자를 식별하고 인증할 수 있습니다. **SSLPEER** 매개변수는 채널의 다른 쪽 끝에 있는 피어 큐 관리자 또는 클라이언트의 인증서의 주제 식별 이름과 비교하는 데 사용할 필터를 지정합니다.

**SET CHLAUTH** 명령 및 **SSLPEER** 매개변수 사용에 대한 자세한 정보는 SET CHLAUTH를 참조하십시오.

디지털 인증서는 인증 기관에 의해 폐기될 수 있습니다. 플랫폼에 따라 LDAP 서버의 CRL 또는 OCSP를 사용하여 인증서의 폐기 상태를 검사할 수 있습니다. 자세한 정보는 318 페이지의 『폐기된 인증서에 대한 작업』의 내용을 참조하십시오.

### MQCSP 구조 사용

MQCSP 연결 보안 매개변수 구조는 MQCONNX 호출에서 지정됩니다. 이 구조에는 애플리케이션에서 제공하는 신임 정보가 포함될 수 있습니다. 애플리케이션은 MQCSP 구조에서 사용자 ID및 비밀번호를 제공할 수 있습니다. IBM MQ 9.3.4에서 애플리케이션은 인증 토큰을 제공할 수도 있습니다. 필요한 경우 보안 엑시트에서 MQCSP를 대체할 수 있습니다.

**경고:** MQCSP 구조의 신임 정보는 때때로 일반 텍스트로 네트워크를 통해 전송됩니다. 클라이언트 애플리케이션 신임 정보가 보호되는지 확인하려면 29 페이지의 『MQCSP 비밀번호 보호』의 내용을 참조하십시오.

자세한 정보는 302 페이지의 『MQCSP 구조를 사용하여 사용자 식별 및 인증』 및 305 페이지의 『인증 토큰에 대한 작업』의 내용을 참조하십시오.

Linux AIX AIX 및 Linux에서 MQCSP 구조에 지정된 사용자 ID및 비밀번호는 운영 체제 또는 PAM (Pluggable Authentication Method) 을 사용하여 인증할 수 있습니다. PAM은 서비스에서 세부사항을 숨기는 사용자 인증을 위한 일반 메커니즘을 제공합니다. 자세한 정보는 329 페이지의 『플러그 가능한 인증 방법 (PAM) 사용』의 내용을 참조하십시오.

### 엑시트에서 식별 및 인증 구현

여러 유형의 사용자 엑시트 프로그램을 사용하여 사용자를 식별하고 인증할 수 있습니다. 자세한 정보는 303 페이지의 『보안 엑시트에서 식별 및 인증 구현』, 304 페이지의 『메시지 엑시트에서 ID 맵핑』, 304 페이지의 『API 엑시트와 API 교차 엑시트에서 ID 맵핑』의 내용을 참조하십시오.

## 권한이 있는 사용자

권한 부여된 사용자는 IBM MQ에 대한 전체 관리 권한이 있는 사용자입니다.

다음 표에 나열된 사용자 이외에도, 큐 관리자의 무결성과 보안을 보장하기 위해 액세스 권한을 부여할 때 특별한 주의를 기울여야 하는 특정 오브젝트와 권한이 있습니다. 다음 권한 중 하나를 부여할 때 추가로 검토해야 합니다.

- SYSTEM 오브젝트에 대한 모든 권한
- 오브젝트를 작성하고 대체하고 삭제하는 관리 권한.

  z/OS z/OS의 경우, 이는 DEFINE, ALTER 및 DELETE 명령을 실행하는 명령 보안 및 명령 자원 보안 권한입니다.

  Multi 다른 모든 플랫폼에서는 이러한 권한이 `+crt`, `+chg` 및 `+dlt` 등의 관리 권한입니다.
- 큐를 지우는 관리 권한.

  z/OS z/OS의 경우, 이는 CLEAR 명령을 실행하는 명령 보안 및 명령 자원 보안 권한입니다.

  Multi 다른 모든 플랫폼에서, 이 권한은 `+clr`입니다.
- 채널을 중지하거나 메시지를 백아웃하거나 커미트하는 관리 권한.

  z/OS z/OS의 경우, 이는 RESET CHANNEL, START CHANNEL 및 STOP CHANNEL과 같은 명령을 실행하는 명령 보안 및 명령 자원 보안 권한입니다.

  Multi 다른 모든 플랫폼에서, 이러한 권한은 `+ctrl` 및 `+ctrlx`입니다.
- 애플리케이션이 권한 검사를 위해 권한을 에스컬레이션할 수 있게 하는 대체 사용자 MQI 권한.

  z/OS z/OS의 경우, 이는 대체 사용자 보안 프로파일에 부여된 임의 권한입니다.

  Multi 다른 모든 플랫폼에서, 이 권한은 `+altusr`입니다.
- 애플리케이션이 메시지의 보안 컨텍스트를 변경할 수 있게 하는 컨텍스트 권한.

  z/OS z/OS의 경우, 이는 컨텍스트 보안 프로파일에 부여된 임의 권한입니다.

  Multi 다른 모든 플랫폼에서, 이러한 권한은 `+setall` 및 `+setid`입니다.

메시징 애플리케이션에는 필요한 큐 또는 토픽에 대한 기본 MQI 권한만 부여하는 것이 일반적인 원칙입니다. 권한을 부여받지 못한 MCAUSER 하에서 실행되는 MCA 채널 및 특정한 다른 특수 유형의 애플리케이션(예: 데드-레터 큐)이 올바르게 작용하려면 애플리케이션에 정상적으로 부여되지 않은 추가 권한이 필요할 수 있습니다.

| 표 67. 플랫폼별 권한이 있는 사용자 | |
|---|---|
| **플랫폼** | **권한이 있는 사용자** |
| Windows 시스템 | • SYSTEM<br>• mqm 그룹의 구성원<br>• 관리자 그룹의 구성원 |
| AIX and Linux 시스템 | • mqm 그룹의 구성원 |
| IBM i 시스템 | • 프로파일 qmqm 및 qmqmadm<br>• qmqmadm 그룹의 모든 구성원<br>• *ALLOBJ 설정으로 정의된 모든 사용자 |
| z/OS | 채널 시작기, 큐 관리자 및 고급 메시지 보안 주소 공간이 실행 중인 사용자 ID입니다. 이러한 사용자 ID가 IBM MQ에 대한 전체 관리 권한을 자동으로 갖지는 않지만 이러한 사용자 ID에 일반적으로 부여되는 액세스 권한 레벨로 인해 권한이 있는 것으로 간주됩니다. |

# MQCSP 구조를 사용하여 사용자 식별 및 인증

MQCONNX 호출에서 MQCSP 연결 보안 매개변수 구조를 지정할 수 있습니다. MQCSP 구조는 메시지 큐 인터페이스 (MQI) 를 사용하여 인증에 사용되는 신임 정보를 제어하는 애플리케이션의 기본 방법입니다.

MQCSP 구조에는 권한 부여 서비스가 사용자를 식별하고 인증하는 데 사용할 수 있는 신임 정보가 포함되어 있습니다.

MQCSP 구조는 애플리케이션이 명시적으로 MQCSP 구조를 제공하지 않는 경우에도 클라이언트 또는 서버 측 보안 엑시트에 의해 수정될 수 있습니다. MQCSP 구조를 명시적으로 제공하지 않는 애플리케이션의 예는 IBM MQ classes for JMS를 사용하는 애플리케이션입니다. MQCSP 구조에 사용자 ID및 비밀번호를 삽입하는 클라이언트 측 보안 엑시트의 예는 75 페이지의 『사용자 ID 및 비밀번호(mqccred)를 삽입하기 위한 클라이언트측 보안 엑시트』의 내용을 참조하십시오.

V 9.4.0 MQCSP 구조에는 사용자 ID및 비밀번호 또는 인증 토큰이 포함되어 있습니다. 다음 제한사항은 MQCSP 구조에서 제공되는 신임 정보에 적용됩니다.

- 애플리케이션 또는 엑시트는 사용자 ID및 비밀번호 또는 인증 토큰을 제공해야 하지만 둘 다 제공해서는 안됩니다.
- 특정 형식 및 요구사항을 충족하는 인증 토큰만 IBM MQ에 액세스하는 데 사용할 수 있습니다. IBM MQ에서 인증 토큰의 요구사항에 대한 자세한 정보는 308 페이지의 『인증 토큰에 대한 요구사항』의 내용을 참조하십시오.
- 인증 토큰의 ID가 애플리케이션의 컨텍스트로 채택되는 경우 토큰은 적합한 사용자 청구를 제공해야 하며 청구 값은 유효한 IBM MQ 사용자 ID여야 합니다. 예를 들어, 사용자 이름은 최대 길이 및 특수 문자 제한사항을 준수해야 합니다. 사용자 ID 채택에 대한 자세한 정보는 302 페이지의 『MQCSP와 AdoptCTX 설정 간의 관계』의 내용을 참조하십시오.

MQCSP 구조에 대한 자세한 정보는 MQCSP-보안 매개변수를 참조하십시오.

**경고:** 클라이언트 애플리케이션에 대한 MQCSP 구조의 신임 정보는 때때로 일반 텍스트로 네트워크를 통해 전송됩니다. 클라이언트 애플리케이션 신임 정보가 보호되는지 확인하려면 29 페이지의 『MQCSP 비밀번호 보호』의 내용을 참조하십시오.

## MQCSP와 AdoptCTX 설정 간의 관계

IBM MQ 는 연결 인증 기능이 사용되는 경우 항상 MQCSP 구조에서 전달되는 신임 정보를 인증합니다. 신임 정보가 성공적으로 인증된 후 IBM MQ 는 연결된 애플리케이션에서 수행되는 조작에 대한 후속 권한 검사를 위해

사용자 ID를 채택할 수 있습니다. 큐 관리자의 **CONNAUTH** 속성에서 참조하는 인증 정보 (AUTHINFO) 오브젝트가 **ADOPTCTX(YES)**로 정의된 경우 MQCSP 신임 정보의 사용자 ID가 채택됩니다.

IBM MQ 에는 권한 검사에 사용할 수 있는 사용자 ID의 길이에 대한 한계가 있습니다. 이러한 한계에 대한 자세한 정보는 83 페이지의 『사용자 ID』의 내용을 참조하십시오. MQCSP 구조에 전달된 사용자 ID가 채택되면 IBM MQ 는 다른 구성 옵션에 따라 다르게 작동합니다.

- LDAP 연결 인증을 사용할 때 IBM MQ 는 사용자 LDAP 레코드의 짧은 사용자 이름 속성에 있는 사용자 ID를 채택합니다. 짧은 사용자 이름 속성은 AUTHINFO 오브젝트의 **SHORTUSR** 속성을 사용하여 설정됩니다.

  예를 들어, **SHORTUSR** 가 `'CN'`로 설정되고 LDAP 레코드가 사용자를 `'CN=Test,SN=MQ,O=IBM,C=UK'`로 나열하는 경우 사용자 ID `Test` 가 사용됩니다.

- OS 연결 인증 또는 PAM 인증을 사용할 때 ADOPTCTX가 YES인 경우, MQCSP 구조에 전달된 사용자 ID는 연결 컨텍스트로 채택될 때 IBM MQ 의 12자사용자 ID 한계를 충족시키기 위해 잘립니다.

  **ChlAuthEarlyAdopt** 가 사용으로 설정되면 사용자 신임 정보가 인증된 후에 잘립니다.

  **ChlAuthEarlyAdopt** 가 사용으로 설정되지 않은 경우 채택 전에 잘림이 발생합니다. Windows에서 사용자가 `user@domain`형식으로 제공되는 경우 이는 사용자가 12자미만인 경우 도메인 스펙이 올바르지 않을 수 있음을 의미합니다.

  예를 들어, 사용자 `` `ibmmq@windowsdomain` `` 가 MQCSP를 통해 제공되는 경우 이 시나리오에서는 `` `ibmmq@window` `` 로 잘립니다. 이는 다음과 같은 오류를 초래합니다.

  ```
  AMQ8074W: SID 'SID' 가 엔티티 'ibmmq@window' 와 일치하지 않으므로 권한 부여에 실패했습니다.
  ```

  이를 기반으로 12자보다 긴 사용자 ID (예: `user@domain`양식의 Windows 도메인 사용자 ID) 를 MQCSP를 통해 전달하는 경우 qm.ini 파일에서 **ChlAuthEarlyAdopt**=*Y* 를 구성하여 이 오류를 방지해야 합니다.

  또는 CONNAUTH AUTHINFO 구성에서 ADOPTCTX (NO) 를 사용하고 CHLAUTH USERMAP 규칙, 보안 엑시트 또는 채널 오브젝트 MCAUSER 설정과 같은 대체 접근 방식을 사용하여 채널의 사용자 ID를 설정하십시오.

## 보안 엑시트에서 식별 및 인증 구현

단방향 또는 상호 인증을 구현하기 위해 보안 엑시트를 사용할 수 있습니다.

보안 엑시트의 1차 목적은 해당 파트너를 인증하기 위해 채널의 각 끝에서 MCA를 사용 가능으로 하는 것입니다. 메시지 채널의 각 끝과 MQI 채널의 서버 측에서 MCA는 일반적으로 연결된 큐 관리자 대신에 작동합니다. MQI 채널의 클라이언트 끝에서 MCA는 일반적으로 IBM MQ MQI client 애플리케이션의 사용자 대신에 작동합니다. 이 경우 상호 인증은 실제로 두 큐 관리자 사이 또는 큐 관리자와 IBM MQ MQI client 애플리케이션의 사용자 사이에 발생합니다.

제공된 보안 엑시트(SSPI 채널 엑시트)는 생성된 인증 토큰을 교환한 다음 Kerberos 등과 같은 신뢰되는 인증 서버에 의해 검사되는 방법으로 상호 인증이 구현되는 방법을 보여줍니다. 자세한 정보는 144 페이지의 『Windows에서 SSPI 채널 엑시트 프로그램』의 내용을 참조하십시오.

상호 인증은 PKI(Public Key Infrastructure) 기술을 사용하여 구현할 수도 있습니다. 각 보안 엑시트가 어떤 무작위 데이터를 생성하고, 그것이 나타내고 있는 큐 관리자나 사용자의 개인 키를 사용하여 서명하고, 서명된 데이터를 보안 메시지로 그 파트너에게 송신합니다. 파트너 보안 엑시트가 큐 관리자나 사용자의 공개 키를 사용하여 디지털 서명을 검사하여 인증을 수행합니다. 디지털 서명을 교환하기 전에, 둘 이상의 알고리즘이 사용 가능한 경우, 보안 엑시트가 메시지 요약을 생성하기 위해 알고리즘에 동의해야 할 수 있습니다.

보안 엑시트가 서명된 데이터를 그 파트너에게 송신하면, 파트너도 그것이 나타내고 있는 큐 관리자나 사용자를 식별하는 어떤 수단을 송신해야 할 수 있습니다. 이것은 식별 이름이거나, 디지털 인증서일 수도 있습니다. 디지털 인증서가 송신되면, 파트너 보안 엑시트가 루트 CA 인증서로 가는 인증서 체인을 통해 작업하여 인증서의 유효성을 검증할 수 있습니다. 이렇게 하면 디지털 서명을 검사하는 데 사용되는 공개 키의 소유권의 보장을 제공합니다.

파트너 보안 엑시트가 인증서 체인의 나머지 인증서를 포함하는 키 저장소에 액세스할 수 있는 경우에만 디지털 인증서의 유효성을 검증할 수 있습니다. 큐 관리자나 사용자를 위한 디지털 인증서가 송신되지 않으면, 파트너 보안 엑시트가 액세스할 수 있는 키 저장소에서 사용 가능해야 합니다. 파트너 보안 엑시트가 서명자의 공개 키를 찾을 수 없으면 디지털 서명을 검사할 수 없습니다.

TLS(Transport Layer Security)는 방금 설명한 것과 같은 PKI 기술을 사용합니다. SSL(Secure Sockets Layer)의 인증 수행에 대한 자세한 정보는 17 페이지의 『TLS(Transport Layer Security) 개념』의 내용을 참조하십시오.

신뢰받는 인증 서버 또는 PKI 지원을 사용할 수 없으면 다른 기술을 사용할 수 있습니다. 보안 엑시트에 구현될 수 있는 공통적인 기술은 대칭 키 알고리즘을 사용합니다.

보안 엑시트 중 하나인 엑시트 A가 난수를 생성하여 파트너 보안 엑시트인 엑시트 B에 보안 메시지로 송신합니다. 엑시트 B가 두 개의 보안 엑시트에만 알려져 있는 키의 사본을 사용하여 숫자를 암호화합니다. 엑시트 B가 암호화된 숫자를 생성한 또 다른 난수와 함께 보안 메시지로 엑시트 A에게 송신합니다. 엑시트 A는 첫 번째 난수가 정확하게 암호화되었는지를 확인하고, 키의 사본을 사용하여 두 번째 난수를 암호화하며, 암호화된 숫자를 엑시트 B에게 보안 메시지로 송신합니다. 그런 후, 엑시트 B가 두 번째 난수가 제대로 암호화되었는지를 확인합니다. 이 교환 동안에, 보안 엑시트 중 어느 것이든지 다른 쪽의 인증에 만족하지 않으면, MCA에게 채널을 닫으라고 지시할 수 있습니다.

이 기술의 장점은 키나 암호가 교환 중에 통신 연결에서 송신되지 않는다는 것입니다. 단점은 공유 키를 보안된 방식으로 분배하는 방법에 대한 문제점의 해결책을 제공하지 못한다는 것입니다. 이 문제점에 대한 한가지 해결책이 436 페이지의 『사용자 엑시트 프로그램에서 기밀성 구현』에 설명되어 있습니다. 비슷한 기술이 SNA에서 두 개의 LU가 세션을 형성하기 위해 바인딩될 때 상호 인증을 위해 사용됩니다. 이 기술은 113 페이지의 『세션 레벨 인증』에 설명되어 있습니다.

상호 인증을 위한 모든 앞선 기술은 단방향 인증을 제공하기 위해 채택될 수 있습니다.

## 메시지 엑시트에서 ID 맵핑

인증을 애플리케이션 레벨에서 구현하는 것이 더 나을 수도 있지만 메시지 엑시트를 사용하여 사용자 ID를 인증하기 위해 정보를 처리할 수 있습니다.

애플리케이션이 메시지를 큐에 넣을 때, 메시지 설명자의 *UserIdentifier* 필드에 애플리케이션과 연관된 사용자 ID가 있습니다. 그러나, 사용자 ID를 인증하는 데 사용될 수 있는 데이터가 없습니다. 이 데이터는 채널의 송신 측에 있는 메시지 엑시트에 의해 추가되고 채널의 수신 측에 있는 메시지 엑시트에 의해 검사됩니다. 예를 들면, 데이터를 인증하는 것은 암호화된 암호이거나 디지털 서명일 수 있습니다.

이 서비스가 애플리케이션 레벨에서 구현되면 더 효율적일 수 있습니다. 기본적인 요구사항은 메시지를 수신하는 애플리케이션 사용자가 메시지를 송신한 애플리케이션의 사용자를 식별하고 인증할 수 있어야 한다는 것입니다. 그러므로, 이 서비스를 애플리케이션 레벨에서 구현하는 것을 고려하는 것이 당연합니다. 자세한 정보는 304 페이지의 『API 엑시트와 API 교차 엑시트에서 ID 맵핑』의 내용을 참조하십시오.

## API 엑시트와 API 교차 엑시트에서 ID 맵핑

메시지를 받는 애플리케이션은 메시지를 전송한 애플리케이션의 사용자를 식별하고 인증할 수 있어야 합니다. 이 서비스는 일반적으로 애플리케이션 레벨에서 가장 잘 구현됩니다. API 엑시트는 다양한 방법으로 서비스를 구현할 수 있습니다.

개별 메시지의 레벨에서, 식별과 인증은 메시지의 송신자와 수신자의 두 사용자와 관계가 있는 서비스입니다. 기본적인 요구사항은 메시지를 수신하는 애플리케이션 사용자가 메시지를 송신한 애플리케이션의 사용자를 식별하고 인증할 수 있어야 한다는 것입니다. 요구사항은 양방향이 아닌 단방향 인증이라는 것을 알아두십시오.

구현된 방법에 따라, 사용자와 애플리케이션은 서비스와 인터페이스되어 있거나, 대화해야 할 수도 있습니다. 또한, 서비스가 사용되는 때와 방법은 사용자와 애플리케이션이 있는 위치와 애플리케이션 자체의 성격에 따라 다를 수 있습니다. 그러므로, 서비스를 링크 레벨이 아니라 애플리케이션 레벨에서 구현하는 것을 고려하는 것이 당연합니다.

이 서비스를 링크 레벨에서 구현하는 것을 고려하는 경우, 다음과 같은 문제를 해결해야 할 수 있습니다.

- 메시지 채널에서 어떻게 서비스를 필요로 하는 메시지에만 서비스를 적용할 것인가?
- 필요한 경우, 사용자와 애플리케이션을 어떻게 서비스와 인터페이스하거나 상호작용할 수 있게 할 것인가?
- 멀티 호핑(multi-hop) 상황에서, 목적지로 가는 둘 이상의 메시지 채널에서 메시지가 어디로 송신될 것인가? 서비스의 구성요소를 어디에서 호출할 것인가?

다음은 식별 및 인증 서비스를 애플리케이션 레벨에서 구현하는 방법의 몇몇 예입니다. 용어 *API* 엑시트는 API 엑시트 또는 API 교차 엑시트를 의미합니다.

- 애플리케이션이 큐에 메시지를 넣으면 API 엑시트가 Kerberos와 같은 트러스트 인증 서버에서 인증 토큰을 확보할 수 있습니다. API 엑시트가 이 토큰을 메시지 안의 애플리케이션 데이터에 추가할 수 있습니다. 메시지가 수신하고 있는 애플리케이션에 의해 검색될 때, 또 다른 API 엑시트가 인증 서버에게 토큰을 검사하여 송신자를 인증하라고 요청할 수 있습니다.
- 애플리케이션이 메시지를 큐에 넣을 때, API 엑시트가 다음 항목을 메시지의 애플리케이션 데이터에 추가할 수 있습니다.
  - 송신자의 디지털 인증서
  - 송신자의 디지털 서명

  메시지 요약을 위한 다른 알고리즘이 사용 가능하면, API 엑시트가 사용했던 알고리즘의 이름을 포함할 수 있습니다.

  메시지가 수신하고 있는 애플리케이션에 의해 검색될 때, 또 다른 API 엑시트가 다음 검사를 수행할 수 있습니다.
  - API 엑시트가 루트 CA 인증서로 가는 인증서 체인을 통해 작업하여 디지털 인증서의 유효성을 검증할 수 있습니다. 이를 수행하려면 API 엑시트가 인증서 체인의 나머지 인증서를 포함하는 키 저장소에 액세스할 수 있어야 합니다. 이 검사는 식별 이름에 의해 식별되는 송신자가 인증서에 있는 공개키의 조작 및 변경되지 않은 순수한 소유자라는 것을 보장합니다.
  - API 엑시트가 인증서에 있는 공개키를 사용하여 디지털 서명을 검사할 수 있습니다. 이 검사는 송신자를 인증합니다.

  송신자의 식별 이름이 전체 디지털 인증서 대신에 송신될 수 있습니다. 이런 경우에, 키 저장소에 송신자의 인증서가 있어야 두 번째 API 엑시트가 송신자의 공개 키를 찾을 수 있습니다. 다른 가능성은 인증서 체인에 있는 모든 인증서를 송신하는 것입니다.
- 애플리케이션이 메시지를 큐에 넣을 때, 메시지 설명자의 *UserIdentifier* 필드에 애플리케이션과 연관된 사용자 ID가 있습니다. 사용자 ID는 송신자를 식별하는 데 사용될 수 있습니다. 인증을 가능하게 하려면, API 엑시트가 암호화된 암호와 같은 데이터를 메시지의 애플리케이션 데이터에 추가할 수 있습니다. 메시지가 수신하고 있는 애플리케이션에 의해 검색될 때, 또 다른 API 엑시트가 메시지와 함께 돌아다닌 데이터를 사용하여 사용자 ID를 인증할 수 있습니다.

  이 기술은 제어되고 트러스트되는 환경 및 트러스트되는 인증 서버나 PKI 지원이 사용 불가능한 환경에서 작성된 메시지에 대해서 충분하다고 간주될 수 있습니다.

## Linux AIX V 9.4.0 인증 토큰에 대한 작업

IBM MQ 9.4.0부터 클라이언트 애플리케이션은 AIX 또는 Linux에서 실행 중인 큐 관리자를 사용하여 인증하기 위한 토큰을 제공할 수 있습니다. 토큰의 사용자 ID를 IBM MQ 자원에 액세스하기 위한 권한 부여에 사용할 수도 있습니다.

JWT (JSON Web Tokens) 는 청구 기반 ID 모델을 채택합니다. ID및 액세스 제어는 청구 및 토큰 발행자의 아이디어로 요약됩니다.

- 청구는 사용자에 대한 정보를 포함하고 사용자가 수행할 수 있는 작업이 아니라 사용자가 누구인지 설정하는 이름 값 쌍입니다.
- 토큰 발행자는 신뢰할 수 있는 써드파티이거나 사용자의 ID만을 기반으로 사용자에 대한 토큰을 발행하는 서버입니다. 토큰 발행자는 사용자가 수행할 수 있는 작업과 관련이 없습니다.

토큰은 청구를 포함하는 단순 구조이며 인터넷을 통해 당사자 간에 쉽게 전송될 수 있습니다. 인증에 토큰을 사용하면 중앙 집중식 ID 관리의 이점이 있습니다. 하나의 신뢰할 수 있는 토큰 발행자를 사용할 수 있으므로 애플리케이션은 각 서비스에 별도로 등록하지 않고도 많은 서비스를 사용하여 인증할 수 있습니다. 신임 정보가 각 서비스에 전송되지 않고 신뢰할 수 있는 발행자에게만 전송되므로 토큰은 증가된 보안을 제공합니다.

JWT는 제안된 인터넷 표준 RFC7519를 통해 정의됩니다.

### IBM MQ 에 대한 토큰 작업 방법

IBM MQ 와 함께 사용되는 토큰은 IBM MQ 가 지원하는 알고리즘으로 서명된 올바른 JWT이어야 합니다. JWT는 JWS (JSON Web Signature) 표준에 따라 서명되어야 합니다. JWE (JSON Web Encryption) 및 JWK (JSON

Web Key) JOSE 기술을 사용하는 토큰은 IBM MQ와 함께 사용할 수 없습니다. 자세한 정보는 308 페이지의 『인증 토큰에 대한 요구사항』의 내용을 참조하십시오.

인증 토큰을 제공하는 애플리케이션은 IBM MQ clients를 지원하는 모든 플랫폼에서 실행할 수 있습니다. 애플리케이션은 C 또는 Java, 로 작성되어야 하며 클라이언트 바인딩을 사용하여 큐 관리자에 연결해야 합니다. 그러나 큐 관리자는 AIX 또는 Linux에서 실행되어야 합니다.

큐 관리자는 키 저장소에서 신뢰할 수 있는 발행자 공개 키 또는 대칭 키에 대해 토큰 서명의 유효성을 검증합니다. 큐 관리자를 설정하려면 311 페이지의 『JWKS 엔드포인트를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』 또는 로컬 키 저장소를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성의 단계를 따르십시오.

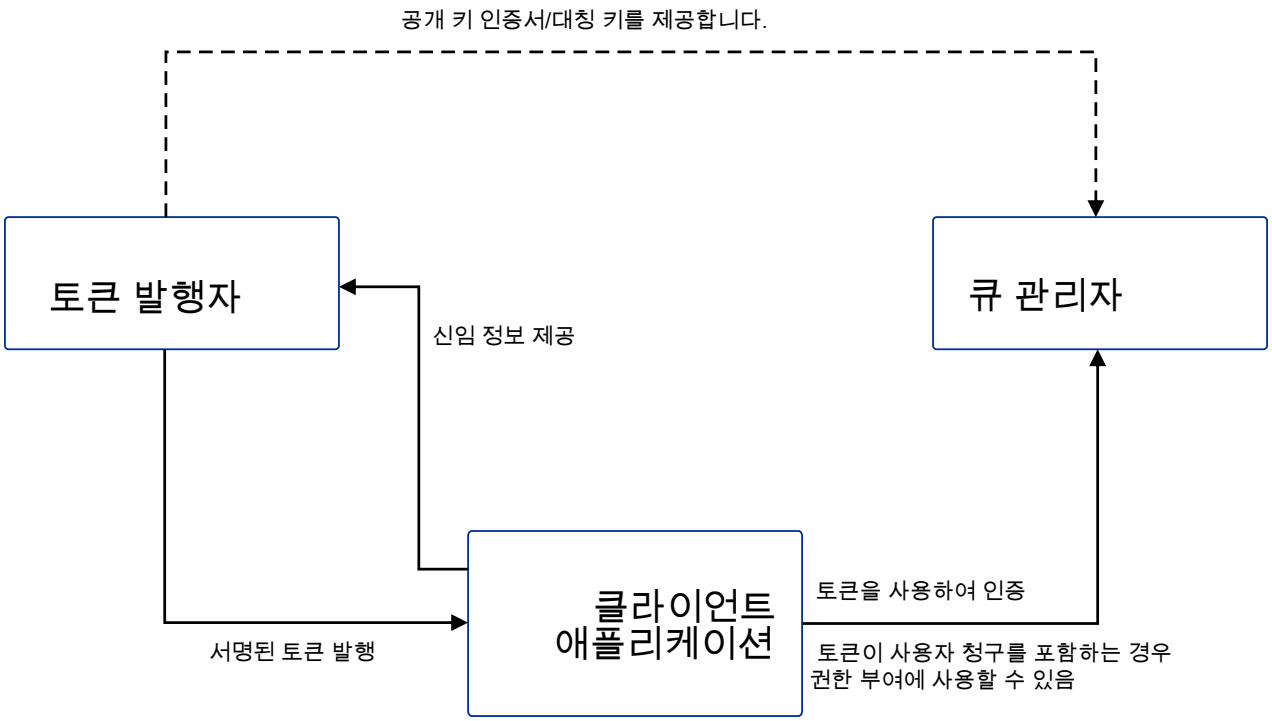
토큰 발행자는 위임된 보안 액세스 권한이 있는 신뢰할 수 있는 당사자입니다. 즉, 애플리케이션 사용자의 ID를 확인합니다. 큐 관리자는 인증 토큰이 올바르고 인증된 사용자에게 IBM MQ 오브젝트에 액세스할 수 있는 권한이 부여되었는지 확인합니다. 큐 관리자는 토큰을 사용하여 처음 연결하기 전에 사용자를 알 수 있지만 알 필요는 없습니다. IBM MQ 관리자는 큐 관리자에 연결하는 애플리케이션에 대한 인증 및 권한 부여를 설정하고 토큰에 포함되어야 하는 항목에 대한 요구사항을 설정해야 합니다.

클라이언트 애플리케이션은 IBM MQ에 연결할 때 인증에 사용하는 발행자로부터 동적으로 토큰을 요청할 수 있습니다. 그런 다음 애플리케이션은 MQCSP 구조또는 선택된 API의 동등한 구조 를 사용하여 연결 시 큐 관리자에 토큰을 전달합니다.

애플리케이션이 인증 토큰을 요청하도록 변경할 수 없고 연결할 때 큐 관리자에게 토큰을 제공할 수 없는 경우, 대신 보안 엑시트를 사용하여 MQCSP 구조에서 토큰을 제공할 수 있습니다.

토큰이 인증 토큰에 대한 요구사항을 충족하고 토큰 서명이 유효한 경우 연결이 설정됩니다. 큐 관리자는 선택적 사용자 청구가 토큰에 포함된 경우 IBM MQ 자원에 액세스하기 위해 권한 검사를 위해 토큰에 포함된 사용자 ID를 사용할 수도 있습니다. 사용자 청구는 큐 관리자가 권한 검사를 위해 채택하는 사용자 ID를 포함하는 토큰 내의 청구입니다. 사용자 청구의 이 이름은 `qm.ini` 파일의 **AuthToken** 스탠자에 있는 **UserClaim** 속성으로 지정됩니다.

자세한 정보는 315 페이지의 『애플리케이션에서 인증 토큰 사용』 및 MQCSP-보안 매개변수를 참조하십시오.




다이어그램은 IBM MQ에서 토큰을 사용하기 위한 예상 플로우의 기본 예제를 표시합니다. 예상 라이프사이클은 다음과 같습니다.

- 이 토큰은 신뢰할 수 있는 발행자가 애플리케이션에 발행합니다. 자세한 정보는 인증 토큰에 대한 요구사항을 참조하십시오.
- 애플리케이션은 연결할 때 토큰을 큐 관리자로 전달합니다. 자세한 정보는 애플리케이션에서 인증 토큰 사용을 참조하십시오.
- 큐 관리자는 키 저장소에서 신뢰할 수 있는 발행자 공개 키 또는 대칭 키에 대해 토큰 서명의 유효성을 검증합니다. 큐 관리자를 설정하려면 311 페이지의 『JWKS 엔드포인트를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』의 단계를 따르십시오.
- 인증 토큰에 올바른 사용자 청구가 포함된 경우 토큰의 사용자를 IBM MQ 자원에 액세스하기 위한 권한 검사에 채택할 수 있습니다. 자세한 정보는 권한 부여를 위해 사용자 채택을 참조하십시오.
- IBM MQ 관리자는 신뢰할 수 있는 토큰 발행자 인증서를 관리합니다. 인증서가 만기되면 토큰 발행자로부터 새 인증서를 확보하여 키 저장소에 추가해야 합니다.
- 큐 관리자를 구성했고 애플리케이션이 연결 중이지만 토큰과 관련된 문제가 발생하는 경우 인증 토큰 문제점 해결 및 토큰 인증 오류 코드를 참조하십시오.

IBM MQ 는 JWT및 JWS 표준을 준수하는 토큰을 제공하는 토큰 발행자와 함께 작동합니다.

아직 토큰을 사용하고 있지 않지만 토큰 서버를 준비하는 데 관련된 내용을 이해하려는 경우 무료 및 오픈 소스 Keycloak 프로젝트에 대한 시작하기 안내서 를 참조하십시오.

**관련 참조**

qm.ini 파일의 AuthToken 스탠자

Linux AIX V 9.4.0

## 인증 토큰에 대한 요구사항

IBM MQ와 함께 사용되는 인증 토큰에 대한 유효성 검증 요구사항, 구조 및 알고리즘입니다.

### 요구사항

IBM MQ 와 함께 사용되는 인증 토큰은 다음 요구사항을 충족해야 합니다.

- 토큰 길이는 최대 길이인 8192자를 초과하지 않아야 합니다. 자세한 정보는 MQCSP의TokenLength (MQLONG)를 참조하십시오.
- 토큰 구조 및 인코딩은 RFC7519의 JWT (JSON Web Token) 스펙 및 RFC7515의 JWS (JSON Web Signature) 스펙에서 정의한 대로 유효합니다.
- 309 페이지의 표 68 에 지정된 필수 토큰 헤더 매개변수가 있으며 매개변수의 값이 유효합니다.
- 310 페이지의 표 69 에 지정된 필수 페이로드 청구가 존재하고 청구 값이 유효합니다.
- 토큰은 IBM MQ 에서 지원하는 310 페이지의 표 70 의 알고리즘으로 서명됩니다.
- 만기 (**exp**) 청구의 값이 현재 시간 이후입니다.
- not before (**nbf**) 청구가 있는 경우 값은 현재 시간 이전입니다.
- 사용자 청구가 있는 경우 값은 310 페이지의 『인증 토큰의 사용자 ID』에 대한 요구사항을 충족해야 합니다.

### 토큰 구조

IBM MQ 는 RFC7519 표준을 준수하는 JWT를 승인합니다. JWT는 RFC7515에 정의된 JWS 표준에 따라 서명되고 인코딩되어야 합니다.

IBM MQ 에서는 JWS 보안 토큰이 다음 세 개의 컴포넌트를 포함할 것으로 예상합니다.

**JOSE 헤더**

해당 컨텐츠를 보안하는 데 사용되는 암호화 알고리즘 및 토큰의 유형을 설명하는 매개변수를 포함하는 JSON 오브젝트입니다.

다음 헤더 예제에서는 인코딩된 오브젝트가 JWT이고 헤더 및 페이로드가 HMAC SHA-256 알고리즘을 사용하여 보호됨을 선언합니다.

```
{
```

```
 "typ":"JWT",
 "alg":"HS256"
}
```

**JWS 페이로드**

JWT 표준에 지정된 대로 청구를 포함하는 JSON 오브젝트입니다. JSON 오브젝트의 각 멤버는 청구입니다. 청구는 토큰 발행자의 ID 또는 베어러의 사용자 ID를 어설션할 수 있습니다.

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

**JWS 서명**

신뢰할 수 있는 발행자가 토큰을 발행했는지 유효성 검증하는 데 사용됩니다.

이러한 컴포넌트는 JWS 보안 토큰에서 마침표 ('.') 로 구분된 base64url-encoded 문자열로 표시됩니다.

JWS 표준을 준수하는 인증 토큰이 서명되어 토큰의 인증을 유효성 검증할 수 있지만 암호화되지 않습니다. 따라서 토큰에 대한 액세스 권한이 있는 모든 사용자가 이를 읽고 재사용할 수 있습니다. 네트워크를 통해 전송될 때 (예를 들어, TLS를 사용하여) 암호화를 사용하여 인증이 보호되도록 큐 관리자에 대한 연결을 구성하십시오. 애플리케이션에서 제공하는 신임 정보를 보호하는 옵션에 대한 자세한 정보는 MQCSP 비밀번호 보호를 참조하십시오.

IBM MQ 는 인증 토큰의 헤더 및 페이로드에서 다음 매개변수 및 청구를 지원합니다. 토큰의 추가 매개변수 또는 청구는 무시됩니다. 토큰에 동일한 이름을 가진 둘 이상의 매개변수 또는 청구가 포함된 경우, 중복 이름을 가진 마지막 매개변수 또는 청구가 사용됩니다.

표 *68.* 토큰 헤더 매개변수 설명

| 토큰 파트 | 매개변수 이름 | 데이터 유형 | 필수 | 설명 |
|---|---|---|---|---|
| 헤더 | **typ** | 문자열 | 예 | 토큰 유형입니다. 이 매개변수의 값은 "JWT" 여야 합니다. |
| | **alg** | 문자열 | 예 | 헤더 및 페이로드를 보안하는 데 사용되는 알고리즘입니다. 이 매개변수의 값은 310 페이지의 표 70의 알고리즘 중 하나여야 합니다. |

| 표 69. 토큰 페이로드 청구 설명 | | | | |
|---|---|---|---|---|
| **토큰 파트** | **매개변수 이름** | **데이터 유형** | **필수** | **설명** |
| 페이로드 | **exp** | 정수 | 예 | 1979년 1월 1일 00:00협정 세계시 (UTC) 이후의 초 수로 표시되는 토큰 만기 시간입니다. 이 시간 이후에는 토큰이 허용되지 않습니다. |
| | **nbf** | 정수 | 아니오 | 1979년 1월 1일 00:00협정 세계시 (UTC) 이후 토큰이 허용되지 않은 시간 (초) 으로 표시되는 시간입니다. |
| | 사용자 청구 이름이 `qm.ini` 파일에서 **AuthToken** 스탠자의 **UserClaim** 필드에 지정되었습니다. | 문자열 | 토큰의 사용자 청구가 권한 부여에 사용되는 경우에만 필요합니다. | 권한 검사를 위해 채택된 사용자 ID를 포함하는 클레임의 이름입니다.<br>예를 들어, 토큰에 사용자 청구 `"AppUser": "MyUserName"`가 있는 경우 `qm.ini` 파일의 **AuthToken** 스탠자에 **UserClaim**=`AppUser` 를 지정해야 합니다. |

인코딩 및 디코딩된 토큰의 좋은 예는 `jwt.io` 웹 사이트의 디버거 페이지를 참조하십시오.

## 알고리즘

IBM MQ 는 JWS 보안 토큰에 대한 JWA (JSON Web Algorithms) 스펙 에 포함된 알고리즘의 서브세트를 지원합니다.

| 표 *70. JWS* 보안 토큰에 대해 *IBM MQ* 에서 지원하는 *JWA (JSON Web* 알고리즘*)* | |
|---|---|
| **alg 매개변수값** | **디지털 서명 또는 MAC 알고리즘** |
| HS256 | SHA-256을 사용하는 HMAC |
| HS384 | SHA-384를 사용하는 HMAC |
| HS512 | SHA-512를 사용하는 HMAC |
| RS256 | SHA-256 을 사용하는 RSASSA-PKCS1-v1_5 |
| RS384 | SHA-384 를 사용하는 RSASSA-PKCS1-v1_5 |
| RS512 | SHA-512 를 사용하는 RSASSA-PKCS1-v1_5 |

## 비대칭 키 인증서 요구사항

토큰이 비대칭 키로 서명된 경우 토큰 발행자의 공개 키 인증서는 큐 관리자가 토큰 인증에 사용하는 키 저장소에 있어야 합니다. 인증 토큰이 수신되면 인증서가 유효 기간 내에 있어야 합니다. 토큰 발행자의 인증서가 취소되지 않았는지 확인하기 위한 검사가 수행되지 않습니다.

## 인증 토큰의 사용자 ID

큐 관리자가 인증 토큰의 사용자 청구에 포함된 사용자 ID를 애플리케이션의 컨텍스트로 채택하도록 구성된 경우 채택된 사용자 ID는 다음 요구사항을 충족해야 합니다.

- 최대 12자를 포함할 수 있습니다.
- 다음 문자 중 하나로 시작해야 합니다.

A-Z a-z

- 다음 문자를 포함할 수 있습니다.

  0-9 A-Z a-z+,-. : = _

- 예약된 사용자 ID UNKNOWN및 NOBODY중 하나가 아니어야 합니다.

**관련 태스크**

**AuthTokens** 를 승인하도록 큐 관리자 구성

**관련 참조**

qm.ini 파일의 AuthToken 스탠자

## Linux AIX V 9.4.0 JWKS 엔드포인트를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성

JWKS 엔드포인트를 사용하여 인증 토큰으로 사용자 및 애플리케이션을 인증하도록 AIX 또는 Linux 에서 실행 중인 IBM MQ 큐 관리자를 구성하십시오.

### 시작하기 전에

토큰이 IBM MQ에서 작동하는 방법에 대한 자세한 정보는 인증 토큰에 대한 작업을 참조하십시오.

큐 관리자를 구성하기 전에 큐 관리자 **CONNAUTH** 속성에서 참조되는 AUTHINFO 오브젝트가 IDPWOS유형인지 확인하십시오. 토큰 인증은 큐 관리자가 OS 사용자 ID및 비밀번호 검사를 위해 구성된 경우에만 사용 가능합니다.

서비스 스탠자의 **SecurityPolicy** 속성이 그룹으로 설정되지 않았는지 확인하십시오. **SecurityPolicy** 가 명시적으로 그룹으로 설정된 경우 토큰 인증을 사용할 수 없습니다. 만약에**SecurityPolicy** 로 설정되어 있습니다  그룹 , 제거**SecurityPolicy** 서비스 스탠자의 속성을 선택한 다음 큐 관리자를 다시 시작하십시오.

### 이 태스크 정보

애플리케이션은 토큰을 사용하여 큐 관리자로 인증할 수 있습니다. IBM MQ 는 제안된 인터넷 표준 RFC7519를 따르는 신뢰할 수 있는 발행자로부터 JSON웹 토큰 (*JWT*) 을 승인합니다. 토큰을 사용하여 ID를 인증한 후 나중에 권한 검사를 위해 채택할 수 있습니다.

토큰을 승인하도록 큐 관리자를 구성하는 가장 간단한 방법은 아래에 설명된 대로 JWKS 엔드포인트를 가리키는 것입니다. 인증 서비스가 이를 제공하지 않고 엔드포인트 또는 JWKS가 다른 이유로 적합하지 않은 경우 312 페이지의 『로컬 키 저장소를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』 의 내용을 참조하십시오.

### 프로시저

1. 인증 서버 관리자에게 다음 세부사항을 문의하십시오.
   - 올바른 JWKS 엔드포인트 (URL) 입니다.
   - 이 서버가 HTTP 트래픽을 암호화하는 데 사용하는 인증서 및/또는 이 인증서에 서명하는 권한입니다.

     **중요사항:** 항상 TLS/HTTPS를 통해 JWKS 정보를 제공해야 하며 큐 관리자가 연결을 신뢰할 수 있도록 하려면 이 정보가 필요합니다.
2. qm.ini 파일에 **HTTPSKeyStore** 를 제공하여 발신 https 연결을 작성하도록 큐 관리자를 구성하십시오.

   자세한 정보는 다음을 참조하십시오.
   - qm.ini 파일의 HTTPSKeyStore 설명.
   - 318 페이지의 『TLS 신뢰 저장소로 사용할 키 저장소 작성』.

   인증 서버가 맞춤형 인증서/CA를 사용하는 경우 이 인증서가 이 HTTPSKeyStore에 올바르게 존재하는지 확인해야 합니다.
3. qm.ini 구성 파일에서 JWKS 스탠자 를 정의하여 JWKS 엔드포인트를 구성하십시오.

   추가 스탠자는 다음을 제공합니다.

- **issuername** 이는 이 권한으로 서명된 토큰에 있는 'iss' 청구와 일치해야 하며, 종종 인증 서비스의 URL을 기반으로 합니다.
- **endpoint** 이는 큐 관리자가 토큰 서명의 유효성을 검증하는 데 사용되는 공개 키를 조회하는 주소입니다.
- **userclaim**이는 토큰이 유효성 검증된 후 IBM MQ 권한 검사에 사용되어야 하는 토큰에서 사용자 정의 필드를 식별하기 위한 선택사항입니다.

**주의:** 이러한 연결에 **ADOPTCTX**(YES) 를 사용하려는 경우에는 반드시 존재해야 합니다.

4. `.ini` 파일 변경이 완료되면 `REFRESH SECURITY TYPE(AUTHINFO)` 명령을 실행하거나 큐 관리자를 재시작하십시오.

   구성에 성공하면 애플리케이션이 서명된 토큰을 사용하여 즉시 연결할 수 있습니다.

   예를 들어, 공용 키를 검색하기 위해 인증 서비스에 접속할 수 없는 문제점이 있는 경우 큐 관리자의 AMQERR01 로그 파일에 문제점이 보고됩니다.

### 결과

JWKS 엔드포인트를 사용하여 인증 토큰을 승인하도록 큐 관리자를 구성했습니다.

**참고:** 키는 인증 서버에서 정기적으로 (15분마다) 새로 고쳐지며, 연결 애플리케이션에서 알 수 없는 키 ID가 표시되는 경우에는 더 자주 새로 고쳐집니다. 일반적으로 이는 인증서가 만료되어 서버 측에서 대체될 때 인증서를 업데이트하기 위해 추가 IBM MQ 구성 조치가 필요하지 않음을 의미합니다. 즉시 새로 고치기를 강제 실행하려면 언제든지 `REFRESH SECURITY TYPE(AUTHINFO)` 명령을 실행하십시오.

**관련 개념**

인증 토큰 문제점 해결

**관련 태스크**

애플리케이션에서 인증 토큰 사용

**관련 참조**

qm.ini 파일의 AuthToken 스탠자

## Linux AIX V 9.4.0 로컬 키 저장소를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성

인증 토큰을 사용하여 사용자 및 애플리케이션을 인증하도록 IBM MQ 큐 관리자를 구성하십시오.

### 시작하기 전에

가능한 경우 토큰 유효성 검증 인증서를 수동으로 구성하는 대신 JWKS 엔드포인트 사용을 고려하십시오. 311 페이지의 『JWKS 엔드포인트를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』의 내용을 참조하십시오. JWKS를 사용하면 일반적으로 초기 구성 및 진행 중인 유지보수가 둘 다 더 단순해집니다.

인증 토큰에 대한 작업에서 토큰이 IBM MQ 에 대해 작업하는 방법에 대해 읽으십시오.

큐 관리자를 구성하기 전에 큐 관리자 **CONNAUTH** 속성에서 참조되는 AUTHINFO 오브젝트가 `IDPWOS`유형인지 확인하십시오. 토큰 인증은 큐 관리자가 OS 사용자 ID및 비밀번호 검사를 위해 구성된 경우에만 사용 가능합니다.

서비스 스탠자의 **SecurityPolicy** 속성이 그룹으로 설정되지 않았는지 확인하십시오. **SecurityPolicy** 가 명시적으로 그룹으로 설정된 경우 토큰 인증을 사용할 수 없습니다. **SecurityPolicy** 가 `Group`으로 설정된 경우, 서비스 스탠자에서 **SecurityPolicy** 속성을 제거한 후 큐 관리자를 재시작하십시오.

### 이 태스크 정보

IBM MQ 9.3.4 에서 애플리케이션은 토큰을 사용하여 큐 관리자로 인증할 수 있습니다. IBM MQ 는 제안된 인터넷 표준 RFC7519를 따르는 신뢰할 수 있는 발행자로부터 JSON웹 토큰 (*JWT*) 을 승인합니다. 토큰을 사용하여 ID를 인증한 후 나중에 권한 검사를 위해 채택할 수 있습니다.

신뢰할 수 있는 발행자의 공개 키 인증서 또는 대칭 키를 큐 관리자의 키 저장소에 저장하여 토큰을 승인하도록 큐 관리자를 구성하십시오. AuthToken 스탠자를 `qm.ini` 파일에 추가하고 큐 관리자가 새 구성을 선택할 수 있도록 보안 구성을 새로 고치십시오.

테스트 환경에서 JWKS를 사용하지 않고 로컬 키 저장소를 구성하거나 큐 관리자에서 인증 서버에 직접 연결할 수 없는 경우 로컬 키 저장소를 구성할 수 있습니다. JWKS 엔드포인트 외에 로컬 키 저장소를 정의할 수도 있습니다.

**참고:** JWKS 엔드포인트 및 로컬 키 저장소 둘 다 표시된 토큰에 대해 일치하는 발행자 및 KID를 제공하는 경우 JWKS 엔드포인트 제공 키가 환경 설정에서 사용됩니다.

이러한 상황에서는 다음과 같이 로컬 키 저장소를 구성하십시오.

## 프로시저

1. 키 저장소를 작성하십시오.
   a) 신뢰할 수 있는 발행자로부터 수신되는 공개 키 인증서 또는 대칭 키에 대한 키 저장소를 작성하십시오. 파일 확장자가 `.kdb` 인 CMS 키 저장소 또는 파일 확장자가 `.p12`인 PKCS#12 키 저장소를 사용할 수 있습니다.

   다음 명령을 실행하여 CMS 키 저장소를 작성하십시오.

   ```
   runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
   -type cms
   ```

   만약**runmqakm** 명령이 오류를 반환합니다. 참조 runmqakm -keydb . 명령이 성공적으로 완료되면 `ls` 명령을 사용하여 디렉토리의 컨텐츠를 나열하십시오.

   ```
   ls -l /var/mqm/qmgrs/qm1/tokenissuer
   ```

   다음 파일이 표시됩니다.

   ```
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.crl
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.kdb
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.rdb
   ```

   b) 필요한 경우 mqm 그룹에 읽기 액세스 권한을 부여할 수 있도록 작성한 키 저장소 파일의 그룹 소유권을 변경하십시오. 처음에는 명령을 실행한 admin 사용자만 작성된 파일에 액세스할 수 있습니다.

   ```
   chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
   ```

   c) 키 저장소 파일의 모드를 변경하여 그룹 mqm에 대한 읽기 권한을 추가하십시오. 예를 들어, 다음 명령은 파일 소유자에 대한 읽기/쓰기 권한과 그룹에 대한 읽기 전용 권한을 추가합니다.

   ```
   chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
   ```

2. **runqmcred** 명령으로 키 저장소 비밀번호를 암호화하고 암호화된 문자열을 파일에 저장하십시오.
   a) 키 저장소 비밀번호를 암호화하는 데 사용되는 초기 키를 포함할 파일을 작성하십시오.

   파일은 텍스트의 단일 행으로 초기 키를 포함해야 합니다. 초기 키의 최대 길이는 256바이트입니다. **INITKEY** 큐 관리자 속성을 사용하여 큐 관리자의 초기 키를 이미 설정한 경우 **INITKEY** 속성의 값을 새 파일에 복사하십시오. 큐 관리자에 대한 초기 키를 아직 설정하지 않은 경우 새 고유 암호화 키를 작성하고 이를 초기 키 파일에 추가하십시오.

   **참고:** 자세한 정보는 INITKEY를 참조하십시오. 초기 키를 지정하지 않으면 기본 키가 사용됩니다. 사용자 고유의 초기 키를 사용하는 것이 더 안전합니다.

   **참고:** 파일의 컨텐츠를 안전하게 유지하기 위해 초기 키 파일에 필요한 최소 권한을 부여하십시오. 초기 키 파일은 키 저장소 비밀번호를 암호화하는 데만 사용됩니다. 따라서 초기 키를 사용하여 비밀번호를 암호화하는 관리자만 초기 키 파일을 읽을 수 있는 액세스 권한이 필요합니다.

   b) 큐 관리자 초기 키가 아직 설정되지 않은 경우, 큐 관리자 **INITKEY** 속성의 값을 313 페이지의 『2.a』 단계에서 작성한 초기 키로 설정하십시오. **ALTER QMGR** 명령을 사용하여 큐 관리자 초기 키를 설정하십시오. 예를 들면, 다음과 같습니다.

```
ALTER QMGR INITKEY('myEncrypt10nK3y')
```

c) **runqmcred** 명령을 실행하여 키 저장소 비밀번호를 암호화하십시오. **-sf** 매개변수를 사용하여 초기 키를 포함하는 파일에 대한 경로를 지정하십시오.

```
runqmcred -sf initial.key
```

프롬프트가 표시되면 키 저장소 비밀번호를 입력하십시오. 암호화된 비밀번호는 명령에 의해 출력됩니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
**************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

마지막 행의 문자열을 복사하여 파일에 저장하십시오.

3. 다음 방법 중 하나를 사용하여 토큰 발행자의 공개 키 인증서 또는 대칭 키를 키 저장소에 추가하십시오.
   - 키 저장소에 RSA 공개 키 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile
```

   - base64 인코딩 대칭 키를 키 저장소에 추가하려면 다음 명령을 실행하십시오.

```
runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile -format ascii
```

   여기서 *keylabel* 은 인증서 또는 비밀 키에 첨부될 레이블이고 *keyfile* 은 인증서 또는 base64 인코딩 비밀 키를 포함하는 파일의 이름입니다.

4. **AuthToken** 스탠자 및 다음 속성을 `qm.ini` 파일에 추가하십시오.
   - **KeyStore** 속성을 사용하여 지정된 키 저장소의 경로입니다.
   - **KeyStorePwdFile** 속성을 사용하여 지정된 키 저장소의 비밀번호를 포함하는 파일입니다.
   - **CertLabel** 속성을 사용하여 지정된 314 페이지의 『3』 단계에서 추가한 인증서 또는 대칭 키의 레이블입니다.

   예를 들면, 다음과 같습니다.

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
   CertLabel=rsakey
```

   여기서 `key.kdb` 은 313 페이지의 『1.a』 단계에서 작성한 키 저장소의 이름이고 `key.pw` 은 314 페이지의 『2.c』 단계에서 작성한 키 저장소의 암호화된 비밀번호를 포함하는 파일입니다.

   **AuthToken** 스탠자에 대한 자세한 정보는 `qm.ini` 파일의AuthToken 스탠자를 참조하십시오.

5. 큐 관리자가 후속 권한 검사에서 사용하기 위해 토큰 사용자 청구에 포함된 사용자 ID를 채택하도록 구성된 경우 **UserClaim** 속성을 **AuthToken** 스탠자에 추가하십시오.

   큐 관리자가 토큰에서 사용자 ID를 채택하도록 구성되었는지 여부를 판별하려면 다음 MQSC 명령을 실행하십시오.

```
DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
```

   여기서 *authinfo_name* 은 큐 관리자 **CONNAUTH** 속성의 값입니다. **ADOPTCTX** 속성의 값이 YES인 경우, 큐 관리자는 토큰의 사용자 ID를 채택하도록 구성되며 **UserClaim** 속성은 **AuthToken** 스탠자에 지정되어야 합니다.

**UserClaim** 속성의 값을 채택할 사용자 ID를 포함하는 토큰 청구의 이름으로 설정하십시오. 예를 들어, 토큰에 "AppUser": "MyUserName"청구가 포함된 경우 **AuthToken** 스탠자에 다음 행을 추가하십시오.

```
UserClaim=AppUser
```

6. qm.ini 파일에서 토큰 구성을 선택하도록 큐 관리자의 보안 구성을 새로 고치십시오. 다음 명령을 실행하여 **runmqsc** 명령을 시작하십시오.

```
runmqsc qm1
```

그런 다음 다음 MQSC 명령을 실행하십시오.

```
REFRESH SECURITY TYPE(CONNAUTH)
```

### 다음에 수행할 작업

개발자가 큐 관리자를 인증하기 위해 애플리케이션에서 토큰을 사용 하는 방법을 이해하는 데 도움이 되도록 개발자와 함께 작업하십시오.

**관련 개념**

인증 토큰 문제점 해결

**관련 태스크**

애플리케이션에서 인증 토큰 사용

**관련 참조**

qm.ini 파일의 AuthToken 스탠자

## Linux AIX V 9.4.0 선택한 토큰 발행자로부터 인증 토큰 얻기

IBM MQ 큐 관리자에 연결할 때 선택한 토큰 발행자로부터 인증 토큰을 얻도록 애플리케이션을 작성하십시오.

### 시작하기 전에

315 페이지의 『애플리케이션에서 인증 토큰 사용』 의 정보를 참조하십시오.

### 프로시저

- 인증 토큰을 얻는 방법과 토큰의 정확한 컨텐츠는 서로 다른 토큰 발행자 간에 다양합니다.

  인증 토큰을 요청하고 얻기 위해 선택한 토큰 발행자와 상호작용하도록 애플리케이션을 작성하십시오. 인증 토큰은 인증 토큰에 대한 IBM MQ 요구사항을 준수해야 합니다. 이러한 요구사항에 대한 자세한 정보는 308 페이지의 『인증 토큰에 대한 요구사항』 의 내용을 참조하십시오.

  토큰 청구에 포함된 사용자 ID를 애플리케이션의 컨텍스트로 채택하려는 경우 인증 토큰도 다음 요구사항을 충족해야 합니다.

  - 인증 토큰은 큐 관리자의 토큰 인증 구성에 있는 사용자 청구 이름과 일치하는 청구를 포함해야 합니다.
  - 사용자 청구의 값은 인증 토큰의 사용자 ID에 대한 요구사항을 충족해야 합니다. 자세한 정보는 310 페이지의 『인증 토큰의 사용자 ID』 의 내용을 참조하십시오.

### 결과

이제 유효성 검증을 위해 IBM MQ 에 제공할 수 있는 올바르게 형식화된 JWT 를 확보했습니다.

**관련 태스크**

**AuthTokens** 를 승인하도록 큐 관리자 구성

**관련 참조**

qm.ini 파일의 AuthToken 스탠자

MQCSP - 보안 매개변수

## Linux AIX V 9.4.0 애플리케이션에서 인증 토큰 사용

IBM MQ 큐 관리자에 연결할 때 인증 토큰을 제공하도록 애플리케이션을 작성하십시오.

## 시작하기 전에

IBM MQ 9.4.0부터 애플리케이션은 큐 관리자에 연결할 때 인증 토큰을 제공할 수 있습니다.

애플리케이션은 다음 요구사항을 충족해야 합니다.

- C 또는 Java 로 작성되어야 합니다 ( IBM MQ classes for JMS/ Jakarta Messaging사용).
- IBM MQ client로 큐 관리자에 연결해야 합니다. 즉, 애플리케이션은 로컬 바인딩을 사용하는 대신 네트워크를 통해 큐 관리자에 연결해야 합니다.
- AIX 또는 Linux에서 실행되는 큐 관리자에 연결해야 합니다.

애플리케이션이 이러한 요구사항을 충족하지 않으면 연결이 실패하고 이유 코드 MQRC_FUNCTION_NOT_SUPPORTED (2298) 가 애플리케이션에 리턴됩니다.

인증 토큰을 제공하는 애플리케이션은 IBM MQ MQI clients를 지원하는 모든 플랫폼에서 실행할 수 있습니다.

자동 클라이언트 재연결을 사용하는 클라이언트는 연결할 때 인증 토큰을 제공할 수 없습니다. 애플리케이션이 인증 토큰을 제공하고 MQCNO 구조에서 MQCNO_RECONNECT 또는 MQCNO_RECONNECT_Q_MGR 옵션을 지정하는 경우, 연결이 실패하고 이유 코드 MQRC_RECONNECT_INCOMPATIBLE (2547) 이 애플리케이션에 리턴됩니다. 자동 클라이언트 다시 연결에 대한 자세한 정보는 자동 클라이언트 다시 연결을 참조하십시오.

이러한 요구사항으로 인해 인증 토큰을 제공하도록 애플리케이션을 작성할 수 없는 경우 클라이언트 보안 엑시트를 사용하여 인증 토큰을 사용하도록 애플리케이션을 마이그레이션할 수도 있습니다. MQCSP 구조에서 인증 토큰을 설정하기 위해 클라이언트 보안 엑시트를 작성할 수 있습니다. 보안 엑시트에 대한 자세한 정보는 클라이언트 연결의 보안 엑시트를 참조하십시오.

IBM MQ 9.4.0부터 JMS 클라이언트 애플리케이션은 연결 시 토큰을 직접 제공할 수 있습니다 ( 315 페이지의 『선택한 토큰 발행자로부터 인증 토큰 얻기』 참조). IBM MQ 9.4.0이전에는 Java 애플리케이션이 엑시트 프로그램을 통해 간접적으로 토큰을 제공할 수 있습니다. 자세한 정보는 Java 클래스 MQCSP를 참조하십시오.

## 이 태스크 정보

**참고:** JWS (JSON Web Signature) 표준을 준수하는 인증 토큰은 토큰의 인증을 유효성 검증할 수 있도록 서명되지만 암호화되지는 않습니다. 따라서 토큰에 대한 액세스 권한이 있는 모든 사용자가 이를 읽고 재사용할 수 있습니다. 인증 토큰이 네트워크를 통해 전송될 때 (예: TLS를 사용하여) 암호화를 사용하여 보호되도록 큐 관리자에 대한 연결을 구성하십시오. 애플리케이션에서 제공하는 신임 정보를 보호하는 옵션에 대한 자세한 정보는 29 페이지의 『MQCSP 비밀번호 보호』의 내용을 참조하십시오.

토큰을 사용하여 연결할 애플리케이션을 수정하기 전에 다음을 확인하십시오.

- 312 페이지의 『로컬 키 저장소를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』 의 단계에 따라 큐 관리자가 인증 토큰을 승인하도록 구성되었습니다.
- 애플리케이션은 인증 서버에서 필요한 대로 유효한 토큰을 얻을 수 있습니다. 315 페이지의 『선택한 토큰 발행자로부터 인증 토큰 얻기』의 내용을 참조하십시오.

애플리케이션이 IBM MQ 큐 관리자에 연결할 때 인증 토큰을 제공하려면 다음 프로세스를 포함하십시오.

## 프로시저

- C (MQI) 애플리케이션에서 인증 토큰을 제공하려면 다음을 수행하십시오.
  애플리케이션은 MQCONN이 아닌 MQCONNX를 사용하여 연결하고 MQCSP 구조를 제공해야 합니다.
  - **AuthenticationType** 필드는 MQCSP_AUTH_ID_TOKEN으로 설정되어야 합니다.
  - 구조의 버전은 MQCSP_VERSION_3으로 설정되어야 합니다.
  - **TokenPtr** 또는 **TokenOffset** 필드는 인증 토큰을 참조해야 합니다.
  - **TokenLength** 필드는 인증 토큰의 길이로 설정되어야 합니다.

  MQCSP 버전 3및 인증 토큰을 사용하여 큐 관리자에 연결하기 위한 C 코드 예:

```
MQCNO cno = {MQCNO_DEFAULT};   /* Connection options  */
MQCSP csp = {MQCSP_DEFAULT};   /* Security parameters */
```

```
char  token[MQ_CSP_TOKEN_LENGTH +1] = {0};  /* Authentication token string */

/* Set the connection options */
cno.SecurityParmsPtr = &csp;
cno.Version = MQCNO_VERSION_5;

/* Set the security parameters */
csp.Version = MQCSP_VERSION_3;
csp.AuthenticationType = MQCSP_AUTH_ID_TOKEN;
csp.TokenPtr = token;
csp.TokenLength = (MQLONG) strlen(token);

/* Connect to the queue manager */
MQCONNX(qmName,                 /* Queue manager name  */
        &cno,                   /* Connection options  */
        &hCon,                  /* Connection handle   */
        &compCode,              /* Completion code     */
        &reason);               /* Reason code         */
```

- Java 애플리케이션에서 인증 토큰을 제공하려면 다음을 수행하십시오.

  IBM MQ classes for JMS/Jakarta Messaging 를 사용하는 애플리케이션은 사용자 이름 및 비밀번호를 사용하는 `createContext`또는 `createConnection` 메소드를 통해 토큰을 제공할 수 있습니다.

  인증 토큰을 제공하려면 다음을 수행하십시오.

  - **UserID** 는 널 또는 빈 문자열로 설정되어야 합니다. 즉, 공백이 없어야 합니다. ""
  - 토큰은 **Password** 문자열로 제공됩니다.

  이는 `ConnectionFactory` 인터페이스의 모든 IBM MQ 구현에 적용됩니다.

  명시적 매개변수 양식 (예: `createContext`(String **userID**, String **password**)) 또는 암시적 매개변수 버전 (예: `createContext`()) 을 사용할 수 있습니다.

  후자의 경우, 비어 있는 **userID** 및 토큰 **Password** 이 먼저 연결 팩토리의 특성으로 제공되어야 합니다.

  인증 토큰을 사용하여 큐 관리자에 연결하는 예제 Java 코드:

```
// Obtain token from authentication provider here:

String myToken = "xxxxxxxxxxxxxxx";

// Acquire instance of an MQ connection Factory:

JmsFactoryFactory ff = JmsFactoryFactory.getInstance(WMQConstants.WMQ_PROVIDER);

JmsConnectionFactory cf = ff.createConnectionFactory();

// Configure any required CF properties here - e.g. MQ Channel details

// Connect to (and authenticate with) the queue manager:

context = cf.createContext(null, myToken); // NOTE - null userID indicates token being
provided
```

  연결이 이유 코드 MQRC_NOT_AUTHORIZED (2035) 또는 MQRC_SECURITY_ERROR (2063) 로 실패하는 경우 큐 관리자 오류 로그에서 실패 원인에 대한 자세한 정보를 포함하는 오류 메시지를 확인하십시오. 인증 토큰 문제점 진단에 대한 자세한 도움말은 인증 토큰 문제점 해결을 참조하십시오.

## 결과

이제 애플리케이션이 큐 관리자에 연결되었습니다. 인증에 사용된 토큰이 만료된 경우에도 연결이 끊어질 때까지 연결된 상태로 유지됩니다. 애플리케이션이 큐 관리자에서 연결을 끊고 다시 연결해야 하는 경우 다시 연결하기 전에 나중에 만기 시간이 있는 새 인증 토큰을 확보해야 할 수 있습니다.

**관련 태스크**

**AuthTokens** 를 승인하도록 큐 관리자 구성

**관련 참조**

qm.ini 파일의 AuthToken 스탠자

MQCSP - 보안 매개변수

## Linux AIX V 9.4.0 TLS 신뢰 저장소로 사용할 키 저장소 작성

발신 TLS 연결을 작성할 때 인증 기관 (CA) 의 공통 세트에서 서명한 인증서의 유효성을 검증할 수 있는 단순 '신뢰 저장소' 를 작성해야 합니다. 예제 TLS 연결은 IBM MQ의 일부 컴포넌트를 구성할 때 사용되는 IBM MQ 클라이언트 채널 또는 HTTPS 연결입니다.

### 이 태스크 정보

**주의:** 사용자 환경에서 신뢰할 인증서 및 인증 기관을 결정하는 것은 엔드-투-엔드 구성의 보안과 관련된 중요한 단계입니다. 이 주제는 IBM MQ 구성요소가 운영 체제에 대해 이미 구성된 동일한 인증서 세트를 신뢰할 수 있도록 하는 공통 단계를 설명하기 위해 제공됩니다. 그러나 확실하지 않은 경우 보안 관리자와 이 프로세스를 논의해야 합니다.

대부분의 UNIX 및 Linux 기반 운영 체제에는 '신뢰할 수 있는' CA 세트가 포함된 파일 시스템 위치가 있습니다. 이 파일 시스템은 운영 체제 설치로 구성되었거나 시스템 관리자가 사용자 정의했을 수 있습니다 (예: 조직에 속하는 내부 CA 포함). 이러한 파일의 위치는 다양하지만 일반적으로 많이 사용되는 운영 체제에 사용되는 일부 값은 다음과 같습니다.

- AIX: `/var/ssl/cert.pem and/or /var/ssl/certs/*.crt`
- RHEL: `/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem`
- Ubuntu: `/etc/ssl/certs/*.pem`

IBM MQ 키 저장소를 작성하고 구성할 때 디렉토리 (예: `/etc/ssl/certs`) 에 있는 모든 인증서 파일을 하나의 명령으로 IBM MQ 키 데이터베이스에 쉽게 추가할 수 있습니다.

### 프로시저

1. 다음 명령을 사용하여 `/etc/ssl/certs` 디렉토리에서 인증서 파일을 추가하십시오.

```
runmqakm -cert -add -file /etc/ssl/certs/*.pem -db mykdb.p12 -stashed
```

2. 옵션: 일부 상황에서는 신뢰 저장소에 대한 '기본' 인증서 세트를 생성하는 것이 유용할 수 있습니다.
   제품과 함께 제공되는 IBM MQ 보안 컴포넌트는 '기본' CA 인증서 세트를 제공합니다.

   **참고:** 이러한 인증서는 자주 업데이트되지 않거나 상대적으로 수명이 짧을 수 있습니다.

   사전 구성된 CA 인증서를 사용하려는 경우에도 **runmqakm** 명령에서 **populate** 및 **ibmcloudtrust** 매개변수를 사용하여 신뢰 저장소를 생성할 수 있습니다.

```
runmqakm -keydb -create -db mqauto.p12 -genpw -stash -type pkcs12 -populate -ibmcloudtrust
```

**관련 개념**
인증 토큰 문제점 해결
**관련 태스크**
애플리케이션에서 인증 토큰 사용
**관련 참조**
qm.ini 파일의 AuthToken 스탠자

## 폐기된 인증서에 대한 작업

디지털 인증서는 인증 기관에 의해 폐기될 수 있습니다. 플랫폼에 따라 LDAP 서버의 CRL 또는 OCSP를 사용하여 인증서의 폐기 상태를 검사할 수 있습니다.

TLS 데이터 교환 중 통신하고 있는 파트너는 디지털 인증서를 사용하여 서로를 인증합니다. 인증은 수신된 인증서가 아직도 신뢰될 수 있는지를 검사하는 것을 포함할 수 있습니다. 인증 기관(CA)은 다음을 포함한 여러 가지 이유로 인증서를 폐기합니다.

- 소유자가 다른 조직으로 이동
- 개인 키가 더 이상 기밀이 아님

CA가 인증서 폐기 목록(CRL)에 폐기된 개인 인증서를 공개합니다. 폐기된 CA 인증서가 권한 취소 목록(ARL)에 공개됩니다.

ALW AIX, Linux, and Windows 플랫폼에서 IBM MQ 지원은 OCSP(온라인 인증서 상태 프로토콜)를 사용하거나 LDAP(Lightweight Directory Access Protocol) 서버에서 CRL과 ARL을 사용하여 해지된 인증서를 확인합니다. OCSP가 기본 메소드입니다.

IBM MQ classes for Java 및 IBM MQ classes for JMS에서는 클라이언트 채널 정의 테이블 파일에서 OCSP 정보를 사용할 수 없습니다. 그러나 온라인 인증서 프로토콜 사용에서 설명한 대로 OCSP를 구성할 수 있습니다.

IBM i IBM i에서 IBM MQ SSL 지원은 LDAP 서버에서만 CRL및 ARL을 사용하여 폐기된 인증서를 검사합니다.

z/OS z/OS에서 IBM MQ SSL 지원은 LDAP 서버에서만 CRL및 ARL을 사용하여 폐기된 인증서를 검사합니다.

인증 기관에 대한 자세한 정보는 12 페이지의 『디지털 인증서』의 내용을 참조하십시오.

## OCSP/CRL 검사

OCSP(Online Certificate Status Protocol)/CRL(Certificate Revocation List) 검사는 원격 수신 인증서에 대해 수행됩니다. 이 프로세스는 관련된 체인 전체를 원격 시스템의 개인 인증서부터 해당 루트 인증서까지 전부 검사합니다.

### openSSL을 사용하여 OCSP 유효성 검증 확인

엔터프라이즈에서 openSSL 을 사용하여 OCSP를 유효성 검증한 후 IBM Global Security Kit (GSKit) TLS 연결을 사용하려고 시도하면 UNKNOWN 상태 경고를 수신합니다.

이는 루트를 제외하고 체인에 있는 모든 인증서가 취소 상태에 대해 GSKit 에 의해 검사되기 때문입니다. GSKit 조작은 RFC 5280을 따르며 이에 대해서는 GSKit 신뢰 정책에 설명되어 있습니다. GSKit 알고리즘은 RFC 5280 및 GSKit 신뢰 정책에 설명된 대로 취소 정보에 대해 사용 가능한 모든 소스를 시도합니다.

### OCSP/CRL 검사가 IBM MQ에서 작동하는 방식

IBM MQ는 인증서 확장에 지정되어 있거나 AUTHINFO 오브젝트에 정의된, 이름 지정된 OCSP 또는 CRL 엔드포인트에 대해 인증서를 검사할 때의 제어 작동에 대해 두 가지 메커니즘을 지원합니다.

- qm.ini 파일의 SSL 스탠자의 **OCSPCheckExtensions**, **CDPCheckExtensions**및 **OCSPAuthentication** 속성
- 큐 관리자의 SSLCRLNL 매개변수와 AUTHINFO OCSP 및 CRLLDAP 구성을 사용하여. 자세한 정보는 ALTER AUTHINFO 및 ALTER QMGR을 참조하십시오.

**주의:**

**AUTHTYPE(OCSP)**을 포함하는 ALTER AUTHINFO 명령은 IBM i 또는 z/OS 큐 관리자에 대한 사용에는 적용되지 않습니다. 그러나 클라이언트가 사용하도록 클라이언트 채널 테이블(CCDT)에 복사할 플랫폼에는 지정할 수 있습니다.

**OCSPCheckExtensions** 및 **CDPCheckExtensions** SSL 스탠자 속성은 IBM MQ가 인증서의 AIA 확장에 지정되어 있는 OCSP 또는 CRL 서버에 대해 인증서를 확인하는지 제어합니다.

사용으로 설정되지 않은 경우에는 인증서 확장에 있는 OCSP 또는 CRL 서버에 접속하지 않습니다.

AUTHINFO 오브젝트를 통해 OCSP 또는 CRL 서버가 설정되었으며 SSLCRLNL **QMGR** 속성을 사용하여 참조된 경우에는 인증서 폐기 처리 중에 IBM MQ가 이러한 서버에 접속하려 시도합니다.

**중요사항:** SSLCRLNL 이름 목록에는 하나의 OCSP AUTHINFO 오브젝트만 정의할 수 있습니다.

다음과 같은 경우

**OCSPCheckExtensions**=NO 및 **CDPCheckExtensions**=*NO*가 설정되었으며,
AUTHINFO 오브젝트에 OCSP 또는 CRL 서버가 정의되지 않은 경우

인증서 폐기 검사가 수행되지 않습니다.

확인이 사용으로 설정된 경우 IBM MQ는 인증서의 폐기 상태를 확인할 때 이름 지정된 OCSP 또는 CRL 서버에 다음 순서대로 접속합니다.

1. **AUTHTYPE(OCSP)** 오브젝트에서 자세히 설명하고 SSLCRLNL **QMGR** 속성에서 참조하는 OCSP 서버입니다.
2. 인증서의 AIA 확장에 설정된 OCSP 서버(**OCSPCheckExtensions**=*YES*인 경우)
3. 인증서의 **CRLDistributionPoints** 확장에 설정된 CRL 서버(**CDPCheckExtensions**=*YES*인 경우)
4. **AUTHINFO(CRLLDAP)** 오브젝트에 설정되었으며 SLCRLNL **QMGR** 속성에서 참조하는 CRL 서버

인증서를 확인하는 중에, 특정 단계에서 OCSP 또는 CRL 서버가 인증서에 대해 결정적인 REVOKED 또는 VALID 응답을 리턴하는 경우에는 추가적인 검사가 수행되지 않으며 표시된 대로의 인증서 상태가 인증서의 신뢰 여부를 결정하는 데 사용됩니다.

OCSP 또는 CRL 서버가 UNKNOWN 결과를 리턴하는 경우에는 OCSP 또는 CRL 서버가 결정적인 결과를 리턴하거나 모든 옵션이 소진될 때까지 처리가 계속됩니다.

인증서를 폐기된 것으로 간주하거나, 상태를 판별할 수 없는 것으로 간주하는 작동은 OCSP 서버와 CRL 서버 간에 다릅니다.

- CRL 서버의 경우에는 CRL을 확보할 수 없는 경우 해당 인증서를 NOT_REVOKED로 간주합니다.
- OCSP 서버의 경우, 이름 지정된 OCSP 서버로부터 폐기 상태를 확보할 수 없으면 작동이 qm.ini 파일의 SSL 스탠자에 있는 **OCSPAuthentication** 속성을 통해 제어됩니다.

  이 속성은 연결을 차단하거나, 연결을 허용하거나, 경고 메시지를 표시하며 연결을 허용하도록 구성할 수 있습니다.

필요한 경우에는 qm.ini 및 mqclient.ini 파일의 SSL 스탠자에 있는 **SSLHTTPProxyName**=*string* 속성을 OCSP 검사에 사용할 수 있습니다. 문자열은 OCSP 검사를 위해 GSKit 에서 사용할 HTTP 프록시 서버의 호스트 이름 또는 네트워크 주소입니다.

폐기 검사를 수행할 때 OCSP 응답자를 대기하는 시간 (초) 을 설정하는 qm.ini 또는 mqclient.ini 파일의 SSL 스탠자에서 **OCSPTimeout** 값을 설정할 수 있습니다.

## ALW 폐기된 인증서 및 OCSP

IBM MQ는 사용할 OCSP(Online Certificate Status Protocol) 응답자를 판별하고, 수신한 응답을 핸들링합니다. OCSP 응답자에 액세스할 수 있도록 하는 단계를 수행해야 할 수도 있습니다.

**참고:** 이 정보는 AIX, Linux, and Windows 시스템의 IBM MQ 에만 적용됩니다.

OCSP를 사용하여 디지털 인증서의 폐기 상태를 검사하기 위해 IBM MQ는 두 개의 방법을 사용하여 어떤 OCSP 응답자에게 연락할지를 판별할 수 있습니다.

- 검사할 인증서에서 AIA(AuthorityInfoAccess) 인증서 확장자를 사용
- 인증 정보 오브젝트에 지정되거나 클라이언트 애플리케이션에 의해 지정된 URL 사용

인증 정보 오브젝트 또는 클라이언트 애플리케이션에서 지정한 URL은 AIA 인증서 확장자에 있는 URL보다 우선합니다.

OCSP 응답자의 URL이 방화벽 뒤에 놓인 경우 OCSP 응답자에 액세스할 수 있도록 방화벽을 재구성하거나 OCSP 프록시 서버를 설정하십시오. SSL 스탠자에 있는 SSLHTTPProxyName 변수를 사용하여 프록시 서버의 이름을 지정하십시오. 클라이언트 시스템에서는 환경 변수 MQSSLPROXY를 사용하여 프록시 서버의 이름을 지정할 수도 있습니다. 자세한 내용은 관련 정보를 참조하십시오.

아마도 테스트 환경에서 실행 중이므로 TLS 인증서의 폐기 여부가 중요하지 않으면, SSL 스탠자에서 OCSPCheckExtensions를 아니오로 설정할 수 있습니다. 이 변수를 설정하면 모든 AIA 인증서 확장자가 무시됩니다. 이 솔루션은 프로덕션 환경에서는 사용할 수 없습니다. 여기에서는 폐기된 인증서를 제시하는 사용자에게 액세스를 허용하지 않습니다.

OCSP 응답자 액세스 호출은 다음 세 가지 결과 중 하나를 초래할 수 있습니다.

**좋음**
인증서가 올바릅니다.

**폐기됨**
인증서가 폐기되었습니다.

**알 수 없음**
이 결과는 세 가지 중 하나의 이유로 발생할 수 있습니다.

- IBM MQ가 OCSP 응답자에 액세스할 수 없습니다.
- OCSP 응답자가 응답을 송신했으나 IBM MQ가 응답의 디지털 서명을 확인할 수 없습니다.
- OCSP 응답자가 인증서에 대한 폐기 데이터를 가지고 있지 않음을 표시하는 응답을 송신했습니다.

IBM MQ에서 알 수 없음의 OCSP 결과를 수신하는 경우 이에 따른 작동은 OCSPAuthentication 속성의 설정에 따라 다릅니다. 큐 관리자의 경우 이 속성은 다음 위치 중 하나에 보관됩니다.

- 

AIX and Linux의 경우 `qm.ini` 파일의 SSL 스탠자.
- 

Windows 레지스트리.

이 속성은 IBM MQ Explorer를 사용하여 설정할 수 있습니다. 클라이언트의 경우 클라이언트 구성 파일의 SSL 스탠자에 속성이 저장됩니다.

알 수 없음 결과를 수신하고 OCSPAuthentication이 REQUIRED(기본값)로 설정되면, IBM MQ는 연결을 거부하고 AMQ9716 유형의 오류 메시지를 발행합니다. 큐 관리자 SSL 이벤트 메시지가 사용 가능한 경우, ReasonQualifier가 MQRQ_SSL_HANDSHAKE_ERROR로 설정된 MQRC_CHANNEL_SSL_ERROR 유형의 SSL 이벤트 메시지가 생성됩니다.

알 수 없음 결과를 수신하고 OCSPAuthentication이 OPTIONAL로 설정되면, IBM MQ에서는 SSL 채널이 시작되고 경고 또는 SSL 이벤트 메시지는 생성되지 않습니다.

알 수 없음 결과를 수신하고 OCSPAuthentication이 WARN으로 설정된 경우, SSL 채널은 시작되지만 IBM MQ는 오류 로그에 AMQ9717 유형의 경고 메시지를 발행합니다. 큐 관리자 SSL 이벤트 메시지가 사용되는 경우에는 ReasonQualifier가 MQRQ_SSL_UNKNOWN_REVOCATION으로 설정된 MQRC_CHANNEL_SSL_WARNING 유형의 SSL 이벤트 메시지가 생성됩니다.

## OCSP 응답의 디지털 서명

OCSP 응답자는 세 가지 방법 중 하나를 사용하여 응답에 서명할 수 있습니다. 응답자는 사용되는 방법에 대해 사용자에게 알려줍니다.

- 검사 중인 인증서를 발행한 동일한 CA 인증서를 사용하여 OCSP 응답에 디지털로 서명할 수 있습니다. 이 경우 추가 인증서를 설정할 필요가 없습니다. TLS 연결을 설정하기 위해 이미 수행한 단계가 OCSP 응답을 확인하기에 충분합니다.
- OCSP 응답은 검사 중인 인증서를 발행한 것과 동일한 인증 기관(CA)이 서명한 또 다른 인증서를 사용하여 디지털로 서명할 수 있습니다. 이 경우 서명 인증서는 OCSP 응답과 함께 전송됩니다. OCSP 응답자로부터 전송된 인증서에는 이 용도로 신뢰할 수 있도록 `id-kp-OCSPSigning`으로 설정된 확장 키 사용 확장(Extended Key Usage Extension)이 있어야 합니다. OCSP 응답이 서명한 인증서와 함께 전송되므로(인증서는 이미 TLS 연결을 위해 신뢰된 CA가 서명함) 다른 추가 인증서 설정이 필요하지 않습니다.
- 검사 중인 인증서와 직접 관련이 없는 다른 인증서를 사용하여 OCSP 응답에 디지털로 서명할 수 있습니다. 이 경우 OCSP 응답은 OCSP 응답자 자체가 발행한 인증서에 의해 서명됩니다. OCSP 점검을 수행하는 큐 관리자나 클라이언트의 키 데이터베이스에 OCSP 응답자 인증서 사본을 추가해야 합니다. 513 페이지의 『AIX, Linux, and Windows 의 키 저장소에 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트 추가』의 내용을 참조하십시오. CA 인증서가 추가되면, 기본적으로 신뢰 루트로서 추가되며 이는 이 컨텍스트에서 필수 설정입니다. 이 인증서를 추가할 수 없으면, IBM MQ는 OCSP 응답에서 디지털 서명을 확인할 수 없고 OCSPAuthentication 값에 따라 OCSP 검사에서는 알 수 없음 결과를 발생시켜 IBM MQ에서 채널을 종료시킬 수 있습니다.

## Java 및 JMS 클라이언트 애플리케이션에서 온라인 인증서 상태 프로토콜(OCSP)

Java API의 제한으로 인해, IBM MQ는 OCSP가 전체 Java 가상 머신(JVM) 프로세스에 대해 사용 가능한 경우에만 TLS 보안 소켓에 대해 온라인 인증서 상태 프로토콜(OCSP) 인증서 폐기 검사를 사용할 수 있습니다. JVM의 모든 보안 소켓에 대해 OCSP를 사용 가능으로 설정하는 방법은 두 가지가 있습니다.

- 테이블 1에 표시되는 OCSP 구성 설정을 포함하도록 JRE java.security 파일을 편집하고 애플리케이션을 재시작하십시오.
- 적용되는 모든 Java Security Manager 정책에 따라 java.security.Security.setProperty() API를 사용하십시오.

최소한 ocsp.enable 및 ocsp.responderURL 값 중 하나를 지정해야 합니다.

| 특성 이름 | 설명 |
|---|---|
| ocsp.enable | 이 특성 값은 `true` 또는 `false`입니다. `true`인 경우 인증서 폐기 검사를 수행할 때 OCSP 검사를 사용할 수 있고, `false` 또는 설정되지 않은 경우 OCSP 검사를 사용할 수 없습니다. |
| ocsp.responderURL | 이 특성 값은 OCSP 응답자의 위치를 식별하는 URL입니다. 다음은 예입니다. `ocsp.responderURL=http://ocsp.example.net:80`. 기본적으로 OCSP 응답자의 위치는 유효성을 검증하는 인증서로부터 암시적으로 판별됩니다. 특성은 (RFC 3280에서 정의된) 권한 정보 액세스 확장 기능이 인증서에 없거나 대체해야 하는 경우 사용됩니다. |
| ocsp.responderCertSubjectName | 이 특성 값은 OCSP 응답자의 인증서 제목 이름입니다. 다음은 예입니다. `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`. 기본적으로 OCSP 응답자의 인증서는 유효성을 검증하는 발행인의 인증서입니다. 이 특성은 기본이 적용되지 않을 경우 OCSP 응답자의 인증서를 식별합니다. 값은 인증서 경로 유효성 검증 도중 제공되는 인증서 세트에서 인증서를 식별하는 문자열 식별 이름(RFC 2253에서 정의됨)입니다. 제목 이름만으로는 인증서를 고유하게 식별하기에 부족한 경우, ocsp.responderCertIssuerName 및 ocsp.responderCertSerialNumber 특성 모두를 대신 사용해야 합니다. 이 특성이 설정되면 특성 ocsp.responderCertIssuerName 및 ocsp.responderCertSerialNumber는 무시됩니다. |
| ocsp.responderCertIssuerName | 이 특성 값은 OCSP 응답자 인증서의 발행인 이름입니다. 다음은 예입니다. `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`. 기본적으로 OCSP 응답자의 인증서는 유효성을 검증하는 발행인의 인증서입니다. 이 특성은 기본이 적용되지 않을 경우 OCSP 응답자의 인증서를 식별합니다. 값은 인증서 경로 유효성 검증 도중 제공되는 인증서 세트에서 인증서를 식별하는 문자열 식별 이름(RFC 2253에서 정의됨)입니다. 이 특성이 설정되면 ocsp.responderCertSerialNumber 특성도 설정해야 합니다. ocsp.responderCertSubjectName 특성이 설정되면 이 특성은 무시합니다. |
| ocsp.responderCertSerialNumber | 이 특성 값은 OCSP 응답자 인증서의 일련 번호입니다. 다음은 예입니다. `ocsp.responderCertSerialNumber=2A:FF:00`. 기본적으로 OCSP 응답자의 인증서는 유효성을 검증하는 발행인의 인증서입니다. 이 특성은 기본이 적용되지 않을 경우 OCSP 응답자의 인증서를 식별합니다. 값은 인증서 경로 유효성 검증 도중 제공되는 인증서 세트에서 인증서를 식별하는 16진 문자열(콜론이나 공백 분리문자가 있을 수 있음)입니다. 이 특성이 설정되면 ocsp.responderCertIssuerName 특성도 설정되어야 합니다. ocsp.responderCertSubjectName 특성이 설정되면 이 특성은 무시합니다. |

이러한 방식으로 OCSP를 사용으로 설정하기 전에는 여러 가지를 고려해야 합니다.

- OCSP 구성의 설정값은 JVM 프로세스에 있는 모든 보안 소켓에 영향을 줍니다. 어떤 경우에는 JVM이 TLS 보안 소켓을 사용하는 다른 애플리케이션 코드와 공유될 때 이 구성에는 원하지 않는 역효과가 있을 수 있습니다. 선택된 OCSP 구성이 동일한 JVM에서 실행하는 모든 애플리케이션에 적합한지 확인하십시오.
- JRE에 유지보수를 적용하면 java.security 파일이 덮어쓰기됩니다. java.security 파일의 덮어쓰기를 피하려면 Java 임시 수정사항 및 제품 유지보수를 적용할 때 주의하십시오. 유지보수를 적용한 후 java.security 변경사

항을 다시 적용해야 하는 경우도 있습니다. 이러한 이유로 java.security.Security.setProperty() API를 대신 사용한 OCSP 구성 설정을 고려할 수 있습니다.

- OCSP 검사 사용 설정은 폐기 검사도 사용으로 설정된 경우에만 효과가 있습니다. 폐기 검사는 `PKIXParameters.setRevocationEnabled()` 메소드에 의해 사용으로 설정됩니다.
- 기본 인터셉터에서 OCSP 검사 사용에 설명된 AMS Java 인터셉터를 사용 중인 경우, 키 저장소 구성 파일의 AMS OCSP 구성과 충돌하는 java.security OCSP 구성을 사용하지 않도록 주의하십시오.

## 인증서 폐기 목록(CRL) 및 권한 취소 목록(ARL) 관련 작업

CRL 및 ARL에 대한 IBM MQ 지원은 플랫폼마다 다릅니다.

각 플랫폼의 CRL 및 ARL 지원은 다음과 같습니다.

- Multi 멀티플랫폼에서 CRL및 ARL 지원은 PKIX X.509 V2 CRL 프로파일 권장사항을 준수합니다.
- z/OS z/OS에서, 시스템 SSL은 Tivoli Public Key Infrastructure(PKI) 제품에 의해 LDAP 서버에 저장된 CRL 및 ARL을 지원합니다.

IBM MQ는 앞서 12시간 동안 액세스된 CRL 및 ARL의 캐시를 유지합니다.

큐 관리자 또는 IBM MQ MQI client가 인증서를 수신하면 이는 인증서가 여전히 유효한지 확인하기 위해 CRL을 검사합니다. IBM MQ는 먼저 캐시가 있는 경우 캐시를 검사합니다. CRL이 캐시에 없으면 IBM MQ에서 사용 가능한 CRL을 찾을 때까지 *SSLCRLNL* 속성을 통해 지정한 인증 정보 오브젝트의 이름 목록에 표시되는 순서대로 IBM MQ에서 LDAP CRL 서버 위치를 조사합니다. 이름 목록이 지정되어 있지 않거나, 빈 값으로 지정되어 있으면, CRL이 검사되지 않습니다.

### *LDAP* 서버 설정

CA의 식별 이름의 계층을 반영하여 LDAP 디렉토리 정보 트리 구조를 구성하십시오. LDAP 데이터 교환 형식 파일을 사용하여 이를 수행하십시오.

LDAP 디렉토리 정보 트리(DIT) 구조를 구성하여 인증서와 CRL을 발행하는 CA의 식별 이름(DN)에 해당하는 계층을 사용하십시오. LDAP 데이터 교환 형식(LDIF)을 사용하는 파일로 DIT 구조를 설정할 수 있습니다. 디렉토리를 업데이트하는 데 LDIF 파일을 사용할 수도 있습니다.

LDIF 파일은 LDAP 디렉토리 안에서 오브젝트를 정의하는 데 필요한 정보를 가지는 ASCII 텍스트 파일입니다. LDIF 파일에는 하나 이상의 항목이 있으며, 이들 각각은 식별 이름(DN), 적어도 하나의 오브젝트 클래스 정의와, 선택적으로 여러 속성 정의를 이룹니다.

`certificateRevocationList;binary` 속성에는 폐기된 사용자 인증서의 목록의 2진 형식이 있습니다. `authorityRevocationList;binary` 속성에는 취소된 CA 인증서의 2진 목록이 포함되어 있습니다. IBM MQ TLS와 함께 사용하기 위해 이러한 속성의 2진 데이터는 DER(Definite Encoding Rules) 형식을 준수해야 합니다. LDIF 파일에 대한 자세한 정보는 LDAP 서버와 함께 제공된 문서를 참조하십시오.

324 페이지의 그림 20 에서는 IBM내의 테스트 조직에서 설정한 식별 이름 "CN=CA1, OU=Test, O=IBM, C=GB"가 있는 가상 인증 기관인 CA1에서 발행한 CRL및 ARL을 로드하기 위해 LDAP 서버에 대한 입력으로 작성할 수 있는 샘플 LDIF 파일을 보여줍니다.

```
dn: o=IBM, c=GB
o: IBM
objectclass: top
objectclass: organization

dn: ou=Test, o=IBM, c=GB
ou: Test
objectclass: organizationalUnit

dn: cn=CA1, ou=Test, o=IBM, c=GB
cn: CA1
objectclass: cRLDistributionPoint
objectclass: certificateAuthority
authorityRevocationList;binary:: (DER format data)
certificateRevocationList;binary:: (DER format data)
caCertificate;binary:: (DER format data)
```

그림 *20.* 인증 기관의 샘플 *LDIF* 파일 이는 구현에 따라 달라질 수 있습니다.

324 페이지의 그림 21 에서는 324 페이지의 그림 20 에 표시된 샘플 LDIF 파일을 CA2에 대한 유사한 파일과 함께 로드할 때 LDAP 서버가 작성하는 DIT 구조를 표시합니다. 이 파일은 PKI 조직에 의해 설정된 가상 인증 기관이며 IBM내에도 있습니다.

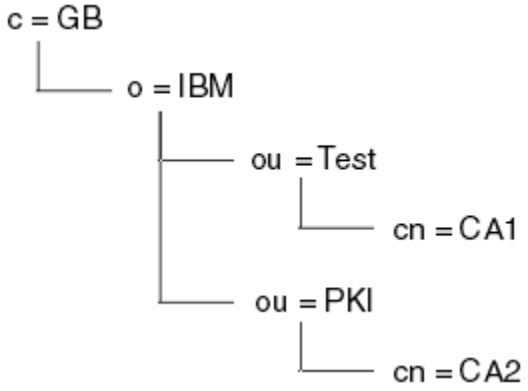



그림 *21. LDAP* 디렉토리 정보 트리*(DIT)* 구조의 예

IBM MQ는 CRL 및 ARL을 모두 검사합니다.

**참고:** LDAP 서버의 액세스 제어 목록을 사용하여 권한이 부여된 사용자가 CRL과 ARL을 보유하는 항목을 읽고 검색하고 비교할 수 있는지 확인하십시오. IBM MQ 는 AUTHINFO 오브젝트의 LDAPUSER및 LDAPPWD 특성을 사용하여 LDAP 서버에 액세스합니다.

*LDAP* 서버 구성 및 업데이트
이 프로시저를 사용하여 LDAP 서버를 구성하거나 업데이트하십시오.

1. 인증 기관에서 DER 형식의 CRL 및 ARL을 확보하십시오.
2. 텍스트 편집기나 LDAP 서버에 제공되는 도구를 사용하여 CA의 식별 이름과 필요한 오브젝트 클래스 정의를 포함하는 하나 이상의 LDIF 파일을 작성하십시오. DER 형식 데이터를 CRL의 `certificateRevocationList;binary` 속성, ARL의 `authorityRevocationList;binary` 속성 값 또는 둘 모두로서 LDIF 파일에 복사하십시오.
3. LDAP 서버를 시작하십시오.
4. 324 페이지의 『2』 단계에서 작성한 LDIF 파일에서 항목을 추가하십시오.

LDAP CRL 서버를 구성한 후에 올바르게 설정되었는지 검사하십시오. 먼저, 채널에서 폐기되지 않은 인증서를 사용해 보고 채널이 올바르게 시작되는지 검사하십시오. 그런 다음, 폐기된 인증서를 사용한 후 채널 시작에 실패하는지 검사하십시오.

인증 기관으로부터 업데이트된 CRL을 자주 얻으십시오. LDAP 서버에서 12시간마다 확보하도록 하십시오.

## 큐 관리자로 *CRL* 및 *ARL*에 액세스

큐 관리자는 LDAP CRL 서버의 주소를 보유하는 하나 이상의 인증 정보 오브젝트와 연관되어 있습니다.

IBM i IBM MQ on IBM i 는 다른 플랫폼과 다르게 작동합니다.

이 절에 있는 인증서 폐기 목록(CRL)에 대한 정보는 권한 취소 목록(ARL)에도 적용됩니다.

큐 관리자에게 각각 LDAP CRL 서버의 주소를 보유하는 인증 정보 오브젝트를 제공하여 큐 관리자에게 CRL에 액세스하는 방법을 알려주십시오. 인증 정보 오브젝트는 *SSLCRLNL* 큐 관리자 속성에 지정된 이름 목록에 보유됩니다.

다음 예에서는 매개변수를 지정하는 데 MQSC가 사용됩니다.

1. AUTHTYPE 매개변수를 CRLLDAP으로 설정한 DEFINE AUTHINFO MQSC 명령을 사용하여 인증 정보 오브젝트를 정의하십시오. IBM i IBM i에서는 CRTMQMAUTI CL 명령도 사용할 수 있습니다.

   AUTHTYPE 매개변수의 CRLLDAP 값은 LDAP 서버에서 CRL에 액세스됨을 나타냅니다. 사용자가 작성한 CRLLDAP 유형을 가진 각 인증 정보 오브젝트는 LDAP 서버의 주소를 보유합니다. 둘 이상의 인증 정보 오브젝트를 가지고 있는 경우, 이러한 오브젝트가 가리키는 LDAP 서버에 동일한 정보가 있어야 합니다. 이렇게 하면 하나 이상의 LDAP 서버가 실패하는 경우에 서비스의 지속성을 제공합니다.

   z/OS 또한 z/OS에서만, 모든 LDAP 서버는 동일 사용자 ID 및 비밀번호를 사용하여 액세스되어야 합니다. 사용되는 사용자 ID와 암호는 이름 목록의 첫 번째 AUTHINFO 오브젝트에 지정된 것입니다.

   모든 플랫폼에서 사용자 ID와 비밀번호는 암호화되지 않은 LDAP 서버에 전송됩니다.
2. DEFINE NAMELIST MQSC 명령을 사용하여, 인증 정보 오브젝트의 이름에 이름 목록을 정의하십시오.
   z/OS z/OS에서는 NLTYPE 이름 목록 속성이 AUTHINFO로 설정되는지 확인하십시오.
3. ALTER QMGR MQSC 명령을 사용하여 큐 관리자에 이름 목록을 제공하십시오. 예를 들면, 다음과 같습니다.

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   여기서, `sslcrlnlname`은 인증 정보 오브젝트의 이름 목록입니다.

   이 명령은 *SSLCRLNL*이라는 큐 관리자 속성을 설정합니다. 이 속성에 대한 큐 관리자의 초기값은 공백입니다.

IBM i IBM i에서 인증 정보 오브젝트를 지정할 수 있지만 큐 관리자는 인증 정보 오브젝트 또는 인증 정보 오브젝트의 이름 목록을 사용할 수 없습니다. IBM i 큐 관리자가 생성한 클라이언트 연결 테이블을 사용하는 IBM MQ 클라이언트만 해당 IBM i 큐 관리자에 지정된 인증 정보를 사용합니다. IBM i 의 *SSLCRLNL* 큐 관리자 속성은 이러한 클라이언트가 사용하는 인증 정보를 판별합니다. IBM i 큐 관리자에게 CRL에 액세스하는 방법을 알리는 방법에 대한 정보는 325 페이지의 『IBM i에서 CRL 및 ARL에 액세스』 의 내용을 참조하십시오.

최대 10개의 대체 LDAP 서버에 대한 연결을 이름 목록에 추가하여 하나 이상의 LDAP 서버가 실패하는 경우에 서비스의 지속성을 보장할 수 있습니다. LDAP 서버에는 동일한 정보가 있어야 합니다.

### IBM i *IBM i*에서 *CRL* 및 *ARL*에 액세스

IBM i에서 CRL 또는 ARL에 액세스하려면 이 프로시저를 사용하십시오.

이 절에 있는 인증서 폐기 목록(CRL)에 대한 정보는 권한 취소 목록(ARL)에도 적용됩니다.

IBM i에서 특정 인증서에 대해 CRL 위치를 설정하려면 다음 단계를 수행하십시오.

1. 260 페이지의 『DCM 액세스』 에 설명된 대로 DCM 인터페이스에 액세스하십시오.
2. 검색 패널의 **CRL 위치 관리** 작업 범주에서 **CRL 위치 추가**를 누르십시오. CRL 위치 관리 페이지가 작업 프레임에 표시됩니다.
3. **CRL 위치 이름** 필드에 CRL 위치 이름을 입력하십시오 (예: `LDAP Server #1`).
4. **LDAP 서버** 필드에 LDAP 서버 이름을 입력하십시오.
5. **SSL(Secure Sockets Layer)** 필드에서, TLS를 사용하여 LDAP 서버에 연결하려면 **예**를 선택하십시오. 그렇지 않으면 **아니오**를 선택합니다.
6. **포트 번호** 필드에 LDAP 서버의 포트 번호를 입력하십시오(예: 389).

7. LDAP 서버가 익명 사용자의 디렉토리 조회를 허용하지 않는 경우, **로그인 식별 이름** 필드에 이 서버의 로그인 식별 이름을 입력하십시오.
8. **확인**을 클릭하십시오. DCM이 CRL 위치를 작성했음을 알려줍니다.
9. 탐색 패널에서 **인증서 저장소 선택**을 클릭하십시오. 인증서 저장소 선택 페이지가 태스크 프레임에 표시됩니다.
10. **다른 시스템 인증서 저장소** 선택란을 선택한 후 **계속**을 클릭하십시오. 인증서 저장소 및 비밀번호 페이지가 표시됩니다.
11. **인증서 저장소 경로 및 파일 이름** 필드에 262 페이지의 『IBM i에서 인증서 저장소 작성』에서 설정한 IFS 경로 및 파일 이름을 입력하십시오.
12. **인증서 저장소 비밀번호** 필드에 비밀번호를 입력하십시오. **계속**을 클릭하십시오. 현재 인증서 저장소 페이지가 태스크 프레임에 표시됩니다.
13. 검색 패널의 **인증서 관리** 작업 범주에서 **CRL 위치 지정 업데이트**를 클릭하십시오. CRL 위치 지정 페이지가 작업 프레임에 표시됩니다.
14. CRL 위치를 지정하려는 CA 인증서의 단일 선택 단추를 선택하십시오. **CRL 위치 지정 업데이트**를 클릭하십시오. CRL 위치 지정 업데이트 페이지가 작업 프레임에 표시됩니다.
15. 인증서에 지정하려는 CRL 위치의 단일 선택 단추를 선택하십시오. **인증 업데이트**를 클릭하십시오. DCM이 지정을 업데이트했음을 알려줍니다.

DCM은 사용자가 인증 기관별로 다른 LDAP 서버를 지정하도록 허용함을 유의하십시오.

*IBM MQ Explorer*를 사용하여 *CRL* 및 *ARL*에 액세스
IBM MQ Explorer를 사용하여 큐 관리자에게 CRL에 액세스하는 방법을 알려줄 수 있습니다.

이 절에 있는 인증서 폐기 목록(CRL)에 대한 정보는 권한 취소 목록(ARL)에도 적용됩니다.

CRL에 LDAP 연결을 설정하려면 다음 프로시저를 사용하십시오.

1. 큐 관리자를 시작했는지 확인하십시오.
2. **인증 정보** 폴더를 마우스 오른쪽 단추로 클릭하고 **새로 작성 -> 인증 정보**를 클릭하십시오. 열리는 등록 정보 시트에서 다음과 같이 하십시오.
   a. 첫 페이지인 **인증 정보 작성**에서 CRL(LDAP) 오브젝트에 이름을 입력하십시오.
   b. **특성 변경**의 **일반** 페이지에서 연결 유형을 선택하십시오. 선택적으로 설명을 입력할 수 있습니다.
   c. **특성 변경**의 **CRL(LDAP)** 페이지를 선택하십시오.
   d. 네트워크 이름이나 IP 주소로 LDAP 서버 이름을 입력하십시오.
   e. 서버가 로그인 세부사항을 필요할 경우, 사용자 ID와 암호(필요 시)를 제공하십시오.
   f. **확인**을 클릭하십시오.
3. 이름 목록 폴더를 마우스 오른쪽 단추로 클릭하고 **새로 작성 -> 이름 목록**을 클릭하십시오. 열리는 등록 정보 시트에서 다음과 같이 하십시오.
   a. 이름 목록에 이름을 입력하십시오.
   b. CRL(LDAP) 오브젝트의 이름을(326 페이지의 『2.a』 단계로부터) 목록에 추가하십시오.
   c. **확인**을 클릭하십시오.
4. 큐 관리자를 마우스 오른쪽 단추로 클릭하고 **특성**을 선택하고, **SSL** 페이지를 선택하십시오.
   a. **인증서 폐기 목록(CRL)에 대해 이 큐 관리자가 수신한 인증서 검사** 선택란을 선택하십시오.
   b. **CRL 이름 목록** 필드에 이름 목록의 이름(326 페이지의 『3.a』 단계로부터)을 입력하십시오.

## *IBM MQ MQI client*를 사용하여 *CRL* 및 *ARL*에 액세스

IBM MQ MQI client에 의해 검사를 위해 CRL을 보유하는 LDAP 서버를 지정하는 세 가지 옵션이 있습니다.

이 절에 있는 인증서 폐기 목록(CRL)에 대한 정보는 권한 취소 목록(ARL)에도 적용됩니다.

LDAP 서버를 지정하는 3가지 방법은 다음과 같습니다.

- 채널 정의 테이블 사용

- MQCONNX 호출에서 SSL 구성 옵션 구조인 MQSCO 사용
- Active Directory를 지원하는 Windows 시스템에서 Active Directory 사용

자세한 내용은 관련 정보를 참조하십시오.

10개까지의 연결을 대체 LDAP 서버에 포함시켜 하나 이상의 LDAP 서버가 실패할 경우 서비스의 지속성을 보장할 수 있습니다. LDAP 서버에는 동일한 정보가 있어야 합니다.

Linux ( zSeries 플랫폼) 에서 실행 중인 IBM MQ MQI client 채널에서 LDAP CRL에 액세스할 수 없습니다.

*OCSP* 응답자 및 *CRL*을 보유하는 *LDAP* 서버의 위치
IBM MQ MQI client 시스템에서 OCSP 응답자의 위치 및 인증서 폐기 목록(CRL)을 보유하는 LDAP(Lightweight Directory Access Protocol) 서버의 위치를 지정할 수 있습니다.

여기에 나와 있는 세 가지 방식으로(아래로 갈수록 우선순위가 낮아짐) 이 위치를 지정할 수 있습니다.

IBM i IBM i의 경우 IBM i에서 CRL 및 ARL 액세스를 참조하십시오.

## IBM MQ MQI client 애플리케이션이 MQCONNX 호출을 실행할 때

**MQCONNX** 호출에 OCSP 응답자 또는 CRL을 보유하는 LDAP 서버를 지정할 수 있습니다.

**MQCONNX** 호출에서 연결 옵션 구조, MQCNO은 SSL 구성 옵션 구조, MQSCO를 참조할 수 있습니다. 대신 MQSCO 구조는 하나 이상의 인증 정보 레코드 구조, MQAIR을 참조할 수 있습니다. 각 MQAIR 구조에는 IBM MQ MQI client가 OCSP 응답자 또는 CRL을 보유하는 LDAP 서버에 액세스하기 위해 필요한 모든 정보가 포함되어 있습니다. 예를 들어, MQAIR 구조의 필드 중 하나는 응답자가 접속할 수 있는 URL입니다. MQAIR 구조에 대한 자세한 정보는 MQAIR - 인증 정보 레코드를 참조하십시오.

## OCSP 응답자 또는 LDAP 서버에 액세스하기 위해 클라이언트 채널 정의 테이블(ccdt) 사용

IBM MQ MQI client가 OCSP 응답자 또는 CRL을 보유하는 LDAP 서버에 액세스할 수 있도록 클라이언트 채널 정의 테이블에 하나 이상의 인증 정보 오브젝트의 속성을 포함하십시오.

서버 큐 관리자에서 하나 이상의 인증 정보 오브젝트를 정의할 수 있습니다. 인증 오브젝트의 속성은 OCSP 응답자(OCSP가 지원되는 플랫폼에서) 또는 CRL을 보유하는 LDAP 서버에 액세스하기 위해 필요한 모든 정보를 포함합니다. 속성 중 하나는 OCSP 응답자 URL을 지정하고, 또 다른 하나는 LDAP 서버가 실행될 수 있는 시스템의 호스트 주소 또는 IP 주소를 지정합니다.

z/OS IBM i AUTHTYPE(OCSP)이 있는 인증 정보 오브젝트는 IBM i 또는 z/OS 큐 관리자에서 사용하기 위해 신청할 수 없지만 클라이언트 용도를 위해 클라이언트 채널 정의 테이블(CCDT)에 복사될 플랫폼에 지정될 수 있습니다.

OCSP 응답자 또는 CRL을 보유하는 LDAP 서버에 액세스하기 위해 IBM MQ MQI client를 사용 가능으로 설정하려면 하나 이상의 인증 정보 오브젝트의 속성이 클라이언트 채널 정의 테이블에 포함될 수 있습니다. 다음 방법 중 하나로 이러한 속성을 포함할 수 있습니다.

Multi

**서버 플랫폼 AIX, Linux, IBM i 및 Windows에서**

하나 이상의 인증 정보 오브젝트의 이름을 포함하는 이름 목록을 정의할 수 있습니다. 그런 다음 큐 관리자 속성, **SSLCRLNL**을 이 이름 목록의 이름으로 설정할 수 있습니다.

CRL을 사용 중인 경우 둘 이상의 LDAP 서버가 보다 높은 사용 가능성을 제공하도록 구성할 수 있습니다. 이는 각 LDAP 서버가 동일한 CRL을 보유하도록 하기 위한 것입니다. 한 LDAP 서버가 필요할 때 사용할 수 없는 경우, IBM MQ MQI client는 다른 서버에 액세스하려고 시도할 수 있습니다.

이름 목록으로 식별되는 인증 정보 오브젝트의 속성이 여기에서는 집합적으로 인증서 폐기 위치로 참조됩니다. 큐 관리자 속성, **SSLCRLNL**을 이름 목록의 이름으로 설정하면 인증서 폐기 위치가 큐 관리자와 연관된 클라이언트 채널 정의 표에 복사됩니다. CCDT를 공유 파일로서 클라이언트 시스템으로부터 액세스할 수 있거나 CCDT가 클라이언트 시스템에 복사되는 경우, 해당 시스템의 IBM MQ MQI client는 CCDT에서 인증서 폐기 위치를 사용하여 OCSP 응답자 또는 CRL을 보유하는 LDAP 서버에 액세스할 수 있습니다.

큐 관리자의 인증서 폐기 위치가 나중에 변경되면 변경사항이 큐 관리자와 연관된 CCDT에 반영됩니다. 큐 관리자 속성 **SSLCRLNL**이 공백으로 설정되면 CCDT에서 인증서 폐기 위치가 제거됩니다. 이들 변경사항은 클라이언트 시스템의 테이블 사본에 반영되지 않습니다.

MQI 채널에서 클라이언트 및 서버 끝의 인증서 폐기 위치가 달라야 하고 서버 큐 관리자가 인증서 폐기 위치를 작성하는 데 사용되는 경우 다음을 수행할 수 있습니다.

1. 서버 큐 관리자에서 클라이언트 시스템에 사용할 인증서 폐기 위치를 작성하십시오.
2. 인증서 폐기 위치가 포함된 CCDT를 클라이언트 시스템에 복사하십시오.
3. 서버 큐 관리자에서 MQI 채널의 서버 측에 필요한 정보로 인증서 폐기 위치를 변경하십시오.
4. 클라이언트 시스템에서 **runmqsc** 명령을 **-n** 매개변수와 함께 사용할 수 있습니다.

Multi

**클라이언트 플랫폼 AIX, Linux, IBM i 및 Windows에서**

CCDT 파일에서 runmqsc 명령을 **-n** 매개변수 및 **DEFINE AUTHINFO** 오브젝트와 함께 사용하여 클라이언트 시스템에 CCDT를 빌드할 수 있습니다. 오브젝트가 정의되는 순서는 이들이 파일에서 사용되는 순서입니다. **DEFINE AUTHINFO** 오브젝트에서 사용할 수 있는 모든 이름은 파일에 보관되지 않습니다. 유일한 자릿수는 **AUTHINFO** 오브젝트를 CCDT 파일에 **DISPLAY**할 때 사용됩니다.

**참고: -n** 매개변수를 지정하는 경우 다른 매개변수를 지정하지 않아야 합니다.

## Windows에서 Active Directory 사용

Windows

Windows 시스템에서 **setmqcrl** 제어 명령을 사용하여 Active Directory에서 현재 CRL 정보를 공개할 수 있습니다.

**setmqcrl** 명령은 OCSP 정보를 공개하지 않습니다.

이 명령 및 해당 구문에 대한 자세한 정보는 setmqcrl의 내용을 참조하십시오.

## *IBM MQ classes for Java* 및 *IBM MQ classes for JMS*를 사용하여 *CRL* 및 *ARL*에 액세스

IBM MQ classes for Java 및 IBM MQ classes for JMS는 다른 플랫폼에서와 다르게 CRL에 액세스합니다.

IBM MQ classes for Java에서 CRL 및 ARL에 대한 작업의 정보는 인증서 폐기 목록 사용을 참조하십시오.

IBM MQ classes for JMS에서 CRL 및 ARL에 대한 작업 정보는 SSLCERTSTORES 오브젝트 특성을 참조하십시오.

## 인증 정보 오브젝트 조작

MQSC 또는 PCF 명령 또는 IBM MQ Explorer를 사용하여 인증 정보 오브젝트를 조작할 수 있습니다.

다음 MQSC 명령이 인증 정보 오브젝트에 대해 수행합니다.

- DEFINE AUTHINFO
- ALTER AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

이러한 명령에 대한 완전한 설명은 MQSC 명령을 참조하십시오.

다음 프로그래밍 가능 명령 형식(PCF) 명령이 인증 정보 오브젝트에 대해 수행합니다.

- 인증 정보 작성
- 인증 정보 복사
- 인증 정보 변경
- 인증 정보 삭제
- 인증 정보 조회

- 인증 정보 이름 조회

이러한 명령의 전체 설명은 프로그램 가능한 명령 형식의 정의를 참조하십시오.

사용 가능한 플랫폼에서 IBM MQ Explorer를 사용할 수도 있습니다.

### Linux AIX 플러그 가능한 인증 방법(PAM) 사용

AIX and Linux 플랫폼에서만 PAM을 사용할 수 있습니다. 일반 AIX 또는 Linux 시스템에는 기존 인증 메커니즘을 구현하는 PAM 모듈 등이 있을 수 있습니다. 비밀번호 유효성 검증의 기본 태스크뿐만 아니라 PAM 모듈은 또한 추가 규칙을 수행하기 위해 호출될 수도 있습니다.

구성 파일은 각 애플리케이션에 어떤 인증 방법이 사용될지를 정의합니다. 예 애플리케이션에는 표준 터미널 로그인, ftp 및 텔넷이 포함됩니다.

PAM의 장점은 애플리케이션이 사용자 ID가 실제로 인증되는 방법에 대해서 알 필요가 없고 관심을 갖지 않는다는 것입니다. 애플리케이션이 올바른 인증 데이터 양식을 PAM에 제공할 수 있는 한 그 뒤에 있는 메커니즘은 투명합니다.

인증 데이터의 양식은 사용 중인 시스템에 달려 있습니다. 예를 들어, IBM MQ 는 MQCONNX API 호출에서 사용되는 MQCSP 구조와 같은 매개변수를 통해 비밀번호를 확보합니다.

**중요사항:** IBM MQ 8.0.0 Fix Pack 3를 설치한 후 필요한 명령 레벨을 설정하기 위해 *802* ( strmqm 명령에서) 의 **-e CMDLEVEL=**레벨 을 사용하여 큐 관리자를 재시작할 때까지 **AUTHENMD** 속성을 설정할 수 없습니다.

#### PAM을 사용하도록 시스템 구성

PAM을 호출할 때 IBM MQ가 사용하는 서비스 이름은 *ibmmq*입니다.

IBM MQ 설치는 다른 운영 체제의 알려진 기본값을 기반으로 운영 체제 사용자로부터의 연결을 사용하는 기본 PAM 구성을 유지보수하려고 시도함을 유의하십시오.

그러나 시스템 관리자가 /etc/pam.conf또는 /etc/pam.d/ibmmq 파일에 정의된 규칙이 여전히 적절한지 확인해야 합니다.

## 오브젝트에 액세스 권한 부여

이 절에는 오브젝트에 대한 액세스를 제어하기 위해 오브젝트 권한 관리자 및 채널 엑시트 프로그램 사용에 대한 정보가 포함되어 있습니다.

ALW AIX, Linux, and Windows 시스템에서. 오브젝트 권한 관리자(OAM)를 사용하여 오브젝트에 대한 액세스를 제어합니다. 이 토픽 콜렉션에는 OAM에 대한 명령 인터페이스 사용에 대한 정보가 포함됩니다.

이 절에는 모든 플랫폼에서 사용자 시스템에 보안을 적용하기 위해 수행할 태스크를 결정하기 위한 체크리스트와, IBM MQ를 관리하고 IBM MQ 오브젝트에 대해 작업할 수 있는 권한을 부여할 때의 고려사항도 포함되어 있습니다.

제공된 보안 메커니즘이 사용자의 필요를 충족시키지 않은 경우 사용자 고유의 채널 엑시트 프로그램을 개발할 수 있습니다.

### 권한 부여에 사용되는 사용자 판별

자원에 액세스할 수 있는 권한은 사용자가 구성원인 그룹에 부여되거나 특정 모드에서 연결과 연관된 사용자에게 직접 부여됩니다. 연결 프로세스 중, 특히 리모트 (클라이언트) 연결의 경우, 이 ID는 큐 관리자의 구성에 의해 변경될 수 있습니다. 이 페이지에는 연결 애플리케이션의 ID및 이러한 기능이 적용되는 우선순위에 영향을 줄 수 있는 IBM MQ 의 다양한 기능 및 해당 구성 옵션이 나열되어 있습니다.

#### 채택된 사용자를 수정할 수 있는 기능

권한을 부여해야 하는 사용자를 설정할 수 있는 다른 기능은 다음과 같습니다.

**애플리케이션 확인 사용자**
IBM MQ에서 원격 연결을 시작하면 프로세스가 실행 중인 운영 체제 사용자가 수신 큐 관리자로 송신됩니다. 이 사용자는 사용자를 수정하는 추가 구성이 없는 경우 권한 검사에 사용할 수 있는 사용자가 있는지 확인하기 위해 전송됩니다.

서버 측 유효성 검증 없이 연결이 해당 ID를 어설션할 수 있으므로 이 사용자를 권한 부여의 기초로 사용하는 것은 권장되지 않습니다. 여기에는 관리 사용자 ('mqm') 도 포함될 수 있습니다.

**채널 MCAUSER 설정**
네트워크 바인딩을 통해 연결하는 애플리케이션은 IBM MQ 채널 정의를 사용하여 이를 수행합니다. 채널 정의는 **MCAUSER** 속성을 지원합니다. 이 속성을 사용하여 연결 애플리케이션에서 어설션한 사용자 대신 권한 부여에 사용할 다른 사용자를 지정할 수 있습니다.

**연결 인증 ADOPTCTX**
애플리케이션은 인증을 위해 큐 관리자에 송신할 사용자 및 비밀번호를 지정할 수 있습니다. 이러한 신임 정보는 연결 인증 기능에 지정된 구성을 사용하여 인증됩니다. 연결 인증을 위한 **ADOPTCTX** 옵션은 사용자가 성공적으로 유효성 검증된 후 권한 부여에 사용되어야 하는지 여부를 제어합니다. YES로 설정하면 인증을 위해 제공되는 사용자가 권한 검사를 위해 채택됩니다.

V 9.4.0 IBM MQ 9.3.4부터 인증을 위해 토큰을 제공할 수 있습니다. **ADOPTCTX** 가 YES로 설정되면 토큰에 포함된 청구에서 사용자가 채택됩니다.

**채널 인증 레코드 MCAUSER**
연결 처리 중에 큐 관리자가 연결과 일치하는 채널 인증 레코드를 찾으려고 시도합니다. 채널 인증 레코드가 일치하고 해당 **USERSRC** 속성 값이 MAP으로 설정된 경우, IBM MQ 는 권한 부여에 사용되는 사용자를 **MCAUSER** 속성의 값으로 변경합니다.

**보안 엑시트**
보안 엑시트는 IBM MQ 보안 처리 중에 작성하고 호출할 수 있는 사용자 정의 함수입니다. 함수가 호출되면 권한 검사에 사용될 연결 사용자와 관련된 여러 필드를 포함하는 MQCD 구조의 사본과 함께 제공됩니다. 보안 엑시트는 이러한 필드를 수정하여 권한을 부여할 사용자를 변경할 수 있습니다.

## 우선 순서

다음 표는 IBM MQ 가 권한을 부여할 사용자를 선택할 때 329 페이지의 『채택된 사용자를 수정할 수 있는 기능』 에 설명된 각 보안 기능의 우선순위를 표시합니다. 순서는 가장 낮은 행부터 가장 높은 행까지입니다. 즉, 첫 번째 행에서 사용자를 설정하는 보안 기능은 다른 행으로 대체됩니다.

표 *71.* 보안 기능의 우선순위

| **순서** | **기능** |
|---|---|
| 1 (최저값) | 애플리케이션 확인 ID |
| 2 | 채널 정의 **MCAUSER** 속성 |
| 3 | **ADOPTCTX(YES)** 를 사용한 연결 인증 |
| 4 | **USERSRC(MAP)** 의 채널 인증 레코드 |
| 5 (최고) | 보안 엑시트 |

## 초기 채택의 영향

연결 인증 및 채널 인증 레코드는 연결 인증 사용자 채택이 수행되는 시기를 제어하는 구성 옵션을 제공합니다. 이 설정을 조기 채택이라고 합니다. 조기 채택이 사용으로 설정된 경우 채널 인증 레코드가 처리되기 전에 연결 인증 ID 채택이 발생합니다 (즉, 채널 인증 레코드가 **CONNAUTH** 채택을 대체함).

사용 안함으로 설정하면 순서가 반대로 됩니다. 즉, **CONNAUTH** 채택 전에 채널 인증 레코드가 처리됩니다. 이 상황에서 연결 인증 채택은 채널 인증이 기록하는 것보다 더 높은 유효 우선순위를 가집니다.

조기 채택의 기본 설정은 사용입니다.

# ALW AIX, Linux, and Windows에서 OAM을 사용하여 오브젝트에 대한 액세스 제어

오브젝트 권한 관리자(OAM)는 IBM MQ 오브젝트에 권한을 부여하고 권한 취소하기 위한 명령 인터페이스를 제공합니다.

375 페이지의 『AIX, Linux, and Windows 에서 IBM MQ 를 관리할 수 있는 권한』 에서 설명한 바와 같이 이러한 명령을 사용할 적합한 권한을 가지고 있어야 합니다. IBM MQ를 관리할 권한이 있는 사용자 ID에는 큐 관리자에 대한 슈퍼유저 권한을 가지고 있습니다. 이는 이들에게 MQI 요청이나 명령을 실행하기 위해 추가 권한을 부여할 필요가 없다는 의미입니다.

## Linux AIX AIX and Linux의 OAM 사용자 기반 권한

UNIX and Linux 시스템에서 오브젝트 권한 관리자(OAM)가 그룹 기반 권한뿐 아니라 사용자 기반 권한도 사용할 수 있습니다.

IBM MQ 8.0 이전에, UNIX and Linux의 액세스 제어 목록(ACL)은 그룹 전용 기반이었습니다. IBM MQ 8.0부터 ACL은 사용자 ID및 그룹을 모두 기반으로 하며 qm.ini 파일의 서비스 스탠자에 설명된 대로 **SecurityPolicy** 속성을 적절한 값으로 설정하여 권한 부여를 위해 사용자 기반 모델 또는 그룹 기반 모델을 사용할 수 있습니다.

### IBM MQ 8.0 이상에서 작동의 변경사항

IBM MQ 8.0부터, 사용자 기반 정책으로 실행할 때 일부 명령은 이전 버전의 제품과 다른 정보를 리턴합니다.

- **dmpmqaut** 및 **dmpmqcfg** 명령은 PCF와 동등한 조작과 마찬가지로 사용자 기반 레코드를 표시합니다.
- IBM MQ Explorer의 OAM 플러그인이 사용자 기반 레코드를 표시하고 사용자 기반 수정을 허용합니다.
- OAM **Inquire** 기능은 사용자가 수행할 수 있는 작업을 표시하는 결과를 리턴합니다.

**setmqaut** 명령에서 **-p** 속성을 사용하면 qm.ini 파일의 서비스 스탠자에 설명된 대로 사용자 기반 권한이 `qm.ini` 파일에서 사용 가능한 경우 동일한 기본 그룹의 모든 사용자에게 액세스를 부여하지 않습니다.

사용자 기반 권한을 적용하기 시작하고 다수의 사용자가 있는 경우, 그룹 기반 모델을 사용하는 것보다 AUTH 큐에 저장된 더 많은 레코드가 있을 수 있으며 권한 부여 프로세스에는 확인할 레코드가 더 많기 때문에 이전에 비해 더 오랜 시간이 걸릴 수 있습니다. 이 시간 증가는 그다지 크지 않아야 합니다. 필요할 경우 사용자와 그룹 권한의 혼합을 사용할 수 있습니다.

### 마이그레이션 고려사항

그룹에서 기존 큐 관리자에 대한 사용자로 모델을 변경할 경우, 즉각적인 효과는 없습니다. 이미 작성된 권한이 계속해서 적용됩니다. 큐 관리자에 연결된 사용자는 이전과 동일한 권한을 받습니다(해당 ID가 속한 모든 그룹의 결합). 새 **setmqaut** 명령이 사용자 ID에 발행되면 이는 즉시 적용됩니다.

사용자 정책을 사용하여 새 큐 관리자를 작성하는 경우, 이 큐 관리자는 해당 큐 관리자를 작성하는 사용자(일반적으로 반드시 `mqm` 사용자 ID일 필요는 없음)에 대한 권한만 가질 수 있습니다. 자동으로 `mqm` 그룹에 부여되는 권한도 있습니다. 그러나 기본 그룹으로서 `mqm`을 가지고 있지 않으면 `mqm` 그룹은 초기 권한부여 세트에 포함되지 않습니다.

사용자에서 그룹 정책으로 이동할 경우, 사용자 기반 권한이 자동으로 삭제되지 않습니다. 그러나 권한 검사 동안에는 더 이상 사용되지 않습니다. 정책을 되돌리기 전에 현재 구성을 저장하고 정책을 변경하고 큐 관리자를 다시 시작한 후 스크립트를 재생하십시오. 이제 이는 그룹 기반 큐 관리자이므로 해당 사용자 ID 규칙이 기본 그룹을 기반으로 저장되는 효과가 나타납니다.

**관련 개념**

OAM(Object Authority Manager)

379 페이지의 『AIX, Linux, and Windows에서 프린시펄 및 그룹』
프린시펄은 그룹에 속할 수 있습니다. 개인이 아닌 그룹에 자원 액세스를 부여하면 필요한 관리의 양을 줄일 수 있습니다. 액세스 제어 목록(ACL)은 그룹 및 사용자 ID 둘 모두를 기반으로 합니다.

**관련 참조**

qm.ini 파일의 Service 스탠자

## ALW AIX, Linux, and Windows 의 IBM MQ 오브젝트에 대한 액세스 제공

**setmqaut** 제어 명령, **SET AUTHREC** MQSC 명령 또는 **MQCMD_SET_AUTH_REC** PCF 명령을 사용하여 사용자 및 사용자 그룹에게 IBM MQ 오브젝트에 대한 액세스를 제공하십시오. IBM MQ Appliance 에서는 **SET AUTHREC** 명령만 사용할 수 있습니다.

**setmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 setmqaut의 내용을 참조하십시오.

**SET AUTHREC** MQSC 명령 및 해당 구문의 전체 정의에 대해서는 SET AUTHREC의 내용을 참조하십시오.

**MQCMD_SET_AUTH_REC** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 설정을 참조하십시오.

이 명령을 실행하려면 큐 관리자가 실행 중이어야 합니다. 프린시펄에 대한 액세스를 변경한 경우에는 변경사항은 OAM에 의해 즉시 반영됩니다.

사용자에게 오브젝트에 대한 액세스를 제공하려면 다음을 지정해야 합니다.

- 작업 중인 오브젝트를 소유하는 큐 관리자의 이름입니다. 큐 관리자의 이름을 지정하지 않는 경우에는 기본 큐 관리자가 가정됩니다.
- 오브젝트의 이름 및 유형(오브젝트를 고유하게 식별하기 위함). 이름을 프로파일로서 지정할 수 있습니다. 이는 오브젝트의 명시적인 이름이거나 와일드카드 문자를 포함하는 일반적인 이름입니다. 일반 프로파일에 대한 자세한 설명과 일반 프로파일에서 와일드카드 문자 사용에 대해서는 333 페이지의 『AIX, Linux, and Windows에서 OAM 일반 프로파일 사용』 의 내용을 참조하십시오.
- 권한이 적용되는 하나 이상의 프린시펄 및 그룹 이름입니다.

  사용자 ID에 공백이 포함되는 경우에는 이 명령을 사용할 때 이를 따옴표에 넣으십시오. Windows 시스템에서 사용자 ID를 도메인 이름으로 규정할 수 있습니다. 실제 사용자 ID에 at 기호(@) 기호가 포함되는 경우 이를 @@로 바꿔서 이것이 사용자 ID와 도메인 이름 간의 구분 기호가 아니라 사용자 ID의 일부분임을 표시하십시오.
- 권한 부여 목록입니다. 목록의 각 항목은 해당 오브젝트에 권한 부여되거나 권한 취소되는 액세스 유형을 지정합니다. 목록의 각 권한 부여는 더하기 부호(+) 또는 빼기 부호(-)가 접두부로 추가되는 키워드로 지정됩니다. 더하기 부호를 사용하여 지정된 권한 부여를 추가하고, 빼기 부호를 사용하여 권한 부여를 제거하십시오. + 또는 - 부호와 키워드 사이에는 공백이 없어야 합니다.

  단일 명령에 여러 권한 부여를 지정할 수 있습니다. 예를 들어, 사용자 또는 그룹이 큐에 메시지를 넣고 이들을 찾아볼 수 있지만 메시지 가져오기에 대한 액세스를 취소하려면 권한 부여 목록은 다음과 같습니다.

```
+browse -get +put
```

### setmqaut 명령 사용 예

다음 예는 오브젝트를 사용하기 위한 권한을 부여하고 취소하기 위해 setmqaut 명령을 사용하는 방법을 보여줍니다.

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

이 예제에서

- saturn.queue.manager는 큐 관리자 이름입니다.
- queue는 오브젝트 유형입니다.
- RED.LOCAL.QUEUE는 오브젝트 이름입니다.
- groupa는 변경할 권한 부여가 있는 그룹의 ID입니다.
- +browse -get +put는 지정된 큐의 권한 부여 목록입니다.

- +browse는 큐에서 메시지를 찾아보기 위한 권한 부여를 추가합니다(찾아보기 옵션과 함께 **MQGET**를 실행하기 위해)
- -get은 큐에서 메시지를 가져오기(**MQGET**) 위한 권한 부여를 제거합니다.
- +put은 메시지를 큐에 넣기(**MQPUT**) 위한 권한 부여를 추가합니다.

다음 명령은 프린시펄 fvuser 및 그룹 groupa 및 groupb로부터 큐 MyQueue에 넣기 권한을 취소합니다. AIX and Linux 시스템에서 이 명령은 또한 fvuser와 같은 기본 그룹에서 모든 프린시펄의 넣기 권한을 취소합니다.

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

### setmqaut 명령을 여러 권한 서비스와 함께 사용

OAM 대신 본인의 권한 서비스를 사용 중인 경우, **setmqaut** 명령에서 이 서비스 이름을 지정하여 이 서비스로 명령을 전달할 수 있습니다. 동시에 실행 중인 여러 설치 가능한 컴포넌트가 있는 경우에는 이 매개변수를 지정해야 합니다. 그렇지 않으면 권한 서비스의 가장 먼저 설치 가능한 컴포넌트에 업데이트가 작성됩니다. 기본적으로 이는 제공된 OAM입니다.

### SET AUTHREC의 사용 참고사항

추가할 권한 목록과 제거할 권한 목록이 겹치면 안됩니다. 예를 들어 같은 명령을 사용하여 표시 권한을 추가하고 표시 권한을 제거할 수는 없습니다. 다른 옵션을 사용하여 권한을 표시한 경우에도 이 규칙이 적용됩니다. 예를 들어 다음 명령은 DSP 권한이 ALLADM 권한과 겹치기 때문에 실패합니다.

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

이 겹치기 동작이 적용되지 않는 경우는 ALL 권한입니다. 다음 명령은 먼저 ALL 권한을 추가한 다음 SETID 권한을 제거합니다.

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

다음 명령은 먼저 ALL 권한을 제거한 다음 DSP 권한을 추가합니다.

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

명령에 제공되는 순서에 관계없이 ALL은 가장 먼저 처리됩니다.

## ALW AIX, Linux, and Windows에서 OAM 일반 프로파일 사용

작성 시 각 개별 오브젝트에 대해 별도의 **setmqaut** 명령 또는 **SET AUTHREC** 명령을 실행하지 않고 단일 조작으로 여러 오브젝트에 대한 사용자의 권한을 설정하려면 OAM 일반 프로파일을 사용하십시오. IBM MQ Appliance 에서는 **SET AUTHREC** 명령만 사용할 수 있습니다.

setmqaut 또는 SET AUTHREC 명령에서 일반 프로파일을 사용하면 해당 프로파일에 맞는 모든 오브젝트에 일반 권한을 설정할 수 있습니다.

이 주제 모음에서는 일반 프로파일의 사용을 보다 자세히 설명합니다.

### OAM 프로파일에서 와일드카드 문자 사용

프로파일을 일반으로 작성하는 방법은 프로파일 이름에 특수 문자(와일드카드 문자)를 사용하는 것입니다. 예를 들어 물음표(?) 와일드카드 문자는 이름에서 단일 문자를 일치시킵니다. 따라서 ABC.?EF를 지정하는 경우 해당 프로파일에 제공하는 권한 부여는 이름 ABC.DEF, ABC.CEF, ABC.BEF 등등이 있는 오브젝트에 적용됩니다.

사용 가능한 와일드카드 문자는 다음과 같습니다.

**?**
단일 문자 대신 물음표(?) 사용. 예를 들어, AB.?D는 오브젝트 AB.CD, AB.ED 및 AB.FD에 적용됩니다.

*****

별표(*)를 다음과 같이 사용하십시오.

- 오브젝트 이름에 있는 규정자와 일치하는 프로파일 이름의 규정자. 규정자는 마침표로 구분되는 오브젝트 이름의 한 부분입니다. 예를 들어, ABC.DEF.GHI에서 규정자는 ABC, DEF 및 GHI입니다.

  예를 들어, ABC.*.JKL은 오브젝트 ABC.DEF.JKL 및 ABC.GHI.JKL에 적용됩니다. (이것은 ABC.JKL에 적용되지 **않습니다**. 이 컨텍스트에 사용되는 *는 항상 하나의 규정자를 표시합니다.)
- 오브젝트 이름에 있는 규정자 내에서 0개 또는 그 이상의 문자와 일치하는 프로파일 이름의 규정자 문자.

  예를 들어, ABC.DE*.JKL는 오브젝트 ABC.DE.JKL, ABC.DEF.JKL 및 ABC.DEGH.JKL에 적용됩니다.

******

다음과 같이 프로파일 이름에서는 이중 별표(**)를 **한 번만** 사용하십시오.

- 모든 오브젝트 이름과 일치하는 전체 프로파일 이름. 예를 들어, -t prcs 를 사용하여 프로세스를 식별한 후 프로파일 이름으로 ** 를 사용하는 경우 모든 프로세스에 대한 권한을 변경합니다.
- 오브젝트 이름에 있는 0개 또는 그 이상의 규정자와 일치하는 프로파일 이름의 시작, 중간 또는 종료 규정자로서 사용됩니다. 예를 들어, **.ABC는 모든 오브젝트를 최종 규정자 ABC로 식별합니다.

완전한 규정자로 이중 별표 ** 만 사용할 수 있습니다.

```
**.DEF
  ABC.**
  A*.**
```

그러나 다음과 같이

```
A**
```

그렇지 않으면 AMQ7226E: 프로파일 이름이 올바르지 않습니다. 라는 메시지를 수신합니다.

**참고:** AIX and Linux 시스템에서 와일드카드 문자를 사용할 때 프로파일 이름을 작은따옴표에 **묶어야 합니다**.

## 프로파일 우선순위

일반 프로파일을 사용할 경우 이해해야 하는 중요한 점은 작성 중인 오브젝트에 적용할 권한을 결정할 때 프로파일에 지정되는 우선순위입니다. 예를 들어, 다음 명령을 발행한 것으로 가정해 보십시오.

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

첫 번째는 프로파일 AB. *; 와 일치하는 이름을 가진 프린시펄 fred에 대한 모든 큐에 넣기 권한을 부여합니다. 두 번째는 AB.C*

이제 AB.CD 큐를 작성한다고 가정합니다. 와일드카드 일치 규칙에 따라 각각 setmqaut가 해당 큐에 적용될 수 있습니다. 따라서, 넣기 또는 가져오기 권한이 있습니까?

응답을 찾기 위해 여러 프로파일을 오브젝트에 적용할 수 있을 때마다 **가장 특정되게 적용되는** 규칙만 적용합니다. 이 규칙을 적용하는 방법은 프로파일 이름을 왼쪽에서 오른쪽으로 비교하는 것입니다. 다를 경우, 일반 문자가 아닌 문자가 일반 문자보다 더 고유한 문자입니다. 따라서, 이 예에서 큐 AB.CD에는 **가져오기** 권한(AB.C*가 AB.*보다 더 특정적임)이 있습니다.

일반 문자를 비교할 때 특이성의 순서는 다음과 같습니다.

1. ?
2. *
3. **

## 프로파일 설정 버리기

**dmpmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 dmpmqaut의 내용을 참조하십시오.

**DISPLAY AUTHREC** MQSC 명령 및 해당 구문의 전체 정의에 대해서는 DISPLAY AUTHREC의 내용을 참조하십시오.

**MQCMD_INQUIRE_AUTH_RECS** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 설정을 참조하십시오.

다음 예는 일반 프로파일의 권한 레코드를 버리기 위해 **dmpmqaut** 제어 명령을 사용하는 방법을 보여줍니다.

1. 이 예는 프린시펄 user1에 대해 큐 a.b.c와 일치하는 프로파일이 있는 모든 권한 레코드를 버립니다.

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

   버리기 결과는 다음과 같습니다.

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

   **참고:** AIX and Linux 의 사용자는 **dmpmqaut** 명령에 -p 옵션을 사용할 수 있지만 권한을 정의할 때 대신 -g groupname 를 사용해야 합니다.

2. 이 예는 큐 a.b.c와 일치하는 프로파일이 있는 모든 권한 레코드를 버립니다.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

   버리기 결과는 다음과 같습니다.

```
profile:     a.b.c
object type: queue
entity:      Administrator
type:        principal
authority:   all
- - - - - - - - - - - - - - - -
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
- - - - - - - - - - - - - - - -
profile:     a.**
object type: queue
entity:      group1
type:        group
authority:   get
```

3. 이 예제는 a.b. * 프로파일에 대한 모든 권한 레코드를 덤프합니다. 표시합니다.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

   버리기 결과는 다음과 같습니다.

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 이 예는 큐 관리자 qmX의 모든 권한 레코드를 버립니다.

```
dmpmqaut -m qmX
```

   버리기 결과는 다음과 같습니다.

```
profile:     q1
object type: queue
entity:      Administrator
type:        principal
authority:   all
- - - - - - - - - - - - - - - -
profile:     q*
object type: queue
entity:      user1
type:        principal
authority:   get, browse
- - - - - - - - - - - - - - - -
profile:     name.*
object type: namelist
entity:      user2
type:        principal
authority:   get
- - - - - - - - - - - - - - - -
profile:     pr1
object type: process
entity:      group1
type:        group
authority:   get
```

5. 이 예는 큐 관리자 qmX의 모든 프로파일 이름 및 오브젝트 유형을 버립니다.

```
dmpmqaut -m qmX -l
```

버리기 결과는 다음과 같습니다.

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

**참고:** IBM MQ for Windows의 경우에만 표시된 모든 프린시펄에는 도메인 정보가 포함됩니다. 예:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

## ALW *AIX, Linux, and Windows*에서 *OAM* 프로파일에 와일드카드 문자 사용

오브젝트 권한 관리자(OAM) 프로파일 이름에서 와일드카드 문자를 사용하여 해당 프로파일을 둘 이상의 오브젝트에 적용할 수 있게 만드십시오.

프로파일을 일반으로 작성하는 방법은 프로파일 이름에 특수 문자(와일드카드 문자)를 사용하는 것입니다. 예를 들어 물음표(?) 와일드카드 문자는 이름에서 단일 문자를 일치시킵니다. 따라서 ABC.?EF를 지정하는 경우 해당 프로파일에 제공하는 권한 부여는 이름 ABC.DEF, ABC.CEF, ABC.BEF 등등이 있는 오브젝트에 적용됩니다.

사용 가능한 와일드카드 문자는 다음과 같습니다.

**?**
단일 문자 대신 물음표(?) 사용. 예를 들어, AB.?D는 오브젝트 AB.CD, AB.ED 및 AB.FD에 적용됩니다.

*****
별표(*)를 다음과 같이 사용하십시오.

- 오브젝트 이름에 있는 규정자와 일치하는 프로파일 이름의 규정자. 규정자는 마침표로 구분되는 오브젝트 이름의 한 부분입니다. 예를 들어, ABC.DEF.GHI에서 규정자는 ABC, DEF 및 GHI입니다.

  예를 들어, ABC.*.JKL은 오브젝트 ABC.DEF.JKL 및 ABC.GHI.JKL에 적용됩니다. (이것은 ABC.JKL에 적용되지 **않습니다**. 이 컨텍스트에 사용되는 *는 항상 하나의 규정자를 표시합니다.)
- 오브젝트 이름에 있는 규정자 내에서 0개 또는 그 이상의 문자와 일치하는 프로파일 이름의 규정자 문자.

  예를 들어, ABC.DE*.JKL는 오브젝트 ABC.DE.JKL, ABC.DEF.JKL 및 ABC.DEGH.JKL에 적용됩니다.

**

다음과 같이 프로파일 이름에서는 이중 별표(**)를 **한 번만** 사용하십시오.

- 모든 오브젝트 이름과 일치하는 전체 프로파일 이름. 예를 들어, `-t prcs` 를 사용하여 프로세스를 식별한 후 프로파일 이름으로 ** 를 사용하는 경우 모든 프로세스에 대한 권한을 변경합니다.
- 오브젝트 이름에 있는 0개 또는 그 이상의 규정자와 일치하는 프로파일 이름의 시작, 중간 또는 종료 규정자로서 사용됩니다. 예를 들어, `**.ABC`는 모든 오브젝트를 최종 규정자 ABC로 식별합니다.

**참고:** AIX and Linux 시스템에서 와일드카드 문자를 사용할 때 프로파일 이름을 작은따옴표에 **묶어야 합니다**.

## ALW *AIX, Linux, and Windows*에서 프로파일 우선순위

둘 이상의 일반 프로파일을 단일 오브젝트에 적용할 수 있습니다. 이 경우이면 가장 특정 규칙이 적용됩니다.

일반 프로파일을 사용할 경우 이해해야 하는 중요한 점은 작성 중인 오브젝트에 적용할 권한을 결정할 때 프로파일에 지정되는 우선순위입니다. 예를 들어, 다음 명령을 발행한 것으로 가정해 보십시오.

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

첫 번째는 프로파일 AB. *; 와 일치하는 이름을 가진 프린시펄 fred에 대한 모든 큐에 넣기 권한을 부여합니다. 두 번째는 AB.C*

이제 AB.CD 큐를 작성한다고 가정합니다. 와일드카드 일치 규칙에 따라 각각 setmqaut가 해당 큐에 적용될 수 있습니다. 따라서, 넣기 또는 가져오기 권한이 있습니까?

응답을 찾기 위해 여러 프로파일을 오브젝트에 적용할 수 있을 때마다 **가장 특정되게 적용되는** 규칙만 적용합니다. 이 규칙을 적용하는 방법은 프로파일 이름을 왼쪽에서 오른쪽으로 비교하는 것입니다. 다를 경우, 일반 문자가 아닌 문자가 일반 문자보다 더 고유한 문자입니다. 따라서, 이 예에서 큐 AB.CD에는 **가져오기** 권한(AB.C*가 AB.*보다 더 특정적임)이 있습니다.

일반 문자를 비교할 때 특이성의 순서는 다음과 같습니다.

1. ?
2. *
3. **

이 MQSC 명령을 사용할 때 동등한 정보에 대해서는 SET AUTHREC의 내용을 참조하십시오.

## ALW *AIX, Linux, and Windows*에서 프로파일 설정 덤프

**dmpmqaut** 제어 명령, **DISPLAY AUTHREC** MQSC 명령 또는 **MQCMD_INQUIRE_AUTH_RECS** PCF 명령을 사용하여 지정된 프로파일과 연관된 현재 권한을 덤프할 수 있습니다. IBM MQ Appliance 에서는 **DISPLAY AUTHREC** 명령만 사용할 수 있습니다.

**dmpmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 dmpmqaut의 내용을 참조하십시오.

**DISPLAY AUTHREC** MQSC 명령 및 해당 구문의 전체 정의에 대해서는 DISPLAY AUTHREC의 내용을 참조하십시오.

**MQCMD_INQUIRE_AUTH_RECS** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 설정을 참조하십시오.

다음 예는 일반 프로파일의 권한 레코드를 버리기 위해 **dmpmqaut** 제어 명령을 사용하는 방법을 보여줍니다.

1. 이 예는 프린시펄 user1에 대해 큐 a.b.c와 일치하는 프로파일이 있는 모든 권한 레코드를 버립니다.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   버리기 결과는 다음과 같습니다.

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   ```

```
type:        principal
authority:   get, browse, put, inq
```

**참고:** AIX and Linux 사용자는 -p 옵션을 사용할 수 없습니다. 이들은 대신 -g groupname을 사용해야 합니다.

2. 이 예는 큐 a.b.c와 일치하는 프로파일이 있는 모든 권한 레코드를 버립니다.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

버리기 결과는 다음과 같습니다.

```
profile:     a.b.c
object type: queue
entity:      Administrator
type:        principal
authority:   all
- - - - - - - - - - - - - - - -
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
- - - - - - - - - - - - - - - -
profile:     a.**
object type: queue
entity:      group1
type:        group
authority:   get
```

3. 이 예제는 a.b. * 프로파일에 대한 모든 권한 레코드를 덤프합니다. 표시합니다.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

버리기 결과는 다음과 같습니다.

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 이 예는 큐 관리자 qmX의 모든 권한 레코드를 버립니다.

```
dmpmqaut -m qmX
```

버리기 결과는 다음과 같습니다.

```
profile:     q1
object type: queue
entity:      Administrator
type:        principal
authority:   all
- - - - - - - - - - - - - - - -
profile:     q*
object type: queue
entity:      user1
type:        principal
authority:   get, browse
- - - - - - - - - - - - - - - -
profile:     name.*
object type: namelist
entity:      user2
type:        principal
authority:   get
- - - - - - - - - - - - - - - -
profile:     pr1
object type: process
entity:      group1
```

```
type:         group
authority:    get
```

5. 이 예는 큐 관리자 qmX의 모든 프로파일 이름 및 오브젝트 유형을 버립니다.

```
dmpmqaut -m qmX -l
```

버리기 결과는 다음과 같습니다.

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

**참고:** IBM MQ for Windows의 경우에만 표시된 모든 프린시펄에는 도메인 정보가 포함됩니다. 예:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

## ALW AIX, Linux, and Windows에서 액세스 설정 표시

특정 프린시펄 또는 그룹이 특정 오브젝트에 대해 갖는 권한을 보려면 **dspmqaut** 제어 명령, **DISPLAY AUTHREC** MQSC 명령 또는 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 명령을 사용하십시오. IBM MQ 어플라이언스에서는 **DISPLAY AUTHREC** 명령만을 사용할 수 있음을 유의하십시오.

이 명령을 실행하려면 큐 관리자가 실행 중이어야 합니다. 프린시펄에 대한 액세스를 변경하면 변경사항은 OAM에 의해 즉시 반영됩니다. 권한 부여는 한 번에 하나의 그룹이나 프린시펄에만 표시될 수 있습니다.

**dmpmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 dmpmqaut의 내용을 참조하십시오.

**DISPLAY AUTHREC** MQSC 명령 및 해당 구문의 전체 정의에 대해서는 DISPLAY AUTHREC의 내용을 참조하십시오.

**MQCMD_INQUIRE_AUTH_RECS** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 설정을 참조하십시오.

다음 예는 **dspmqaut** 제어 명령을 사용하여 GpAdmin 그룹이 QueueMan1 큐 관리자에 있는 이름이 Annuities인 프로세스 정의에 가지고 있는 권한 부여를 표시하는 방법을 보여줍니다.

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

## ALW AIX, Linux, and Windows 에서 IBM MQ 오브젝트에 대한 액세스 변경 및 취소

사용자 또는 그룹이 오브젝트에 대해 가지고 있는 액세스 레벨을 변경하려면 **setmqaut** 제어 명령, **DELETE AUTHREC** MQSC 명령 또는 **MQCMD_DELETE_AUTH_REC** PCF 명령을 사용하십시오. MQ Appliance IBM MQ 어플라이언스에서는 **DELETE AUTHREC** 명령만 사용할 수 있습니다.

사용자를 그룹에서 제거하는 프로세스는 다음에서 설명됩니다.

- Windows 137 페이지의 『Windows에서 그룹 작성 및 관리』
- AIX 136 페이지의 『AIX에서 그룹 작성 및 관리』
- Linux 136 페이지의 『Linux에서 그룹 작성 및 관리』

.

IBM MQ 오브젝트를 작성하는 사용자 ID에는 해당 오브젝트에 대한 전체 제어 권한이 부여됩니다. 로컬 mqm 그룹(또는 Windows 시스템의 관리자)에서 이 사용자 ID를 제거하는 경우 이러한 권한은 취소되지 않습니다. **setmqaut** 제어 명령 또는 **MQCMD_DELETE_AUTH_REC** PCF 명령을 사용하여 오브젝트를 mqm 또는 관리자 그룹에서 제거한 후에 이를 작성한 사용자 ID의 오브젝트에 대한 액세스를 취소하십시오.

setmqaut 제어 명령 및 해당 구문의 전체 정의에 대해서는 setmqaut의 내용을 참조하십시오.

**DELETE AUTHREC** MQSC 명령 및 해당 구문의 전체 정의에 대해서는 DELETE AUTHREC의 내용을 참조하십시오.

**MQCMD_DELETE_AUTH_REC** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 설정을 참조하십시오.

Windows Windows의 경우 IBM MQ 8.0에서 **setmqaut**의 **-u** *SID* 매개변수를 사용하여 언제든지 특정 Windows 사용자 계정에 해당하는 OAM 항목을 삭제할 수 있습니다.

IBM MQ 8.0 이전에는 사용자 프로파일을 삭제하기 전에 특정 Windows 사용자 계정에 해당하는 OAM 항목을 삭제했어야 했습니다. 사용자 계정을 제거한 후 OAM 항목을 제거하는 것이 불가능했습니다.

### ALW AIX, Linux, and Windows 시스템에서 보안 액세스 검사 방지

참고: 이 주제에서는 사용하도록 권장되지 않는 기능에 대해 설명합니다. 보안 검사를 끄기 위해 오브젝트 권한 관리자 (OAM) 를 사용 안함으로 설정할 수 있습니다. 이는 테스트 환경에 적합할 수 있습니다. 사용 안함으로 설정되면 큐 관리자가 더 이상 권한 부여 또는 연결 인증 검사를 수행할 수 없습니다. TLS, 채널 인증 레코드 및 보안 엑시트를 계속 사용할 수 있습니다. OAM을 사용 안함으로 설정하거나 제거하면 OAM을 기존 큐 관리자에 추가할 수 없습니다.

보안 검사를 수행하지 않기로 결정한 경우(예를 들어, 테스트 환경에서) 다음 두 방법 중 하나를 사용하여 OAM을 사용 안함으로 설정할 수 있습니다.

- 큐 관리자를 작성하기 전에 운영 체제 환경 변수 **MQSNOAUT**를 설정하십시오.

  **MQSNOAUT** 환경 변수 설정의 영향 및 AIX, Linux, and Windows에서 **MQSNOAUT** 를 설정하는 방법에 대한 정보는 환경 변수 설명을 참조하십시오.
- 서비스를 제거하려면 큐 관리자 구성 파일을 편집하십시오.

**경고:** OAM이 제거되면 이를 기존 큐 관리자에 다시 넣을 수 없습니다. 이는 OAM은 오브젝트 작성 시에 배치되어야 하기 때문입니다. IBM MQ OAM을 제거한 후 다시 사용하려면 큐 관리자를 다시 빌드해야 합니다.

OAM이 사용 불가능한 상태에서 **setmqaut** 또는 **dspmqaut** 명령을 사용하는 경우 다음 사항에 유의하십시오.

- OAM은 지정된 프린시펄 또는 그룹을 유효성 검증하지 않습니다. 즉, 명령이 올바르지 않은 값을 승인할 수 있음을 의미합니다.
- OAM은 보안 검사를 수행하지 않고 모든 프린시펄 및 그룹이 모든 적용 가능한 오브젝트 조작을 수행할 권한이 있음을 나타냅니다.
- 인증 검사를 위해 OAM에 전달된 신임 정보는 유효성 검증되지 않습니다.

**관련 개념**

AIX, Linux, and Windows용 설치 가능 서비스 및 컴포넌트

**관련 태스크**

설치 가능 서비스 구성

**관련 참조**

설치 가능 서비스 참조 정보

## 자원에 대한 필수 액세스 부여

IBM MQ 시스템에 보안을 적용하기 위해 수행해야 하는 태스크를 판별하려면 이 토픽을 사용하십시오.

## 이 태스크 정보

이 태스크 동안에 적합한 보안 레벨을 IBM MQ 설치의 요소에 적용하기 위해 어떤 조치가 필요한지를 결정하십시오. 언급된 각 개별 태스크는 모든 플랫폼에 대한 단계별 지시사항을 제공합니다.

## 프로시저

1. 큐 관리자에 대한 액세스를 특정 사용자로 제한해야 합니까?
   a) 아니오: 추가 조치를 취하지 마십시오.
   b) 예: 다음 질문으로 이동하십시오.
2. 이러한 사용자가 큐 관리자 자원의 서브세트에서 부분 관리 액세스를 필요로 합니까?
   a) 아니오: 다음 질문으로 이동하십시오.
   b) 예: 341 페이지의 『큐 관리자 자원의 서브세트에서 부분 관리 액세스 부여』의 내용을 참조하십시오.
3. 이러한 사용자가 큐 관리자 자원의 서브세트에서 전체 관리 액세스를 필요로 합니까?
   a) 아니오: 다음 질문으로 이동하십시오.
   b) 예: 349 페이지의 『큐 관리자 자원의 서브세트에서 전체 관리 액세스 부여』의 내용을 참조하십시오.
4. 이러한 사용자가 모든 큐 관리자 자원에 대한 읽기 전용 액세스를 필요로 합니까?
   a) 아니오: 다음 질문으로 이동하십시오.
   b) 예: 355 페이지의 『큐 관리자에서 모든 자원에 읽기 전용 액세스 부여』의 내용을 참조하십시오.
5. 이러한 사용자가 모든 큐 관리자 자원에서 전체 관리 액세스를 필요로 합니까?
   a) 아니오: 다음 질문으로 이동하십시오.
   b) 예: 356 페이지의 『큐 관리자에서 모든 자원에 전체 관리 액세스 부여』의 내용을 참조하십시오.
6. 사용자 애플리케이션을 큐 관리자에 연결해야 합니까?
   a) 아니오: 357 페이지의 『큐 관리자에 대한 연결성 제거』에 설명한 대로 연결성을 사용 안함으로 설정하십시오.
   b) 예: 358 페이지의 『사용자 애플리케이션이 큐 관리자에 연결할 수 있도록 허용』의 내용을 참조하십시오.

## Multi z/OS 큐 관리자 자원의 서브세트에서 부분 관리 액세스 부여

특정 사용자에게 큐 관리자 자원의 전체가 아닌 일부에 대한 부분 관리 액세스를 제공해야 합니다. 수행해야 하는 조치를 판별하려면 이 테이블을 사용하십시오.

표 72. 큐 관리자 자원의 서브세트에 부분 관리 액세스 부여

| 사용자가 관리해야 하는 오브젝트 유형 | 수행 조치 |
|---|---|
| 큐 | 342 페이지의 『일부 큐에 제한된 관리 액세스 부여』에 설명된 대로 필수 큐에 부분 관리 액세스를 부여하십시오. |
| 토픽 | 343 페이지의 『일부 토픽에 제한된 관리 액세스 부여』에 설명된 대로 필수 토픽에 부분 관리 액세스를 부여하십시오. |
| 채널 | 344 페이지의 『일부 채널에 제한된 관리 액세스 부여』에 설명된 대로 필수 채널에 부분 관리 액세스를 부여하십시오. |
| 큐 관리자 | 345 페이지의 『큐 관리자에 제한된 관리 액세스 부여』에 설명된 대로 큐 관리자에 부분 관리 액세스를 부여하십시오. |

| 표 72. 큐 관리자 자원의 서브세트에 부분 관리 액세스 부여 (계속) | |
|---|---|
| **사용자가 관리해야 하는 오브젝트 유형** | **수행 조치** |
| 프로세스 | 346 페이지의 『일부 프로세스에 제한된 관리 액세스 부여』에 설명된 대로 필수 프로세스에 부분 관리 액세스를 부여하십시오. |
| 이름 목록 | 347 페이지의 『일부 이름 목록에 제한된 관리 액세스 부여』에 설명된 대로 필수 이름 목록에 부분 관리 액세스를 부여하십시오. |
| 서비스 | 348 페이지의 『일부 서비스에 제한된 관리 액세스 부여』에 설명된 대로 필수 서비스에 부분 관리 액세스를 부여하십시오. |

## 일부 큐에 제한된 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 큐에 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 조치를 위해 일부 큐에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

**참고:** MQ Appliance IBM MQ Appliance에서는 **SET AUTHREC** 명령만 사용할 수 있습니다.

## 프로시저

- ALW

  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
  ```

- IBM i

  IBM i의 경우 다음 명령을 실행하십시오.

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS z/OS의 경우 ID 컨텍스트 또는 모든 컨텍스트를 전달하려면 다음 명령을 실행하십시오.

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  큐에 대해 사용자가 수행할 수 있는 MQSC 명령을 판별하려면 각 MQSC 명령에 대해 다음 명령을 실행하십시오.

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
  PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  사용자가 DISPLAY QUEUE 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
  PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- ALW AIX, Linux, and Windows 시스템에서 다음 권한의 결합입니다. +chg, +clr, +dlt, +dsp. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.
- IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMDLT, *ADMDSP. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.
- z/OS z/OS에서 값 ALTER, CLEAR, DELETE 또는 MOVE 중 하나입니다.

**참고:** 큐에 +crt를 간접적으로 부여하면 사용자 또는 그룹을 관리자로 만듭니다. 일부 큐에 제한된 관리 액세스를 부여하기 위해 +crt 권한을 사용하지 마십시오.

**QType**

DISPLAY 명령의 경우 값 QUEUE, QLOCAL, QALIAS, QMODEL, QREMOTE 또는 QCLUSTER의 중 하나입니다.

*ReqdAction*의 다른 값의 경우 값 QLOCAL, QALIAS, QMODEL 또는 QREMOTE 중 하나입니다.

## 일부 토픽에 제한된 관리 액세스 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 토픽에 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보
일부 조치를 위해 일부 토픽에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

이러한 명령은 지정된 토픽에 액세스를 부여합니다. 사용자가 토픽에서 어떤 MQSC 명령을 수행할 수 있는지를 판별하려면 각 MQSC 명령에 대해 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

사용자가 DISPLAY TOPIC 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- ALW AIX, Linux, and Windows 시스템에서 다음 권한의 결합입니다. +chg, +clr, +crt, +dlt, +dsp. +ctrl. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.
- IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.
- z/OS z/OS에서 값 ALTER, CLEAR, DEFINE, DELETE 또는 MOVE 중 하나입니다.

## 일부 채널에 제한된 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 채널에 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 조치를 위해 일부 채널에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- IBM i
  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS에서:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

이러한 명령은 지정된 채널에 액세스를 부여합니다. 사용자가 채널에서 어떤 MQSC 명령을 수행할 수 있는지를 판별하려면 각 MQSC 명령에 대해 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

사용자가 DISPLAY CHANNEL 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- ALW AIX, Linux, and Windows에서 다음 권한의 조합입니다. +chg, +clr, +crt, +dlt, +dsp. +ctrl, +ctrlx. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.
- IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.
- z/OS z/OS에서 값 ALTER, CLEAR, DEFINE, DELETE 또는 MOVE 중 하나입니다.

## 큐 관리자에 제한된 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에 대한 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 조치를 위해 큐 관리자에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- IBM i
  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS에서:

사용자가 큐 관리자에서 어떤 MQSC 명령을 수행할 수 있는지를 판별하려면 각 MQSC 명령에 대해 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

사용자가 DISPLAY QMGR 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- ALW AIX, Linux, and Windows에서 다음 권한의 결합입니다. +chg, +clr, +crt, +dlt, +dsp. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.

  +set가 MQI 권한이고 일반적으로 관리적인 것으로 간주되지 않더라도 큐 관리자에 +set를 부여하면 간접적으로 전체 관리 권한으로 이어질 수 있습니다. 일반 사용자 및 애플리케이션에 +set를 부여하지 마십시오.
- IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.

## 일부 프로세스에 제한된 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 프로세스에 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 조치를 위해 일부 프로세스에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows에서:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
  ```
- IBM i
  IBM i에서:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```
- z/OS z/OS에서:

```
RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

이러한 명령은 지정된 채널에 액세스를 부여합니다. 사용자가 채널에서 어떤 MQSC 명령을 수행할 수 있는지를 판별하려면 각 MQSC 명령에 대해 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

사용자가 DISPLAY PROCESS 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- ALW AIX, Linux, and Windows에서 다음 권한의 결합입니다. +chg, +clr, +crt, +dlt, +dsp. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.
- IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.
- z/OS z/OS에서 값 ALTER, CLEAR, DEFINE, DELETE 또는 MOVE 중 하나입니다.

## 일부 이름 목록에 제한된 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 이름 목록에 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 조치를 위해 일부 이룸 목록에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- IBM i
  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS에서:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

이들 명령은 지정된 이름 목록에 대한 액세스 권한을 부여합니다. 사용자가 이름 목록에서 수행할 수 있는 MQSC 명령을 판별하려면 각 MQSC 명령에 대해 다음 명령을 발행하십시오.

```
RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

사용자가 DISPLAY NAMELIST 명령을 사용할 수 있게 하려면 다음 명령을 발행하십시오.

```
RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- ALW AIX, Linux, and Windows에서 다음 권한의 결합입니다. +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.
- IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.
- z/OS z/OS에서 값 ALTER, CLEAR, DEFINE, DELETE 또는 MOVE 중 하나입니다.

## 일부 서비스에 제한된 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 서비스에 부분 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 조치를 위해 일부 서비스에 제한된 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

z/OS z/OS에는 서비스 오브젝트가 존재하지 않습니다.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS에서:

  이러한 명령은 지정된 서비스에 액세스를 부여합니다. 사용자가 서비스에서 어떤 MQSC 명령을 수행할 수 있는지를 판별하려면 각 MQSC 명령에 대해 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  사용자가 DISPLAY SERVICE 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

  **ReqdAction**
  그룹이 수행할 수 있는 조치:

  - ALW AIX, Linux, and Windows 시스템에서 다음 권한의 결합입니다. +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.
  - IBM i IBM i에서 다음 권한의 결합입니다. *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. 권한 *ALLADM은 이러한 개별 권한 모두와 동등합니다.

## 큐 관리자 자원의 서브세트에서 전체 관리 액세스 부여

특정 사용자에게 큐 관리자 자원의 전체가 아닌 일부에 대한 전체 관리 액세스를 제공해야 합니다. 수행해야 하는 조치를 판별하려면 이 테이블을 사용하십시오.

표 73. 큐 관리자 자원의 서브세트에 전체 관리 액세스 부여

| 사용자가 관리해야 하는 오브젝트 유형 | 수행 조치 |
|---|---|
| 큐 | 350 페이지의 『일부 큐에 전체 관리 액세스 부여』에 설명된 대로 필수 큐에 전체 관리 액세스를 부여하십시오. |
| 토픽 | 350 페이지의 『일부 토픽에 전체 관리 액세스 부여』에 설명된 대로 필수 토픽에 전체 관리 액세스를 부여하십시오. |
| 채널 | 351 페이지의 『일부 채널에 전체 관리 액세스 부여』에 설명된 대로 필수 채널에 전체 관리 액세스를 부여하십시오. |
| 큐 관리자 | 352 페이지의 『큐 관리자에 전체 관리 액세스 부여』에 설명된 대로 큐 관리자에 전체 관리 액세스를 부여하십시오. |

| 표 73. 큐 관리자 자원의 서브세트에 전체 관리 액세스 부여 (계속) | |
|---|---|
| **사용자가 관리해야 하는 오브젝트 유형** | **수행 조치** |
| 프로세스 | 353 페이지의 『일부 프로세스에 전체 관리 액세스 부여』에 설명된 대로 필수 프로세스에 전체 관리 액세스를 부여하십시오. |
| 이름 목록 | 353 페이지의 『일부 이름 목록에 전체 관리 액세스 부여』에 설명된 대로 필수 이름 목록에 전체 관리 액세스를 부여하십시오. |
| 서비스 | 354 페이지의 『일부 서비스에 전체 관리 액세스 부여』에 설명된 대로 필수 서비스에 전체 관리 액세스를 부여하십시오. |

## 일부 큐에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 큐에 전체 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 큐에 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW

  AIX, Linux, and Windows에서:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
  ```

- IBM i

  IBM i에서:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  z/OS에서:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름.

  z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 일부 토픽에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 토픽에 전체 관리 액세스를 부여하십시오.

### 이 태스크 정보

일부 조치를 위해 일부 토픽에 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
```

- IBM i
  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS에서:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름.

  z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 일부 채널에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 채널에 전체 관리 액세스를 부여하십시오.

### 이 태스크 정보

일부 채널에 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- IBM i
  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS에서:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름.

  z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

### 큐 관리자에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에 대한 전체 관리 액세스를 부여하십시오.

### 이 태스크 정보

일부 큐 관리자에 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- IBM i
  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS에서:

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름.

  z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 일부 프로세스에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 프로세스에 전체 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 프로세스에 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW

  AIX, Linux, and Windows에서:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
  ```

- IBM i

  IBM i에서:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  z/OS에서:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름.

  z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 일부 이름 목록에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 이름 목록에 전체 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 이름 목록에 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW

  AIX, Linux, and Windows에서:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
  ```

- IBM i

  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

  z/OS에서:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## 일부 서비스에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 일부 서비스에 전체 관리 액세스를 부여하십시오.

## 이 태스크 정보

일부 서비스 전체 관리 액세스를 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW

  AIX, Linux, and Windows에서:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- IBM i

  IBM i에서:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

  z/OS에서:

```
RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## 큐 관리자에서 모든 자원에 읽기 전용 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 모든 자원에 읽기 전용 액세스를 부여하십시오.

### 이 태스크 정보

역할 기반 권한 추가 마법사 또는 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

권한 세부사항을 변경한 후 REFRESH SECURITY 명령을 사용하여 보안 새로 고치기를 수행하십시오.

### 프로시저

- 마법사 사용:
  a) IBM MQ Explorer 네비게이터 분할창에서 큐 관리자를 마우스의 오른쪽 단추로 클릭하고 **오브젝트 권한 > 역할 기반 권한 추가**를 클릭하십시오.
  역할 기반 권한 추가 마법사가 열립니다.
- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ** -t queue -g GroupName +browse +dsp
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get
+put
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

  SYSTEM.ADMIN.COMMAND.QUEUE 및 SYSTEM.MQEXPLORER.REPLY.MODEL 은 IBM MQ Explorer를 사용하려는 경우에만 필요합니다.
- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*Q) USER('GroupName') AUT(*ADMDSP *BROWSE) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*TOPIC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CHL) USER('GroupName') AUT(*ADMDSP *INQ) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CLTCN) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*AUTHINFO) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*LSR) USER('GroupName') AUT(*ADMDSP)MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*NMLIST) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*SVC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ('object-name') OBJTYPE(*MQM) USER('GroupName') AUT(*ADMDSP *CONNECT *INQ)
MQMNAME('QMgrName')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQQUEUE) ID(GroupName) ACCESS(READ)
RDEFINE MXTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MXTOPIC) ID(GroupName) ACCESS(READ)
RDEFINE MQPROC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

```
RDEFINE MQNLIST QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## 큐 관리자에서 모든 자원에 전체 관리 액세스 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 모든 자원에 전체 관리 액세스를 부여하십시오.

### 이 태스크 정보

역할 기반 권한 추가 마법사 또는 운영 체제에 적합한 명령을 사용할 수 있습니다.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

**참고:** ALW

1. **runmqsc** 를 사용하여 IBM MQ Explorer대신 큐 관리자를 관리하는 경우 SYSTEM.MQSC.REPLY.QUEUE를 사용하며 SYSTEM.MQEXPLORER.REPLY.MODEL 큐.
2. 큐 관리자의 모든 자원에 대한 사용자 액세스 권한을 부여하는 경우, 해당 사용자에게 qm.ini 파일에 대한 읽기 액세스 권한이 없으면 사용자가 실행할 수 없는 몇 가지 명령이 있습니다. qm.ini 파일을 읽을 수 있는 mqm 사용자가 아닌 사용자에 대한 제한사항 때문입니다.

   사용자가 qm.ini 파일에 대한 읽기 액세스 권한을 부여한 경우가 아니면 다음 명령을 실행할 수 없습니다.

   - TLS를 사용하도록 구성된 채널 정의
   - qm.ini에 정의된 자동 구성 삽입 변수를 사용하여 채널 정의

### 프로시저

- 마법사를 사용하는 경우 IBM MQ Explorer 네비게이터 분할창에서 큐 관리자를 마우스 오른쪽 단추로 클릭하고 **오브젝트 권한 > 역할 기반 권한 추가**를 클릭하십시오.
  역할 기반 권한 추가 마법사가 열립니다.
- Linux AIX
  AIX and Linux 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n '**' -t queue -g GroupName +alladm +browse
setmqaut -m QMgrName -n @class -t queue -g GroupName +crt
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get +put
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
```

```
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +connect
```

@class 에 대한 자세한 정보는 **setmqaut** 의 내용을 참조하십시오.

- Windows
  Windows 시스템의 경우 AIX and Linux 시스템에서와 같은 명령을 발행하지만 @class 대신에 프로파일 이름 @CLASS를 사용하십시오.
- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## 큐 관리자에 대한 연결성 제거

사용자 애플리케이션이 큐 관리자에 연결하기를 원치 않는 경우에는 이에 대한 연결 권한을 제거하십시오.

### 이 태스크 정보

사용자의 운영 체제에 적합한 명령을 사용하여 큐 관리자에 연결하기 위한 모든 사용자의 권한을 취소하십시오.

멀티플랫폼에서 DELETE AUTHREC 명령도 사용할 수 있습니다.

**참고:** IBM MQ 어플라이언스에서는 **DELETE AUTHREC** 명령만을 사용할 수 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
```

```
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
```

PERMIT 명령을 발행하지 마십시오.

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름.

z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**GroupName**
액세스를 거부할 그룹의 이름

## 사용자 애플리케이션이 큐 관리자에 연결할 수 있도록 허용

사용자 애플리케이션이 큐 관리자에 연결할 수 있도록 허용하고 싶습니다. 취해야 할 조치를 판별하려면 이 토픽의 테이블을 사용하십시오.

먼저 클라이언트 애플리케이션이 큐 관리자에 연결하는지 여부를 판별하십시오.

큐 관리자에 연결할 애플리케이션이 모두 클라이언트 애플리케이션이 아니면 365 페이지의 『큐 관리자에 대한 원격 액세스 사용 안함으로 설정』에 설명된 대로 원격 액세스를 사용 안함으로 설정하십시오.

큐 관리자에 연결할 애플리케이션의 하나 이상의 클라이언트 애플리케이션이면 358 페이지의 『큐 관리자에 대한 원격 연결성 보안』에 설명된 대로 원격 연결성을 보안 설정하십시오.

두 경우 모두 365 페이지의 『연결 보안 설정』에 설명된 대로 연결 보안을 설정하십시오.

큐 관리자에 연결 중인 각 사용자의 자원에 대한 액세스를 제어하려면 다음 테이블을 참조하십시오. 첫 번째 열의 명령문이 참이면 두 번째 열에 나열된 조치를 취하십시오.

| 명령문 | 수행 조치 |
|---|---|
| 큐를 사용하는 애플리케이션이 있습니다. | 366 페이지의 『큐에 대한 사용자 액세스 제어』 참조 |
| 토픽을 사용하는 애플리케이션이 있습니다. | 371 페이지의 『토픽에 대한 사용자 액세스 제어』의 내용을 참조하십시오. |
| 큐 관리자 오브젝트에서 조회한 애플리케이션이 있습니다. | 373 페이지의 『큐 관리자에서 조사하기 위한 권한 부여』의 내용을 참조하십시오. |
| 프로젝트 오브젝트를 사용하는 애플리케이션이 있습니다. | 373 페이지의 『프로세스에 액세스하기 위한 권한 부여』 참조 |
| 이름 목록을 사용하는 애플리케이션이 있습니다. | 374 페이지의 『이름 목록에 액세스하기 위한 권한 부여』 참조 |

## 큐 관리자에 대한 원격 연결성 보안

TLS, 보안 엑시트, 채널 인증 레코드 또는 이러한 메소드의 결합을 사용하여 큐 관리자에 대한 원격 연결성을 보안 설정할 수 있습니다.

### 이 태스크 정보

클라이언트 워크스테이션에서 클라이언트 연결 채널을 사용하거나 서버에서 서버 연결 채널을 사용하여 큐 관리자에 클라이언트를 연결합니다. 다음 방법 중 하나로 이러한 연결을 보안 설정하십시오.

### 프로시저

1. 채널 인증 레코드와 함께 TLS 사용:
   a) 모든 식별 이름(DN)을 USERSRC(NOACCESS)에 맵핑하려면 SSLPEERMAP 채널 인증 레코드를 사용하여 DN이 채널을 열지 못하도록 하십시오.
   b) 특정 DN 또는 DN 세트를 USERSRC(CHANNEL)에 맵핑하려면 SSLPEERMAP 채널 인증 레코드를 사용하여 채널을 열도록 허용하십시오.

2. 보안 엑시트와 함께 TLS 사용:
   a) 서버 연결 채널의 MCAUSER를 권한이 없는 사용자 ID로 설정하십시오.
   b) MQCD 구조에서 엑시트로 전달된 SSLPeerNamePtr 및 SSLPeerNameLength 필드에서 수신하는 TLS DN의 값에 따라 MCAUSER 값을 지정하도록 보안 엑시트를 작성하십시오.
3. 고정된 채널 정의 값과 함께 TLS 사용:
   a) 서버 연결 채널의 SSLPEER를 특정 값으로 설정하거나 값 범위를 줄이십시오.
   b) 서버 연결 채널의 MCAUSER를 채널을 실행해야 하는 사용자 ID로 설정하십시오.
4. TLS를 사용하지 않는 채널에서 채널 인증 레코드 사용:
   a) 주소 맵핑 채널 인증 레코드를 ADDRESS(*) 및 USERSRC(NOACCESS)와 함께 사용하여 IP 주소가 채널을 열지 못하도록 하십시오.
   b) 이러한 주소에 대한 주소 맵핑 채널 인증 레코드를 USERSRC(CHANNEL)와 함께 사용하여 특정 IP 주소가 채널을 열도록 허용하십시오.
5. 보안 엑시트 사용:
   a) 사용자가 선택하는 특성을 기반으로(예: 발신 IP 주소) 연결을 인증하려면 보안 엑시트를 작성하십시오.
6. 특정 상황에 필요한 경우 보안 엑시트와 함께 채널 인증 레코드를 사용하거나 세 메소드를 모두 사용할 수 있습니다.

특정 *IP* 주소 차단
채널 인증 레코드를 사용하여 특정 채널이 IP 주소의 인바운드 연결을 승인하지 않도록 하거나 전체 큐 관리자가 IP 주소의 액세스를 허용하지 않도록 할 수 있습니다.

## 시작하기 전에
다음 명령을 실행하여 채널 인증 레코드를 사용하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 이 태스크 정보
특정 채널에서 인바운드 연결을 승인하지 않고 올바른 채널 이름을 사용하는 경우에만 연결을 승인하도록 하려는 경우 IP 주소를 차단하도록 한 가지 규칙 유형을 사용할 수 있습니다. 전체 큐 관리자에 대한 IP 주소 액세스를 허용하지 않으려면 일반적으로 방화벽을 사용하여 이를 영구적으로 차단합니다. 그러나 다른 규칙 유형을 사용하여 예를 들어 방화벽 업데이트를 기다리는 동안 임시로 몇 개의 주소를 차단할 수 있습니다.

## 프로시저
- 특정 채널을 사용하지 않도록 IP 주소를 차단하려면 MQSC 명령 **SET CHLAUTH**, 또는 PCF 명령 **Set Channel Authentication Record**를 사용하여 채널 인증 레코드를 설정하십시오.

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

명령에 3개 부분이 있습니다.

**SET CHLAUTH (*generic-channel-name*)**
전체 큐 관리자, 단일 채널 또는 채널 범위의 연결을 차단하고 싶은지 여부를 제어하려면 이 명령 부분을 사용합니다. 여기에 넣는 것이 어떤 영역이 포함되는지를 판별합니다.

예를 들면, 다음과 같습니다.

- SET CHLAUTH('*') - 큐 관리자에서 모든 채널을 차단합니다. 즉, 전체 큐 관리자입니다.
- SET CHLAUTH('SYSTEM.*') - SYSTEM으로 시작하는 모든 채널을 차단합니다.
- SET CHLAUTH('SYSTEM.DEF.SVRCONN') - SYSTEM.DEF.SVRCONN 채널을 차단합니다.

**CHLAUTH 규칙 유형**
명령의 이 부분을 사용하여 명령의 유형을 지정하고, 단일 주소 또는 주소 목록을 제공할지 여부를 판별합니다.

예를 들면, 다음과 같습니다.

- TYPE(ADDRESSMAP) - 단일 주소 또는 와일드카드 주소를 제공하려면 ADDRESSMAP을 사용합니다. 예를 들어, ADDRESS('192.168.*')은 192.168로 시작하여 IP 주소에서 나오는 모든 연결을 차단합니다.

  패턴이 있는 IP 주소 필터링에 대한 자세한 정보는 일반 IP 주소를 참조하십시오.
- TYPE(BLOCKADDR) - 차단할 주소 목록을 제공하고 싶은 경우 BLOCKADDR을 사용합니다.

**추가 매개변수**

이러한 매개변수는 명령의 두 번째 부분에서 사용한 규칙의 유형에 의존합니다.

- TYPE(ADDRESSMAP)의 경우 ADDRESS를 사용합니다.
- TYPE(BLOCKADDR)의 경우 ADDRLIST를 사용합니다.

**관련 참조**

SET CHLAUTH

큐 관리자가 실행 중이 아닌 경우 특정 *IP* 주소를 일시적으로 차단

큐 관리자가 실행 중이 아니므로 MQSC 명령을 실행할 수 없을 때 특정 IP 주소 또는 주소 범위를 차단하고 싶을 수도 있습니다. blockaddr.ini 파일을 수정하여 예외적으로 IP 주소를 일시적으로 차단할 수 있습니다.

## 이 태스크 정보

blockaddr.ini 파일에는 큐 관리자가 사용하는 BLOCKADDR 정의의 사본이 포함됩니다. 이 파일은 리스너가 큐 관리자보다 먼저 시작된 경우 리스너가 판독합니다. 이러한 상황에서 리스너는 blockaddr.ini 파일에 수동으로 추가한 값을 사용합니다.

그러나 큐 관리자가 시작될 때 BLOCKADDR 정의 세트를 blockaddr.ini 파일에 기록하여 사용자가 수행했을 수 있는 모든 수동 편집을 겹쳐씁니다. 마찬가지로, **SET CHLAUTH** 명령을 사용하여 BLOCKADDR 정의를 추가하거나 삭제할 때마다 blockaddr.ini 파일이 업데이트됩니다. 그러므로 큐 관리자가 실행 중일 때 **SET CHLAUTH** 명령을 사용하는 방법으로만 BLOCKADDR 정의를 영구적으로 변경할 수 있습니다.

## 프로시저

1. 텍스트 편집기에서 blockaddr.ini 파일을 여십시오.

   파일은 큐 관리자의 데이터 디렉토리에 있습니다.
2. IP 주소를 단순 키워드-값 쌍으로서 추가하십시오. 여기서 키워드는 Addr입니다.

   패턴이 있는 IP 주소 필터링에 대한 자세한 정보는 일반 IP 주소를 참조하십시오.

   예를 들면, 다음과 같습니다.

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

**관련 태스크**

359 페이지의 『특정 IP 주소 차단』

채널 인증 레코드를 사용하여 특정 채널이 IP 주소의 인바운드 연결을 승인하지 않도록 하거나 전체 큐 관리자가 IP 주소의 액세스를 허용하지 않도록 할 수 있습니다.

**관련 참조**

SET CHLAUTH

특정 사용자 *ID* 차단

채널이 종료되도록 하는 사용자 ID가 확인된 경우 해당 사용자 ID를 지정하여 특정 사용자가 채널을 사용하지 못하게 할 수 있습니다. 채널 인증 레코드를 설정하여 이를 수행합니다.

## 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 프로시저

MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

TYPE(BLOCKUSER)에서 제공되는 사용자 목록만 SVRCONN 채널에 적용되고 큐 관리자 채널에 대해서는 큐 관리자가 적용되지 않습니다.

*userID1* 및 *userID2*는 채널을 사용하지 못하도록 할 각 사용자의 ID입니다. 특수 값 *MQADMIN을 지정하여 권한이 부여된 관리 사용자를 참조할 수도 있습니다. 권한이 있는 사용자에 대한 자세한 정보는 301 페이지의 『권한이 있는 사용자』의 내용을 참조하십시오. *MQADMIN에 대한 자세한 정보는 SET CHLAUTH를 참조하십시오.

**관련 참조**

SET CHLAUTH

*MCAUSER* 사용자 *ID*에 리모트 큐 관리자 맵핑

채널이 연결 중인 원래 큐 관리자에 따라 채널 인증 레코드를 사용하여 채널의 MCAUSER 속성을 설정할 수 있습니다.

## 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 이 태스크 정보

선택적으로 규칙이 적용되는 IP 주소를 제한할 수 있습니다.

이 기술은 서버 연결 채널에는 적용되지 않습니다. 다음 명령에서 서버-연결 채널의 이름을 지정하는 경우 이는 아무런 효과가 없습니다.

## 프로시저

- MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

  ```
  SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
  ) USERSRC(MAP) MCAUSER(user)
  ```

  *generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

  *generic-partner-qmgr-name*은 큐 관리자의 이름이거나, 큐 관리자 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

  *user*는 지정된 큐 관리자에서 모든 연결에 사용할 사용자 ID입니다.
- 특정 IP 주소로 이 명령을 제한하려면 다음과 같이 **ADDRESS** 매개변수를 포함시키십시오.

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user) ADDRESS(
generic-ip-address)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*generic-ip-address*는 단일 주소이거나, 와일드카드로서 별표(*)를 또는 주소와 일치하는 범위를 나타내는 하이픈(-)을 포함하는 패턴입니다. 일반 IP 주소에 대한 자세한 정보는 일반 IP 주소를 참조하십시오.

**관련 참조**

SET CHLAUTH

*MCAUSER* 사용자 *ID*에 클라이언트 사용자 *ID* 맵핑

클라이언트에서 수신한 사용자 ID에 따라 채널 인증 레코드를 사용하여 서버 연결 채널의 MCAUSER 속성을 변경할 수 있습니다.

## 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 이 태스크 정보

이 기술은 서버 연결 채널에만 적용됩니다. 다른 채널 유형에는 영향을 주지 않습니다.

## 프로시저

MQSC 명령 **SET CHLAUTH** 또는 PCF 명령 **Set Channel Authentication Record** 를 사용하여 채널 인증 레코드를 설정하시시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*client-user-name*은 클라이언트 애플리케이션과 연관된 사용자 ID입니다. 값은 클라이언트 애플리케이션에서 확인하거나, 조기 채택을 사용하는 연결 인증에서 변경하거나, 채널 엑시트를 통해 설정할 수 있습니다.

*user*는 클라이언트 사용자 이름 대신 사용될 사용자 ID입니다.

**관련 참조**

SET CHLAUTH

Channels 스탠자의 속성(ChlauthEarlyAdopt)

*MCAUSER* 사용자 *ID*에 *SSL* 또는 *TLS* 식별 이름 맵핑

수신한 식별 이름(DN)에 따라 채널 인증 레코드를 사용하여 채널의 MCAUSER 속성을 설정할 수 있습니다.

## 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 프로시저

MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*generic-ssl-peer-name*은 SSLPEER 값의 표준 IBM MQ 규칙을 따르는 문자열입니다. SSLPEER 값의 IBM MQ 규칙을 참조하십시오.

*user*는 지정된 DN을 사용하는 모든 연결에 사용할 사용자 ID입니다.

*generic-issuer-name*은 일치시킬 인증서의 발행인 DN을 가리킵니다. 이 매개변수는 선택적이지만 여러 인증 기관이 사용 중인 경우 잘못된 인증서와 가짜로 일치하는 것을 피하기 위해서 이를 사용해야 합니다.

**관련 참조**

SET CHLAUTH

리모트 큐 관리자로부터의 액세스 차단

채널 인증 레코드를 사용하여 리모트 큐 관리자가 채널을 시작하지 못하도록 할 수 있습니다.

## 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 이 태스크 정보

이 기술은 서버 연결 채널에는 적용되지 않습니다. 다음 명령에서 서버-연결 채널의 이름을 지정하는 경우 이는 아무런 효과가 없습니다.

## 프로시저

MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*generic-partner-qmgr-name*은 큐 관리자의 이름이거나, 큐 관리자 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

**관련 참조**

SET CHLAUTH

클라이언트 사용자 *ID*에 대한 액세스 차단

채널 인증 레코드를 사용하여 클라이언트 사용자 ID가 채널에 연결하지 못하도록 설정할 수 있습니다.

## 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 이 태스크 정보

이 기술은 서버 연결 채널에만 적용됩니다. 다른 채널 유형에는 영향을 주지 않습니다.

### 프로시저

MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*client-user-name*은 클라이언트 애플리케이션과 연관된 사용자 ID입니다. 값은 클라이언트 애플리케이션에서 확인하거나, 조기 채택을 사용하는 연결 인증에서 변경하거나, 채널 엑시트를 통해 설정할 수 있습니다.

**관련 참조**

SET CHLAUTH

*SSL* 또는 *TLS* 식별 이름의 액세스 차단

TLS 식별 이름(DN)이 채널을 시작하는 것을 막기 위해 채널 인증 레코드를 사용할 수 있습니다.

### 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

### 프로시저

MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*generic-ssl-peer-name*은 SSLPEER 값의 표준 IBM MQ 규칙을 따르는 문자열입니다. SSLPEER 값의 IBM MQ 규칙을 참조하십시오.

*generic-issuer-name*은 일치시킬 인증서의 발행인 DN을 가리킵니다. 이 매개변수는 선택적이지만 여러 인증 기관이 사용 중인 경우 잘못된 인증서와 가짜로 일치하는 것을 피하기 위해서 이를 사용해야 합니다.

**관련 참조**

SET CHLAUTH

*MCAUSER* 사용자 *ID*에 *IP* 주소 맵핑

연결을 수신한 IP 주소에 따라 채널 인증 레코드를 사용하여 채널의 MCAUSER 속성을 설정할 수 있습니다.

### 시작하기 전에

다음과 같이 채널 인증 레코드를 사용할 수 있는지 확인하십시오.

```
ALTER QMGR CHLAUTH(ENABLED)
```

### 프로시저

MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**을 사용하여 채널 인증 레코드를 설정하십시오. 예를 들어, 다음 MQSC 명령을 실행할 수 있습니다.

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.

*user*는 지정된 DN을 사용하는 모든 연결에 사용할 사용자 ID입니다.

*generic-ip-address*는 연결되는 주소이거나, 와일드카드로서 별표(*)를 또는 주소와 일치하는 범위를 나타내는 하이픈(-)을 포함하는 패턴입니다.

**관련 참조**

SET CHLAUTH

## 큐 관리자에 대한 원격 액세스 사용 안함으로 설정

클라이언트 애플리케이션이 큐 관리자에 연결하는 것을 원치 않으면 이에 대한 원격 액세스를 사용 안함으로 설정하십시오.

### 이 태스크 정보

다음 방법 중 하나를 사용하여 클라이언트 애플리케이션이 큐 관리자에 연결하는 것을 막으십시오.

### 프로시저

- MQSC 명령 **DELETE CHANNEL**을 사용하여 모든 서버 연결 채널을 삭제하십시오.
- MQSC 명령 **ALTER CHANNEL**을 사용하여 채널의 메시지 채널 에이전트 사용자 ID(MCAUSER)를 액세스 권한이 없는 사용자 ID로 설정하십시오.

## 연결 보안 설정

비즈니스 요구사항이 있는 각 사용자 또는 사용자 그룹에 큐 관리자에 연결하는 권한을 부여하십시오.

### 이 태스크 정보

연결 보안을 설정하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows에서:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```

- IBM i
  IBM i에서:

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```

- z/OS
  z/OS에서:

  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  ```

이러한 명령은 배치, CICS, IMS 및 채널 시작기(CHIN)에 연결하기 위한 권한을 제공합니다. 특정 연결 유형을 사용하지 않으면 관련 명령을 생략하십시오.

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**관련 개념**

187 페이지의 『Connection security profiles for the channel initiator』
Profiles for checking connections from the channel initiator are composed of the queue manager or queue sharing group name followed by the word *CHIN*. Give the user ID used by the channel initiator started task address space READ access to the connection profile.

## 큐에 대한 사용자 액세스 제어

큐에 대한 애플리케이션 액세스를 제어하고 싶습니다. 취할 조치를 판별하려면 이 토픽을 사용하십시오.

첫 번째 열의 각 신뢰 명령문에 대해 두 번째 열에 표시된 조치를 취하십시오.

| 명령문 | Action |
|---|---|
| 애플리케이션은 큐로부터 메시지를 받습니다. | 366 페이지의 『큐로부터 메시지를 가져오는 권한 부여』 참조 |
| 애플리케이션이 컨텍스트를 설정합니다. | 367 페이지의 『컨텍스트를 설정하기 위한 권한 부여』 참조 |
| 애플리케이션이 컨텍스트를 전달합니다. | 368 페이지의 『컨텍스트를 전달하기 위한 권한 부여』 참조 |
| 애플리케이션이 메시지를 클러스터된 큐에 넣습니다. | 446 페이지의 『리모트 클러스터 큐에 메시지를 넣는 권한 부여』 참조 |
| 애플리케이션이 메시지를 로컬 큐에 넣습니다. | 369 페이지의 『메시지를 로컬 큐에 넣기 위한 권한 부여』 참조 |
| 애플리케이션이 메시지를 모델 큐에 넣습니다. | 369 페이지의 『메시지를 모델 큐에 넣기 위한 권한 부여』 참조 |
| 애플리케이션이 메시지를 리모트 큐에 넣습니다. | 370 페이지의 『메시지를 리모트 클러스터 큐에 넣기 위한 권한 부여』 참조 |

### 큐로부터 메시지를 가져오는 권한 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 큐 또는 큐 세트로부터 메시지를 가져오는 권한을 부여하십시오.

#### 이 태스크 정보

일부 큐로부터 메시지를 가져오는 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

#### 프로시저

- Windows
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

컨텍스트를 설정하기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 넣고 있는 메시지에 컨텍스트를 설정하기 위한 권한을 부여하십시오.

## 이 태스크 정보

일부 큐에 컨텍스트를 설정하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령 중 하나를 실행하십시오.
  - ID 컨텍스트만을 설정하려면:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
```

  - 모든 컨텍스트를 설정하려면:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
```

  **참고:** `setid` 또는 `setall` 권한을 사용하려면 적절한 큐 오브젝트와 큐 관리자 오브젝트 모두에 권한을 부여해야 합니다.

- IBM i
  IBM i의 경우 다음 명령 중 하나를 실행하십시오.
  - ID 컨텍스트만을 설정하려면:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
QMgrName ')
```

  - 모든 컨텍스트를 설정하려면:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS의 경우 다음 명령 세트 중 하나를 실행하려면:
  - ID 컨텍스트만을 설정하려면:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  - 모든 컨텍스트를 설정하려면:

```
RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

컨텍스트를 전달하기 위한 권한 부여
컨텍스트를 검색된 메시지에서 비즈니스 요구사항이 있는 각 사용자 그룹에 넣는 중인 메시지로 전달하기 위한 권한을 부여하십시오.

## 이 태스크 정보

일부 큐에 컨텍스트를 전달하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령 중 하나를 실행하십시오.
  - ID 컨텍스트만을 전달하려면:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
```

  - 모든 컨텍스트를 전달하려면:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
```

- IBM i
  IBM i의 경우 다음 명령 중 하나를 실행하십시오.
  - ID 컨텍스트만을 전달하려면:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
QMgrName ')
```

  - 모든 컨텍스트를 전달하려면:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS의 경우 ID 컨텍스트 또는 모든 컨텍스트를 전달하려면 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

메시지를 로컬 큐에 넣기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 로컬 큐 또는 큐 세트에 메시지를 넣기 위한 권한을 부여하십시오.

## 이 태스크 정보

일부 로컬 큐에 메시지를 넣기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

메시지를 모델 큐에 넣기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 모델 큐 또는 모델 큐 세트에 메시지를 넣기 위한 권한을 부여하십시오.

### 이 태스크 정보

모델 큐는 동적 큐를 작성하는 데 사용됩니다. 그러므로 모델 및 동적 큐 둘 모두에 권한을 부여해야 합니다. 이러한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW

  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

  ```
  setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
  ```

- IBM i

  IBM i의 경우 다음 명령을 실행하십시오.

  ```
  GRTMQMAUT OBJ(' ModelQueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  ```

- z/OS

  z/OS의 경우 다음 명령을 실행하십시오.

  ```
  RDEFINE MQQUEUE QMgrName.ModelQueueName UACC(NONE)
  PERMIT QMgrName.ModelQueueName CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ModelQueueName**
  동적 큐가 기반으로 하는 모델의 이름입니다.

  **ObjectProfile**
  권한 부여를 변경할 동적 큐 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

메시지를 리모트 클러스터 큐에 넣기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 리모트 클러스터 큐 또는 큐 세트에 메시지를 넣기 위한 권한을 부여하십시오.

### 이 태스크 정보

리모트 클러스터 큐에 메시지를 넣기 위해 리모트 큐의 로컬 정의에 넣거나 완전한 리모트 큐에 넣을 수 있습니다. 리모트 큐의 로컬 정의를 사용 중인 경우에는 로컬 오브젝트에 넣기 위한 권한이 필요합니다. 369 페이지의 『메시지를 로컬 큐에 넣기 위한 권한 부여』 의 내용을 참조하십시오. 완전한 리모트 큐를 사용 중인 경우에는 리모트 큐에 넣기 위한 권한이 필요합니다. 운영 체제에 적합한 명령을 사용하여 이 권한을 부여하십시오.

기본 동작은 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대한 액세스 제어를 수행하는 것입니다. 이 작동은 여러 전송 큐를 사용 중인 경우에도 적용된다는 점을 유의하십시오.

이 주제에 설명된 특정 동작은 보안 스탠자 주제에 설명된 대로 qm.ini 파일에서 **ClusterQueueAccessControl** 속성을 *RQMName*으로 구성하고 큐 관리자를 재시작한 경우에만 적용됩니다.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -t rqmname -n
ObjectProfile -g GroupName +put
```

  리모트 클러스터 큐에 대해서만 *rqmname* 오브젝트를 사용할 수 있음을 유의하십시오.
- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
QMgrName')
```

  리모트 클러스터 큐에 대해서만 RMTMQMNAME 오브젝트를 사용할 수 있음을 유의하십시오.
- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  리모트 클러스터 큐에 대해서만 리모트 큐 관리자(또는 큐 공유 그룹)의 이름을 사용할 수 있음을 유의하십시오.

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한 부여를 변경하려는 리모트 큐 관리자 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 토픽에 대한 사용자 액세스 제어

토픽에 대한 애플리케이션의 액세스를 제어해야 합니다. 취할 조치를 판별하려면 이 토픽을 사용하십시오.

첫 번째 열의 각 신뢰 명령문에 대해 두 번째 열에 표시된 조치를 취하십시오.

표 *74.* 토픽에 대한 사용자 액세스 제어

| **명령문** | **Action** |
|---|---|
| 애플리케이션이 토픽에 메시지를 발행합니다. | 371 페이지의 『토픽에 메시지를 발행하기 위한 권한 부여』 참조 |
| 애플리케이션이 토픽을 구독합니다. | 372 페이지의 『토픽을 구독하기 위한 권한 부여』 참조 |

토픽에 메시지를 발행하기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 메시지를 토픽 또는 토픽 세트에 발행하기 위한 권한을 부여하십시오.

### 이 태스크 정보

메시지를 일부 토픽에 발행하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

토픽을 구독하기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 토픽 또는 토픽 세트를 구독하기 위한 권한을 부여하십시오.

### 이 태스크 정보

일부 토픽을 구독하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## 큐 관리자에서 조사하기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 큐 관리자에서 조사하기 위한 권한을 부여하십시오.

## 이 태스크 정보
큐 관리자에서 조사하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  이러한 명령은 지정된 큐 관리자에 액세스를 부여합니다. 사용자가 MQINQ 명령을 사용하도록 허용하려면 다음 명령을 실행하십시오.

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 프로세스에 액세스하기 위한 권한 부여
비즈니스 요구사항이 있는 각 사용자 그룹에 프로세스 또는 프로세스 세트에 액세스하기 위한 권한을 부여하십시오.

## 이 태스크 정보
일부 프로세스에 액세스하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
')
```

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

## 이름 목록에 액세스하기 위한 권한 부여

비즈니스 요구사항이 있는 각 사용자 그룹에 이름 목록 또는 이름 목록 세트에 액세스하기 위한 권한을 부여하십시오.

## 이 태스크 정보

일부 이름 목록에 액세스하기 위한 권한을 부여하려면 운영 체제에 적합한 명령을 사용하십시오.

Multi 멀티플랫폼에서 SET AUTHREC 명령을 사용할 수도 있습니다.

### 프로시저

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
+all
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- z/OS

z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile
CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
큐 관리자 이름. z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## ALW AIX, Linux, and Windows 에서 IBM MQ 를 관리할 수 있는 권한

IBM MQ 관리자는 모든 IBM MQ 명령을 사용하고 다른 사용자에 대한 권한을 부여할 수 있습니다. 관리자가 리모트 큐 관리자에게 명령을 발행할 때 관리자는 리모트 큐 관리자에 대해 필요한 권한이 있어야 합니다. 추가 고려사항은 Windows 시스템에 적용됩니다.

IBM MQ 관리자에게는 모든 IBM MQ 명령(다른 사용자에 대한 IBM MQ 권한을 부여하기 위한 명령 포함)을 사용할 권한이 있습니다.

IBM MQ 관리자가 되려면 **mqm** 그룹이라는 특수 그룹의 구성원이어야 합니다.

Windows 또는 Windows 에서만 로컬 계정이 Windows 시스템에서 관리자 그룹의 구성원인 경우 IBM MQ를 관리할 수 있습니다.

**주의:** 관리자 명령을 사용하여 mqm 그룹에 Azure AD 사용자를 추가할 수 있습니다. 예를 들어, `net localgroup mqm AzureAD\<your userID> /add` 명령을 사용하십시오. 그런 다음, IBM MQ 관리 명령을 실행하거나 IBM MQ Explorer를 사용하십시오.

**mqm** 그룹은 IBM MQ가 설치될 때 자동으로 작성됩니다. 추가 사용자가 관리를 수행할 수 있도록 이들을 그룹에 추가할 수 있습니다. 이 그룹의 모든 구성원은 모든 자원에 대한 액세스 권한을 가집니다. 이 액세스 권한은 **mqm** 그룹에서 사용자를 제거하고 **REFRESH SECURITY** 명령을 실행해야만 취소할 수 있습니다.

관리자는 IBM MQ를 관리하기 위한 제어 명령을 사용할 수 있습니다. 이러한 제어 명령 중 하나는 **setmqaut**이고, 이는 다른 사용자들이 IBM MQ 자원에 액세스하거나 제어할 수 있도록 권한을 부여하는 데 사용됩니다. 권한 레코드 관리를 위한 PCF 명령은 큐 관리자에 대해 dsp 및 chg 권한을 부여받은 비관리자가 사용할 수 있습니다. PCF 명령을 사용한 권한 관리에 대한 자세한 정보는 PCF(Programmable Command Format)를 참조하십시오.

관리자는 리모트 큐 관리자가 처리하는 MQSC 명령에 대한 필수 권한이 있어야 합니다. IBM MQ Explorer는 PCF 명령을 발행하여 관리 태스크를 수행합니다. 관리자는 IBM MQ Explorer를 사용하여 로컬 시스템에서 큐 관리자를 관리하는 데 추가 권한이 필요하지 않습니다. IBM MQ Explorer가 또 다른 시스템에서 큐 관리자를 관리하는 데 사용되면 관리자에게는 PCF 명령이 리모트 큐 관리자에 의해 처리되기 위해서 필요한 권한이 있어야 합니다.

**주의:** IBM MQ Script (MQSC) 명령을 실행하는 제어 명령 **runmqsc**를 사용하기 위해 관리자일 필요는 없습니다.

MQSC 명령을 리모트 큐 관리자에게 송신하는 데 간접적인 모드로 **runmqsc**가 사용되면, 각 MQSC 명령은 이스케이프 PCF 명령 안에 캡슐화됩니다.

PCF 및 MQSC 명령이 처리될 때의 권한 확인에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 큐 관리자, 큐, 프로세스, 이름 목록 및 인증 정보 오브젝트에 대해 작동하는 PCF 명령의 경우 IBM MQ 오브젝트에 대해 작업하는 권한을 참조하십시오. PCF 나가기 명령 내에서 캡슐화된 동등한 MQSC 명령에 대해서 이 절을 참조하십시오.
- 채널, 채널 이니시에이터, 리스너 및 클러스터에 대해 실행되는 PCF 명령은 채널 보안을 참조하십시오.
- 권한 레코드에 대해 작동하는 PCF 명령의 경우 PCF 명령에 대한 권한 검사를 참조하십시오.

- z/OS IBM MQ for z/OS에서 명령 서버가 처리하는 MQSC 명령의 경우 z/OS에서 명령 보안 및 명령 자원 보안을 참조하십시오.

또한 Windows 시스템에서 SYSTEM 계정에는 IBM MQ 자원에 대한 전체 액세스 권한이 있습니다.

AIX and Linux 플랫폼에서는 제품에서만 사용할 **mqm**의 특수 사용자 ID도 작성됩니다 권한을 부여받지 못한 사용자에게 사용 가능해서는 안됩니다. 모든 IBM MQ 오브젝트는 사용자 ID **mqm**이 소유하고 있습니다.

Windows 시스템에서 관리자 그룹의 구성원은 SYSTEM 계정과 마찬가지로 큐 관리자도 관리할 수 있습니다. 도메인 내에서 활성 상태인 모든 권한 있는 사용자 ID가 포함된 도메인 제어기에 도메인 **mqm** 그룹을 작성하고 이 그룹을 로컬 **mqm** 그룹에 추가할 수도 있습니다. **crtmqm**과 같은 일부 명령은 IBM MQ 오브젝트에서 권한을 조작하므로 이러한 오브젝트에 대해 작업하기 위한 권한이 필요합니다(다음 절에 설명된 대로). **mqm** 그룹의 구성원은 모든 오브젝트에 대해 작업할 수 있는 권한이 있지만, Windows 시스템에서 같은 이름의 로컬 사용자와 도메인 인증 사용자가 있는 경우 권한이 거부되는 경우가 있을 수 있습니다. 이 프로그램은 379 페이지의 『AIX, Linux, and Windows에서 프린시펄 및 그룹』에서 설명합니다.

사용자 계정 제어(UAC) 기능이 포함된 Windows 버전은 사용자가 관리자 그룹의 구성원일 경우에도 특정 운영 체제 기능에 대해 수행할 수 있는 조치를 제한합니다. 사용자 ID가 관리자 그룹에 있지만 **mqm** 그룹에는 없는 경우 승격된 명령 프롬프트를 사용하여 IBM MQ admin 명령 (예: **crtmqm**) 을 실행해야 합니다. 그렇지 않으면 `AMQ7077: 요청된 조작을 수행할 권한이 없습니다` 오류가 생성됩니다. 상승된 명령 프롬프트를 열려면 시작 메뉴 항목 또는 명령 프롬프트의 경우 아이콘을 마우스 오른쪽 단추로 클릭하고 **관리자로서 실행**을 선택하십시오.

다음 조치를 취하기 위해 **mqm** 그룹의 구성원일 필요는 없습니다.

- PCF 나가기 명령 내에 PCF 명령 또는 MQSC 명령을 발행하는 애플리케이션 프로그램에서 명령을 발행하십시오(채널 이니시에이터를 조작하지 않는 경우). (이러한 명령은 108 페이지의 『채널 시작기 정의 보호』에 설명되어 있습니다.)
- 애플리케이션 프로그램에서 MQI 호출을 발행하십시오 (`MQCONNX` 호출에 빠른 경로 바인딩을 사용하지 않는 경우).
- `crtmqcvx` 명령을 사용하여 데이터 유형 구조에 대한 데이터 변환을 수행하는 코드의 단편을 작성하십시오.
- `dspmq` 명령을 사용하여 큐 관리자를 표시하십시오.
- `dspmqtrc` 명령을 사용하여 IBM MQ 형식화된 추적 출력을 표시하십시오.

그룹과 사용자 ID 둘 모두에 12자 제한이 적용됩니다.

일반적으로, UNIX and Linux 플랫폼은 사용자 ID의 길이를 12자로 제한합니다. AIX 5.3에서는 이 한계를 높였지만 IBM MQ에서는 모든 UNIX and Linux 플랫폼에서 12자 제한을 유지합니다. 12자보다 많은 사용자 ID를 사용하면 IBM MQ는 이를 `UNKNOWN` 값으로 바꿉니다. 사용자 ID를 `UNKNOWN` 값으로 정의하지 마십시오.

## ALW AIX, Linux, and Windows에서 mqm 그룹 관리

mqm 그룹의 사용자에게는 IBM MQ에 대한 전체 관리 권한이 부여됩니다. 이러한 이유로 애플리케이션과 일반 사용자를 mqm 그룹에 등록해서는 안됩니다. mqm 그룹에는 IBM MQ 관리자의 계정만을 포함해야 합니다.

이러한 태스크가 다음에 설명되어 있습니다.

- Windows Windows에서 그룹 작성 및 관리
- AIX AIX에서 그룹 작성 및 관리
- Linux Linux에서 그룹 작성 및 관리

Windows 도메인 제어기가 Windows 2000 또는 Windows 2003 이상에서 실행되는 경우 도메인 관리자는 IBM MQ가 사용할 특수 계정을 설정했을 수도 있습니다. 자세한 정보는 Prepare IBM MQ Wizard 를 사용하여 IBM MQ 구성 및 IBM MQ 에 대한 Windows 도메인 계정 작성 및 설정의 내용을 참조하십시오.

## ALW AIX, Linux, and Windows 에서 IBM MQ 오브젝트에 대한 작업을 수행할 수 있는 권한

모든 오브젝트는 IBM MQ에 의해 보호되고, 프린시펄에는 이에 액세스하기 위한 적합한 권한이 제공되어야 합니다. 서로 다른 프린시펄은 다른 오브젝트마다 다른 액세스 권한이 필요합니다.

큐 관리자, 큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트는 모두 MQI 호출 또는 PCF 명령을 사용하는 애플리케이션에서 액세스됩니다. 이러한 자원은 모두 IBM MQ에 의해 보호되고, 애플리케이션에는 이에 액세스하기 위한 권한이 제공되어야 합니다. 요청을 하는 엔티티는 사용자, MQI 호출을 실행하는 애플리케이션 프로그램 또는 PCF 명령을 실행하는 관리 프로그램일 수 있습니다. 요청자의 ID를 프린시펄이라고 합니다.

서로 다른 프린시펄 그룹에 동일한 오브젝트에 대한 서로 다른 유형의 액세스 권한을 부여할 수 있습니다. 예를 들어, 특정 큐의 경우 한 그룹이 넣기 및 가져오기 조작 둘 모두를 수행하도록 허용될 수도 있고 다른 그룹은 큐를 찾아보기만이 허용될 수도 있습니다(찾아보기 옵션이 있는 MQGET). 마찬가지로, 일부 그룹에 큐에 대한 가져오기(get) 및 넣기(put) 권한을 주는 동시에 큐 속성 변경 또는 큐 삭제를 할 수 없도록 할 수 있습니다.

일부 조작은 특히 민감하여 권한이 있는 사용자로 제한해야 합니다. 예를 들면, 다음과 같습니다.

- 특정 큐(예: 트랜스미션 큐)나 커맨드 큐 SYSTEM.ADMIN.COMMAND.QUEUE에 액세스
- 전체 MQI 컨텍스트 옵션을 사용하는 프로그램 실행
- 애플리케이션 큐 작성 및 삭제

오브젝트에 대한 전체 액세스 권한은 오브젝트를 작성한 사용자 ID에 및 mqm 그룹의 모든 구성원(및 Windows 시스템에서 로컬 관리자 그룹의 구성원)에게 자동으로 제공됩니다.

**관련 개념**

375 페이지의 『AIX, Linux, and Windows 에서 IBM MQ 를 관리할 수 있는 권한』
IBM MQ 관리자는 모든 IBM MQ 명령을 사용하고 다른 사용자에 대한 권한을 부여할 수 있습니다. 관리자가 리모트 큐 관리자에게 명령을 발행할 때 관리자는 리모트 큐 관리자에 대해 필요한 권한이 있어야 합니다. 추가 고려사항은 Windows 시스템에 적용됩니다.

## ALW AIX, Linux, and Windows에서 보안 검사가 수행되는 시기

보안 검사는 일반적으로 큐 관리자에 연결할 때, 오브젝트를 열거나 닫을 때 및 메시지를 넣거나 가져올 때 수행됩니다.

일반 애플리케이션에 대한 보안 검사는 다음과 같습니다.

**큐 관리자에 연결(MQCONN 또는 MQCONNX 호출)**

이 때 애플리케이션이 특정 큐 관리자와 처음으로 연관됩니다. 큐 관리자는 운영 환경을 조사하여 애플리케이션과 연관된 사용자 ID를 검색합니다. IBM MQ는 그러면 사용자 ID가 큐 관리자에 연결되는지 확인하고 나중에 검사하기 위해 사용자 ID를 보존합니다.

사용자는 IBM MQ에 사인온할 필요가 없습니다. IBM MQ는 사용자가 기본 운영 체제에 사인온했고 이에 의해 인증되었음을 가정합니다.

**오브젝트 열기(MQOPEN 또는 MQPUT1 호출)**

IBM MQ 오브젝트는 오브젝트를 열고 이에 대해 명령을 실행하여 액세스합니다. 모든 자원 검사는 오브젝트에 실제로 액세스할 때가 아니라 오브젝트를 열 때 수행됩니다. 이는 **MQOPEN** 요청이 필요한 액세스 유형을 지정해야 함을 의미합니다. 예를 들어, 사용자가 오브젝트를 찾아보기만을 원하는지 또는 메시지를 큐에 넣기 등과 같은 업데이트를 수행하고 싶어하는지 여부입니다.

IBM MQ는 **MQOPEN** 요청에 이름이 지정된 자원을 검사합니다. 알리어스 큐 또는 리모트 큐 오브젝트의 경우, 권한은 알리어스 큐나 리모트 큐가 해석하는 큐가 아니라 오브젝트 자체에 대해 사용한 권한입니다. 이는 사용자가 여기에 액세스할 권한이 필요하지 않음을 의미합니다. 큐 작성 권한을 권한이 있는 사용자로 제한하십시오. 그렇게 하지 않으면 사용자가 알리어스를 작성하는 것만으로 일반 액세스 제어를 무시할 수도 있습니다. 리모트 큐에서 큐와 큐 관리자 이름을 명시적으로 참조하는 경우, 리모트 큐 관리자와 연관된 트랜스미션 큐를 검사합니다.

동적 큐에 대한 권한은 구동된 모델 큐 권한을 기본으로 하므로 반드시 동일할 필요는 없습니다. 이에 대해서는 참고 124 페이지의 『1』에서 설명됩니다.

액세스 검사를 위해 큐 관리자가 사용하는 사용자 ID는 큐 관리자에 연결된 응용프로그램의 운영 환경에서 확보한 사용자 ID입니다. 적절히 권한이 부여된 애플리케이션은 대체 사용자 ID를 지정하는 **MQOPEN** 호출을 발행할 수 있습니다. 그런 다음 대체 사용자 ID에 대해 액세스 제어 검사를 수행합니다. 이러한 조치로 애플리케이션과 연관된 사용자 ID가 변경되지 않으며 액세스 제어 검사에 사용된 사용자 ID가 변경됩니다.

**메시지 넣기 및 가져오기(MQPUT 또는 MQGET 호출)**
액세스 제어 검사가 수행되지 않습니다.

**오브젝트 닫기(MQCLOSE)**
**MQCLOSE**가 동적 큐를 삭제하는 결과를 낳지 않는 한 액세스 제어 검사가 수행되지 않습니다. 이 경우 사용자 ID가 큐를 삭제할 권한이 있는지 검사합니다.

**토픽 구독하기(MQSUB)**
애플리케이션이 토픽을 구독할 때 이는 수행해야 할 조작의 유형을 지정합니다. 새 구독을 작성하거나, 기존 구독을 변경하거나, 이를 변경하지 않고 기존 구독을 재개하는 것입니다. 조작의 유형마다 큐 관리자는 애플리케이션에 연관된 사용자 ID에 해당 조작을 수행할 권한이 있는지 검사합니다.

애플리케이션이 토픽을 구독할 때 구독한 애플리케이션이 있는 토픽 트리 이상의 위치에서 발견된 토픽 오브젝트에 대해 권한 검사가 수행됩니다. 권한 검사에는 두 개 이상의 토픽 오브젝트에 대한 검사가 포함될 수 있습니다.

큐 관리자가 권한 검사를 위해 사용하는 사용자 ID는 애플리케이션이 큐 관리자에 연결할 때 운영 체제로부터 얻은 사용자 ID입니다.

큐 관리자는 관리 큐가 아니라 구독자 큐에서 권한 검사를 수행합니다.

## ALW AIX, Linux, and Windows 에서 IBM MQ 에 의해 액세스 제어가 구현되는 방법

IBM MQ는 오브젝트 권한 관리자를 사용하여 기본 운영 체제가 제공하는 보안 서비스를 사용합니다. IBM MQ는 액세스 제어 목록을 작성하고 유지보수하기 위한 명령을 제공합니다.

권한 서비스 인터페이스라는 액세스 제어 인터페이스는 IBM MQ의 일부입니다. IBM MQ는 액세스 제어 관리자의 구현을 제공합니다(오브젝트 권한 관리자*(OAM)*로 알려진 권한 서비스 인터페이스 준수). 이는 사용자가 별도로 지정하지 않는 한 자동으로 설치되고 사용자가 작성하는 각 큐 관리자에서 사용 가능으로 설정됩니다(340 페이지의 『AIX, Linux, and Windows 시스템에서 보안 액세스 검사 방지』에 설명된 대로). OAM은 권한 서비스 인터페이스를 준수하는 사용자 또는 벤더가 작성한 컴포넌트로 대체될 수 있습니다.

OAM은 운영 체제 사용자 및 그룹 ID를 사용하여 기본 운영 체제의 보안 기능을 사용합니다. 사용자는 연결 권한이 있는 경우에만 IBM MQ 오브젝트에 액세스할 수 있습니다. 331 페이지의 『AIX, Linux, and Windows에서 OAM을 사용하여 오브젝트에 대한 액세스 제어』에서는 이 권한을 부여하고 취소하는 방법을 설명합니다.

OAM은 제어하는 각 자원에 대해 액세스 제어 목록(ACL)을 유지보수합니다. 권한 부여 데이터는 SYSTEM.AUTH.DATA.QUEUE라는 로컬 큐에 저장됩니다. 이 큐에 대한 액세스 권한은 mqm 그룹의 사용자로 제한되며, Windows에서는 관리자 그룹의 사용자 및 SYSTEM ID로 로그인한 사용자로 제한됩니다. 큐에 대한 사용자 액세스는 변경될 수 없습니다.

IBM MQ는 액세스 제어 목록을 작성하고 유지보수하기 위한 명령을 제공합니다. 이러한 명령에 대한 자세한 정보는 331 페이지의 『AIX, Linux, and Windows에서 OAM을 사용하여 오브젝트에 대한 액세스 제어』의 내용을 참조하십시오.

IBM MQ는 OAM에 프린시펄, 자원 이름 및 액세스 유형을 포함하는 요청을 전달합니다. OAM은 유지보수하는 ACL을 기본으로 액세스 권한을 부여하거나 거부합니다. IBM MQ는 OAM의 결정을 따릅니다. OAM이 결정을 내릴 수 없는 경우 IBM MQ는 액세스를 허용하지 않습니다.

## ALW AIX, Linux, and Windows에서 사용자 ID 식별

오브젝트 권한 관리자는 자원에 대한 액세스를 요청하고 있는 프린시펄을 식별합니다. 프린시펄로서 사용되는 사용자 ID는 컨텍스트에 따라 다릅니다.

오브젝트 권한 관리자(OAM)는 특정 자원에 대한 액세스를 요청 중인 사용자를 식별할 수 있어야 합니다. IBM MQ는 이 ID를 가리키기 위해 프린시펄이라는 용어를 사용합니다. 프린시펄은 애플리케이션이 처음 큐 관리자에 연결할 때 설정됩니다. 이는 연결 애플리케이션과 연관된 사용자 ID로부터 큐 관리자에 의해 판별됩니다. (애플

리케이션이 큐 관리자에 연결하지 않고 XA 호출을 실행하면 xa_open 호출을 실행한 애플리케이션과 연관된 사용자 ID가 큐 관리자에 의해 권한 검사에 사용됩니다.)

AIX and Linux 시스템에서 권한 루틴은 실제(로그인한) 사용자 ID 또는 애플리케이션과 연관된 유효 사용자 ID를 검사합니다. 검사된 사용자 ID는 바인드 유형에 의존할 수 있습니다. 자세한 내용은 설치 가능 서비스를 참조하십시오.

IBM MQ는 시스템으로부터 받은 사용자 ID를 각 메시지의 메시지 헤더(MQMD 구조)에 사용자의 ID로서 전파합니다. 이 ID는 메시지 컨텍스트 정보의 일부이고 381 페이지의 『AIX, Linux, and Windows에서 컨텍스트 권한』 에서 설명됩니다. 애플리케이션은 컨텍스트 정보를 변경할 권한이 부여되지 않는 한 이 정보를 변경할 수 없습니다.

## ALW *AIX, Linux, and Windows*에서 프린시펄 및 그룹

프린시펄은 그룹에 속할 수 있습니다. 개인이 아닌 그룹에 자원 액세스를 부여하면 필요한 관리의 양을 줄일 수 있습니다. 액세스 제어 목록(ACL)은 그룹 및 사용자 ID 둘 모두를 기반으로 합니다.

예를 들어, 특정 애플리케이션을 실행하고 싶어하는 사용자로 구성된 그룹을 정의할 수도 있습니다. 기타 사용자는 해당 사용자 ID를 적절한 그룹에 추가하여 필요한 모든 자원에 대한 액세스를 제공받을 수 있습니다.

그룹을 정의하고 관리하는 이 프로세스는 특정 플랫폼별로 설명됩니다.

- AIX AIX에서 그룹 작성 및 관리
- Linux Linux에서 그룹 작성 및 관리
- Windows Windows에서 그룹 작성 및 관리

프린시펄은 둘 이상의 그룹(해당 그룹 세트)에 속할 수 있습니다. 이는 해당 그룹 세트에서 각 그룹에 부여된 모든 권한의 집합을 가지고 있습니다. 이러한 권한은 캐싱되므로 프린시펄의 그룹 멤버십에 작성한 변경사항은 사용자가 MQSC 명령 **REFRESH SECURITY**(또는 해당 PCF에 상당하는 명령)을 실행하지 않는 한 큐 관리자가 재시작될 때까지는 인식되지 않습니다.

### Linux AIX AIX and Linux 시스템

ACL(액세스 제어 목록)은 사용자 ID와 그룹을 모두 기반으로 하며 다음 중 하나를 설정하여 인증에 사용할 수 있습니다.**SecurityPolicy** 에 설명된 대로 적절한 값에 속성을 지정합니다. 서비스 스탠자qm.ini 파일 .

권한 부여를 위해 사용자 기반 모델 을 사용할 수 있으며, 이를 통해 사용자와 그룹을 모두 사용할 수 있습니다. 그러나 setmqaut 명령에 사용자를 지정할 때 새 권한은 사용자에게만 적용되고 해당 사용자가 속한 그룹에는 적용되지 않습니다. 자세한 정보는 331 페이지의 『AIX and Linux의 OAM 사용자 기반 권한』 의 내용을 참조하십시오.

권한 부여를 위해 그룹 기반 모델을 사용하는 경우, 사용자 ID가 속하는 기본 그룹이 ACL에 포함됩니다. 개별 사용자 ID는 포함되지 않고 권한은 해당 그룹의 모든 구성원에게 부여됩니다. 이로 인해 동일 그룹에서 또 다른 프린시펄의 권한을 변경하여 프린시펄의 권한을 부주의하게 변경할 수 있음을 유의하십시오.

모든 사용자가 명목상으로 nobody 기본 사용자 그룹에 지정되고, 기본적으로 이 그룹에는 권한 부여가 제공되지 않습니다. nobody 그룹의 권한을 변경하여 특정 권한이 없는 사용자에게 IBM MQ 자원에 대한 액세스를 부여할 수 있습니다.

에서IBM MQ 9.3.0 , 당신은`UserExternal` 의 옵션**SecurityPolicy** 비운영 체제 사용자 이름을 생성하는 속성입니다. 비운영 체제 사용자 이름을 작성하는 경우 해당 사용자는 nobody 그룹에만 속한 것으로 간주됩니다. 이 옵션에 대한 자세한 정보는 crtmqm 및 qm.ini 파일의 Service 스탠자를 참조하십시오.

`UNKNOWN` 값으로 사용자 ID를 정의하지 마십시오. `UNKNOWN` 값은 사용자 ID가 너무 길 때 사용되므로 임의의 사용자 ID는 `UNKNOWN`의 액세스 권한을 사용합니다.

LDAP 사용에 대한 정보는 386 페이지의 『설정 권한 부여』 의 내용을 참조하십시오.

사용자는 최대 12자를 포함할 수 있고 그룹 이름은 12자를 포함할 수 있습니다.

### Windows Windows 시스템

ACL은 사용자 ID 및 그룹 둘 모두를 기반으로 합니다. AIX and Linux에서도 검사는 동일합니다. 동일한 사용자 ID가 있는 다른 도메인에서는 다른 사용자를 가질 수 있습니다. IBM MQ는 이러한 사용자에게 다른 액세스 레벨이 부여될 수 있도록 사용자 ID가 도메인 이름에 의해 규정되도록 허용합니다.

그룹 이름에는 선택적으로 다음 형식으로 지정된 도메인 이름을 포함할 수 있습니다.

```
GroupName@domain domain_name\group_name
```

글로벌 그룹은 두 가지 경우에만 OAM에 의해 검사됩니다.

1. 큐 관리자 보안 스탠자에는 `GroupModel=GlobalGroups` 설정이 포함됩니다. 보안을 참조하십시오.
2. 큐 관리자는 대체 보안 액세스 그룹을 사용 중입니다. **crtmqm** 의 내용을 참조하십시오.

사용자 ID는 최대 20자까지 포함하고 도메인 이름은 최대 15자, 그룹 이름은 최대 64자까지 포함할 수 있습니다.

OAM은 먼저 로컬 보안 데이터베이스를 검사한 다음 1차 도메인의 데이터베이스를 검사하고 마지막으로 신뢰하는 도메인의 데이터베이스를 검사합니다. 발견된 첫 번째 사용자 ID가 검사를 위해 OAM에 의해 사용됩니다. 이러한 각 사용자 ID에는 특정 컴퓨터에 여러 그룹 멤버십이 있을 수 있습니다.

일부 제어 명령(예: **crtmqm**)은 오브젝트 권한 관리자(OAM)를 사용하여 IBM MQ 오브젝트에서 권한을 변경합니다. OAM은 특정 사용자 ID의 권한을 판별하기 위해 앞의 단락에 지정된 순서로 보안 데이터베이스를 검색합니다. 그 결과 OAM에 의해 판별된 권한은 사용자 ID가 로컬 mqm 그룹의 구성원이라는 사실을 대체할 수도 있습니다. 예를 들어, 글로벌 그룹을 통해 로컬 mqm 그룹의 멤버십을 가지고 있는 도메인 제어기에 의해 인증된 사용자 ID로부터 **crtmqm** 명령을 실행하는 경우 시스템에 로컬 mqm 그룹에 없고 이름이 같은 로컬 사용자가 있는 경우 명령이 실패합니다.

설정에 대한 자세한 내용은**SecurityPolicy** 속성Windows , 보다 서비스 스탠자qm.ini 파일 .

## Windows *Windows 보안 ID(SID)*

Windows 의 IBM MQ 는 사용 가능한 SID를 사용합니다. Windows SID가 권한 부여 요청과 함께 제공되지 않은 경우에는 IBM MQ는 사용자 이름만을 기반으로 사용자를 식별하지만 잘못된 권한이 부여되는 결과가 나올 수도 있습니다.

Windows 시스템에서 사용자 ID를 보충하기 위해 보안 ID(SID)가 사용됩니다. SID에는 사용자가 정의된 Windows 보안 계정 관리자(SAM) 데이터베이스에서 전체 사용자 계정 세부사항을 식별합니다. IBM MQ for Windows에서 메시지가 작성되면 IBM MQ는 SID를 메시지 디스크립터에 저장합니다. IBM MQ on Windows 이 권한 검사를 수행할 때 SID를 사용하여 SAM 데이터베이스에서 전체 정보를 조회합니다. (이 조회가 성공하려면 사용자가 정의된 SAM 데이터베이스에 액세스 할 수 있어야 합니다.)

기본적으로, Windows SID에 권한 부여 요청이 제공되지 않은 경우에는, IBM MQ는 사용자 이름만을 기반으로 사용자를 식별합니다. 이는 보안 데이터베이스를 다음 순서로 검색하여 수행됩니다.

1. 로컬 보안 데이터베이스
2. 1차 도메인의 보안 데이터베이스
3. 신뢰되는 도메인의 보안 데이터베이스

사용자 이름이 고유하지 않은 경우 잘못된 IBM MQ 권한이 부여될 수도 있습니다. 이 문제를 방지하려면 각 권한 요청에 SID를 포함하십시오. SID는 IBM MQ가 사용자 신임 정보를 설정하기 위해 사용합니다.

모든 권한 요청에 SID가 포함되도록 지정하려면 **regedit**를 사용하십시오. SecurityPolicy를 NTSIDsRequired로 설정하십시오.

## ALW AIX, Linux, and Windows에서 대체 사용자 권한

사용자 ID가 IBM MQ 오브젝트에 액세스할 때 또 다른 사용자의 권한을 사용할 수 있는 사용자 ID를 지정할 수 있습니다. 이는 대체 사용자 권한이라 불리고 이를 임의의 IBM MQ 오브젝트에서 사용할 수 있습니다.

대체 사용자 권한은 서버가 프로그램으로부터 요청을 수신하고 프로그램에 요청의 필수 권한이 있는지 확인하고 싶은 경우에 필수입니다. 서버에는 필수 권한이 있을 수 있지만 프로그램이 요청한 조치에 대한 권한이 있는지 여부를 알아야 합니다.

예를 들어, 사용자 ID PAYSERV 하에서 실행 중인 서버 프로그램이 큐에서 사용자 ID USER1에 의해 큐에 놓여진 요청 메시지를 검색한다고 가정해 보십시오. 서버 프로그램이 요청 메시지를 가져오면 요청을 처리하고 요청 메시지에 지정된 응답 대상 큐로 응답을 다시 넣습니다. 자체 사용자 ID(PAYSERV)를 사용하여 응답 대상 큐를 여는 권한을 부여하는 대신에 서버는 다른 사용자 ID를 지정할 수 있습니다. 이 경우에는 USER1입니다. 이 예에서는

대체 사용자 권한을 사용하여 PAYSERV가 응답 대상 큐를 열 때 대체 사용자 ID로서 USER1을 지정하도록 허용되는지 여부를 제어할 수 있습니다.

대체 사용자 ID는 오브젝트 디스크립터의 **AlternateUserId** 필드에 지정됩니다.

### Linux Linux에서 특정 그룹 멤버십 문제 해결

일부 시스템은 일반 **getgrent** 운영 체제 API 호출 시리즈를 통해 그룹 정보를 리턴하는 속도가 느립니다. 또한 엔터프라이즈에 검색할 수천 개의 그룹이 있고 mqm 사용자가 속한 그룹을 찾는 경우 느린 응답으로 인해 내부 큐 관리자 제한시간 초과가 발생할 수 있습니다. 이 문제점을 방지하기 위한 대체 운영 체제 API가 있습니다.

더 빠른 대체 API를 사용하고 한 번의 호출에서 모든 그룹을 리턴하려면 환경 변수 MQS_GETGROUPLIST_API를 설정하십시오.

사용자의 보조 그룹에 연결 액세스 권한을 부여하고 MQS_GETGROUPLIST_API 변수를 사용으로 설정하면 문제점이 완화되는 경우 RC2035 오류가 수신될 수 있습니다.

그런 다음 IBM MQ 는 **getgrent** API 대신 **getgrouplist** API를 사용합니다.

**getgrouplist**를 사용으로 설정하려면 다음을 수행하십시오.

1. 큐 관리자 중지
2. export MQS_GETGROUPLIST_API=1 명령을 실행하십시오.
3. 큐 관리자를 재시작하십시오.

실패한 시나리오를 재시도하고 문제점이 해결된 경우 mqm 사용자의 `.bashrc / .profile` 파일을 수정하여 이 환경 변수를 추가하거나 큐 관리자를 시작하는 데 사용하는 스크립트에 환경 변수를 추가할 수 있습니다.

시스템이 NIS 또는 LDAP과 같은 여러 저장소의 운영 체제에 대한 사용자 또는 그룹 정보를 병합하는 경우 그룹 또는 사용자 ID가 운영 체제 레벨 권한을 설치하고 설정하는 데 사용되므로 로컬 저장소를 포함한 모든 저장소에서 일치하는지 확인하십시오.

### ALW AIX, Linux, and Windows에서 컨텍스트 권한

컨텍스트는 특정 메시지에 적용되는 정보이며, 메시지의 일부인 메시지 설명자 MQMD에 포함되어 있습니다. 애플리케이션은 MQOPEN 또는 MQPUT 호출이 만들어질 때 컨텍스트 데이터를 지정할 수 있습니다.

컨텍스트 정보는 다음과 같은 두 개의 섹션으로 구성됩니다.

**ID 섹션**
메시지 송신자. 이는 UserIdentifier, AccountingToken 및 ApplIdentityData 필드로 구성됩니다.

**원본 섹션**
메시지 송신자 및 큐에 넣는 시기. 이는 PutApplType, PutApplName, PutDate, PutTime 및 ApplOriginData 필드로 구성됩니다.

애플리케이션은 MQOPEN 또는 MQPUT 호출이 만들어질 때 컨텍스트 데이터를 지정할 수 있습니다. 이 데이터는 애플리케이션에서 생성할 수 있으며, 다른 메시지에서 전달되거나 기본적으로 큐 관리자가 생성할 수 있습니다. 예를 들어, 서버 프로그램이 요청자의 ID를 검사하기 위해 컨텍스트 데이터를 사용하여 메시지가 권한 있는 사용자 ID 하에서 실행 중인 애플리케이션에서 나왔는지 여부를 테스트할 수 있습니다.

서버 프로그램은 UserIdentifier를 사용하여 대체 사용자의 사용자 ID를 판별할 수 있습니다. 컨텍스트 권한을 사용하여 사용자가 MQOPEN 또는 MQPUT1 호출에서 컨텍스트 옵션을 지정할 수 있는지 여부를 제어할 수 있습니다.

컨텍스트 옵션에 대한 정보는 컨텍스트 정보 제어 를 참조하고, 컨텍스트와 관련된 메시지 디스크립터 필드에 대한 설명은 MQMD-메시지 디스크립터 를 참조하십시오.

## 보안 엑시트에서 액세스 제어 구현

MCAUserIdentifier 또는 오브젝트 권한 관리자를 사용하여 보안 엑시트에서 액세스 제어를 구현할 수 있습니다.

### MCAUserIdentifier

현재 상태인 채널의 모든 인스턴스는 관련된 채널 정의 구조, 즉 MQCD를 가지고 있습니다. MQCD의 필드의 초기 값은 IBM MQ 관리자가 작성하는 채널 정의에 의해 판별됩니다. 특히, 필드 중 하나인 *MCAUserIdentifier*의 초기 값은 DEFINE CHANNEL 명령에서 MCAUSER 매개변수의 값에 의해 판별되거나, 채널 정의가 다른 방법으로 작성되는 경우에는 MCAUSER에 해당하는 값에 의해 판별됩니다.

MQCD 구조가 MCA에 의해 호출될 때 채널 엑시트 프로그램으로 전달됩니다. 보안 엑시트가 MCA에 의해 호출될 때, 보안 엑시트가 *MCAUserIdentifier*의 값을 변경할 수 있으며, 채널 정의에 지정된 모든 값을 바꿉니다.

Multi 멀티플랫폼에서 *MCAUserIdentifier*의 값이 공백이 아닌 한, 큐 관리자는 MCA가 큐 관리자에 연결한 후에 큐 관리자의 자원에 액세스하려고 시도할 때 *MCAUserIdentifier*의 값을 권한 검사를 위한 사용자 ID로서 사용합니다. *MCAUserIdentifier*의 값이 공백인 경우, 큐 관리자는 MCA의 기본 사용자 ID를 대신 사용합니다. 이는 RCVR, RQSTR, CLUSRCVR 및 SVRCONN 채널에 적용됩니다. MCA를 전송하기 위해 *MCAUserIdentifier*의 값이 공백이 아닐지라도 기본 사용자 ID가 항상 권한 검사에 사용됩니다.

z/OS z/OS에서 큐 관리자는 공백이 아닌 경우 권한 검사를 위해 *MCAUserIdentifier* 의 값을 사용할 수도 있습니다. 수신 MCA와 서버 연결 MCA의 경우, 큐 관리자가 권한 검사를 위해 *MCAUserIdentifier*의 값을 사용할지 여부는 다음에 따라 결정됩니다.

- 채널 정의에 있는 PUTAUT 매개변수의 값
- 검사에 사용된 RACF 프로파일
- 채널 시작기 주소 공간 사용자 ID의 RESLEVEL 프로파일로의 액세스 레벨

송신 MCA의 경우에는, 다음에 따라 달라집니다.

- 송신 MCA가 호출자인지 응답자인지 여부
- 채널 시작기 주소 공간 사용자 ID의 RESLEVEL 프로파일로의 액세스 레벨

보안 엑시트가 *MCAUserIdentifier*에 저장하는 사용자 ID는 다양한 방법으로 얻을 수 있습니다. 다음은 몇 가지 예입니다.

- MQI 채널의 클라이언트 끝에 보안 엑시트가 없는 경우, IBM MQ 클라이언트 애플리케이션과 연관된 사용자 ID는 클라이언트 애플리케이션이 MQCONN 호출을 실행할 때 클라이언트 연결 MCA에서 서버 연결 MCA로 흐릅니다.서버 연결 MCA는 이 사용자 ID를 채널 정의 구조인 MQCD에서 *RemoteUserIdentifier* 필드에 저장합니다. *MCAUserIdentifier*의 값이 이 때에 비어 있으면, MCA가 *MCAUserIdentifier*에 같은 사용자 ID를 저장합니다. MCA가 사용자 ID를 *MCAUserIdentifier*에 저장하지 않는 경우 보안 엑시트는 *MCAUserIdentifier*를 *RemoteUserIdentifier*의 값으로 설정하여 이를 나중에 수행할 수 있습니다.

  클라이언트 시스템에서 흐르는 사용자 ID가 새 보안 도메인을 입력하고 있고 서버 시스템에서 유효하지 않는 경우에는 보안 엑시트는 유효한 도메인을 위해 사용자 ID를 대체하고 대체된 사용된 ID를 *MCAUserIdentifier*에 저장합니다.
- 사용자 ID가 보안 메시지에서 파트너 보안 엑시트에 의해 송신될 수 있습니다.

  메시지 채널에서 송신 MCA에 의해 호출되는 보안 엑시트의 경우 송신 MCA가 실행 중인 사용자 ID를 송신할 수 있습니다. 그런 후, 수신 MCA에 의해 호출되는 보안 엑시트가 사용자 ID를 *MCAUserIdentifier*에 저장할 수 있습니다. 마찬가지로 MQI 채널에서 채널의 클라이언트 끝에 있는 보안 엑시트는 IBM MQ MQI client 애플리케이션과 연관된 사용자 ID를 전송할 수 있습니다. 그런 후, 채널의 서버 측에 있는 보안 엑시트가 사용자 ID를 *MCAUserIdentifier*에 저장할 수 있습니다. 이전 예에 있는 것처럼, 사용자 ID가 대상 시스템에서 유효하지 않으면, 보안 엑시트가 유효한 사용자 ID로 그것을 대체하고 *MCAUserIdentifier*에 대체된 사용자 ID를 저장할 수 있습니다.

  디지털 인증서가 식별과 인증 서비스의 일부로 수신되면, 보안 엑시트가 인증서의 식별 이름을 대상 시스템에서 유효한 사용자 ID로 맵핑할 수 있습니다. 그런 후, 사용자 ID를 *MCAUserIdentifier*에 저장할 수 있습니다.
- 채널에서 TLS가 사용되면 파트너의 식별 이름(DN)이 MQCD의 SSLPeerNamePtr 필드에 있는 엑시트로 전달되고 해당 인증서 발행인의 DN이 MQCXP의 SSLRemCertIssNamePtr 필드에 있는 엑시트로 전달됩니다.

*MCAUserIdentifier* 필드, 채널 정의 구조, MQCD 및 채널 엑시트 매개변수 구조, MQCXP에 대한 자세한 정보는 채널 엑시트 호출 및 데이터 구조를 참조하십시오. MQI 채널에서 클라이언트 시스템에서 흐르는 사용자 ID에 대한 정보는 액세스 제어를 참조하십시오.

**참고:** IBM WebSphere MQ 7.1 릴리스 이전에 구성된 보안 엑시트 애플리케이션은 업데이트가 필요할 수 있습니다. 자세한 정보는 채널 보안 엑시트 프로그램을 참조하십시오.

### IBM MQ 오브젝트 권한 관리자 사용자 인증

IBM MQ MQI client 연결에서 보안 엑시트는 오브젝트 권한 관리자(OAM) 사용자 인증에 사용된 MQCSP 구조를 수정하거나 작성하는 데 사용될 수 있습니다. 이에 대해서는 메시지 채널의 채널 엑시트 프로그램에 설명되어 있습니다.

## 메시지 엑시트에서 액세스 제어 구현

한 사용자 ID를 다른 사용자 ID로 대체하기 위해 메시지 엑시트를 사용해야 할 수도 있습니다.

메시지를 서버 애플리케이션에 전송하는 클라이언트 애플리케이션을 고려하십시오. 서버 애플리케이션은 메시지 디스크립터에서 *UserIdentifier* 필드로부터 사용자 ID를 추출하고, 대체 사용자 권한이 있는 경우 클라이언트 대신에 IBM MQ 자원에 액세스할 때 큐 관리자에게 이 사용자 ID를 사용하도록 요청합니다.

채널 정의에서 PUTAUT 매개변수가 CTX (또는 z/OS의 경우 ALTMCA) 로 설정된 경우, 각 수신 메시지의 *UserIdentifier* 필드에 있는 사용자 ID는 MCA가 목적지 큐를 열 때 권한 검사에 사용됩니다.

어떤 상황에서는 보고 메시지가 생성되면, 보고가 필요한 메시지의 *UserIdentifier* 필드에 있는 사용자 ID의 권한이 필요하게 됩니다. 특히, COD(confirm-on-delivery) 보고서와 만기 보고서는 항상 이 권한이 필요합니다.

이런 상황 때문에, 메시지가 새 보안 도메인에 들어갈 때 *UserIdentifier* 필드에서 하나의 사용자 ID를 다른 것 대신에 사용할 필요가 있을 수 있습니다. 이는 채널의 수신 측에 있는 메시지 엑시트에 의해 수행될 수 있습니다. 또는, 수신되는 메시지의 *UserIdentifier* 필드에 있는 사용자 ID가 새 보안 도메인에 정의되어 있는지를 확인할 수 있습니다.

수신되는 메시지에 메시지를 송신한 애플리케이션의 사용자에 대한 디지털 인증서가 있으면, 메시지 엑시트가 인증서의 유효성을 검증하고 인증서의 식별 이름을 수신하고 있는 시스템에서 유효한 사용자 ID로 맵핑할 수 있습니다. 그런 후, 메시지 설명자에 있는 *UserIdentifier* 필드를 이 사용자 ID로 설정할 수 있습니다.

메시지 엑시트가 수신되는 메시지에 있는 *UserIdentifier* 필드의 값을 변경하는 것이 필요하면, 메시지 엑시트가 메시지의 송신자를 동시에 인증하는 것이 적당할 수 있습니다. 자세한 정보는 304 페이지의 『메시지 엑시트에서 ID 맵핑』의 내용을 참조하십시오.

## API 엑시트 및 API 교차 엑시트에서 액세스 제어 구현

API 또는 API 교차 엑시트는 IBM MQ가 제공한 액세스 제어를 보충하기 위해 액세스 제어를 제공할 수 있습니다. 특히 엑시트는 메시지 레벨에서 액세스 제어를 제공할 수 있습니다. 엑시트는 애플리케이션이 특정 기준을 충족하는 메시지만을 큐에 넣고, 큐에서 가져오도록 확실히 할 수 있습니다.

다음 예를 고려하십시오.

- 메시지에 임의의 주문에 대한 정보가 있습니다. 애플리케이션이 메시지를 큐에 넣을 때, API 또는 API 교차 엑시트는 주문의 총 값이 몇몇 미리 정해진 한계 미만인지 확인할 수 있습니다.
- 메시지가 리모트 큐 관리자로부터 목적지 큐에 도착합니다. 애플리케이션이 큐로부터 메시지를 가져오려고 시도할 때 API 또는 API 교차 엑시트는 메시지의 송신자가 메시지를 큐에 전송할 권한이 있는지 확인할 수 있습니다.

## Multi 큐 보안 스트리밍

관리자는 스트리밍 큐 기능을 사용하여 메시지를 원래 큐에 넣을 때마다 중복 메시지가 배치되는 보조 큐가 있는 로컬(또는 모델) 큐를 구성할 수 있습니다. 큐 스트리밍 권한과 관련하여 고려할 요소는 두 가지가 있습니다.

### 중복 메시지 스트리밍을 위한 큐 구성 권한

한 큐에서 보조 큐로 중복 메시지를 스트리밍하려면 메시지 스트리밍을 사용으로 설정할 권한이 있어야 합니다. 큐의 **STREAMQ** 속성을 구성할 수 있으려면 다음 권한이 있어야 합니다.

1. **STREAMQ** 속성을 대체하는 큐의 CHG 권한
2. 중복 메시지를 넣을 큐의 CHG 권한

구성 시 이 두 권한 검사를 조합하면 원래 큐에 대한 CHG 권한만 있는 사용자가 권한이 없는 다른 큐에 메시지를 넣을 수 없습니다.

### 하나 이상의 큐를 열고 메시지를 넣을 권한

애플리케이션이 보조 큐로 구성된 큐를 열 때, **STREAMQ** 속성을 통해 권한을 검사하여 애플리케이션 사용자가 원래 큐에 대해 PUT 권한이 있는지 확인합니다.

**참고:** 별명 큐에 사용되는 권한 모델과 비슷하게 보조 큐에서는 애플리케이션 사용자의 추가 권한 검사가 수행되지 않습니다.

원본 또는 보조 큐의 메시지를 사용하는 애플리케이션에는 사용 중인 큐에서만 GET 또는 BROWSE 권한이 필요합니다.

put 또는 get 시간에는 추가 권한 검사가 수행되지 않습니다.

### 예

다음 예에서는 admin 사용자가 원래 큐인 INQUIRIES.QUEUE를 구성하여 중복 메시지를 로컬 큐인 ANALYTICS.queue로 스트리밍할 수 있지만, admin이 PURCHASES.QUEUE로 메시지를 복제하지 못하게 설정하는 올바른 권한을 보여줍니다.

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(ANALYTICS.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(PURCHASES.QUEUE) PRINCIPAL('admin') AUTHRMV(CHG)
```

그러면 admin 사용자가 다음 명령을 실행할 수 있습니다.

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

그러나 같은 사용자가 다음 명령을 실행하여

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

중복 메시지를 PURCHASES.QUEUE에 넣도록 INQUIRIES.QUEUE를 구성하는 경우 다음 오류가 발생합니다.

```
AMQ8135E 권한이 부여되지 않았습니다.
```

INQUIRIES.QUEUE가 메시지를 ANALYTICS.QUEUE에 복제하도록 구성된 경우 다음 권한 레코드를 사용하면 appuser 사용자로 실행 중인 애플리케이션을 사용하여 INQUIRIES.QUEUE에 메시지를 넣고 ANALYTICS.QUEUE에 메시지를 복제할 수 있습니다.

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

**참고:** appuser는 ANALYTICS.QUEUE에 대한 권한 레코드가 필요하지 않습니다. 중복 메시지는 큐 관리자가 큐에 넣습니다.

**관련 개념**
스트리밍 큐

## z/OS Streaming queues security on z/OS

The streaming queues feature allows an administrator to configure a local (or model) queue with a secondary queue, where duplicate messages are placed, whenever a message is put to the original queue. There are two aspects to consider regarding queue streaming authorities.

### Authority to configure a queue for streaming duplicate messages

If you want to enable message streaming of duplicate messages from one queue to a secondary queue, you must have permission to do so. Permission to configure the **STREAMQ** attribute of a queue requires that you have the following profiles setup:

1. ALTER access level to MQADMIN or MXADMIN for the queue they are altering the **STREAMQ** attribute for
2. ALTER access level to MQADMIN or MXADMIN for the queue you want to stream messages to

The combination of these security checks at configuration time ensures that a user, who only has ALTER access on the original queue, cannot cause messages to be put to another queue on which they have no permissions.

### Authority to open the queue or queues and put messages

When an application opens a queue that has been configured with a secondary queue, through its **STREAMQ** attribute, an authority check is made that the application user has UPDATE authority on the original queue.

**Note:** No additional authority check is made for the application user on the secondary queue, which is similar to the authority model used for alias queues.

Applications consuming messages from either the original or the secondary queue require UPDATE or READ authority, only on the queue they are consuming from.

No additional authority checks are made at put or get time.

### Example

The following example shows the correct profiles being set to allow user ADMIN to configure an original queue, INQUIRIES.QUEUE, to stream messages to local queue ANALYTICS.QUEUE using RACF:

```
RDEFINE MQCMDS <QMGR>.ALTER.QLOCAL UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.ALTER.QLOCAL CLASS(MQCMDS) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.INQUIRIES.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.INQUIRIES.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.ANALYTICS.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.ANALYTICS.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)
```

User ADMIN is then able to issue the following command:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

but if the same user issues the following command without setting up the correct security profiles:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

to configure INQUIRIES.QUEUE to put duplicate messages to PURCHASES.QUEUE, they receive the following error:

```
CSQM166I <QMGR> CSQMAQLC QLOCAL(INQUIRIES.QUEUE) NOT AUTHORIZED
```

**Related concepts**

Streaming queues

## Multi LDAP 권한 부여

LDAP 권한 부여를 사용하여 로컬 사용자 ID에 대한 필요를 제거할 수 있습니다.

### 지원 플랫폼에서 LDAP 권한 부여의 가용성

LDAP 권한 부여는 멀티플랫폼에서 사용 가능합니다.

**주의:**

IBM MQ 9.0 GA(General Availability)부터 이 기능은 이전 릴리스에서 마이그레이션되었든 새 항목이든 상관없이 모든 큐 관리자에서 사용 가능합니다.

## LDAP 권한 부여의 개요

LDAP 권한 부여를 사용하면 **setmqaut** 및 **DISPLAY AUTHREC**와 같은 권한 부여 구성을 처리하는 명령이 식별 이름을 처리할 수 있습니다. 이전에는 사용자는 신임 정보를 로컬 운영 체제에서 사용자 및 그룹에 대해 존재하는 최대 사용 가능한 문자와 비교하여 인증되었습니다.

**주의: DEFINE AUTHINFO** 명령을 실행한 경우 큐 관리자를 재시작해야 합니다. 큐 관리자를 다시 시작하지 않으면 setmqaut 명령이 올바른 결과를 리턴하지 않습니다.

사용자가 식별 이름이 아니라 사용자 ID를 제공하는 경우에는 사용자 ID가 처리됩니다. 예를 들어, 채널에 PUTAUT(CTX)와 함께 수신 메시지가 있으면 사용자 ID의 문자는 LDAP 식별 이름으로 맵핑되고 적합한 권한 검사가 수행됩니다.

**DISPLAY CONN** 등과 같은 다른 명령은 해당 사용자 ID가 실제 로컬 OS에 존재하지 않을 수도 있지만 계속해서 작업하고 사용자 ID의 실제 값을 보여줍니다.

Linux AIX LDAP 권한이 있는 경우 큐 관리자는 qm.ini 파일의 **SecurityPolicy** 속성에 관계없이 AIX and Linux 플랫폼에서 항상 보안의 사용자 모델을 사용합니다. 따라서 개별 사용자에 대한 권한을 설정하면 해당 사용자에게만 영향을 미치고 해당 사용자의 그룹에 속하는 다른 사용자에게는 영향을 미치지 않습니다.

OS 모델에서처럼 사용자는 여전히 개별 및 사용자가 속하는 모든 그룹(있는 경우) 둘 모두에 지정된 결합된 권한을 가지고 있습니다.

예를 들어, 다음 레코드가 LDAP 저장소에 정의되었다고 가정하십시오.

- **inetOrgPerson** 클래스에서:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- **groupOfNames** 클래스에서:

```
dn="cn=Application Group A, ou=groups, o=yourcompany, c=yourcountry"
        longname=ApplicationGroupA [longer than 12 characters]
        members="cn=JaneDoe, ou=users, o=yourcompany, c=yourcountry",
                "cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
```

인증 용도를 위해 이 LDAP 서버를 사용하는 큐 관리자는 해당 **CONNAUTH** 값이 IDPWLDAP 유형의 **AUTHINFO** 오브젝트를 가리키고 관련 이름-해석 속성이 다음과 같이 적절하게 설정되도록 정의되어 있어야 합니다.

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

인증을 위해 이 구성이 제공되면 애플리케이션은 MQCNO 호출 내에서 사용된 CSPUserID 필드를 다음 값 세트 하나로 채울 수 있습니다.

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

또는

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

어느 경우든 시스템은 제공된 값을 사용하여 "jodoe"의 OS 컨텍스트를 인증할 수 있습니다.

## Multi 설정 권한 부여

권한 부여를 사용하기 위해 약칭이나 **USRFIELD**를 사용하는 방법입니다.

385 페이지의 『LDAP 권한 부여』에 설명된 여러 형식에 대해 작업하는 방법은 권한 부여 명령으로 계속되고 추가적으로 확장하여 shortname 또는 USRFIELD를 꾸밈없는 방식으로 사용할 수 있습니다.

문자열은 권한 부여를 위한 사용자(프린시펄)의 이름을 지정할 때 LDAP 레코드에 특정 속성을 지정합니다.

**중요사항:** 이 문자는 운영 체제 사용자 ID에 사용할 수 없으므로 문자열은 = 문자를 포함해서는 안됩니다.

잠재적으로 shortname인 권한 부여를 위해 프린시펄 이름을 OAM에 전달하는 경우에는 문자열은 12자에 맞아야 합니다. 맵핑 알고리즘은 먼저 LDAP 조회에서 SHORTUSR 속성을 사용하여 이를 DN으로 해석하려고 시도합니다.

이것이 UNKNOWN_ENTITY 오류로 실패하거나 제공된 문자열이 shortname일 수 없으면 USRFIELD 속성을 사용하여 LDAP 조회를 구성하려는 추가 시도가 작성됩니다.

**주의:** DEFINE AUTHINFO 명령을 실행한 경우 큐 관리자를 재시작해야 합니다. 큐 관리자를 다시 시작하지 않으면 setmqaut 명령이 올바른 결과를 리턴하지 않습니다.

사용자 권한 부여 처리를 위해 다음 setmqaut 명령 설정은 모두 동등합니다.

표 75. 사용자 권한 부여 설정

| 명령 | 참고 |
|---|---|
| `setmqaut -m QM -t qmgr -p jodoe +connect` | 이는 SHORTUSR을 통해 해석되는 평평한 완전하지 않은 이름입니다. |
| `setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect` | 또한 USRFIELD를 통해 동일 엔티티로 해석되는 평평한 완전하지 않은 이름입니다. |
| `setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect` | 이름 지정된 속성 사용. |
| `setmqaut -m QM -t qmgr -p "phone=1234567" +connect` | AUTHINFO 오브젝트에 구성된 속성일 필요가 없는 또 다른 이름 지정된 속성을 사용. |

SET AUTHREC MQSC 명령을 **setmqaut** 명령의 대안으로 사용할 수 있습니다.

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

또는 Set Authority Record (MQCMD_SET_AUTH_REC) PCF 명령을 다음 문자열을 포함하는 MQCACF_PRINCIPAL_ENTITY_NAMES 요소와 함께 사용하십시오.

```
"cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

그룹을 처리할 때 그룹 이름의 양식을 12자에 맞춰야 한다는 요구사항이 없기 때문에 shortname 처리에 대한 모호성이 없습니다. 그러므로 그룹의 SHORTUSR 속성에 상응하는 속성이 없습니다.

즉, 387 페이지의 표 76에 설명된 구문 예는 올바름을 의미하고 AUTHINFO 오브젝트를 확장된 속성과 함께 구성했고 다음으로 설정한 것으로 가정합니다.

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

표 76. 그룹 권한 부여 설정

| 명령 | 참고 |
|---|---|
| `setmqaut -m QM -t qmgr -g ApplicationGroupA +connect` | 해석하기 위해 GRPFIELD 사용 |
| `setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect` | 단일 속성 이름 지정 |

| 표 76. 그룹 권한 부여 설정 (계속) | |
|---|---|
| **명령** | **참고** |
| `setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect` | 전체 DN 사용 |

SET AUTHREC MQSC 명령을 앞의 **setmqaut** 명령에 대한 대안으로 사용할 수 있습니다.

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

또는 권한 레코드 설정(MQCMD_SET_AUTH_REC) PCF 명령을 다음 문자열을 포함하는 MQCACF_GROUP_ENTITY_NAMES 요소와 함께 사용하십시오.

```
 "ApplicationGroupA"
```

**중요사항:**

사용자나 그룹이건 이름을 참조하기 위해 사용하는 형식은 고유한 DN을 파생하는 것이 가능해야 합니다.

따라서 예를 들어, 둘 모두 "shortu=jodoe"를 가지고 있는 두 개의 개별 레코드가 있으면 안됩니다.

단일 고유 DN을 판별할 수 없으면 OAM은 MQRC_UNKNOWN_ENTITY를 리턴합니다.

Multi

## 권한 부여 표시

사용자 또는 그룹의 권한 부여를 표시하는 다양한 방법

### dspmqaut 명령

사용자 또는 그룹에 사용 가능한 권한 부여를 표시하기 위한 가장 단순한 방법은 dspmqaut 명령을 사용하는 것입니다.

사용자 또는 그룹을 식별하기 위해 구문 변화에 대한 조회를 사용할 수 있습니다. 명령 출력은 명령행에 제공된 형식으로 ID를 반복하는 점을 유의하십시오. 출력은 전체 해석된 DN에 대해서는 보고하지 않습니다.

예를 들면, 다음과 같습니다.

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

또는

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

### dmpmqaut 및 dmpmqcfg 명령

dmpmqaut 명령 및 해당 MQSC 또는 PCF와 동등한 명령은 386 페이지의 『설정 권한 부여』에 설명된 **setmqaut** 테이블과 같이 지원되는 형식으로 프린시펄 또는 그룹을 지정할 수 있습니다. 그러나 **dspmqaut**와 달리, **dmpmqaut** 명령은 항상 전체 DN을 보고합니다.

```
dmpmqaut -m QM -t qmgr -p jodoe
------------------------------------
profile: self
object type:qmgr
```

```
entity:cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry
entity type: principal
authority: connect
```

마찬가지로, 선택된 레코드에서 필터링이 없는 dmpmqcfg 명령은 항상 나중에 재실행할 수 있는 형식으로 전체 DN을 표시합니다.

```
dmpmqcfg -m QM -x authrec
------------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi

# LDAP 권한 부여 사용 시의 기타 고려사항

IBM MQ 9.0.0에서 LDAP 권한 부여를 사용할 때 알아야 하는 메시지 큐 인터페이스(MQI)와 기타 MQSC 및 PCF 명령의 변경사항에 대한 간략한 설명입니다.

## ADOPTCTX

애플리케이션이 인증 정보를 제공하거나 ADOPTCTX 속성을 YES로 설정할 요구사항이 없습니다.

애플리케이션이 명백하게 인증하지 않거나 **ADOPTCTX**가 활성 CONNAUTH 오브젝트에 대해 NO로 설정된 경우에는 운영 체제 사용자 ID로부터 애플리케이션과 연관된 ID 컨텍스트를 구합니다.

권한 부여를 적용해야 하는 할 때, 해당 컨텍스트는 setmqaut 명령에서와 같은 규칙을 사용하여 LDAP ID에 맵핑됩니다.

## MQI 호출에 대한 입력 매개변수

MQOPEN, MQPUT1 및 MQSUB에는 대체 사용자 ID를 지정할 수 있는 구조가 있습니다.

이러한 필드가 사용되면 12자 사용자 ID는 **setmqaut**, **dmpmqaut** 및 **dspmqaut** 명령에서와 같은 명령을 사용하여 DN에 맵핑됩니다.

MQPUT 및 MQPUT1은 또한 적합하게 권한이 부여된 프로그램이 MQMD UserIdentifier 필드를 설정하도록 허용합니다. 이 필드의 값은 PUT 프로세스 동안에 치안이 유지되지 않고 값으로 설정될 수도 있습니다.

그러나 평소처럼 **UserIdentifier** 값이 메시지 처리의 마지막 단계에서 권한 부여에 사용될 수 있습니다. 예를 들어, PUTAUT(CTX)가 수신 채널에 정의될 때입니다.

이 때 ID는 LDAP 또는 OS 기반일 수 있는 해당 수신 큐 관리자의 구성을 사용하여 권한 부여가 검사됩니다.

## MQI 호출에 대한 출력 매개변수

MQI 구조에서 프로그램에 사용자 ID가 제공될 때마다 이는 연결과 연관된 12자의 약칭 버전입니다.

예를 들어, API 엑시트의 **MQAXC.UserId** 값은 LDAP 맵핑으로부터 돌아온 약칭입니다.

## 기타 관리 MQSC 및 PCF 명령

DISPLAY CONN USERID 등과 같이 오브젝트 상태에서 사용자 정보를 보여주는 명령은 컨텍스트와 연관된 12자 약칭 이름을 리턴합니다. 전체 DN은 표시되지 않습니다.

CHLAUTH 맵핑 규칙 또는 채널의 MCAUSER 값과 같이 ID의 어설션을 허용하는 명령은 이러한 속성에 정의된 최대 길이까지의 값을 취합니다(현재 64자).

구문에는 변경이 없습니다. 해당 ID에 대해 권한 부여가 필요할 때 이는 **setmqaut**, **dmpmqaut** 및 **dspmqaut** 명령에서와 같은 규칙을 사용하여 DN에 내부적으로 맵핑됩니다.

즉 채널 정의에서 MCAUSER 값은 DISPLAY CHSTATUS와 같은 문자열을 표시하지 않지만 이들이 동일한 ID를 참조한다는 의미입니다.

예를 들면, 다음과 같습니다.

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

그런 다음 DISPLAY CHSTATUS(*) ALL은 모든 연결에 대해 SHORTUSR 값, *MCAUSER(jodoe)*를 표시합니다.

## Multi OS와 LDAP 권한 부여 모델 간 전환

서로 다른 플랫폼에서 서로 다른 권한 부여 메소드 간에 전환하는 방법입니다.

큐 관리자의 CONNAUTH 속성은 AUTHINFO 오브젝트를 가리킵니다. d오브젝트가 IDPWLDAP 유형이고 LDAP 저장소가 인증에 사용될 때.

이제 OS 기반 권한 부여를 계속할 수 있게 해주거나 LDAP 권한 부여에 대해 작업할 수 있게 해주는 동일 오브젝트에 권한 부여 메소드를 적용할 수 있습니다.

### IBM i, AIX and Linux

Linux IBM i AIX

큐 관리자는 언제든지 OS와 LDAP 모델 사이에서 스위치할 수 있습니다. 구성을 변경하고 REFRESH SECURITY TYPE (CONNAUTH) 명령을 사용하여 해당 구성을 활성으로 만들 수 있습니다.

예를 들어, 이 오브젝트가 이미 인증을 위한 연결 정보로 구성된 경우:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

### Windows

Windows

권한 구성 변경에 OS 및 LDAP 모델 간 전환이 포함된 경우 변경사항을 적용하려면 큐 관리자를 다시 시작해야 합니다. 그렇지 않으면 REFRESH SECURITY TYPE (CONNAUTH) 명령을 사용하여 변경사항을 활성화해야 합니다.

### 처리 규칙

OS에서 LDAP 권한 부여로 전환하면 설정된 기존 OS 권한이 비활성이 되고 표시되지 않게 됩니다.

**dmpmqaut** 등과 같은 명령은 이러한 OS 규칙을 표시하지 않습니다. 마찬가지로 LDAP에서 OS로 전환할 때 정의된 LDAP 권한 부여는 비활성이되고 보이지 않게 되고 원본 OS 규칙을 복원합니다.

어떤 이유로든 **dmpmqcfg** 명령을 사용하여 큐 관리자의 정의를 백업하고 싶은 경우에는 해당 백업에는 백업 시에 적용되는 권한 부여 메소드에 대해 정의된 규칙만을 포함할 것입니다.

## Multi LDAP 관리

각 플랫폼이 LDAP를 관리하는 방법의 개요.

LDAP 권한 부여를 사용할 때 운영 체제에서 mqm 그룹(또는 이에 상당하는 그룹)의 멤버십은 그다지 중요하지 않습니다. 해당 그룹의 구성원은 특정 명령행 명령을 처리할 수 있는지 여부만을 제어합니다.

특히, strmqm 및 endmqm 명령을 실행하려면 해당 그룹에 있어야 합니다.

일단 큐 관리자가 실행 중이면 이제는 완전한 권한이 있는 계정에는 한계가 있습니다. **strmqm** 명령을 실행한 사용자의 사용자 ID와는 별개로, OS mqm(또는 이에 상당하는) 그룹에 속하는 다른 사용자는 특수 권한을 받지 않습니다.

다른 사용자의 권한 부여는 이들이 속하는 LDAP 그룹에 따라 다릅니다. **setmqaut** 등과 같은 명령에서 mqm 그룹 이름을 자격 없이 사용하면 LDAP 그룹에 맵핑하도록 허용되지 않습니다.

### AIX and Linux

Linux AIX

일단 큐 관리자가 실행 중이면 자동으로 완전한 권한이 있는 계정만이 큐 관리자를 시작한 실제 사용자입니다.

mqm ID가 여전히 존재하고 파일 등과 같은 OS 자원의 소유자로서 사용됩니다. mqm은 큐 관리자를 실행 중인 유효 ID이기 때문입니다. 그러나 mqm 사용자는 OAM에 의해 제어되는 관리 태스크를 자동으로 수행할 수 없을 것입니다.

### Windows

Windows

Windows에서 자동으로 전체 권한이 제공되는 계정은 큐 관리자를 시작한 OS 사용자와 코어 큐 관리자 프로세스를 실행하는 사용자입니다. 예를 들어, 큐 관리자가 Windows 서비스로 시작된 경우의 MUSR_MQADMIN입니다.

LDAP 권한 부여 모드에서 실행할 때 Windows는 AIX and Linux 플랫폼과 매우 유사하게 작동합니다. 이는 12자 축약 이름 및 전체 DN을 다룹니다.

### IBM i

IBM i

IBM i에서 자동으로 권한이 있는 계정은 큐 관리자 및 QMQM ID를 시작한 계정입니다.

큐 관리자를 시작한 사용자 ID는 시스템을 시작하기 위해서만 필요하므로 두 ID 모두 필요합니다. 일단 실행 중이면 큐 관리자 프로세스에는 QMQM 권한만이 있습니다.

### MQADMIN 권한을 제공하는 샘플 스크립트

Linux AIX

큐 관리자에서 전체 관리를 수행할 수 있는 그룹이 있는 것이 유용하므로 샘플 스크립트는 AIX and Linux 플랫폼에 다음과 같이 탑재되어 있습니다.

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

이 샘플은 두 개의 매개변수를 사용합니다.

- 큐 관리자 이름
- LDAP 그룹 이름

샘플은 setmqaut 명령을 처리하여 모든 오브젝트에 전체 권한을 부여합니다. 이는 관리 역할을 위해 IBM MQ Explorer OAM 마법사가 생성한 것과 동일한 스크립트입니다. 예를 들어, 코드는 다음과 같이 시작됩니다.

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

## 메시지의 기밀성

메시지를 암호화하면 메시지의 컨텐츠가 기밀로 유지됩니다. 사용자의 필요에 따라 IBM MQ에서 메시지를 암호화하는 다양한 방법이 있습니다.

포인트-투-포인트 메시징 인프라에 대해 애플리케이션 레벨의 엔드-투-엔드 데이터 보호가 필요한 경우 Advanced Message Security를 사용하여 메시지를 암호화하거나 자체 API 엑시트 또는 API 교차 엑시트를 작성할 수 있습니다.

가장 안전한 솔루션은 애플리케이션이 메시지를 배치한 위치에서 이용 애플리케이션이 가져오는 위치까지 메시지를 암호화하는 엔드 투 엔드 암호화를 제공하는 것입니다. 이 방법은 AMS(101 페이지의 『Advanced Message Security 계획』)를 사용하거나 API 엑시트 또는 API 교차 엑시트를 작성하여 수행할 수 있습니다. 자세한 정보는 436 페이지의 『사용자 엑시트 프로그램에서 기밀성 구현』의 내용을 참조하십시오.

네트워크를 통해 전송되는 동안에만 메시지를 암호화해야 하는 경우에는 TLS를 사용할 수 있습니다. 자세한 정보는 22 페이지의 『IBM MQ의 TLS 보안 프로토콜』의 내용을 참조하십시오. 또는 사용자 고유의 보안 엑시트, 메시지 엑시트 또는 송신 및 수신 엑시트 프로그램을 작성하여 암호화를 수행할 수 있습니다.

z/OS 큐 관리자에서 저장 메시지를 암호화해야 하는 경우 해당 큐 관리자에서 z/OS 데이터 세트 암호화를 사용할 수 있습니다. 자세한 정보는 437 페이지의 『Confidentiality for data at rest on IBM MQ for z/OS with data set encryption』의 내용을 참조하십시오.

**관련 태스크**

TLS를 사용하여 두 개의 큐 관리자 연결

클라이언트를 큐 관리자에 안전하게 연결

## CipherSpec 사용 가능

**DEFINE CHANNEL** 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec을 사용 가능으로 설정합니다.

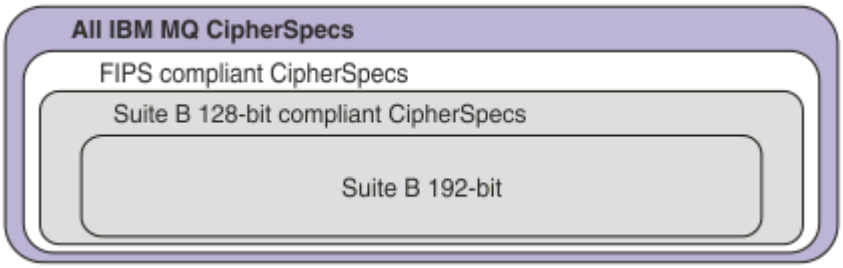
**참고:** AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

IBM MQ에 사용할 수 있는 CipherSpec 중 일부는 FIPS를 준수합니다. FIPS 준수 CipherSpec 중 일부는 스위트 B도 준수하지만 TLS_RSA_WITH_3DES_EDE_CBC_SHA와 같은 일부는 이를 준수하지 않습니다.

모든 스위트 B 준수 CipherSpec은 FIPS도 준수합니다. 모든 스위트 B 준수 CipherSpec은 두 개의 그룹에 해당합니다. 128비트(예: ECDHE_ECDSA_AES_128_GCM_SHA256) 및 192비트(예: ECDHE_ECDSA_AES_256_GCM_SHA384)입니다.

다음 다이어그램은 이러한 서브세트 간의 관계를 설명합니다.




제품은 모든 플랫폼에서 TLS 1.3 보안 프로토콜을 지원합니다.

이러한 각 플랫폼에서 사용할 수 있는 CipherSpec은 393 페이지의 표 77에 나열됩니다. 이러한 CipherSpec을 사용하는 데 대한 정보는 395 페이지의 『IBM MQ에서 TLS 1.3 사용』 및 396 페이지의 『IBM MQ MQI client와 TLS 1.3』의 내용을 참조하십시오.

구성과 향후 마이그레이션을 쉽게 수행하도록 IBM MQ에서는 알리어스 CipherSpec 세트도 제공합니다. 알리어스 CipherSpec을 사용하도록 기존 보안 구성을 마이그레이션하는 것은 향후 추가적으로 구성을 변경할 필요 없이 암호 추가 및 삭제에 적응할 수 있음을 의미합니다. 이러한 알리어스 CipherSpec은 393 페이지의 표 77의 알리어스 CipherSpec 섹션에 나열되어 있습니다. 알리어스 CipherSpec을 사용하도록 마이그레이션하는 데 관한 자세한 정보는 알리어스 CipherSpec을 사용하도록 기존 보안 구성 마이그레이션을 참조하십시오.

396 페이지의 『IBM MQ에서 사용으로 설정된 기본 CipherSpec 값』에서 설명한 대로, 기본 CipherSpec을 구성할 수 있습니다. 또는 다음에서 채널과 함께 사용할 수 있는 대체 CipherSpec 세트도 제공할 수 있습니다.

- Multi IBM MQ for Multiplatforms(404 페이지의 『IBM MQ for Multiplatforms에서 정렬된 사용 가능한 CipherSpec의 사용자 정의 목록 제공』에서 설명).
- z/OS IBM MQ for z/OS(405 페이지의 『IBM MQ for z/OS에서 정렬된 사용 가능한 CipherSpec의 사용자 정의 목록 제공』에서 설명).

IBM MQ에서 사용하기 위해 다시 사용 가능하게 할 더 이상 사용되지 않는 CipherSpec이 필요한 경우 406 페이지의 『더 이상 사용되지 않는 CipherSpec』에 나열됩니다.

## IBM MQ TLS 지원과 함께 사용할 수 있는 CipherSpec

IBM MQ 큐 관리자와 함께 사용할 수 있는 암호 스펙은 다음 표에 자동으로 나열됩니다. 개인 인증서를 요청할 때 공용 및 개인 키 쌍의 키 크기를 지정합니다. TLS 데이터 교환 동안에 사용되는 키 크기는 인증서에 저장된 크기입니다(표에 명시된 것처럼 CipherSpec으로 판별되는 경우는 제외).

표 *77. IBM MQ TLS* 지원과 함께 사용할 수 있는 *CipherSpec*

| 플랫폼 지원[395 페이지의 『1』] | CipherSpec 이름 | 16진 코드 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 (암호화 비트) | FIPS [395 페이지의 『2』] | 스위트 B |
|---|---|---|---|---|---|---|---|
| **알리어스 CipherSpec** | | | | | | | |
| 모두 | ANY_TLS13_OR_HIGHER [395 페이지의 『3』] [395 페이지의 『4』] | 해당사항 없음 | 조정됨 | 조정됨 | 조정됨 | 조정됨 | 조정됨 |
| 모두 | ANY_TLS13 [395 페이지의 『4』] [395 페이지의 『5』] | 해당사항 없음 | TLS 1.3 | 조정됨 | 조정됨 | 조정됨 | 조정됨 |
| 모두 | ANY_TLS12_OR_HIGHER [395 페이지의 『4』] [395 페이지의 『6』] | 해당사항 없음 | 조정됨 | 조정됨 | 조정됨 | 조정됨 | 조정됨 |
| 모두 | ANY_TLS12 [395 페이지의 『7』] | 해당사항 없음 | TLS 1.2 | 조정됨 | 조정됨 | 조정됨 | 조정됨 |
| 모두 | ANY [395 페이지의 『8』] | 해당사항 없음 | 조정됨 | 조정됨 | 조정됨 | 조정됨 | 조정됨 |
| **CipherSpecs for TLS 1.3** | | | | | | | |
| 모두 | TLS_AES_128_GCM_SHA256 | 1301 | TLS 1.3 | GCM | AES-128 with GCM (128) | 예 | 아니오 |
| 모두 | TLS_AES_256_GCM_SHA384 | 1302 | TLS 1.3 | GCM | AES-256 with GCM (256) | 예 | 아니오 |
| 모두 | TLS_CHACHA20_POLY1305_SHA256 | 1303 | TLS 1.3 | POLY1305 | CHACHA20 (256) | 아니오 | 아니오 |
| ALW | TLS_AES_128_CCM_SHA256 | 1304 | TLS 1.3 | CBC-MAC | CTR (128)이 있는 AES-128 | 예 | 아니오 |
| ALW | TLS_AES_128_CCM_8_SHA256 [395 페이지의 『10』] | 1305 | TLS 1.3 | CBC-MAC | CTR (128)이 있는 AES-128 | 예 | 아니오 |
| **CipherSpecs for TLS 1.2** | | | | | | | |

| 표 77. *IBM MQ TLS* 지원과 함께 사용할 수 있는 *CipherSpec (계속)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **플랫폼 지원**395 페이지의 『1』 | **CipherSpec 이름** | **16진 코드** | **사용되는 프로토콜** | **MAC 알고리즘** | **암호화 알고리즘 (암호화 비트)** | **FIPS**395 페이지의 『2』 | **스위트 B** |
| 모두 | TLS_RSA_WITH_AES_128_CBC_SHA256395 페이지의 『9』 | 003C | TLS 1.2 | SHA-256 | AES (128) | 예 | 아니오 |
| 모두 | TLS_RSA_WITH_AES_256_CBC_SHA256 395 페이지의 『9』 395 페이지의 『11』 | 003D | TLS 1.2 | SHA-256 | AES (256) | 예 | 아니오 |
| 모두 | TLS_RSA_WITH_AES_128_GCM_SHA256 395 페이지의 『9』 395 페이지의 『12』 | 009C | TLS 1.2 | SHA-256 과 AEAD GCM | AES (128) | 예 | 아니오 |
| 모두 | TLS_RSA_WITH_AES_256_GCM_SHA384395 페이지의 『9』 395 페이지의 『11』 395 페이지의 『12』 | 009D | TLS 1.2 | SHA-384 와 AEAD GCM | AES (256) | 예 | 아니오 |
| 모두 | ECDHE_ECDSA_AES_128_CBC_SHA256 395 페이지의 『9』 | C023 | TLS 1.2 | SHA-256 | AES (128) | 예 | 아니오 |
| 모두 | ECDHE_ECDSA_AES_256_CBC_SHA384 395 페이지의 『9』 395 페이지의 『11』 | C024 | TLS 1.2 | SHA-384 | AES (256) | 예 | 아니오 |
| 모두 | ECDHE_RSA_AES_128_CBC_SHA256 395 페이지의 『9』 | C027 | TLS 1.2 | SHA-256 | AES (128) | 예 | 아니오 |
| 모두 | ECDHE_RSA_AES_256_CBC_SHA384 395 페이지의 『9』 395 페이지의 『11』 | C028 | TLS 1.2 | SHA-384 | AES (256) | 예 | 아니오 |
| Multi | ECDHE_ECDSA_AES_128_GCM_SHA256 395 페이지의 『11』 395 페이지의 『12』 | C02B | TLS 1.2 | SHA-256 과 AEAD GCM | AES (SHA384) | 예 | 128 비트 |
| Multi | ECDHE_ECDSA_AES_256_GCM_SHA384 395 페이지의 『11』 395 페이지의 『12』 | C02C | TLS 1.2 | SHA-384 와 AEAD GCM | AES (SHA384) | 예 | 192 비트 |
| 모두 | ECDHE_RSA_AES_128_GCM_SHA256 395 페이지의 『12』 | C02F | TLS 1.2 | SHA-256 과 AEAD GCM | AES (128) | 예 | 아니오 |
| 모두 | ECDHE_RSA_AES_256_GCM_SHA384 395 페이지의 『11』 395 페이지의 『12』 | C030 | TLS 1.2 | AEAD AES-128 GCM | AES (SHA384) | 예 | 아니오 |

표 77. *IBM MQ TLS* 지원과 함께 사용할 수 있는 *CipherSpec (계속)*

| 플랫폼 지원[395 페이지의 『1』] | CipherSpec 이름 | 16진 코드 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 (암호화 비트) | FIPS [395 페이지의 『2』] | 스위트 B |
|---|---|---|---|---|---|---|---|

**참고:**

1. 각 플랫폼 아이콘에 포함되는 플랫폼 목록은 제품 문서에서 사용되는 아이콘을 참조하십시오.
2. FIPS 인증 플랫폼에서 CipherSpec이 FIPS 인증 CipherSpec인지 여부를 지정합니다. FIPS에 대한 설명은 FIPS(Federal Information Processing Standards)를 참조하십시오.
3. ALW ANY_TLS13_OR_HIGHER 알리어스 CipherSpec은 원격 끝이 허용할 것이나 TLS 1.3 이상의 프로토콜을 사용해서만 연결되는 최상위 레벨의 보안을 협상합니다.
4. IBM i IBM i에서 TLS 1.3 또는 ANY CipherSpec을 사용하려면 기반 운영 체제 버전이 TLS 1.3을 지원해야 합니다. 자세한 정보는 TLSv1.3에 대한 시스템 TLS 지원을 참조하십시오.
5. ALW ANY_TLS13 알리어스 CipherSpec은 각 플랫폼에 대해 이 표에 나열된 대로 TLS 1.3 프로토콜을 사용하는 허용 가능한 CipherSpec의 서브세트를 나타냅니다.
6. ALW ANY_TLS12_OR_HIGHER 알리어스 CipherSpec은 원격 끝이 허용할 것이나 TLS 1.2 이상의 프로토콜을 사용해서만 연결되는 최상위 레벨의 보안을 협상합니다.
7. ANY_TLS12 CipherSpec은 각 플랫폼에 대해 이 표에 나열된 대로 TLS 1.2 프로토콜을 사용하는 허용 가능한 CipherSpec의 서브세트를 나타냅니다.
8. ALW ANY 알리어스 CipherSpec은 원격 끝이 허용할 최상위 레벨의 보안을 협상합니다.
9. IBM i 이러한 CipherSpec은 시스템 값 QSSLCSLCTL이 *OPSSYS로 설정된 IBM i 7.4 시스템에서 사용 가능하지 않습니다.
10. ALW 이러한 CipherSpec은 16-옥텟 무결성 검사 값(ICV) 대신에 8-옥텟 ICV를 사용합니다.
11. 탐색기가 사용하는 JRE에 제한 없는 적절한 정책 파일이 적용되지 않는 한, 이 CipherSpec을 사용하여 IBM MQ Explorer에서 큐 관리자로의 연결을 보호할 수 없습니다.
12. ALW Following a recommendation by GSKit, TLS 1.2 GCM CipherSpecs have a restriction which means that after 2^24.5 TLS records are sent, using the same session key, the connection is terminated with message AMQ9288E. 이 GCM 제한사항은 사용 중인 FIPS 모드에 관계없이 활성화됩니다.

    이 오류가 발생하지 않도록 하려면 TLS 1.2 GCM 암호를 사용하지 않고 비밀 키 재설정을 사용으로 설정하거나 환경 변수 GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE 가 설정된 IBM MQ 큐 관리자 또는 클라이언트를 시작하십시오. GSKit 라이브러리의 경우, 연결의 양쪽에 이 환경 변수를 설정하고 이 환경 변수를 큐 관리자 연결 및 큐 관리자 연결에 큐 관리자 연결에 적용해야 합니다. 이 설정은 관리되지 않는 .NET 클라이언트에 영향을 주지만 Java 또는 관리 .NET 클라이언트에는 영향을 주지 않습니다. 자세한 정보는 AES-GCM 암호 제한을 참조하십시오.

    z/OS 이 제한은 IBM MQ for z/OS에는 적용되지 않습니다.

## IBM MQ에서 TLS 1.3 사용

제품은 모든 플랫폼에서 TLS 1.3 을 지원합니다.

IBM MQ 9.2.0 이상에서 작성된 큐 관리자는 기본적으로 TLS 1.3을 지원합니다. IBM MQ의 이전 버전에서 마이그레이션된 큐 관리자에서는 TLS 1.3을 사용 가능하게 해야 합니다. **AllowTLSV13**=TRUE 특성을 설정하여 마이그레이션된 큐 관리자에서 TLS 1.3을 사용 가능하게 할 수 있습니다.

- Multi IBM MQ for Multiplatforms 큐 관리자의 경우 qm.ini 파일을 편집하고 SSL 스탠자 아래에 **AllowTLSV13**=TRUE 특성을 추가하십시오.

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS IBM MQ for z/OS 큐 관리자의 경우 큐 관리자 시동 JCL에 지정된 QMINI 데이터 세트를 편집하고 TransportSecurity 아래에 **AllowTLSV13**=TRUE 특성을 추가하십시오.

```
TransportSecurity:
    AllowTLSV13=TRUE
```

TLS 1.3이 사용 가능하면 TLS 1.3 스펙에 따라 취약한 CipherSpec와 통신하려고 하면 IBM MQ에서 사용 가능 여부에 상관없이 해당 시도는 거부됩니다. TLS 1.3이 약한 것으로 간주하는 CipherSpec은 다음 기준을 하나 이상 충족합니다.

- SSL 3.0 프로토콜을 사용합니다.
- RC4 또는 RC2를 암호화 알고리즘으로 사용합니다.
- 암호화 키 크기(비트)가 112 이하입니다.

이러한 제한사항은 더 이상 사용되지 않는 CipherSpec에 대한 표 1에 참고 [3]으로 플래그 지정되어 있습니다.

이러한 CipherSpec을 계속해서 사용해야 하는 경우에는 TLS 1.3 모드를 사용 안함으로 설정해야 합니다.

- ALW 큐 관리자의 qm.ini 파일을 편집하고 **AllowTLSV13** 특성의 설정을 다음으로 변경하십시오.

```
SSL:
  AllowTLSV13=FALSE
```

- z/OS 큐 관리자의 QMINI 데이터 세트를 편집하고 **AllowTLSV13** 특성의 설정을 다음으로 변경하십시오.

```
TransportSecurity:
    AllowTLSV13=FALSE
```

## IBM MQ MQI client와 TLS 1.3

ALW

IBM MQ MQI client를 사용하는 경우, **AllowTLSV13**의 값은 애플리케이션에서 사용 중인 mqclient.ini 파일의 SSL 스탠자에 명시적으로 지정되지 않은 한 다음과 같이 추정됩니다.

- 약한 CipherSpec이 사용으로 설정된 경우에는 **AllowTLSV13**이 FALSE로 설정되며 TLS 1.3 CipherSpec을 사용할 수 없습니다.
- 그 외의 경우에는 **AllowTLSV13**이 TRUE로 설정되며 새 TLS 1.3 CipherSpec 및 알리어스 CipherSpec을 사용할 수 있습니다.

## IBM MQ에서 사용으로 설정된 기본 CipherSpec 값

새 IBM MQ 큐 관리자의 기본 구성에서는 IBM MQ가 TLS 1.2 및 TLS 1.3 프로토콜과 CipherSpec을 사용하는 다양한 암호화 알고리즘에 대한 지원을 제공합니다. 호환성을 목적으로 IBM MQ가 SSL 3.0 및 TLS 1.0 프로토콜과 보안 취약성에 민감하거나 취약하다고 알려진 여러 암호화 알고리즘을 사용하도록 구성할 수도 있습니다. 기본 구성에서 사용 가능한 CipherSpecs 의 목록은 유지보수를 적용하여 변경할 수 있습니다.

다음 제어를 사용하여 CipherSpec의 사용을 제한하거나 허용하기 위해 IBM MQ를 구성할 수 있습니다.

- SSLFIPS를 사용하는 FIPS 140-2 준수 CipherSpec만 허용.
- ALW SUITEB를 사용하는 NSA Suite B 준수 CipherSpec만 허용.

- Multi **AllowedCipherSpecs**를 사용하여 사용자 정의 CipherSpec 목록을 허용합니다.
- ALW **AMQ_ALLOWED_CIPHERS** 환경 변수를 사용하여 사용자 정의 CipherSpec 목록을 허용합니다.
- ALW **AllowWeakCipher** 또는 **AMQ_SSL_WEAK_CIPHER_ENABLE** 환경 변수를 사용하여 더 이상 사용되지 않는 CipherSpec의 사용을 허용합니다.
- z/OS CHINIT JCL의 DD 문을 사용하여 더 이상 사용되지 않는 CipherSpec의 사용을 허용.

**참고: AllowedCipherSpecs** 또는 **AMQ_ALLOWED_CIPHERS**를 사용하여 CipherSpec의 사용자 정의 목록을 지정하면 이는 더 이상 사용되지 않는 CipherSpec의 사용을 대체합니다. 사용자 정의 CipherSpec 목록과 함께 NSA Suite B 또는 FIPS 140-2 제한사항을 사용할 경우, 사용자 정의 목록에 Suite B 또는 FIPS 140-2 설정에서 허용하는 CipherSpec만 포함되어 있는지 확인하십시오.

**관련 개념**

43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』
이 토픽은 IBM MQ에서 CipherSpec과 디지털 인증서 사이의 관계를 소개하여 보안 정책에 대해 적합한 CipherSpec 및 디지털 인증서를 선택하는 방법에 대한 정보를 제공합니다.

20 페이지의 『CipherSpecs 및 CipherSuites』
암호화 보안 프로토콜은 안전 연결에서 사용되는 알고리즘에 동의해야 합니다. CipherSpec 및 CipherSuite는 특정 알고리즘 결합을 정의합니다.

41 페이지의 『스위트 B에 대해 IBM MQ 구성』
IBM MQ은(는) AIX, Linux, and Windows 플랫폼에서 NSA Suite B 표준을 준수하여 작동하도록 구성할 수 있습니다.

31 페이지의 『FIPS(Federal Information Processing Standard)』
이 주제에서는 미국 국립 표준 기술국의 FIPS(Federal Information Processing Standard) Cryptomodule Validation Program 및 TLS 채널에서 사용할 수 있는 암호화 기술을 소개합니다.

**관련 태스크**

알리어스 CipherSpe를 사용하도록 기존 보안 구성 마이그레이션

**관련 참조**

DEFINE CHANNEL

ALTER CHANNEL

채널 변경, 복사 및 작성

## ALW AES-GCM 암호 제한

TLS 암호화에 사용될 때 AES-GCM 암호에 부과되는 제한사항에 대한 안내서입니다. 이러한 제한사항은 IETF및 NIST 조직에 의해 부과되며 AES-GCM 암호를 사용할 때 3개이상의$^{24.5}$ TLS 레코드를 안전하게 전송하는 데 동일한 세션 키를 사용하지 않아야 합니다.

이러한 제한사항에 대한 자세한 정보는 RFC 9325섹션 4.4 키 사용 한계 및 RFC 8446섹션 5.5를 참조하십시오.

IBM MQ 는 암호화 기능을 직접 구현하지 않습니다. 대신 여러 다른 암호화 라이브러리를 사용하여 TLS및 Advanced Message Security 기능을 제공합니다. Windows, Linux및 AIX 운영 체제에서 IBM MQ 가 사용하는 암호화 라이브러리는 IBM Global Security Kit (GSKit)입니다. 애플리케이션의 경우 C및 비관리 .NET 라이브러리는 암호화 기능을 위해 GSKit 를 사용합니다. GSKit 에 의한 AES-GCM 암호화 알고리즘의 구현에는 표준 그룹에 의해 지정되는 제한사항이 포함됩니다. 또한 이러한 제한사항은 기본적으로 사용 가능합니다. 이와 같이 IBM MQ TLS 통신은 AES-GCM 암호를 사용할 때 동일한 세션 키를 사용하여 3개이상의$^{24.5}$ TLS 레코드가 전송되면 종료됩니다.

**참고:** 다른 암호화 라이브러리가 사용되고 이러한 라이브러리가 동일한 제한사항을 구현하지 않았기 때문에 IBM i, IBM Z 또는 IBM MQ for HPE NonStop 플랫폼 또는 Java/JMS관리 .NET 애플리케이션에는 이 제한사항이 없습니다.

IBM MQ 채널이 동일한 세션 키를 사용하여 3개이상의$^{24.5}$ TLS 레코드가 전송될 수 있을 정도로 오랫동안 실행 중인 상태로 남아 있는 경우 기본 암호화 라이브러리가 연결을 종료합니다. 이로 인해 채널이 종료되고 AMQ9288E 오류 메시지가 생성됩니다. 이러한 방식으로 통신이 종료된 애플리케이션은 수행 중인 IBM MQ 조작에서 `MQRC_CONNECTION_BROKEN` 리턴 코드를 수신합니다.

연결 종료는 통신의 양쪽 끝에서 수행할 수 있지만 암호화 기능을 위해 GSKit 를 사용하는 끝에서만 수행할 수 있습니다.

### 제한사항 완화를 위한 조언

이 제한사항으로 인해 종료되는 통신을 방지하거나 처리하는 방법에 대한 일부 옵션은 다음과 같습니다.

**다시 연결 가능한 클라이언트 사용**
연결이 실패하는 경우 자동으로 다시 연결을 시도하도록 애플리케이션을 구성할 수 있습니다. 여기에는 GCM 제한사항으로 인해 종료된 연결이 포함됩니다. 다시 연결하도록 구성된 경우, 클라이언트 애플리케이션은 임의의 실패 지점에서 자동으로 복원되고 오브젝트를 열기 위한 모든 핸들이 복원됩니다. 이는 애플리케이션 코드로 돌아가지 않고 수행됩니다.

자세한 정보는 자동 클라이언트 다시 연결을 참조하십시오.

**비밀 키 재설정 값 설정**
IBM MQ 는 구성 가능한 바이트 수가 채널을 통해 전송된 후 세션 키 재설정을 요청하도록 구성될 수 있습니다. 이 한계에 도달하면 IBM MQ 는 암호화 계층이 세션 키 재설정을 수행하여 새 세션 키가 되도록 요청합니다.

지정된 값은 전송되는 바이트 수이며, 이는 IBM MQ에서 전송되는 메시지의 크기와 관련됩니다. 전송되는 TLS 레코드 수에 대한 제한이 있습니다. TLS 레코드는 네트워크의 최대 전송 단위 (MTU) 에 따라 최대 바이트 수를 전송할 수 있으므로 메시지 바이트와 TLS 레코드 간에 직접 맵핑이 없습니다. 이 값보다 큰 전송된 메시지는 다중 TLS 레코드로 전송됩니다. MTU값은 네트워크 간에 다양합니다. 또한 IBM MQ 메시지 데이터 전송 외부에서 TLS 레코드를 전송해야 하는 다른 이유가 있습니다 (예: IBM MQ 하트비트 검사, TLS 경보, 기타 IBM MQ 프로토콜 메시지). 이러한 추가 TLS 레코드는 최대 TLS 레코드 수에 계수되지만 IBM MQ 비밀 키 재설정 값에서 계수되지 않습니다.

비밀 키 재설정을 사용하여 정기적으로 세션 키를 재설정하면 AES-GCM 제한으로 인해 채널이 종료되지 않을 수 있습니다.

자세한 정보는 SSL및 TLS 비밀 키 재설정을 참조하십시오.

**TLS 1.3 cipherspecs 사용**
TLS 1.3 프로토콜을 사용할 때 AES-GCM 제한사항이 여전히 존재하는 반면, TLS 1.3 프로토콜은 TLS 통신을 인터럽트할 필요 없이 자동으로 세션 키 재설정을 수행하도록 지원합니다. 이를 통해 GSKit 는 IBM MQ 에서 비밀 키 재설정을 요청하지 않고도 필요한 경우 세션 키 재설정을 관리할 수 있습니다.

자세한 정보는 392 페이지의 『CipherSpec 사용 가능』 의 IBM MQ 에서 TLS 1.3 사용 의 내용을 참조하십시오.

**AES-GCM 제한 사용 안함**
필요한 경우 AES-GCM 제한을 사용하지 않도록 환경 변수 **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** 를 설정하여 제한을 사용 안함으로 설정할 수 있습니다. 이를 수행하면 동일한 세션 키를 사용하여 임의의 수의 TLS 레코드를 전송할 수 있습니다. 이 완화를 선택하는 경우 보안 통신을 위해 GSKit 를 사용하는 통신의 각 끝에 환경 변수를 설정해야 합니다.

**경고:** 이 옵션은 세 개 이상의$^{24.5}$ TLS 레코드가 전송된 후에는 공격자가 전송된 레코드에 대한 분석을 수행하여 사용 중인 세션 키를 판별할 수 있으므로 권장되지 않습니다. 세션 키가 판별되면 해당 세션 키를 사용하는 모든 기존 및 향후 통신이 손상됩니다.

## TLS 데이터 교환에서 CipherSpec 순서

CipherSpec의 순서는 예를 들어, ANY* CipherSpec 중 하나를 사용할 때 여러 개의 가능한 CipherSpec 사이에서 선택할 때 사용됩니다.

TLS 데이터 교환 중 클라이언트 및 서버는 환경 설정의 순서에 따라 지원하는 CipherSpec 및 프로토콜을 교환합니다. 양측에서 우선순위를 지정한 공통 CipherSpec이 선택되고 TLS 통신에 사용됩니다. CipherSpec 프로토콜을 선택하는 경우 버전도 고려합니다. 예를 들어, 서버가 TLS 1.3 CipherSpec 이전의 TLS 1.2 CipherSpec을 나열하는 경우 클라이언트가 TLS 1.3을 지원하고 사용 가능한 공통 TLS 1.3 CipherSpec이 있는 한, TLS 1.3의 우선시할 수 있습니다.

IBM MQ 가 TLS에 대해 구성되면 CipherSpecs 를 다음 표에 표시된 순서대로 설정합니다 (가장 선호되는 순서에서 가장 선호되지 않는 순서로).

**참고: AllowedCipherSpecs** 속성을 통해 CipherSpec이 사용 가능하지 않으면 TLS 데이터 교환 중에 사용하도록 구성되지 않습니다.

**AllowedCipherSpecs** 속성이 지정되지 않은 경우 다음 표에 표시되는 사용 가능한 암호의 기본 목록이 사용됩니다.

표 *78. IBM MQ 9.2.0*의 *CipherSpec* 순서

| 플랫폼 | CipherSpec | 프로토콜 | 16진 코드 | 기본적으로 사용됨 |
|---|---|---|---|---|
| 모두 | TLS_CHACHA20_POLY1305_SHA256 | TLS 1.3 | 1303 | 예 |
| 모두 | TLS_AES_256_GCM_SHA384 | TLS 1.3 | 1302 | 예 |
| 모두 | TLS_AES_128_GCM_SHA256 | TLS 1.3 | 1301 | 예 |
| ALW | TLS_AES_128_CCM_SHA256 | TLS 1.3 | 1304 | 예 |
| ALW | TLS_AES_128_CCM_8_SHA256 | TLS 1.3 | 1305 | 예 |
| 모두 | TLS_RSA_WITH_AES_256_GCM_SHA384 | TLS 1.2 | 009D | 예 |
| Multi | ECDHE_ECDSA_AES_256_GCM_SHA384 | TLS 1.2 | C02C | 예 |
| 모두 | ECDHE_RSA_AES_256_GCM_SHA384 | TLS 1.2 | C030 | 예 |
| 모두 | TLS_RSA_WITH_AES_256_CBC_SHA256 | TLS 1.2 | 003D | 예 |
| 모두 | ECDHE_ECDSA_AES_256_CBC_SHA384 | TLS 1.2 | C024 | 예 |
| 모두 | ECDHE_RSA_AES_256_CBC_SHA384 | TLS 1.2 | C028 | 예 |
| 모두 | TLS_RSA_WITH_AES_128_GCM_SHA256 | TLS 1.2 | 009C | 예 |
| Multi | ECDHE_ECDSA_AES_128_GCM_SHA256 | TLS 1.2 | C02B | 예 |
| 모두 | ECDHE_RSA_AES_128_GCM_SHA256 | TLS 1.2 | C02F | 예 |
| 모두 | TLS_RSA_WITH_AES_128_CBC_SHA256 | TLS 1.2 | 003C | 예 |
| 모두 | ECDHE_ECDSA_AES_128_CBC_SHA256 | TLS 1.2 | C023 | 예 |

| 표 78. IBM MQ 9.2.0의 CipherSpec 순서 (계속) | | | | |
|---|---|---|---|---|
| **플랫폼** | **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
| 모두 | ECDHE_RSA_AES_128_CBC_SHA256 | TLS 1.2 | C027 | 예 |
| ALW | ECDHE_ECDSA_3DES_EDE_CBC_SHA256 | TLS 1.2 | C008 | 아니오 |
| Multi | ECDHE_RSA_3DES_EDE_CBC_SHA256 | TLS 1.2 | C012 | 아니오 |
| ALW | TLS_RSA_WITH_RC4_128_SHA256 | TLS 1.2 | 0005 | 아니오 |
| ALW | ECDHE_ECDSA_RC4_128_SHA256 | TLS 1.2 | C007 | 아니오 |
| Multi | ECDHE_RSA_RC4_128_SHA256 | TLS 1.2 | C011 | 아니오 |
| 모두 | TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | 003B | 아니오 |
| ALW | ECDHE_ECDSA_NULL_SHA256 | TLS 1.2 | C006 | 아니오 |
| Multi | ECDHE_RSA_NULL_SHA256 | TLS 1.2 | C010 | 아니오 |
| ALW | TLS_RSA_WITH_NULL_NULL | TLS 1.2 | 0000 | 아니오 |
| ALW z/OS | TLS_RSA_WITH_AES_256_CBC_SHA | TLS 1.0 | 0035 | 아니오 |
| ALW z/OS | TLS_RSA_WITH_AES_128_CBC_SHA | TLS 1.0 | 002F | 아니오 |
| IBM i | AES_SHA_US | TLS 1.0 | 002E | 아니오 |
| 모두 | TLS_RSA_WITH_3DES_EDE_CBC_SHA | TLS 1.0 | 000A | 아니오 |
| 모두 | TLS_RSA_WITH_RC4_128_SHA | TLS 1.0 | 0005 | 아니오 |
| IBM i | TLS_RSA_WITH_RC4_128_MD5 | TLS 1.0 | 0004 | 아니오 |
| 모두 | TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | 0009 | 아니오 |
| IBM i | TLS_RSA_EXPORT_WITH_RC4_40_MD5 | TLS 1.0 | 0003 | 아니오 |

| 표 78. IBM MQ 9.2.0의 CipherSpec 순서 (계속) | | | | |
|---|---|---|---|---|
| **플랫폼** | **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
| IBM i | TLS_RSA_EXPORT_WITH_RC2_40_MD5 | TLS 1.0 | 0006 | 아니오 |
| IBM i | TLS_RSA_WITH_NULL_SHA | TLS 1.0 | 0002 | 아니오 |
| IBM i | TLS_RSA_WITH_NULL_MD5 | TLS 1.0 | 0001 | 아니오 |
| 모두 | TRIPLE_DES_SHA_US | SSL v3 | 000A | 아니오 |
| 모두 | RC4_SHA_US | SSL v3 | 0005 | 아니오 |
| 모두 | RC4_MD5_US | SSL v3 | 0004 | 아니오 |
| 모두 | DES_SHA_EXPORT | SSL v3 | 0009 | 아니오 |
| 모두 | RC4_MD5_EXPORT | SSL v3 | 0003 | 아니오 |
| 모두 | RC2_MD5_EXPORT | SSL v3 | 0006 | 아니오 |
| 모두 | NULL_SHA | SSL v3 | 0002 | 아니오 |
| 모두 | NULL_MD5 | SSL v3 | 0001 | 아니오 |
| ALW | FIPS_WITH_3DES_EDE_CBC_SHA | SSL v3 | FEFF | 아니오 |
| ALW | RC4_56_SHA_EXPORT1024 | SSL v3 | 0064 | 아니오 |
| ALW | DES_SHA_EXPORT1024 | SSL v3 | 0062 | 아니오 |
| ALW | FIPS_WITH_DES_CBC_SHA | SSL v3 | FEFE | 아니오 |

이 목록은 z/OS 에서 IBM MQ 가 사용하는 암호화 라이브러리가 제공하는 기본 목록으로 프로토콜을 정렬하여 구성되었으며 z/OS 및 분산 플랫폼에서 일관됩니다.

## 순서 변경

다른 순서가 필요한 경우 다음 규칙과 함께 IBM MQ for Multiplatforms z/OS 또는 IBM MQ for z/OS의 TransportSecurity 스탠자 에서 SSL 스탠자의 **AllowedCipherSpecs** 속성을 사용하여 CipherSpecs 의 새 순서를 제공할 수 있습니다.

- 상위 프로토콜 버전은 목록에서의 위치에 관계없이 항상 사용됩니다.
- 목록에 제공된 경우 사용 안함으로 설정된 모든 CipherSpec이 다시 사용 가능합니다.
- TLS 서버의 목록 순서는 TLS 클라이언트보다 우선순위가 높습니다.
- TLS 1.3이 사용 가능한 경우 특정 CipherSpec은 지원되지 않습니다.

예를 들어, IBM MQ for Multiplatforms에서 큐 관리자에 다음이 구성된 경우:

```
SSL:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

z/OS 그리고 IBM MQ for z/OS에서 큐 관리자에 다음이 구성된 경우:

```
TransportSecurity:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

CipherSpec 사용 순서는 다음과 같습니다.

- ANY_TLS12에 연결된 클라이언트는 TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256을 사용할 가능성이 높습니다.
- ANY_TLS12_OR_HIGHER에 연결된 클라이언트는 TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256을 사용할 가능성이 높습니다(클라이언트에서 TLS 1.3을 지원한다고 가정함).
- TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA에 연결된 클라이언트는 해당 CipherSpec을 사용합니다.

## IBM MQ의 이전 버전

IBM MQ 9.2.0 이전에서는 다음 CipherSpec 순서가 사용되었습니다.

표 *79. IBM MQ 9.2.0* 이전 *CipherSpec* 순서

| 플랫폼 | CipherSpec | 프로토콜 | 기본적으로 사용됨 |
|---|---|---|---|
| ALW z/OS | TLS_RSA_WITH_AES_128_CBC_SHA | TLS 1.0 | 아니오 |
| IBM i | AES_SHA_US | TLS 1.0 | 아니오 |
| ALW z/OS | TLS_RSA_WITH_AES_256_CBC_SHA | TLS 1.0 | 아니오 |
| 모두 | RC4_SHA_US | SSL v3 | 아니오 |
| 모두 | TLS_RSA_WITH_RC4_128_SHA | TLS 1.0 | 아니오 |
| 모두 | RC4_MD5_US | SSL v3 | 아니오 |
| IBM i | TLS_RSA_WITH_RC4_128_MD5 | TLS 1.0 | 아니오 |
| 모두 | TRIPLE_DES_SHA_US | SSL v3 | 아니오 |
| 모두 | TLS_RSA_WITH_3DES_EDE_CBC_SHA | TLS 1.0 | 아니오 |
| ALW | DES_SHA_EXPORT1024 | SSL v3 | 아니오 |
| 모두 | RC4_56_SHA_EXPORT1024 | SSL v3 | 아니오 |
| 모두 | RC4_MD5_EXPORT | SSL v3 | 아니오 |
| IBM i | TLS_RSA_EXPORT_WITH_RC4_40_MD5 | TLS 1.0 | 아니오 |
| 모두 | RC2_MD5_EXPORT | SSL v3 | 아니오 |

| 표 79. IBM MQ 9.2.0 이전 CipherSpec 순서 (계속) | | | |
|---|---|---|---|
| **플랫폼** | **CipherSpec** | **프로토콜** | **기본적으로 사용됨** |
| IBM i | TLS_RSA_EXPORT_WITH_RC2_40_MD5 | TLS 1.0 | 아니오 |
| 모두 | DES_SHA_EXPORT | SSL v3 | 아니오 |
| 모두 | TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | 아니오 |
| 모두 | NULL_SHA | SSL v3 | 아니오 |
| IBM i | TLS_RSA_WITH_NULL_SHA | TLS 1.0 | 아니오 |
| 모두 | NULL_MD5 | SSL v3 | 아니오 |
| IBM i | TLS_RSA_WITH_NULL_MD5 | TLS 1.0 | 아니오 |
| ALW | FIPS_WITH_DES_CBC_SHA | SSL v3 | 아니오 |
| ALW | FIPS_WITH_3DES_EDE_CBC_SHA | SSL v3 | 아니오 |
| 모두 | TLS_RSA_WITH_AES_128_CBC_SHA256 | TLS 1.2 | 예 |
| 모두 | TLS_RSA_WITH_AES_256_CBC_SHA256 | TLS 1.2 | 예 |
| 모두 | TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | 아니오 |
| 모두 | TLS_RSA_WITH_AES_128_GCM_SHA256 | TLS 1.2 | 예 |
| 모두 | TLS_RSA_WITH_AES_256_GCM_SHA384 | TLS 1.2 | 예 |
| ALW | ECDHE_ECDSA_RC4_128_SHA256 | TLS 1.2 | 아니오 |
| ALW | ECDHE_ECDSA_3DES_EDE_CBC_SHA256 | TLS 1.2 | 아니오 |
| Multi | ECDHE_RSA_RC4_128_SHA256 | TLS 1.2 | 아니오 |
| Multi | ECDHE_RSA_3DES_EDE_CBC_SHA256 | TLS 1.2 | 아니오 |
| 모두 | ECDHE_ECDSA_AES_128_CBC_SHA256 | TLS 1.2 | 예 |
| 모두 | ECDHE_ECDSA_AES_256_CBC_SHA384 | TLS 1.2 | 예 |
| 모두 | ECDHE_RSA_AES_128_CBC_SHA256 | TLS 1.2 | 예 |
| 모두 | ECDHE_RSA_AES_256_CBC_SHA384 | TLS 1.2 | 예 |

| 표 79. IBM MQ 9.2.0 이전 CipherSpec 순서 (계속) | | | |
|---|---|---|---|
| **플랫폼** | **CipherSpec** | **프로토콜** | **기본적으로 사용됨** |
| Multi | ECDHE_ECDSA_AES_128_GCM_SHA256 | TLS 1.2 | 예 |
| Multi | ECDHE_ECDSA_AES_256_GCM_SHA384 | TLS 1.2 | 예 |
| 모두 | ECDHE_RSA_AES_128_GCM_SHA256 | TLS 1.2 | 예 |
| 모두 | ECDHE_RSA_AES_256_GCM_SHA384 | TLS 1.2 | 예 |
| Multi | ECDHE_RSA_NULL_SHA256 | TLS 1.2 | 아니오 |
| ALW | ECDHE_ECDSA_NULL_SHA256 | TLS 1.2 | 아니오 |
| ALW | TLS_RSA_WITH_NULL_NULL | TLS 1.2 | 아니오 |
| ALW | TLS_RSA_WITH_RC4_128_SHA256 | TLS 1.2 | 아니오 |
| Multi | TLS_AES_128_GCM_SHA256 | TLS 1.3 | 예 |
| Multi | TLS_AES_256_GCM_SHA384 | TLS 1.3 | 예 |
| Multi | TLS_CHACHA20_POLY1305_SHA256 | TLS 1.3 | 예 |
| ALW | TLS_AES_128_CCM_SHA256 | TLS 1.3 | 예 |
| ALW | TLS_AES_128_CCM_8_SHA256 | TLS 1.3 | 예 |

**중요사항:** 2020년 7월 23일을 기준으로, 다음 AllowedCipherSpecs 속성은 현재 기본적으로 사용되는 CipherSpec만 사용합니다. 그러나 이 날짜 이후 더 이상 사용되지 않는 CipherSpec가 잘못 다시 사용 가능해지지 않도록 보장하기 위해 다음 AllowedCipherSpecs 속성에서 사용하는 CipherSpec을 최신 데이터로 확인해야 합니다.

이 순서의 CipherSpec을 리턴해야 하는 경우 다음 **AllowedCipherSpecs** SSL/TransportSecurity 스탠자 속성 값을 사용하여 이를 수행할 수 있습니다.

```
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,ECDHE_ECDSA_AES_128_CBC_SHA256,
ECDHE_ECDSA_AES_256_CBC_SHA384,ECDHE_RSA_AES_128_CBC_SHA256,ECDHE_RSA_AES_256_CBC_SHA384,
ECDHE_ECDSA_AES_128_GCM_SHA256,ECDHE_ECDSA_AES_256_GCM_SHA384,ECDHE_RSA_AES_128_GCM_SHA256,
ECDHE_RSA_AES_256_GCM_SHA384
```

## IBM MQ for Multiplatforms에서 정렬된 사용 가능한 CipherSpec의 사용자 정의 목록 제공

Multi

ALW **AMQ_ALLOWED_CIPHERS** 환경 변수 또는 `.ini` 파일의 **AllowedCipherSpecs** SSL 스탠자 속성을 사용하여 IBM MQ 채널과 함께 사용할 수 있도록 사용자의 환경 설정 순서대로 사용 가능한 CipherSpecs의 대체 세트를 제공할 수 있습니다. 다음 이유 중 하나로 인해 이 설정을 사용할 수 있습니다.

- 이름 지정된 CipherSpec 중 하나를 사용하지 않는 한, IBM MQ 리스너에서 수신 채널 시작 요청을 승인하지 않도록 제한하기 위해.
- TLS 데이터 교환에서 사용되는 CipherSpec의 우선순위 순서를 변경하기 위해.

이 기능을 사용하여 ANY* CipherSpecs에 포함된 CipherSpecs 를 제어할 수 있습니다.

**AMQ_ALLOWED_CIPHERS** 환경 변수 또는 **AllowedCipherSpecs** SSL 스탠자 속성은 다음 항목을 값으로 허용합니다.

- 단일 CipherSpec 이름.
- 다시 사용으로 설정하기 위한 CipherSpec 이름의 쉼표로 구분된 목록.
- 특수 값 ALL. 모든 CipherSpec을 나타냅니다.

**참고: ALL** CipherSpec을 사용해서는 안 됩니다. 이는 SSL 3.0 및 TLS 1.0 프로토콜뿐만 아니라 다수의 취약한 암호화 알고리즘을 사용하기 때문입니다.

이 설정이 구성된 경우 이는 기본 CipherSpec 목록을 대체하며 IBM MQ가 약한 암호 사용 중단 설정을 무시하도록 합니다(아래 내용 참조).

- IBM MQ 리스너가 이름 지정된 CipherSpec 중 하나를 사용하는 SSL/TLS 제안만 허용합니다.
- IBM MQ 채널이 공백 SSLCIPH 값, 또는 이름 지정된 CipherSpec 중 하나만 허용합니다.
- SSLCIPH 값에 대해 **runmqsc**의 Tab 키를 사용한 자동 완성 기능을 사용하면 해당 값이 이름 지정된 CipherSpec 중 하나로 제한됩니다.

예를 들어, 채널이 정의/변경되고 리스너가 ECDHE_RSA_AES_128_GCM_SHA256 또는 ECDHE_ECDSA_AES_256_GCM_SHA384를 허용하도록 하려는 경우 `qm.ini` 파일에서 다음을 설정할 수 있습니다.

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

또한 이 목록의 CipherSpec은 TLS 데이터 교환 중 사용되는 CipherSpec 우선순위를 판별하는 데 사용됩니다. 예를 들어, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256의 목록을 지정하는 경우 데이터 교환 중에 클라이언트가 이러한 CipherSpec 둘 다를 지정하여 연결하는 경우(즉, ANY_TLS12에 연결된 클라이언트) TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec은 TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec보다 우선 선택됩니다.

java.security 파일 설정을 사용하여 AMQP 또는 MQTT 채널에서 사용되는 암호를 제한할 수 있습니다.

## IBM MQ for z/OS에서 정렬된 사용 가능한 CipherSpec의 사용자 정의 목록 제공

z/OS

QMINI 데이터 세트의 **AllowedCipherSpecs** TransportSecurity 스탠자 속성을 사용하여 IBM MQ 채널과 함께 사용할 수 있는 CipherSpecs 의 대체 세트를 환경 설정 순서대로 제공할 수 있습니다. 다음 이유로 이를 수행할 수 있습니다.

- 이름 지정된 CipherSpec 중 하나를 사용하지 않는 한, IBM MQ 리스너에서 수신 채널 시작 요청을 승인하지 않도록 제한하기 위해.
- TLS 데이터 교환에서 사용되는 CipherSpec의 우선순위 순서를 변경하기 위해.

이 기능을 사용하여 ANY* CipherSpec에 포함된 CipherSpec을 제어할 수 있습니다. **AllowedCipherSpecs** 속성은 다음을 승인합니다.

- 단일 CipherSpec 이름.
- 다시 사용으로 설정하기 위한 CipherSpec 이름의 쉼표로 구분된 목록.
- 특수 값 ALL. 모든 CipherSpec을 나타냅니다.

  **참고: ALL** CipherSpec을 사용해서는 안 됩니다. 이는 SSL 3.0 및 TLS 1.0 프로토콜뿐만 아니라 다수의 취약한 암호화 알고리즘을 사용하기 때문입니다. 이 설정이 구성된 경우 이는 기본 CipherSpec 목록을 대체하며 IBM

MQ가 약한 암호 사용 중단 설정을 무시하도록 합니다(410 페이지의 『z/OS에서 더 이상 사용되지 않는 CipherSpec 사용』 참조).

IBM MQ 리스너는 이름 지정된 CipherSpec 중 하나를 사용하는 SSL/TLS 제안만 승인하며, IBM MQ 채널은 빈 SSLCIPH 값 또는 이름 지정된 CipherSpec 중 하나만 허용합니다.

예를 들어, 채널을 정의/대체하고 리스너가 ECDHE_RSA_AES_128_GCM_SHA256 또는 ECDHE_RSA_AES_256_GCM_SHA384를 허용하도록 하려는 경우 다음을 설정할 수 있습니다.

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

또한 이 목록의 CipherSpec은 TLS 데이터 교환 중 사용되는 CipherSpec 우선순위를 판별하는 데 사용됩니다. 예를 들어, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256의 목록을 지정하는 경우 데이터 교환 중에 클라이언트가 이러한 CipherSpec 둘 다를 지정하여 연결하는 경우(즉, ANY_TLS12에 연결된 클라이언트) TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec은 TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec보다 우선 선택됩니다.

## Deprecated 더 이상 사용되지 않는 CipherSpec

필요한 경우 IBM MQ와 함께 사용할 수 있는 더 이상 사용되지 않는 CipherSpec 목록입니다.

IBM MQ TLS 지원과 함께 사용할 수 있는 더 이상 사용되지 않는 CipherSpec은 다음 표에 나열되어 있습니다.

표 *80.* 더 이상 사용되지 않는 *CipherSpec*을 *IBM MQ*와 함께 다시 사용으로 설정할 수 있습니다.

| 플랫폼 지원409 페이지의 『1』 | CipherSpec 이름 | 16진 코드 | 사용되는 프로토콜 | 데이터 무결성 | 암호화 알고리즘 (암호화 비트) | FIPS 409 페이지의 『2』 | 스위트 B | 더 이상 사용되지 않는 경우 업데이트 |
|---|---|---|---|---|---|---|---|---|
| **CipherSpecs for SSL 3.0** | | | | | | | | |
| IBM i | AES_SHA_US 409 페이지의 『3』 | 002F | SSL 3.0 | SHA-1 | AES (128) | 아니오 | 아니오 | 9.0.0.0 |
| 모두 | DES_SHA_EXPORT 409 페이지의 『3』 409 페이지의 『4』 409 페이지의 『5』 | 0009 | SSL 3.0 | SHA-1 | DES (56) | 아니오 | 아니오 | 9.0.0.0 |
| ALW | DES_SHA_EXPORT1024 409 페이지의 『3』 409 페이지의 『6』 | 0062 | SSL 3.0 | SHA-1 | DES (56) | 아니오 | 아니오 | 9.0.0.0 |
| ALW | FIPS_WITH_DES_CBC_SHA 409 페이지의 『3』 | FEFE | SSL 3.0 | SHA-1 | DES (56) | 409 페이지의 『7』 없음 | 아니오 | 9.0.0.0 |
| ALW | FIPS_WITH_3DES_EDE_CBC_SHA 409 페이지의 『3』 | FEFF | SSL 3.0 | SHA-1 | 3DES (168) | 409 페이지의 『8』 없음 | 아니오 | 9.0.0.1 및 9.0.1 |
| 모두 | NULL_MD5 409 페이지의 『3』 | 0001 | SSL 3.0 | MD5 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| 모두 | NULL_SHA 409 페이지의 『3』 | 0002 | SSL 3.0 | SHA-1 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| 모두 | RC2_MD5_EXPORT 409 페이지의 『3』 409 페이지의 『4』 409 페이지의 『5』 | 0006 | SSL 3.0 | MD5 | RC2 (40) | 아니오 | 아니오 | 9.0.0.0 |

표 80. 더 이상 사용되지 않는 *CipherSpec*을 *IBM MQ*와 함께 다시 사용으로 설정할 수 있습니다. *(계속)*

| 플랫폼 지원409 페이지의 『1』 | CipherSpec 이름 | 16진 코드 | 사용되는 프로토콜 | 데이터 무결성 | 암호화 알고리즘 (암호화 비트) | FIPS 409 페이지의 『2』 | 스위트 B | 더 이상 사용되지 않는 경우 업데이트 |
|---|---|---|---|---|---|---|---|---|
| 모두 | RC4_MD5_EXPORT 409 페이지의 『4』 409 페이지의 『3』 | 0003 | SSL 3.0 | MD5 | RC4 (40) | 아니오 | 아니오 | 9.0.0.0 |
| 모두 | RC4_MD5_US 409 페이지의 『3』 | 0004 | SSL 3.0 | MD5 | RC4 (128) | 아니오 | 아니오 | 9.0.0.0 |
| 모두 | RC4_SHA_US 409 페이지의 『3』 409 페이지의 『5』 | 0005 | SSL 3.0 | SHA-1 | RC4 (128) | 아니오 | 아니오 | 9.0.0.0 |
| ALW | RC4_56_SHA_EXPORT1024 409 페이지의 『3』 409 페이지의 『6』 | 0064 | SSL 3.0 | SHA-1 | RC4 (56) | 아니오 | 아니오 | 9.0.0.0 |
| 모두 | TRIPLE_DES_SHA_US 409 페이지의 『3』 409 페이지의 『5』 | 000A | SSL 3.0 | SHA-1 | 3DES (168) | 아니오 | 아니오 | 9.0.0.1 및 9.0.1 |
| **CipherSpecs for TLS 1.0** | | | | | | | | |
| IBM i | TLS_RSA_EXPORT_WITH_RC2_40_MD5 409 페이지의 『3』 | 0006 | TLS 1.0 | MD5 | RC2 (40) | 아니오 | 아니오 | 9.0.0.0 |
| IBM i | TLS_RSA_EXPORT_WITH_RC4_40_MD5409 페이지의 『3』 409 페이지의 『4』 | 0003 | TLS 1.0 | MD5 | RC4 (40) | 아니오 | 아니오 | 9.0.0.0 |
| 모두 | TLS_RSA_WITH_DES_CBC_SHA 409 페이지의 『3』 | 0009 | TLS 1.0 | SHA-1 | DES (56) | 409 페이지의 『9』 없음 | 아니오 | 9.0.0.0 |
| IBM i | TLS_RSA_WITH_NULL_MD5 409 페이지의 『3』 | 0001 | TLS 1.0 | MD5 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| IBM i | TLS_RSA_WITH_NULL_SHA 409 페이지의 『3』 | 0002 | TLS 1.0 | SHA-1 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| IBM i | TLS_RSA_WITH_RC4_128_MD5 409 페이지의 『3』 | 0004 | TLS 1.0 | MD5 | RC4 (128) | 아니오 | 아니오 | 9.0.0.0 |
| z/OS ALW | TLS_RSA_WITH_AES_128_CBC_SHA 409 페이지의 『10』 | 002F | TLS 1.0 | SHA-1 | AES (128) | 예 | 아니오 | 9.0.5 |
| z/OS ALW | TLS_RSA_WITH_AES_256_CBC_SHA 409 페이지의 『6』 409 페이지의 『10』 | 0035 | TLS 1.0 | SHA-1 | AES (256) | 예 | 아니오 | 9.0.5 |
| 모두 | TLS_RSA_WITH_3DES_EDE_CBC_SHA | 000A | TLS 1.0 | SHA-1 | 3DES (168) | 예 | 아니오 | 9.0.0.1 및 9.0.1 |
| **CipherSpecs for TLS 1.2** | | | | | | | | |
| ALW | ECDHE_ECDSA_NULL_SHA256 409 페이지의 『3』 | C006 | TLS 1.2 | SHA-1 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| ALW | ECDHE_ECDSA_RC4_128_SHA256 409 페이지의 『3』 | C007 | TLS 1.2 | SHA-1 | RC4 (128) | 아니오 | 아니오 | 9.0.0.0 |

| 표 80. 더 이상 사용되지 않는 CipherSpec을 IBM MQ와 함께 다시 사용으로 설정할 수 있습니다. (계속) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **플랫폼 지원**409 페이지의 『1』 | **CipherSpec 이름** | **16진 코드** | **사용되는 프로토콜** | **데이터 무결성** | **암호화 알고리즘 (암호화 비트)** | **FIPS**409 페이지의 『2』 | **스위트 B** | **더 이상 사용되지 않는 경우 업데이트** |
| ALW IBM i | ECDHE_RSA_NULL_SHA256 409 페이지의 『3』 | C010 | TLS 1.2 | SHA-1 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| ALW IBM i | ECDHE_RSA_RC4_128_SHA256 409 페이지의 『3』 | C011 | TLS 1.2 | SHA-1 | RC4 (128) | 아니오 | 아니오 | 9.0.0.0 |
| ALW | TLS_RSA_WITH_NULL_NULL 409 페이지의 『3』 | 0000 | TLS 1.2 | 없음 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| 모두 | TLS_RSA_WITH_NULL_SHA256 409 페이지의 『3』 | 003B | TLS 1.2 | SHA-256 | 없음 | 아니오 | 아니오 | 9.0.0.1 |
| ALW | TLS_RSA_WITH_RC4_128_SHA256 409 페이지의 『3』 | 0005 | TLS 1.2 | SHA-1 | RC4 (128) | 아니오 | 아니오 | 9.0.0.0 |
| ALW | ECDHE_ECDSA_3DES_EDE_CBC_SHA256 | C0008 | TLS 1.2 | SHA-1 | 3DES (168) | 예 | 아니오 | 9.0.0.1 및 9.0.1 |
| ALW IBM i | ECDHE_RSA_3DES_EDE_CBC_SHA256 | C012 | TLS 1.2 | SHA-1 | 3DES (168) | 예 | 아니오 | 9.0.0.1 및 9.0.1 |

표 80. 더 이상 사용되지 않는 *CipherSpec*을 *IBM MQ*와 함께 다시 사용으로 설정할 수 있습니다. *(계속)*

| 플랫폼 지원[409 페이지의 『1』] | CipherSpec 이름 | 16진 코드 | 사용되는 프로토콜 | 데이터 무결성 | 암호화 알고리즘 (암호화 비트) | FIPS [409 페이지의 『2』] | 스위트 B | 더 이상 사용되지 않는 경우 업데이트 |
|---|---|---|---|---|---|---|---|---|

**참고:**

1. 각 플랫폼 아이콘에 포함되는 플랫폼 목록은 제품 문서에서 사용되는 아이콘을 참조하십시오.
2. FIPS 인증 플랫폼에서 CipherSpec이 FIPS 인증 CipherSpec인지 여부를 지정합니다. FIPS에 대한 설명은 FIPS(Federal Information Processing Standards)를 참조하십시오.
3. ALW TLS 1.3이 사용으로 설정되면(`qm.ini`의 AllowTLSV13 특성을 통해) 이러한 CipherSpec이 사용 안함으로 설정됩니다.

   z/OS IBM MQ for z/OS 9.2.0 이상에서 작성된 큐 관리자는 기본적으로 TLS 1.3을 사용으로 설정하므로, 이러한 CipherSpec이 사용 안함으로 설정됩니다. 필요한 경우 TLS V1.3을 해제하여 이러한 CipherSpec을 사용으로 설정할 수 있습니다. 이는 큐 관리자 JCL에 있는 QMINI 데이터 세트의 TransportSecurity 스탠자에 **AllowTLSV13**==*FALSE*를 추가하여 수행됩니다. 이전 버전에서 IBM MQ for z/OS 9.2.0으로 마이그레이션된 큐 관리자는 기본적으로 TLS 1.3을 사용으로 설정하지 않으므로, 이러한 CipherSpec이 사용으로 설정됩니다.
4. 최대 데이터 교환 키 크기는 512비트입니다. SSL 데이터 교환 중에 교환된 인증서 중 한 개의 키 크기가 512비트를 초과할 경우, 데이터 교환 중에 사용할 수 있도록 임시 512비트 키가 생성됩니다.
5. 이 CipherSpec은 IBM MQ classes for Java 또는 IBM MQ classes for JMS에서 더 이상 지원되지 않습니다. 자세한 정보는 IBM MQ classes for Java의 SSL/TLS 암호 스펙 및 암호 스위트 또는 IBM MQ classes for JMS의 SSL/TLS 암호 스펙 및 암호 스위트를 참조하십시오.
6. 데이터 교환 키 크기는 1024비트입니다.
7. Deprecated 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다. 이름 `FIPS_WITH_DES_CBC_SHA`은(는) 히스토리이며 이 CipherSpec이 이전에(더 이상) FIPS를 준수하지 않는다는 사실을 반영합니다. 이 CipherSpec은 더 이상 사용되지 않으므로 앞으로는 사용하지 않는 것이 좋습니다.
8. Deprecated 이름 `FIPS_WITH_3DES_EDE_CBC_SHA`은(는) 히스토리이며 이 CipherSpec이 이전에(더 이상) FIPS를 준수하지 않는다는 사실을 반영합니다. 이 CipherSpec은 더 이상 사용되지 않습니다.
9. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다.
10. 이러한 CipherSpec만 다시 사용하는 경우 CSQXWEAK DD문을 사용할 필요가 없습니다.

## IBM MQ for Multiplatforms에서 더 이상 사용되지 않는 CipherSpec 사용

Multi

기본적으로 채널 정의에 더 이상 사용되지 않는 CipherSpec을 지정하도록 허용되지 않습니다. IBM MQ for Multiplatforms에서 더 이상 사용되지 않는 CipherSpec을 지정하려 시도하면 AMQ8242: SSLCIPH definition wrong 메시지가 수신되며 PCF가 `MQRCCF_SSL_CIPHER_SPEC_ERROR`를 리턴합니다.

더 이상 사용되지 않는 CipherSpec으로 채널을 시작할 수 없습니다. 더 이상 사용되지 않는 CipherSpec을 사용하여 수행하려고 하면 시스템에서 `MQCC_FAILED (2)`와 `MQRC_SSL_INITIALIZATION_ERROR (2393)`의 **Reason** 을(를) 클라이언트에 리턴합니다.

환경 변수 **AMQ_SSL_WEAK_CIPHER_ENABLE**을(를) 설정하여 서버에서 런타임 시 채널을 정의하기 위해 더 사용되지 않는 하나 이상의 CipherSpec을 다시 사용할 수 있습니다.

**AMQ_SSL_WEAK_CIPHER_ENABLE** 환경 변수는 다음을 승인합니다.

- 단일 CipherSpec 이름 또는
- 다시 사용으로 설정하기 위한 CipherSpec 이름의 쉼표로 구분된 목록.
- 특수 값 ALL. 모든 CipherSpec을 나타냅니다.

**주의:** ALL이 올바른 옵션이어도 엔터프라이즈에서 요구하는 특정 상황에서**만** 이를 사용해야 합니다. ALL CipherSpec을 다시 사용 가능하게 하면 SSL 3.0 및 TLS 1.0 프로토콜뿐만 아니라, 많은 취약한 암호화 알고리즘이 사용 가능해집니다.

예를 들어, ECDHE_RSA_RC4_128_SHA256을 다시 사용으로 설정하려면 다음 환경 변수를 설정하십시오.

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

또는 다음과 같이 설정하여 `qm.ini` 파일에서 SSL 스탠자를 변경하십시오.

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

## z/OS에서 더 이상 사용되지 않는 CipherSpec 사용

z/OS

기본적으로 채널 정의에 더 이상 사용되지 않는 CipherSpec을 지정하도록 허용되지 않습니다. z/OS에서 더 이상 사용되지 않는 CipherSpec을 지정하려고 하면 메시지 CSQM102E, 메시지 CSQX616E 또는 CSQX674E를 수신합니다.

다음 메시지 중 하나가 표시되고 엔터프라이즈에서 보안이 약한 CipherSpec의 사용을 다시 활성화해야 하는 경우 이 절에 나열된 지침을 따르십시오.

**주의:** 다음 지시사항에서 더미 정의(DD)문이 적용되려면 SSLTASKS가 0이 아닌 값이어야 합니다. SSLTASKS를 변경해야 하는 경우 채널 시작기를 재순환해야 합니다.

IBM MQ for z/OS에서 중단된 CipherSpec 또는 약점을 제어하는 최신 메소드는 다음과 같습니다.

- 약한 CipherSpec을 사용할 수 있게 설정하려면 CSQXWEAK(이)라는 더미 데이터 정의(DD) 명령문을 채널 시작기 JCL에 추가하여 수행합니다. 단독으로 지정된 경우 TLS 1.2 프로토콜과 연관된 보안이 약한 CipherSpec만 사용으로 설정합니다. 예를 들어 다음과 같습니다.

  ```
  //CSQXWEAK DD DUMMY
  ```

  **참고:** 더 이상 사용되지 않는 모든 CipherSpec에서 이 DD 명령문을 사용할 필요는 없으며, 앞의 표에 있는 노트 10을 참조하십시오.
- SSLv3 CipherSpec을 다시 사용할 수 있게 설정하려면 CSQXSSL3(이)라는 더미 DD문을 채널 시작기 JCL에 추가하여 수행합니다. 모든 SSLv3 CipherSpec은 **약함**으로 간주되므로 CSQXWEAK도 지정해야 합니다.

  ```
  //CSQXSSL3 DD DUMMY
  ```
- 더 이상 사용되지 않는 TLS V1 CipherSpec을 다시 사용하도록 설정하려면 채널 시작기 JCL에 TLS10ON(TLS V1.0 ON)라는 더미 DD문을 추가하여 수행합니다. 단독으로 지정된 경우 TLS 1.0 프로토콜과 연관된 보안이 강한 CipherSpec만 사용으로 설정합니다.

  ```
  //TLS10ON DD DUMMY
  ```

  CSQXWEAK(으)로 지정된 경우, TLS 1.0과 연관된 **약한** CipherSpec을 사용할 수도 있습니다.
- 더 이상 사용되지 않는 TLS V1 CipherSpec을 명시적으로 끄려는 경우, TLS10OFF(TLS V1.0 OFF)(이)라는 더미 DD문을 채널 시작기 JCL에 추가하여 수행합니다. 예를 들면 다음과 같습니다.

  ```
  //TLS10OFF DD DUMMY
  ```

**System SSL** 기본 암호 스펙 목록에 나열된 암호 스펙을 사용하여 리스너와 협상만 수행하려는 경우 CHINT JCL에서 다음 DD 문을 정의해야 합니다.

```
JCL: //GSKDCIPS DD DUMMY
```

**중요사항:** IBM MQ for z/OS 9.2.0 이상에서 이전에 나열된 DD 카드 및 **AllowTLSV13**의 값은 채널 시작기 시작 중에 메시지를 표시하여 사용 가능한 프로토콜과 그렇지 않은 프로토콜을 표시할 때 고려됩니다. 따라서 이전에 나열된 DD 카드 중 하나가 지정되어도 이는 이러한 설정의 조합으로 인해 특정 프로토콜이 다른 프로토콜과 함께 사용할 수 없음을 의미할 수 있습니다. 예를 들어, TLS 1.3이 사용 가능한 경우 프로토콜 SSL 3.0은 허용되지 않습니다.

DD 변경이 적합하지 않은 경우 약한 CipherSpec과 SSLv3 지원 모두를 강제로 사용으로 다시 설정할 수 있는 대체 메커니즘이 있습니다. 추가 정보는 IBM 서비스를 참조하십시오.

**관련 개념**

43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』
이 토픽은 IBM MQ에서 CipherSpec과 디지털 인증서 사이의 관계를 소개하여 보안 정책에 대해 적합한 CipherSpec 및 디지털 인증서를 선택하는 방법에 대한 정보를 제공합니다.

**관련 참조**

DEFINE CHANNEL

ALTER CHANNEL

## 알리어스 CipherSpec 설정 간 관계

이 정보에서는 클라이언트 및 서버 구성에서 알리어스 CipherSpec의 서로 다른 조합일 때 예상되는 동작을 설명합니다. 여기에서, 클라이언트는 통신을 시작하는 엔티티(예: 클라이언트 애플리케이션 또는 큐 관리자 송신자 채널)를 말하며, 서버는 클라이언트에서 통신을 수신하는 엔티티(예: 서버-연결 채널 또는 수신자 채널)를 말합니다.

### 최소 프로토콜 대 고정 프로토콜 CipherSpec

IBM MQ는 두 가지 서로 다른 CipherSpec 유형을 지원합니다.

**최소 프로토콜**
: 최소 레벨 CipherSpec은 상한을 설정하지 않는 유형입니다(예: ANY, ANY_TLS12_OR_HIGHER 또는 ANY_TLS13_OR_HIGHER).

**고정 프로토콜**
: 고정 프로토콜 CipherSpec은 특정 프로토콜(예: ANY_TLS12 및 ANY_TLS13) 또는 특정 알고리즘(예: ECDHE_ECDSA_3DES_EDE_CBC_SHA256)을 식별하는 유형입니다.

최소 및 고정 프로토콜 CipherSpecs 은 모든 플랫폼에서 지원됩니다.

구성을 최대한 단순하게 하면서 보안을 유지하려면 채널 양측에서 **최소 프로토콜** CipherSpec을 사용하는 것이 좋습니다. 이는 통신의 양측이 더 높은 TLS 프로토콜 버전을 지원하는 경우 양측의 구성을 변경할 필요없이 자동으로 새 버전을 지원하고 사용할 수 있도록 합니다.

시작 측에서 **최소 프로토콜** CipherSpec을 사용하지만, 수신 측에서 **고정 프로토콜** CipherSpec을 사용하면 연결이 거부될 수 있으며 다음이 발생합니다.

- Multi 메시지 AMQ9631 및 AMQ9641이 발행됩니다.
- z/OS 메시지 CSQX631E 및 CSQX641E가 발행됩니다.

다음 표에서는 서로 다른 알리어스 CipherSpec 설정 및 예상되는 결과 사이의 관계를 보여줍니다. 412 페이지의 표 81 에서는 클라이언트, 서버 또는 둘 다에서 TLS 1.3 이 사용으로 설정되지 않은 경우 예상되는 동작을 보여줍니다. 412 페이지의 표 82 에서는 클라이언트 및 서버 모두에서 TLS 1.3 이 사용으로 설정된 경우 예상되는 동작을 보여줍니다. 두 경우 모두 클라이언트의 CipherSpecs 는 테이블의 Y축에 표시되고 서버의 CipherSpecs 는 테이블의 X축에 표시됩니다.

**참고:** 다음 표에서 실패할 가능성 있음으로 표시된 셀은 연결의 한 부분에 **최소 프로토콜** CipherSpec을 지정하고 다른 부분에 특정(**고정 프로토콜**) CipherSpec을 지정하는 경우 충돌 가능성이 있음을 표시합니다.

예를 들어, 클라이언트와 서버가 ANY CipherSpec을 사용하도록 설정되었으며 서버 채널이 특정 CipherSpec을 사용하도록 설정되었다고 가정해 보십시오.

- 클라이언트와 서버 모두에서 지원하는 가장 강력한 CipherSpec이 채널에 구성된 특정 CipherSpec과 일치하는 경우에는 TLS 데이터 교환이 성공적으로 해결됩니다.

- 그러나 클라이언트와 서버 모두에서 지원하는 더 강력한 CipherSpec이 있는 경우 TLS 데이터 교환은 이것이 채널에 지정된 CipherSpec과 일치하지 않더라도 이를 사용하여 해결하려 하며, 따라서 TLS 데이터 교환이 실패합니다.

표 *81.* 클라이언트, 서버 또는 둘 다에서 *TLS 1.3* 이 사용으로 설정되지 않은 경우 예상되는 동작

| | **서버** | | | |
|---|---|---|---|---|
| **클라이언트** | **특정 TLS 1.2 CipherSpec** | **ANY** | **ANY_ TLS12** | **ANY_TLS12_ OR_HIGHER** |
| **특정 TLS 1.2 CipherSpec** | 연결 | 연결 | 연결 | 연결 |
| **모두** | 실패할 수 있음 | 연결 | 연결 | 연결 |
| **ANY_ TLS12** | 실패할 수 있음 | 연결 | 연결 | 연결 |
| **ANY_TLS12_ OR_HIGHER** | 실패할 수 있음 | 연결 | 연결 | 연결 |

표 *82.* 클라이언트 및 서버 모두에서 *TLS 1.3* 이 사용으로 설정된 경우 예상되는 동작

| | **서버** | | | | | | |
|---|---|---|---|---|---|---|---|
| **클라이언트** | **특정 TLS 1.2 CipherSpec** | **특정 TLS 1.3 CipherSpec** | **ANY** | **ANY_TLS 12** | **ANY_TLS 13** | **ANY_TLS12_ OR_HIGHER** | **ANY_TLS13_ OR_HIGHER** |
| **특정 TLS 1.2 CipherSpec** | 연결 | **실패** | 연결 | 연결 | **실패** | 연결 | **실패** |
| **특정 TLS 1.3 CipherSpec** | **실패** | 연결 | 연결 | **실패** | 연결 | 연결 | 연결 |
| **모두** | **실패** | 실패할 수 있음 | 연결 | **실패** | 연결 | 연결 | 연결 |
| **ANY_TLS12** | 실패할 수 있음 | **실패** | 연결 | 연결 | **실패** | 연결 | **실패** |
| **ANY_TLS13** | **실패** | 실패할 수 있음 | 연결 | **실패** | 연결 | 연결 | 연결 |
| **ANY_TLS12_ OR_HIGHER** | **실패** | 실패할 수 있음 | 연결 | **실패** | 연결 | 연결 | 연결 |
| **ANY_TLS13_ OR_HIGHER** | **실패** | 실패할 수 있음 | 연결 | **실패** | 연결 | 연결 | 연결 |

**관련 개념**

43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』
이 토픽은 IBM MQ에서 CipherSpec과 디지털 인증서 사이의 관계를 소개하여 보안 정책에 대해 적합한 CipherSpec 및 디지털 인증서를 선택하는 방법에 대한 정보를 제공합니다.

20 페이지의 『CipherSpecs 및 CipherSuites』
암호화 보안 프로토콜은 안전 연결에서 사용되는 알고리즘에 동의해야 합니다. CipherSpec 및 CipherSuite는 특정 알고리즘 결합을 정의합니다.

392 페이지의 『CipherSpec 사용 가능』
**DEFINE CHANNEL** 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec을 사용 가능으로 설정합니다.

**관련 태스크**

NY_TLS12_OR_HIGHER CipherSpec을 사용하도록 기존 보안 구성 마이그레이션

## IBM MQ Explorer를 사용하여 CipherSpec에 대한 정보 확보

IBM MQ Explorer를 사용하여 CipherSpec의 설명을 표시할 수 있습니다.

392 페이지의 『CipherSpec 사용 가능』 에 있는 CipherSpec에 대한 정보를 확보하려면 다음 프로시저를 사용하십시오.

1. IBM MQ Explorer를 열고 **큐 관리자** 폴더를 펼치십시오.
2. 큐 관리자를 시작했는지 확인하십시오.
3. 작업하려는 큐 관리자를 선택하고 **채널**을 클릭하십시오.
4. 작업하려는 채널을 마우스 오른쪽 단추로 클릭하고 **특성**을 선택하십시오.
5. **SSL** 특성 페이지를 선택하십시오.
6. 작업하려는 CipherSpec을 목록에서 선택하십시오. 설명이 목록 아래의 창에 표시됩니다.

## z/OS IBM i CipherSpec을 지정하기 위한 대안

운영 체제가 TLS 지원을 제공하는 플랫폼에서는, 392 페이지의 『CipherSpec 사용 가능』에 포함되지 않은 새 CipherSpec을 시스템이 지원할 수도 있습니다.

새 CipherSpec을 SSLCIPH 매개변수에 지정할 수 있으나, 제공하는 값은 플랫폼에 따라 다릅니다. 모든 경우에 스펙은 올바르고 시스템이 실행 중인 TLS 버전에 의해 지원되는 TLS CipherSpec에 해당해야 합니다.

**참고:** 이 절은 AIX, Linux, and Windows 시스템에 적용되지 않습니다. CipherSpecs 가 IBM MQ 제품과 함께 제공되므로 새 CipherSpecs 가 쉽먼트 후에 사용 가능하게 되지 않기 때문입니다.

### IBM i IBM i

16진 값을 나타내는 2자의 문자열.

허용되는 값에 대한 자세한 정보는 보안 세션을 위한 문자 정보 설정의 사용법 참고사항 절에서 3번 항목을 참조하십시오.

**주의:** 암호가 사용될 값에서 명확하지 않으므로 **SSLCIPH**에 16진 암호 값을 지정하지 말아야 하며 사용할 프로토콜의 선택은 불확실합니다. 16진 암호 값을 사용하면 CipherSpec 불일치 오류가 발생할 수 있습니다.

**CHGMQMCHL** 또는 **CRTMQMCHL** 명령을 사용하여 값을 지정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

또한 **ALTER QMGR** MQSC 명령을 사용하여 **SSLCIPH** 매개변수를 설정할 수도 있습니다.

### z/OS z/OS

16진 값을 나타내는 4문자의 문자열입니다. 16진 코드는 TLS 프로토콜에 정의된 값에 해당합니다.

자세한 정보는 암호 스위트 정의를 참조하십시오. 여기에는 4자리 16진 코드 양식의 지원되는 모든 TLS 1.0, TLS 1.2 및 TLS 1.3 암호 스펙의 목록이 있습니다.

**참고:** Deprecated 보안이 약한 CipherSpec 또는 더 이상 사용되지 않는 프로토콜(예: SSL V3.0 또는 TLS 1.0)에 속한 CipherSpec을 사용하려면 채널 시작기 시작 JCL에 관련 DD 카드를 지정해야 합니다. 자세한 정보는 406 페이지의 『더 이상 사용되지 않는 CipherSpec』의 내용을 참조하십시오.

## IBM MQ 클러스터에 대한 고려사항

IBM MQ 클러스터를 사용하면 392 페이지의 『CipherSpec 사용 가능』에서 CipherSpec 이름을 사용하는 것이 가장 안전합니다. 대체 스펙을 사용하는 경우 스펙이 다른 플랫폼에서는 유효하지 않을 수도 있음을 유의하십시오. 자세한 정보는 449 페이지의 『SSL/TLS 및 클러스터』의 내용을 참조하십시오.

## IBM MQ MQI client를 위해 CipherSpec 지정

IBM MQ MQI client를 위해 CipherSpec을 지정하기 위한 세 가지 옵션이 있습니다.

해당 옵션은 다음과 같습니다.

- 채널 정의 테이블 사용
- MQCD 구조, MQCD_VERSION_7 이상 또는 MQCONNX 호출에서 SSLCipherSpec 필드 사용.
- Active Directory를 지원하는 Windows 시스템에서 Active Directory 사용

## IBM MQ classes for Java 및 IBM MQ classes for JMS를 사용하여 CipherSuite 지정

IBM MQ classes for Java 및 IBM MQ classes for JMS는 다른 플랫폼에서와 다르게 CipherSuite를 지정합니다.

IBM MQ classes for Java에서 CipherSuite 지정에 대한 정보는 Java에 대한 TLS(Transport Layer Security) 지원을 참조하십시오.

IBM MQ classes for JMS에서 CipherSuite 지정에 대한 정보는 IBM MQ classes for JMS에서 TLS(Transport Layer Security) 사용을 참조하십시오.

## IBM MQ.NET용 CipherSpec 지정

IBM MQ.NET의 경우 MQEnvironment 클래스를 사용하거나 연결 특성의 해시 테이블에서 MQC.SSL_CIPHER_SPEC_PROPERTY를 사용하여 CipherSpec을 지정할 수 있습니다.

.NET 비관리 클라이언트를 위한 CipherSpec 지정에 대한 정보는 비관리 .NET 클라이언트에 TLS 사용을 참조하십시오.

.NET 관리 클라이언트를 위한 CipherSpec 지정에 대한 정보는 관리 .NET 클라이언트에 대한 CipherSpec 지원을 참조하십시오.

## z/OS IBM MQ for z/OS와 함께 AT-TLS 사용

AT-TLS(Application Transparent Transport Layer Security)는 해당 애플리케이션에서 TLS 지원을 구현하거나 TLS가 사용 중임을 인식할 필요 없이 z/OS 애플리케이션에 대한 TLS 지원을 제공합니다. AT-TLS는 z/OS에서만 사용할 수 있습니다.

AT-TLS는 모든 버전의 IBM MQ for z/OS에서 사용할 수 있습니다.

IBM MQ for z/OS에서 AT-TLS를 사용하기 전에 관련된 417 페이지의 『제한사항』을 이해해야 합니다.

AT-TLS(Application Transparent Transport Layer Security)를 사용하려면 z/OS Communications Server에서 TLS가 투명하게 사용으로 설정된 TCP/IP 연결을 결정하는 데 사용하는 규칙 세트가 포함된 정책문을 정의합니다.

IBM MQ for z/OS에는 채널에 CipherSpec으로 구성된 SSLCIPH 매개변수가 있어야 하는 자체 TLS 구현이 있습니다.

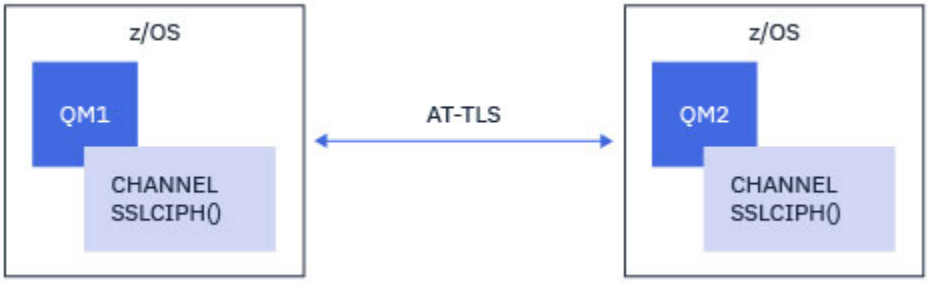
채널에서 TLS를 사용하도록 결정할 때 IBM MQ 관리자가 AT-TLS 또는 IBM MQ TLS를 사용하도록 결정할 수 있습니다. AT-TLS가 다른 미들웨어에 사용되는지 여부 또는 성능 영향으로 인해 결정되는 경우가 많습니다. AT-TLS 및 IBM MQ TLS의 성능에 대한 기본 비교는 MP16: IBM MQ for z/OS 용량 계획 및 성능 조정을 참조하십시오.

### 시나리오

IBM MQ에서 AT-TLS 사용은 다음 시나리오에서 지원됩니다.

#### 시나리오 1

채널의 양쪽에서 AT-TLS를 사용하는 두 개의 IBM MQ for z/OS 큐 관리자 사이. 즉, 두 채널 모두 SSLCIPH 속성을 지정하지 않습니다. 이 접근 방식은 모든 메시지 채널에서 사용할 수 있습니다.



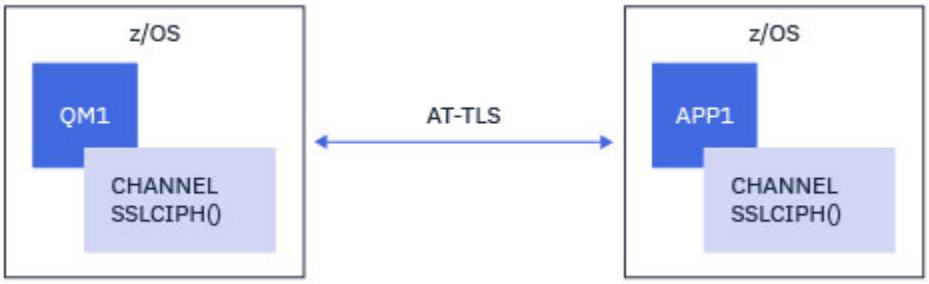

이 시나리오의 구현은 채널의 한쪽마다 하나씩, 두 개의 AT-TLS 정책을 정의하는 것으로 구성됩니다. 이러한 정책은 시나리오 3 또는 시나리오 4에서 사용되는 정책과 동일합니다.

예를 들어, 채널이 이름 지정된 단일 CipherSpec 사용에서 AT-TLS 사용으로 변경된 경우 아웃바운드 채널은 418 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』의 정책을 사용하고 인바운드 채널은 426 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』의 정책을 사용합니다.

채널이 알리어스 CipherSpec 사용에서 AT-TLS 사용으로 변경된 경우 아웃바운드 채널은 422 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』의 정책을 사용하고 인바운드 채널은 430 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』의 정책을 사용합니다.

### 시나리오 2

채널의 양쪽에서 AT-TLS를 사용하는 IBM MQ for z/OS 큐 관리자와 z/OS에서 실행 중인 IBM MQ Java 클라이언트 애플리케이션 사이. 즉, 서버 연결 채널과 클라이언트 연결 채널 모두 SSLCIPH 속성을 지정하지 않습니다.



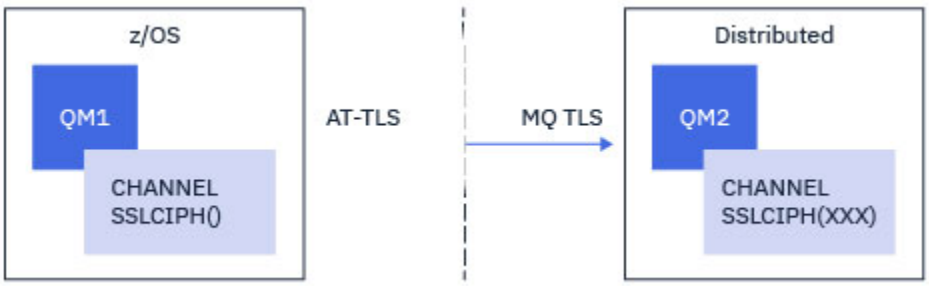

이 시나리오의 구현은 채널의 한쪽마다 하나씩, 두 개의 AT-TLS 정책을 정의하는 것으로 구성됩니다. 이러한 정책은 시나리오 3 또는 시나리오 4에서 사용되는 정책과 동일합니다.

예를 들어, 채널이 이름 지정된 단일 CipherSpec 사용에서 AT-TLS 사용으로 변경된 경우 클라이언트 연결 채널은 418 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』의 정책을 사용하고 서버 연결 채널은 426 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』의 정책을 사용합니다.

채널이 알리어스 CipherSpec 사용에서 AT-TLS 사용으로 변경된 경우 클라이언트 연결 채널은 422 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』의 정책을 사용하고 서버 연결 채널은 430 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』의 정책을 사용합니다.

### 시나리오 3

사이IBM MQ for z/OS 큐 관리자 및 큐 관리자가 실행 중IBM MQ for Multiplatforms , 여기서IBM MQ for z/OS 큐 관리자는 AT-TLS를 사용하며IBM MQ for Multiplatforms 큐 관리자가 사용하는IBM MQ SSLCIPH 속성을 단일 이름으로 지정하여 TLSCipherSpec . 이는 클러스터 송신자 및 클러스터 수신자 이외의 모든 메시지 채널 유형에 적용됩니다.

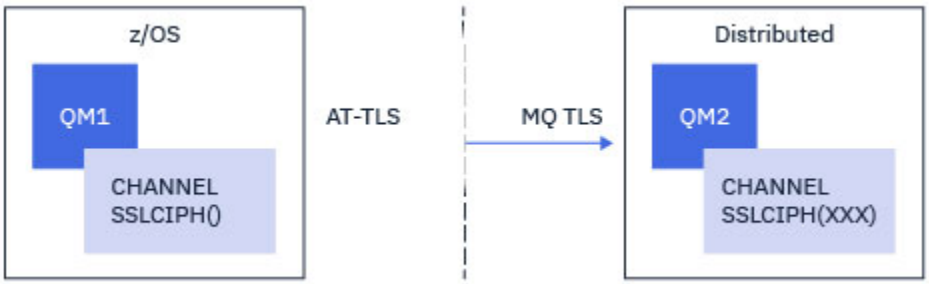



IBM MQ for z/OS 관리자에서 IBM MQ for Multiplatforms로의 아웃바운드 채널에 대한 AT-TLS 구성 예는 418 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』을 참조하고 IBM MQ for Multiplatforms 큐 관리자에서 IBM MQ for z/OS 큐 관리자로의 인바운드 연결에 대한 AT-TLS 구성 예는 426 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』을 참조하십시오.

두 큐 관리자가 모두 z/OS에 있는 경우 동일한 AT-TLS 구성을 사용할 수 있지만 오른쪽의 큐 관리자가 AT-TLS를 사용하도록 구성되지 않았습니다.

### 시나리오 4

IBM MQ for z/OS 큐 관리자와 IBM MQ for Multiplatforms에서 실행 중인 큐 관리자 사이, 여기서 IBM MQ for z/OS 큐 관리자는 AT-TLS를 사용하고 IBM MQ for Multiplatforms 큐 관리자는 알리어스 CipherSpec으로 SSLCIPH 속성을 지정하여 IBM MQ TLS를 사용합니다. 이는 클러스터 송신자 및 클러스터 수신자 이외의 모든 메시지 채널 유형에 적용됩니다.

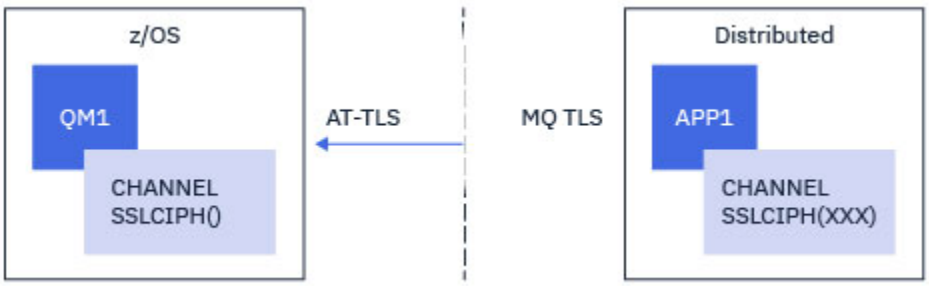



IBM MQ for z/OS 관리자에서 IBM MQ for Multiplatforms로의 아웃바운드 채널에 대한 AT-TLS 구성 예는 422 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』을 참조하고 IBM MQ for Multiplatforms 큐 관리자에서 IBM MQ for z/OS 큐 관리자로의 인바운드 연결에 대한 AT-TLS 구성 예는 430 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』 및 430 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』을 참조하십시오.

두 큐 관리자가 모두 z/OS에 있는 경우 동일한 AT-TLS 구성을 사용할 수 있지만 오른쪽의 큐 관리자가 AT-TLS를 사용하도록 구성되지 않았습니다.

### 시나리오 5

IBM MQ for z/OS 큐 관리자와 IBM MQ for Multiplatforms에서 실행 중인 클라이언트 애플리케이션 사이, 여기서 IBM MQ for z/OS 큐 관리자는 AT-TLS를 사용하고 클라이언트 애플리케이션은 이름 지정된 단일 CipherSpec으로 SSLCIPH 속성을 지정하여 IBM MQ TLS를 사용합니다.

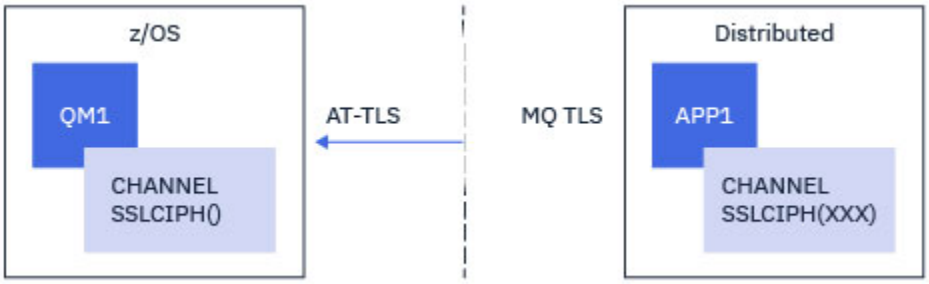



이 시나리오에는 인바운드 메시지 채널에서 사용되는 것과 동일한 요구사항을 충족하는 단일 AT-TLS 정책이 필요합니다. 426 페이지의 『이름 지정된 단일 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』을 참조하십시오.

클라이언트 애플리케이션이 Java 애플리케이션이고 z/OS에서 실행 중인 경우 동일한 AT-TLS 구성을 사용할 수 있지만 AT-TLS를 사용하도록 구성되지 않았습니다.

### 시나리오 6

IBM MQ for z/OS 큐 관리자와 IBM MQ for Multiplatforms에서 실행 중인 클라이언트 애플리케이션 사이, 여기서 IBM MQ for z/OS 큐 관리자는 AT-TLS를 사용하고 클라이언트 애플리케이션은 알리어스 CipherSpec으로 SSLCIPH 속성을 지정하여 IBM MQ TLS를 사용합니다.




이 시나리오에는 인바운드 메시지 채널에서 사용되는 것과 동일한 요구사항을 충족하는 단일 AT-TLS 정책이 필요합니다. 430 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』을 참조하십시오.

클라이언트 애플리케이션이 Java 애플리케이션이고 z/OS에서 실행 중인 경우 동일한 AT-TLS 구성을 사용할 수 있지만 AT-TLS를 사용하도록 구성되지 않았습니다.

## 제한사항

IBM MQ for z/OS는 AT-TLS를 인식하지 않으므로 이전 시나리오에 적용되는 몇 가지 제한사항이 있습니다.

- IBM MQ TLS와 결합된 AT-TLS는 클러스터 송신자 및 클러스터 수신자 채널에서 작동하지 않습니다.
- IBM MQ for z/OS 큐 관리자는 AT-TLS를 사용 중임을 인식하지 못하고 파트너 큐 관리자 또는 클라이언트로부터 인증서 정보를 수신하지 않습니다. 따라서 다음 속성은 AT-TLS를 사용하는 채널의 z/OS 측에 영향을 미치지 않습니다.
  - SSLCAUTH 및 SSLPEER 채널 속성
  - SSLRKEYC 큐 관리자 속성
  - CHLAUTH 규칙의 SSLPEERMAP 속성
- TLS 비밀 키 재협상을 사용하려면 채널의 양쪽에서 IBM MQ TLS를 사용해야 합니다. 따라서 AT-TLS를 사용하여 IBM MQ for z/OS 큐 관리자에 연결하는 경우 IBM MQ for Multiplatforms 큐 관리자 또는 클라이언트에서 TLS 비밀 키 재협상이 사용으로 설정되지 않아야 합니다.

큐 관리자에 대한 TLS 시크릿 키 재조정을 사용하지 않으려면 큐 관리자 SSLRKEYC 매개변수를 0(으)로 설정하십시오. 클라이언트의 경우, 클라이언트 유형에 따라 관련 매개변수를 0(으)로 설정하십시오. 이를 수행하는 방법에 대한 세부사항은 435 페이지의 『SSL 및 TLS 비밀 키 재설정』을 참조하십시오.

## AT-TLS 구성 명령문

AT-TLS는 명령문 세트를 사용하여 구성됩니다. 이 주제에 설명된 시나리오에서 사용되는 명령문은 다음과 같습니다.

**TTLSRule**
TCP/IP 연결을 TLS 구성과 일치시키기 위한 기준 세트를 지정합니다. 이는 다시 다른 명령문 유형을 참조합니다.

**TTLSGroupAction**
참조하는 TTLSRule이 사용으로 설정되었는지 여부를 지정합니다.

**TTLSEnvironmentAction**
참조하는 TTLSRule에 대한 자세한 구성을 지정하고 여러 가지 다른 명령문을 참조합니다.

**TTLSKeyringParms**
AT-TLS에서 사용할 키 링을 참조합니다.

**TTLSCipherParms**
사용될 암호 스위트를 정의합니다.

**TTLSEnvironmentAdvancedParms**
사용으로 설정된 TLS 또는 SSL 프로토콜을 정의합니다.

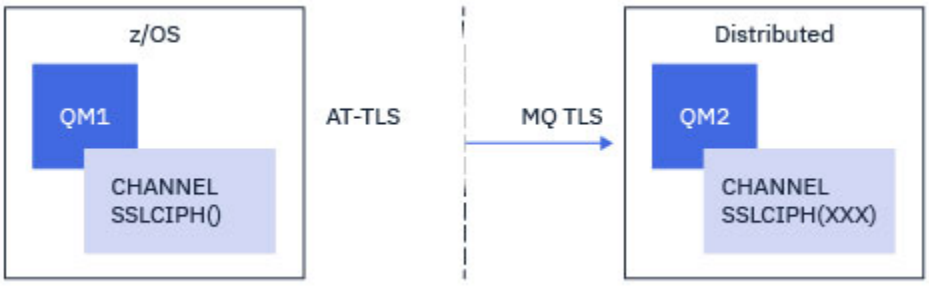
**주의:** 여기에 문서화되지 않은 AT-TLS를 사용하는 다른 AT-TLS 정책문이 있으며 필요에 따라 IBM MQ와 함께 사용할 수 있습니다. 그러나 IBM MQ는 이 주제에 설명된 정책으로만 테스트되었습니다.

## z/OS 이름 지정된 단일 *CipherSpec*을 사용하여 *IBM MQ for Multiplatforms* 큐 관리자로의 아웃바운드 채널에 *AT-TLS* 구성

IBM MQ for z/OS 큐 관리자에서 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS를 설정하는 방법입니다. 이 경우 z/OS 큐 관리자의 채널은 SSLCIPH 속성이 설정되지 않은 송신자 채널이고 비z/OS 큐 관리자의 채널은 SSLCIPH 속성이 이름 지정된 단일 CipherSpec으로 설정된 수신자 채널입니다.

알리어스 CipherSpec을 사용하는 예는 422 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS 구성』의 내용을 참조하십시오.

이 예에서 TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec을 사용하는 기존 송신자-수신자 채널 쌍은 송신자 채널이 IBM MQ TLS 대신 AT-TLS를 사용하도록 조정됩니다.




구성을 약간 조정하여 다른 TLS 프로토콜 및 CiperSpec을 사용할 수 있습니다. 클러스터 송신자 및 클러스터 수신자 채널 이외의 다른 메시지 채널 유형은 AT-TLS 구성을 변경하지 않고 사용할 수 있습니다.

## 프로시저

**1단계: 채널 중지**

**2단계: AT-TLS 정책 작성 및 적용**

이 시나리오를 위해 다음 AT-TLS문을 작성해야 합니다.

1. 채널 시작기 주소 공간에서 대상 수신자 채널의 IP 주소 및 포트 번호로의 아웃바운드 연결을 일치시키는 TTLSRule문. 이러한 값은 송신자 채널의 CONNAME에 사용된 정보와 일치해야 합니다. 여기서는 특정 채널 시작기 작업 이름과 일치시키기 위한 추가 필터링이 포함되었습니다.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   앞의 규칙은 CSQ1CHIN 작업에서 포트 1414의 IP 주소 123.456.78.9로 이동하는 연결과 일치합니다.

   고급 필터링 옵션은 TTLSRule에 설명되어 있습니다.

2. 규칙을 사용으로 설정하는 TTLSGroupAction문. TTLSRule 는 **TTLSGroupActionRef** 특성을 사용하여 TTLSGroupAction 를 참조합니다.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** 특성에 의해 TTLSRule 와 연관된 TTLSEnvironmentAction 문. TTLSEnvironmentAction은 TLS 환경을 구성하고 사용할 키 링을 지정합니다.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. TTLSKeyringParms 문은 **TTLSKeyringParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관되며 AT-TLS에서 사용되는 키 링을 정의합니다.

   키 링에는 원격 비z/OS 큐 관리자가 신뢰하는 인증서가 포함되어야 합니다. 이 키 링은 채널 시작기에서 사용되는 키 링과 동일한 방식으로 정의할 수 있습니다. 240 페이지의 『Configuring your z/OS system to use TLS』의 내용을 참조하십시오.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관된 TTLSCipherParms 문.

   이 명령문에는 대상 수신자 채널에 사용된 IBM MQ CipherSpec 이름과 동일해야 하는 단일 암호 스위트 이름이 포함되어야 합니다.

   **참고:** AT-TLS 암호 스위트 이름이 IBM MQ CipherSpec 이름과 반드시 일치하는 것은 아닙니다. 그러나 다음 표에서 IBM MQ CipherSpec 이름을 찾고 TTLSCipherParms 주제의 표 2에서 16진 코드 열을 확장된 문자 열과 상호 참조하여 IBM MQ CipherSpec 이름과 일치하는 AT-TLS 암호 스위트 이름을 찾을 수 있습니다.

표 *83. IBM MQ for z/OS 9.2.0*에서 *z/OS*의 *CipherSpec*

| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
|---|---|---|---|
| TLS_CHACHA20_POLY1305_SHA256 | TLS 1.3 | 1303 | 예 |
| TLS_AES_256_GCM_SHA384 | TLS 1.3 | 1302 | 예 |
| TLS_AES_128_GCM_SHA256 | TLS 1.3 | 1301 | 예 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 | TLS 1.2 | 009D | 예 |
| ECDHE_RSA_AES_256_GCM_SHA384 | TLS 1.2 | C030 | 예 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 | TLS 1.2 | 003D | 예 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 | TLS 1.2 | C024 | 예 |
| ECDHE_RSA_AES_256_CBC_SHA384 | TLS 1.2 | C028 | 예 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 | TLS 1.2 | 009C | 예 |
| ECDHE_RSA_AES_128_GCM_SHA256 | TLS 1.2 | C02F | 예 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 | TLS 1.2 | 003C | 예 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 | TLS 1.2 | C023 | 예 |
| ECDHE_RSA_AES_128_CBC_SHA256 | TLS 1.2 | C027 | 예 |
| TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | 003B | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA | TLS 1.0 | 0035 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA | TLS 1.0 | 002F | 아니오 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA | TLS 1.0 | 000A | 아니오 |
| TLS_RSA_WITH_RC4_128_SHA | TLS 1.0 | 0005 | 아니오 |
| TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | 0009 | 아니오 |
| TRIPLE_DES_SHA_US | SSL v3 | 000A | 아니오 |
| RC4_SHA_US | SSL v3 | 0005 | 아니오 |
| RC4_MD5_US | SSL v3 | 0004 | 아니오 |

| 표 *83. IBM MQ for z/OS 9.2.0*에서 *z/OS*의 *CipherSpec (계속)* | | | |
|---|---|---|---|
| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
| DES_SHA_EXPORT | SSL v3 | 0009 | N |
| RC4_MD5_EXPORT | SSL v3 | 0003 | 아니오 |
| RC2_MD5_EXPORT | SSL v3 | 0006 | 아니오 |
| NULL_SHA | SSL v3 | 0002 | 아니오 |
| NULL_MD5 | SSL v3 | 0001 | 아니오 |

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms 문은 **TTLSEnvironmentAdvancedParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관됩니다.

   이 명령문을 사용하여 사용 가능한 SSL및 TLS 프로토콜을 지정할 수 있습니다. IBM MQ 를 사용하면 TTLSCipherParms 문에서 사용되는 암호 스위트 이름과 일치하는 단일 프로토콜만 사용해야 합니다.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

전체 명령문 세트는 다음과 같으며 정책 에이전트에 적용되어야 합니다.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction              CSQ1-GROUP-ACTION
{
  TTLSEnabled                ON
}

TTLSEnvironmentAction              CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                    CLIENT
  TTLSKeyringParmsRef              CSQ1-KEYRING
  TTLSCipherParmsRef               CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms             CSQ1-KEYRING
{
  Keyring                    MQCHIN/CSQ1RING
}

TTLSCipherParms              CSQ1-CIPHERPARM
{
  V3CipherSuites             TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

**단계 3: z/OS 채널에서 SSLCIPH 제거**

다음 명령을 사용하여 z/OS 채널에서 CipherSpec 을 제거하십시오.

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

**4단계: 채널 시작**

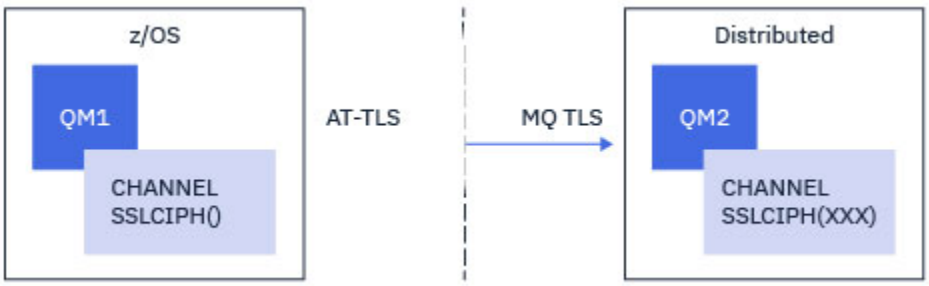
채널이 시작되면 AT-TLS와 IBM MQ TLS의 조합을 사용하게 됩니다.

**주의:** 앞의 AT-TLS 명령문은 최소 구성일 뿐입니다. 여기에 문서화되지 않은 AT-TLS를 사용하는 다른 AT-TLS 정책문이 있으며 필요에 따라 IBM MQ와 함께 사용할 수 있습니다. 그러나 IBM MQ 는 설명된 정책으로만 테스트되었습니다.

## z/OS 알리어스 *CipherSpec*을 사용하여 *IBM MQ for Multiplatforms* 큐 관리자로의 아웃바운드 채널에 *AT-TLS* 구성

IBM MQ for z/OS 큐 관리자에서 IBM MQ for Multiplatforms 큐 관리자로의 아웃바운드 채널에 AT-TLS를 설정하는 방법입니다. 이 경우 z/OS 큐 관리자의 채널은 SSLCIPH 속성이 설정되지 않은 송신자 채널이고 비z/OS 큐 관리자의 채널은 SSLCIPH 속성이 알리어스 CipherSpec으로 설정된 수신자 채널입니다.

이 예에서 ANY_TLS13 알리어스 CipherSpec을 사용하는 기존 송신자-수신자 채널 쌍은 송신자 채널이 IBM MQ TLS 대신 AT-TLS를 사용하도록 조정됩니다.




구성을 약간 조정하여 다른 TLS 프로토콜 및 CiperSpec을 사용할 수 있습니다. 클러스터 송신자 및 클러스터 수신자 채널 이외의 다른 메시지 채널 유형은 AT-TLS 구성을 변경하지 않고 사용할 수 있습니다.

## 프로시저

**1단계: 채널 중지**

**2단계: AT-TLS 정책 작성 및 적용**

이 시나리오를 위해 다음 AT-TLS문을 작성해야 합니다.

1. 채널 시작기 주소 공간에서 대상 수신자 채널의 IP 주소 및 포트 번호로의 아웃바운드 연결을 일치시키는 TTLSRule문. 이러한 값은 송신자 채널의 CONNAME에 사용된 정보와 일치해야 합니다. 여기서는 특정 채널 시작기 작업 이름과 일치시키기 위한 추가 필터링이 포함되었습니다.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   앞의 규칙은 CSQ1CHIN 작업에서 포트 1414의 IP 주소 123.456.78.9로 이동하는 연결과 일치합니다.

   고급 필터링 옵션은 TTLSRule에 설명되어 있습니다.

2. 규칙을 사용으로 설정하는 TTLSGroupAction문. TTLSRule 는 **TTLSGroupActionRef** 특성을 사용하여 TTLSGroupAction 를 참조합니다.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** 특성에 의해 TTLSRule 와 연관된 TTLSEnvironmentAction 문. TTLSEnvironmentAction은 TLS 환경을 구성하고 사용할 키 링을 지정합니다.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. TTLSKeyringParms 문은 **TTLSKeyringParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관되며 AT-TLS에서 사용되는 키 링을 정의합니다.

키 링에는 원격 비z/OS 큐 관리자가 신뢰하는 인증서가 포함되어야 합니다. 이 키 링은 채널 시작기에서 사용되는 키 링과 동일한 방식으로 정의할 수 있습니다. 240 페이지의 『Configuring your z/OS system to use TLS』의 내용을 참조하십시오.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관된 TTLSCipherParms 문.

   이 명령문에는 하나 이상의 암호 스위트 이름이 포함되어야 하며, 이 중 하나 이상이 대상 수신자 채널에서 사용되는 알리어스 CipherSpec이 암시하는 CipherSpec 세트와 호환되어야 합니다.

   **참고:** AT-TLS 암호 스위트 이름이 IBM MQ CipherSpec 이름과 반드시 일치하는 것은 아닙니다. 그러나 다음 표에서 IBM MQ CipherSpec 이름을 찾고 TTLSCipherParms 주제의 표 2에서 16진 코드 열을 확장된 문자열과 상호 참조하여 IBM MQ CipherSpec 이름과 일치하는 AT-TLS 암호 스위트 이름을 찾을 수 있습니다.

표 *84. IBM MQ for z/OS 9.2.0*에서 *z/OS*의 *CipherSpec*

| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
|---|---|---|---|
| TLS_CHACHA20_POLY1305_SHA256 | TLS 1.3 | 1303 | 예 |
| TLS_AES_256_GCM_SHA384 | TLS 1.3 | 1302 | 예 |
| TLS_AES_128_GCM_SHA256 | TLS 1.3 | 1301 | 예 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 | TLS 1.2 | 009D | 예 |
| ECDHE_RSA_AES_256_GCM_SHA384 | TLS 1.2 | C030 | 예 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 | TLS 1.2 | 003D | 예 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 | TLS 1.2 | C024 | 예 |
| ECDHE_RSA_AES_256_CBC_SHA384 | TLS 1.2 | C028 | 예 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 | TLS 1.2 | 009C | 예 |
| ECDHE_RSA_AES_128_GCM_SHA256 | TLS 1.2 | C02F | 예 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 | TLS 1.2 | 003C | 예 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 | TLS 1.2 | C023 | 예 |
| ECDHE_RSA_AES_128_CBC_SHA256 | TLS 1.2 | C027 | 예 |
| TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | 003B | 아니오 |

| 표 84. IBM MQ for z/OS 9.2.0에서 z/OS의 CipherSpec (계속) | | | |
|---|---|---|---|
| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
| TLS_RSA_WITH_AES_256_CBC_SHA | TLS 1.0 | 0035 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA | TLS 1.0 | 002F | 아니오 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA | TLS 1.0 | 000A | 아니오 |
| TLS_RSA_WITH_RC4_128_SHA | TLS 1.0 | 0005 | 아니오 |
| TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | 0009 | 아니오 |
| TRIPLE_DES_SHA_US | SSL v3 | 000A | 아니오 |
| RC4_SHA_US | SSL v3 | 0005 | 아니오 |
| RC4_MD5_US | SSL v3 | 0004 | 아니오 |
| DES_SHA_EXPORT | SSL v3 | 0009 | N |
| RC4_MD5_EXPORT | SSL v3 | 0003 | 아니오 |
| RC2_MD5_EXPORT | SSL v3 | 0006 | 아니오 |
| NULL_SHA | SSL v3 | 0002 | 아니오 |
| NULL_MD5 | SSL v3 | 0001 | 아니오 |

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

**주의:** 큐 관리자 및 AT-TLS 정책 모두 TLS 1.3을 지원하는 경우, 하나 이상의 TLS 1.3 CipherSpec 을 포함하는 알리어스 CipherSpecs 만 채널이 시작되도록 허용합니다. 예를 들어, ANY_TLS12 를 사용하면 TTLSCipherParms 가 TLS 1.2 CipherSpecs를 포함하지만 ANY_TLS12_OR_HIGHER 또는 ANY_TLS13 을 사용하면 채널을 시작할 수 있는 경우에도 채널이 시작되지 않습니다. 설명은 411 페이지의 『알리어스 CipherSpec 설정 간 관계』 의 내용을 참조하십시오.

6. TTLSEnvironmentAdvancedParms 문은 **TTLSEnvironmentAdvancedParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관됩니다.

   이 명령문은 사용 가능한 SSL및 TLS 프로토콜을 지정하는 데 사용할 수 있으며 TTLSCipherParms 문의 암호 스위트와 일치해야 합니다.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

전체 명령문 세트는 다음과 같으며 정책 에이전트에 적용되어야 합니다.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

**단계 3: z/OS 채널에서 SSLCIPH 제거**

다음 명령을 사용하여 z/OS 채널에서 CipherSpec 을 제거하십시오.

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

**4단계: 채널 시작**

채널이 시작되면 AT-TLS와 IBM MQ TLS의 조합을 사용하게 됩니다.

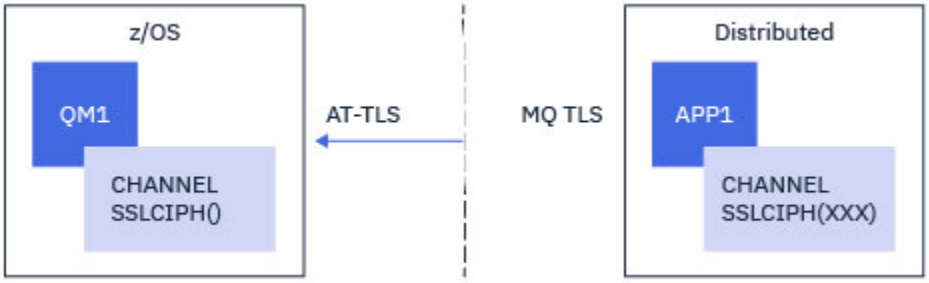
**주의:** 앞의 AT-TLS 명령문은 최소 구성일 뿐입니다. 여기에 문서화되지 않은 AT-TLS를 사용하는 다른 AT-TLS 정책문이 있으며 필요에 따라 IBM MQ와 함께 사용할 수 있습니다. 그러나 IBM MQ 는 설명된 정책으로만 테스트되었습니다.

### z/OS 이름 지정된 단일 *CipherSpec*을 사용하여 *IBM MQ for Multiplatforms* 큐 관리자로부터의 인바운드 채널에 *AT-TLS* 구성

IBM MQ for Multiplatforms 큐 관리자에서 IBM MQ for z/OS 큐 관리자로의 인바운드 채널에 AT-TLS를 설정하는 방법입니다. 이 경우 z/OS 큐 관리자의 채널은 SSLCIPH 속성이 설정되지 않은 수신자 채널이고 비z/OS 큐 관리자의 채널은 SSLCIPH 속성이 이름 지정된 단일 CipherSpec으로 설정된 송신자 채널입니다.

알리어스 CipherSpec을 사용하는 예는 430 페이지의 『알리어스 CipherSpec을 사용하여 IBM MQ for Multiplatforms 큐 관리자로부터의 인바운드 채널에 AT-TLS 구성』의 내용을 참조하십시오.

이 예에서 TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec을 사용하는 기존 송신자-수신자 채널 쌍은 수신자 채널이 IBM MQ TLS 대신 AT-TLS를 사용하도록 조정됩니다.




구성을 약간 조정하여 다른 TLS 프로토콜 및 CiperSpec을 사용할 수 있습니다. 클러스터 송신자 및 클러스터 수신자 채널 이외의 다른 메시지 채널 유형은 AT-TLS 구성을 변경하지 않고 사용할 수 있습니다.

## 프로시저

### 1단계: 채널 중지

### 2단계: AT-TLS 정책 작성 및 적용

이 시나리오를 위해 다음 AT-TLS문을 작성해야 합니다.

1. 송신자 채널의 IP 주소에서 채널 시작기 주소 공간으로의 인바운드 연결을 일치시키는 TTLSRule문. 여기서는 특정 채널 시작기 작업 이름과 일치시키기 위한 추가 필터링이 포함되었습니다.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   앞의 규칙은 원격 IP 주소 123.456.78.9에서 로컬 포트 1414의 CSQ1CHIN 작업으로 들어오는 연결과 일치합니다.

   고급 필터링 옵션은 TTLSRule에 설명되어 있습니다.

2. 규칙을 사용으로 설정하는 TTLSGroupAction문. TTLSRule 는 **TTLSGroupActionRef** 특성을 사용하여 TTLSGroupAction 를 참조합니다.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction 문은 **TTLSEnvironmentActionRef** 특성에 의해 TTLSRule 와 연관됩니다. TTLSEnvironmentAction은 TLS 환경을 구성하고 사용할 키 링을 지정합니다.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS는 SSLCAUTH 채널 속성을 사용하는 것과 동일한 상호 인증을 제공하는 기능을 제공합니다. 이는 인바운드 TTLSEnvironmentAction 문에 대해 **HandshakeRole** 값이 *ServerWithClientAuth* 인 TTLSEnvironmentAction 문을 사용하여 수행됩니다.

4. TTLSKeyringParms 문은 **TTLSKeyringParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관되며 AT-TLS에서 사용되는 키 링을 정의합니다.

   키 링에는 원격 비z/OS 큐 관리자가 신뢰하는 인증서가 포함되어야 합니다. 이 키 링은 채널 시작기에서 사용되는 키 링과 동일한 방식으로 정의할 수 있습니다. 240 페이지의 『Configuring your z/OS system to use TLS』 의 내용을 참조하십시오.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관된 TTLSCipherParms 문.

   이 명령문에는 원격 송신자 채널에 사용된 IBM MQ CipherSpec 이름과 동일해야 하는 단일 암호 스위트 이름이 포함되어야 합니다.

   **참고:** AT-TLS 암호 스위트 이름이 IBM MQ CipherSpec 이름과 반드시 일치하는 것은 아닙니다. 그러나 다음 표에서 IBM MQ CipherSpec 이름을 찾고 TTLSCipherParms 주제의 표 2에서 16진 코드 열을 확장된 문자열과 상호 참조하여 IBM MQ CipherSpec 이름과 일치하는 AT-TLS 암호 스위트 이름을 찾을 수 있습니다.

표 *85. IBM MQ for z/OS 9.2.0*에서 *z/OS*의 *CipherSpec*

| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
|---|---|---|---|
| TLS_CHACHA20_POLY1305_SHA256 | TLS 1.3 | 1303 | 예 |
| TLS_AES_256_GCM_SHA384 | TLS 1.3 | 1302 | 예 |
| TLS_AES_128_GCM_SHA256 | TLS 1.3 | 1301 | 예 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 | TLS 1.2 | 009D | 예 |
| ECDHE_RSA_AES_256_GCM_SHA384 | TLS 1.2 | C030 | 예 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 | TLS 1.2 | 003D | 예 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 | TLS 1.2 | C024 | 예 |
| ECDHE_RSA_AES_256_CBC_SHA384 | TLS 1.2 | C028 | 예 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 | TLS 1.2 | 009C | 예 |
| ECDHE_RSA_AES_128_GCM_SHA256 | TLS 1.2 | C02F | 예 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 | TLS 1.2 | 003C | 예 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 | TLS 1.2 | C023 | 예 |

| 표 85. IBM MQ for z/OS 9.2.0에서 z/OS의 CipherSpec (계속) | | | |
|---|---|---|---|
| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
| ECDHE_RSA_AES_128_CBC_SHA256 | TLS 1.2 | C027 | 예 |
| TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | 003B | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA | TLS 1.0 | 0035 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA | TLS 1.0 | 002F | 아니오 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA | TLS 1.0 | 000A | 아니오 |
| TLS_RSA_WITH_RC4_128_SHA | TLS 1.0 | 0005 | 아니오 |
| TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | 0009 | 아니오 |
| TRIPLE_DES_SHA_US | SSL v3 | 000A | 아니오 |
| RC4_SHA_US | SSL v3 | 0005 | 아니오 |
| RC4_MD5_US | SSL v3 | 0004 | 아니오 |
| DES_SHA_EXPORT | SSL v3 | 0009 | N |
| RC4_MD5_EXPORT | SSL v3 | 0003 | 아니오 |
| RC2_MD5_EXPORT | SSL v3 | 0006 | 아니오 |
| NULL_SHA | SSL v3 | 0002 | 아니오 |
| NULL_MD5 | SSL v3 | 0001 | 아니오 |

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms 문은 **TTLSEnvironmentAdvancedParmsRef** 특성에 의해 `TTLSEnvironmentAction` 와 연관됩니다.

   이 명령문을 사용하여 사용 가능한 SSL및 TLS 프로토콜을 지정할 수 있습니다. IBM MQ 를 사용하면 `TTLSCipherParms` 문에서 사용되는 암호 스위트 이름과 일치하는 단일 프로토콜만 사용해야 합니다.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

전체 명령문 세트는 다음과 같으며 정책 에이전트에 적용되어야 합니다.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction              CSQ1-GROUP-ACTION
{
  TTLSEnabled                ON
}

TTLSEnvironmentAction              CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                    SERVER
  TTLSKeyringParmsRef              CSQ1-KEYRING
  TTLSCipherParmsRef               CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms             CSQ1-KEYRING
{
  Keyring                    MQCHIN/CSQ1RING
}

TTLSCipherParms              CSQ1-CIPHERPARM
{
  V3CipherSuites             TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

**단계 3: z/OS 채널에서 SSLCIPH 제거**

다음 명령을 사용하여 z/OS 채널에서 CipherSpec 을 제거하십시오.

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

**4단계: 채널 시작**

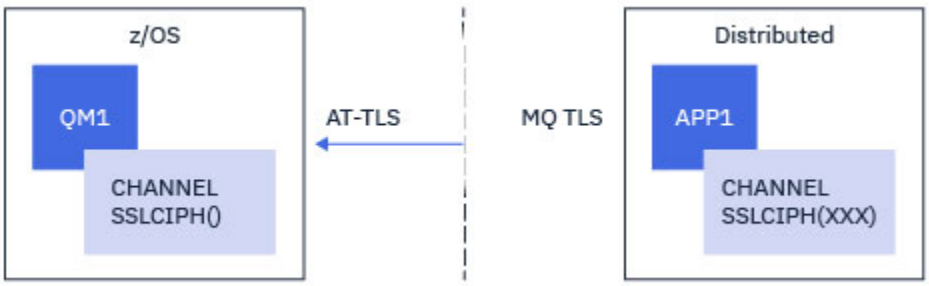
채널이 시작되면 AT-TLS와 IBM MQ TLS의 조합을 사용하게 됩니다.

**주의:** 앞의 AT-TLS 명령문은 최소 구성일 뿐입니다. 여기에 문서화되지 않은 AT-TLS를 사용하는 다른 AT-TLS 정책문이 있으며 필요에 따라 IBM MQ와 함께 사용할 수 있습니다. 그러나 IBM MQ 는 설명된 정책으로만 테스트되었습니다.

## z/OS 알리어스 *CipherSpec*을 사용하여 *IBM MQ for Multiplatforms* 큐 관리자로부터의 인바운드 채널에 *AT-TLS* 구성

IBM MQ for Multiplatforms 큐 관리자에서 IBM MQ for z/OS 큐 관리자로의 인바운드 채널에 AT-TLS를 설정하는 방법입니다. 이 경우 z/OS 큐 관리자의 채널은 SSLCIPH 속성이 설정되지 않은 수신자 채널이고 비z/OS 큐 관리자의 채널은 SSLCIPH 속성이 알리어스 CipherSpec으로 설정된 송신자 채널입니다.

이 예에서 TLS 1.3 CipherSpec을 사용하는 기존 송신자-수신자 채널 쌍은 수신자 채널이 IBM MQ TLS 대신 AT-TLS를 사용하도록 조정됩니다.




구성을 약간 조정하여 다른 TLS 프로토콜 및 CiperSpec을 사용할 수 있습니다. 클러스터 송신자 및 클러스터 수신자 채널 이외의 다른 메시지 채널 유형은 AT-TLS 구성을 변경하지 않고 사용할 수 있습니다.

## 프로시저

### 1단계: 채널 중지

### 2단계: AT-TLS 정책 작성 및 적용

이 시나리오를 위해 다음 AT-TLS문을 작성해야 합니다.

1. 송신자 채널의 IP 주소에서 채널 시작기 주소 공간으로의 인바운드 연결을 일치시키는 TTLSRule문. 여기서는 특정 채널 시작기 작업 이름과 일치시키기 위한 추가 필터링이 포함되었습니다.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   앞의 규칙은 원격 IP 주소 123.456.78.9에서 로컬 포트 1414의 CSQ1CHIN 작업으로 들어오는 연결과 일치합니다.

   고급 필터링 옵션은 TTLSRule에 설명되어 있습니다.

2. 규칙을 사용으로 설정하는 TTLSGroupAction문. TTLSRule 는 **TTLSGroupActionRef** 특성을 사용하여 TTLSGroupAction 를 참조합니다.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction 문은 **TTLSEnvironmentActionRef** 특성에 의해 TTLSRule 와 연관됩니다. TTLSEnvironmentAction은 TLS 환경을 구성하고 사용할 키 링을 지정합니다.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS는 SSLCAUTH 채널 속성을 사용하는 것과 동일한 상호 인증을 제공하는 기능을 제공합니다. 이는 인바운드 TTLSEnvironmentAction 문에 대해 **HandshakeRole** 값이 *ServerWithClientAuth* 인 TTLSEnvironmentAction 문을 사용하여 수행됩니다.

4. TTLSKeyringParms 문은 **TTLSKeyringParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관되며 AT-TLS에서 사용되는 키 링을 정의합니다.

   키 링에는 원격 비z/OS 큐 관리자가 신뢰하는 인증서가 포함되어야 합니다. 이 키 링은 채널 시작기에서 사용되는 키 링과 동일한 방식으로 정의할 수 있습니다. 240 페이지의 『Configuring your z/OS system to use TLS』 의 내용을 참조하십시오.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관된 TTLSCipherParms 문.

   이 명령문에는 원격 송신자 채널에 설정된 알리어스 CipherSpec에 포함된 하나 이상의 암호 스위트 이름이 포함되어야 합니다.

   **참고:** AT-TLS 암호 스위트 이름이 IBM MQ CipherSpec 이름과 반드시 일치하는 것은 아닙니다. 그러나 다음 표에서 IBM MQ CipherSpec 이름을 찾고 TTLSCipherParms 주제의 표 2에서 16진 코드 열을 확장된 문자열과 상호 참조하여 IBM MQ CipherSpec 이름과 일치하는 AT-TLS 암호 스위트 이름을 찾을 수 있습니다.

표 *86. IBM MQ for z/OS 9.2.0*에서 *z/OS*의 *CipherSpec*

| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
|---|---|---|---|
| TLS_CHACHA20_POLY1305_SHA256 | TLS 1.3 | 1303 | 예 |
| TLS_AES_256_GCM_SHA384 | TLS 1.3 | 1302 | 예 |
| TLS_AES_128_GCM_SHA256 | TLS 1.3 | 1301 | 예 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 | TLS 1.2 | 009D | 예 |
| ECDHE_RSA_AES_256_GCM_SHA384 | TLS 1.2 | C030 | 예 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 | TLS 1.2 | 003D | 예 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 | TLS 1.2 | C024 | 예 |
| ECDHE_RSA_AES_256_CBC_SHA384 | TLS 1.2 | C028 | 예 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 | TLS 1.2 | 009C | 예 |
| ECDHE_RSA_AES_128_GCM_SHA256 | TLS 1.2 | C02F | 예 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 | TLS 1.2 | 003C | 예 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 | TLS 1.2 | C023 | 예 |

| 표 86. IBM MQ for z/OS 9.2.0에서 z/OS의 CipherSpec (계속) | | | |
|---|---|---|---|
| **CipherSpec** | **프로토콜** | **16진 코드** | **기본적으로 사용됨** |
| ECDHE_RSA_AES_128_CBC_SHA256 | TLS 1.2 | C027 | 예 |
| TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | 003B | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA | TLS 1.0 | 0035 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA | TLS 1.0 | 002F | 아니오 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA | TLS 1.0 | 000A | 아니오 |
| TLS_RSA_WITH_RC4_128_SHA | TLS 1.0 | 0005 | 아니오 |
| TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | 0009 | 아니오 |
| TRIPLE_DES_SHA_US | SSL v3 | 000A | 아니오 |
| RC4_SHA_US | SSL v3 | 0005 | 아니오 |
| RC4_MD5_US | SSL v3 | 0004 | 아니오 |
| DES_SHA_EXPORT | SSL v3 | 0009 | N |
| RC4_MD5_EXPORT | SSL v3 | 0003 | 아니오 |
| RC2_MD5_EXPORT | SSL v3 | 0006 | 아니오 |
| NULL_SHA | SSL v3 | 0002 | 아니오 |
| NULL_MD5 | SSL v3 | 0001 | 아니오 |

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

**주의:** 큐 관리자 및 AT-TLS 정책 모두 TLS 1.3을 지원하는 경우, 하나 이상의 TLS 1.3 CipherSpec 을 포함하는 알리어스 CipherSpecs 만 채널이 시작되도록 허용합니다. 예를 들어, ANY_TLS12 를 사용하면 TTLSCipherParms 가 TLS 1.2 CipherSpecs를 포함하지만 ANY_TLS12_OR_HIGHER 또는 ANY_TLS13 을 사용하면 채널을 시작할 수 있는 경우에도 채널이 시작되지 않습니다. 설명은 411 페이지의 『알리어스 CipherSpec 설정 간 관계』 의 내용을 참조하십시오.

6. TTLSEnvironmentAdvancedParms 문은 **TTLSEnvironmentAdvancedParmsRef** 특성에 의해 TTLSEnvironmentAction 와 연관됩니다.

   이 명령문은 사용 가능한 SSL및 TLS 프로토콜을 지정하는 데 사용할 수 있으며 TTLSCipherParms 문의 암호 스위트와 일치해야 합니다.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

전체 명령문 세트는 다음과 같으며 정책 에이전트에 적용되어야 합니다.

```
TTLSRule                      REMOTE-TO-CSQ1
{
  LocalAddr                   ALL
  LocalPortRange              1414
  RemoteAddr                  123.456.78.9
  Jobname                     CSQ1CHIN
  Direction                   INBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

#### 단계 3: z/OS 채널에서 SSLCIPH 제거

다음 명령을 사용하여 z/OS 채널에서 CipherSpec 을 제거하십시오.

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

#### 4단계: 채널 시작

채널이 시작되면 AT-TLS와 IBM MQ TLS의 조합을 사용하게 됩니다.

**주의:** 앞의 AT-TLS 명령문은 최소 구성일 뿐입니다. 여기에 문서화되지 않은 AT-TLS를 사용하는 다른 AT-TLS 정책문이 있으며 필요에 따라 IBM MQ와 함께 사용할 수 있습니다. 그러나 IBM MQ 는 설명된 정책으로만 테스트되었습니다.

# SSL 및 TLS 비밀 키 재설정

IBM MQ는 큐 관리자 및 클라이언트에서 보안 키의 재설정을 지원합니다.

보안 키는 암호화되지 않은 데이터 바이트의 지정된 수가 채널을 통해 흐를 때 재설정됩니다. 채널 하트비트가 사용 가능하면 채널 하트비트 후에 데이터를 송신하거나 수신하기 전에 보안 키가 재설정됩니다.

이 키 재설정 값은 항상 IBM MQ 채널의 한쪽을 시작하여 설정됩니다.

## 큐 관리자

큐 관리자의 경우, **SSLRKEYC** 매개변수와 함께 **ALTER QMGR** 명령을 사용하여 키 재조정 중에 사용되는 값을 설정하십시오.

IBM i에서는 **SSLRSTCNT** 매개변수와 함께 **CHGMQM** 를 사용하십시오.

## MQI 클라이언트

기본적으로 MQI 클라이언트는 보안 키를 재협상하지 않습니다. 세 가지 방법 중 하나를 사용하여 MQI 클라이언트가 키를 재협상하게 만들 수 있습니다. 다음 목록에서 방법은 우선순위의 순서대로 표시됩니다. 여러 값을 지정하는 경우에는 우선순위가 가장 높은 값이 사용됩니다.

1. MQCONNX 호출에서 MQSCO 구조의 KeyResetCount 필드를 사용합니다.
2. 환경 변수 **MQSSLRESET**사용.
3. 클라이언트 구성 파일의 SSL 스탠자에서 **SSLKeyResetCount** 속성을 설정합니다.

이 변수는 0에서 999 999 999까지의 정수로 설정할 수 있으며 이 숫자는 TLS 보안 키가 재조정되기 전에 TLS 대화 내에서 송신 및 수신된 암호화되지 않은 바이트 수를 나타냅니다. 0 값을 지정하면 TLS 보안 키가 재조정되지 않음을 나타냅니다. TLS 비밀 키 재설정 계수를 1B - 32KB 범위로 지정하는 경우 TLS 채널은 32KB의 비밀 키 재설정 계수를 사용합니다. 이는 작은 TLS 보안 키 재설정 값으로 인해 발생할 수 있는 지나친 키 재설정을 피하기 위한 것입니다.

0보다 큰 값이 지정되고 채널 하트비트가 채널에 사용되는 경우 보안 키는 메시지 데이터가 채널 하트비트 후에 송신 또는 수신되기 전에도 재결정됩니다.

각각의 재조정이 이루어지면 다음 비밀 키 재조정까지의 바이트 수가 재설정됩니다.

## Java

IBM MQ classes for Java의 경우 애플리케이션은 다음 방법으로 보안 키를 재설정할 수 있습니다.

- MQEnvironment 클래스에 sslResetCount 필드 설정
- 해시 테이블 오브젝트에 환경 특성 MQC.SSL_RESET_COUNT_PROPERTY 설정. 그러면 애플리케이션이 해시 테이블을 MQEnvironment 클래스의 `properties` 필드에 지정하거나 해시 테이블을 해당 구성자의 MQQueueManager 오브젝트로 전달합니다.

애플리케이션이 이러한 방법 중 하나 이상을 사용하는 경우 일반 서열 규칙이 적용됩니다. 서열 규칙에 대해서는 클래스 com.ibm.mq.MQEnvironment를 참조하십시오.

sslResetCount 필드 또는 환경 변수 MQC.SSL_RESET_COUNT_PROPERTY의 값은 비밀 키를 협상하기 전에 IBM MQ classes for Java 클라이언트 코드가 송신하고 수신한 총 바이트 수를 나타냅니다. 송신된 바이트 수는 암호화 전의 바이트 수이며, 수신된 바이트 수는 복호화 후의 바이트 수입니다. 바이트 수에는 IBM MQ classes for Java 클라이언트가 송신하고 수신한 제어 정보도 포함됩니다.

재설정 수가 기본값인 0인 경우 비밀 키는 재협상되지 않습니다. CipherSuite가 지정되지 않으면 재설정 수가 무시됩니다.

## JMS

IBM MQ classes for JMS의 경우 SSLRESETCOUNT 특성은 암호화에 사용된 보안 키가 재협상되기 전에 연결에 의해 전송 및 수신된 총 바이트 수를 나타냅니다. 송신된 바이트 수는 암호화 전의 바이트 수이며, 수신된 바이트 수는 복호화 후의 바이트 수입니다. 바이트 수에는 IBM MQ classes for JMS에 의해 송신되거나 수신된 제어 정

보도 포함됩니다. 예를 들어, 4MB 데이터가 플로우된 후 재조정되는 비밀 키가 포함된 SSL 또는 TLS 사용 MQI 채널을 통한 연결을 작성하는 데 사용될 수 있는 ConnectionFactory 오브젝트를 구성하려면 JMSAdmin에 다음 명령을 실행하십시오.

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

SSLRESETCOUNT 값이 기본값인 0인 경우 비밀 키가 재협상되지 않습니다. SSLCIPHERSUITE를 설정하지 않으면 SSLRESETCOUNT 특성이 무시됩니다.

### .NET

.NET 비관리 클라이언트의 경우, 정수 특성 **SSLKeyResetCount** 는 비밀 키가 재조정되기 전에 TLS 대화 내에서 송신 및 수신된 암호화되지 않은 바이트 수를 표시합니다. IBM MQ classes for .NET에서 오브젝트 특성 사용에 대한 자세한 정보는 속성 값 가져오기 및 설정을 참조하십시오.

.NET 관리 클라이언트의 경우, SSLStream 클래스가 비밀 키 재설정/재협상을 지원하지 않습니다. 그러나 다른 IBM MQ 클라이언트와 일관성을 유지하기 위해 IBM MQ 관리 .NET 클라이언트는 애플리케이션이 **SSLKeyResetCount**를 설정하도록 허용합니다. 자세한 정보는 비밀 키 재설정 또는 재협상을 참조하십시오.

### XMS .NET

XMS .NET 비관리 클라이언트의 경우 IBM MQ 큐 관리자에 대한 보안 연결을 참조하십시오.

**관련 참조**

ALTER QMGR
큐 관리자 표시
메시지 큐 관리자 변경(CHGMQM)
메시지 큐 관리자 표시(DSPMQM)

## 사용자 엑시트 프로그램에서 기밀성 구현

### 보안 엑시트에서 기밀성 구현

보안 엑시트가 채널에서 플로우되는 데이터를 암호화하고 해독하기 위해 대칭 키를 생성하고 분배하여 기밀성 서비스에서 역할을 할 수 있습니다. 이것을 수행하는 공통적인 기술은 PKI 기법을 사용합니다.

한 보안 엑시트가 무작위 데이터 값을 생성하고, 파트너 보안 엑시트가 나타내고 있는 큐 관리자나 사용자의 공개 키로 그것을 암호화하며, 암호화된 데이터를 보안 메시지로 그 파트너에게 송신합니다. 파트너 보안 엑시트가 그것이 나타내고 있는 큐 관리자나 사용자의 개인 키로 무작위 데이터 값을 해독합니다. 각 보안 엑시트가 이제 그들 모두에게 알려져 있는 알고리즘을 사용하여 다른 엑시트와 독립적으로 대칭 키를 도출하는 데 무작위 데이터 값을 사용할 수 있습니다. 또는 무작위 데이터 값을 키로 사용할 수도 있습니다.

첫 번째 보안 엑시트가 이 때에 그 파트너를 인증하지 않으면, 파트너에 의해 송신되는 다음 보안 메시지가 대칭 키로 암호화된 예상 값을 가질 수 있습니다. 첫 번째 보안 엑시트가 이제 파트너 보안 엑시트가 예상한 값을 제대로 암호화할 수 있었는지를 검사하여 그 파트너를 인증할 수 있습니다.

둘 이상의 알고리즘이 사용 가능한 경우, 이 기회를 사용하여 보안 엑시트가 채널에서 플로우되는 데이터를 암호화하고 암호를 해독하기 위한 알고리즘에 동의할 수 있습니다.

### 메시지 엑시트에서 기밀성 구현

채널의 송신 측에 있는 메시지 엑시트가 메시지에 있는 응용프로그램 데이터를 암호화하고 채널의 수신 측에 있는 다른 메시지 엑시트가 데이터를 해독할 수 있습니다. 성능을 이유로 보통 대칭 키 알고리즘이 이 목적으로 사용됩니다. 대칭 키가 생성되고 분배될 수 있는 방법에 대한 자세한 정보는 436 페이지의 『사용자 엑시트 프로그램에서 기밀성 구현』의 내용을 참조하십시오.

임베드된 메시지 설명자를 포함하는 트랜스미션 큐 헤더인 MQXQH 등의 메시지 헤더는 메시지 엑시트에 의해 암호화돼서는 안됩니다. 이것은 메시지 엑시트가 송신 측에 의해 호출된 후, 또는 메시지 엑시트가 수신 측에 의해

호출되기 전에 메시지 헤더의 데이터 변환이 일어나기 때문입니다. 헤더가 암호화되면, 데이터 변환이 실패하고 채널이 중지됩니다.

### 송신 및 수신 엑시트에서 기밀성 구현

송신과 수신 엑시트는 채널에서 플로우되는 데이터를 암호화하고 해독하는 데 사용될 수 있습니다. 다음과 같은 이유로 인해 이 서비스를 제공하기 위해서는 메시지 엑시트보다 이들이 더 적합합니다.

- 메시지 채널에서는 메시지의 응용프로그램 데이터뿐만 아니라 메시지 헤더도 암호화될 수 있습니다.
- 송신과 수신 엑시트는 메시지 채널뿐만 아니라 MQI 채널에서도 사용될 수 있습니다. MQI 호출의 매개변수들에는 MQI 채널에서 플로우되는 동안에 보호되어야 하는 민감한 응용프로그램 데이터가 있을 수 있습니다. 그러므로 두 종류의 채널 모두에서 동일한 송신 및 수신 엑시트를 사용할 수 있습니다.

### API 엑시트 및 API 교차 엑시트에서 기밀성 구현

메시지의 애플리케이션 데이터는 메시지가 송신 애플리케이션에 의해 놓일 때 API 또는 API 교차 엑시트에 의해 암호화될 수 있고 메시지가 수신 애플리케이션에 의해 검색될 때 두 번째 엑시트에 의해 복호화될 수 있습니다. 성능을 이유로 보통 대칭 키 알고리즘이 이 목적으로 사용됩니다. 그러나, 애플리케이션 레벨에서는 많은 사용자가 서로 메시지를 송신할 수도 있는 상황에서, 문제는 의도된 메시지 수신자만이 메시지의 암호를 해독할 수 있게 보장하는 방법입니다. 한 가지 솔루션은 메시지를 서로 송신하는 사용자의 각 쌍에 다른 대칭 키를 사용하는 것입니다. 그러나, 이 솔루션은 관리하기에 어렵고 시간이 많이 소요될 수 있습니다. 특히, 사용자가 다른 조직에 속하는 경우가 그렇습니다. 이 문제점을 해결하는 표준 방법은 디지털 봉합으로 알려져 있으며 PKI 기술을 사용합니다.

애플리케이션이 메시지를 큐에 넣을 때 API 또는 API 교차 엑시트는 임의 대칭 키를 생성하고 메시지에서 애플리케이션 데이터를 암호화하기 위해 키를 사용합니다. 엑시트는 대칭 키를 의도한 수신자의 공개 키로 암호화합니다. 그런 후, 메시지의 애플리케이션 데이터를 암호화된 애플리케이션 데이터와 암호화된 대칭 키로 바꿉니다. 이런 식으로, 의도된 수신자만이 대칭 키, 즉 애플리케이션 데이터의 암호를 해독할 수 있습니다. 암호화된 메시지에 둘 이상의 의도된 수신자가 있는 경우 엑시트는 각 의도한 수신자의 대칭 키의 사본을 암호화할 수 있습니다.

애플리케이션 데이터를 암호화하고 복호화하기 위한 서로 다른 알고리즘을 사용할 수 있는 경우 엑시트에는 사용된 알고리즘의 이름을 포함할 수 있습니다.

## z/OS Confidentiality for data at rest on IBM MQ for z/OS with data set encryption

IBM MQ for z/OS can harden customer and configuration data by writing the data to the active log data sets, the archive log data sets, page sets, boot strap data sets (BSDS), and shared message data sets (SMDS).

z/OS provides efficient, policy-based encryption of data sets. IBM MQ for z/OS supports z/OS data set encryption for:

- Active log data sets; see note “1” on page 437
- Archive log data sets; see note “2” on page 437
- Page sets; see note “1” on page 437
- BSDS; see note “2” on page 437
- CSQINP* data sets; see note “2” on page 437
- SMDS; see note “1” on page 437

This provides confidentiality of data at rest on an individual z/OS queue manager.

**Notes:**

1. From IBM MQ for z/OS 9.2.0, z/OS data set encryption for active logs. page sets, and SMDS are supported.
2. Data set encryption for archive logs, BSDS and CSQINP* data sets is supported on all versions of IBM MQ for z/OS.

3. IBM MQ Advanced Message Security provides an alternative mechanism of protecting data at rest. In addition AMS also protects data in memory and in flight

See Using the z/OS data set encryption enhancements for more information about z/OS data set encryption.

Configuration of z/OS data set encryption is outside of the control of IBM MQ for z/OS. Encryption settings take effect when the data set is created.

This means that any existing data sets need to be recreated before a new data set encryption policy can be used.

IBM MQ for z/OS can run with a mixture of encrypted and non-encrypted data sets, but a standard configuration would encrypt all, or none, of the data sets used.

## z/OS Overview of steps to encrypt an IBM MQ for z/OS data set

How you encrypt an IBM MQ for z/OS data set.

### Before you begin

You must ensure that you have configured z/OS data set encryption correctly in your enterprise. If you are setting up data set encryption in a queue sharing group, you must configure z/OS data set encryption for data sharing.

**Note:** A z/OS encrypted data set must be an extended format data set.

### Procedure

1. Set up encryption key and `key-label` in RACF to use to encrypt the data set.
2. Create a profile for `key-label` in the RACF CSFKEYS class.
3. Grant READ access to the user Id of the queue manager, and any other user Ids that need access to the encrypted data.

   This might include user IDs that are used to run print utilities against the data set. For example, the user running CSQUTIL SCOPY would need to decrypt the relevant page set.
4. Associate the encryption `key-label` with the data set name.

   You can do this by using an SMS data class, or a RACF DFP segment, for the data set name or high-level qualifier.

   You can also associate the `key-label` with the data set when the data set is allocated.
5. Rename any existing data set using IDCAMS ALTER.
6. Re-allocate the data set with the appropriate attributes.
7. Copy the contents of the renamed data set to the new data set using IDCAMS REPRO.

   The data is encrypted by the action of copying it into the data set.
8. Repeat steps “4” on page 438 to “6” on page 438 for any other data sets that need to be encrypted.

## z/OS Example of how to encrypt queue manager active logs

The following topics guide you through the process of enabling data set encryption on existing active logs.

**Note:** The process for other data sets is similar to that for active logs.

In this example:

- Queue manager CSQ1 is run under user QMCSQ1, and has active log data sets CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, and so on
- The hardware and software environment is capable of using z/OS data set encryption
- RACF is used as the SAF
- The queue manager has been stopped

Carry out the procedure in the following order:

1. "Configuring the data set encryption key for the queue manager" on page 439
2. "Configuring data set encryption for the log data sets" on page 439

## z/OS Configuring the data set encryption key for the queue manager

How you configure a data set encryption key for a queue manager.

### About this task

This task is a prerequisite for "Configuring data set encryption for the log data sets" on page 439.

### Procedure

1. Set up an AES-256 bit encryption DATA key with a label, for example, CSQ1DSKY, using the z/OS key generator utility program (KGUP).
2. Define the RACF CSFKEYS profile for the CSQ1DSKY encryption key, by issuing the following command:

   ```
   RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
   ```

3. Configure the ICSF segment of the profile to allow the key to be used as a protected key, by issuing the following command:

   ```
   RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
   ```

4. Allow the queue manager to use the encryption key by giving QMCSQ1 READ access to the profile, by issuing the following command:

   ```
   PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
   ```

   Give the same access to any administrative user that needs to read or write the encrypted data set.
5. Refresh the CSFKEYS class by issuing the following command.

   ```
   SETROPTS RACLIST(CSFKEYS) REFRESH
   ```

### What to do next

Configure data set encryption for the data sets as described in "Configuring data set encryption for the log data sets" on page 439

## z/OS Configuring data set encryption for the log data sets

How you configure the encryption on the log data sets.

### Before you begin

Ensure that you have read:

Overview of steps to encrypt an IBM MQ for z/OS data set, and carried out the procedure in "Configuring the data set encryption key for the queue manager" on page 439

### About this task

This method uses the DFP segment of a RACF generic profile, so that you can use the encryption key for all new data sets that match the profile.

Alternatively, you can configure and use an SMS data class, or the key label can be specified directly when allocating the data set.

As previously described, in this example, queue manager CSQ1 is run under user QMCSQ1, and has active log data sets CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, and so on.

## Procedure

1. Create the generic profile if it does not exist, by issuing the following command:

```
ADDSD 'CSQ1.LOGS.*' UACC(NONE)
```

2. Permit the queue manager user alter access on the profile, by issuing the following command:

```
PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
```

   Also, permit the appropriate access needed for any administrative user.

3. Add the DFP segment with the encryption key label by issuing the following command:

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

   **Note:** You must use the same encryption key that you used in configuring the data set encryption key for the queue manager.

4. Refresh the generic dataset profiles by issuing the following command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Rename each log data set to a backup, then recreate and restore the data, using IDCAMS. The following JCL fragment converts CSQ1.LOGS.LOGCOPY1.DS001:

   a) Rename the data set to a back-up

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME DATASET TO BACKUP                                    */
  /*-------------------------------------------------------------*/
  ALTER 'CSQ1.LOGS.LOGCOPY1.DS001'                  -
        NEWNAME('CSQ1.BAK.LOGS.LOGCOPY1.DS001')
```

   b) Redefine the data set.

   The new data set will be encrypted due to the RACF profile.

   **Note:** Replace ++EXTDCLASS++ with the name of the extended format data class you want to use for the data set.

```
//REDEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* REDEFINE THE DATASET                                        */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                    -
        (NAME(CSQ1.LOGS.LOGCOPY1.DS001)             -
        LINEAR                                      -
        SHAREOPTIONS(2 3)                           -
        MODEL(CSQ1.BAK.LOGS.LOGCOPY1.DS001)         -
        DATACLAS(++EXTDCLASS++))
```

   c) Copy the data from the backup into the recreated data set.

   This step encrypts the data:

```
//RESTORE  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RESTORE DATA INTO ENCRYPTED LOG                             */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)     -
        OUTDATASET(CSQ1.LOGS.LOGCOPY1.DS001)
```

### What to do next

Repeat Step “5” on page 440 for all active log data sets.

Only a single encryption key is required, and all data sets can be associated with the same key label.

Restart queue manager CSQ1. Use the output from the DISPLAY LOG command to verify that the log data sets have been encrypted.

## z/OS Considerations for z/OS data set encryption in a queue sharing group

Each queue manager in a queue sharing group (QSG) must be able to read the logs, BSDS, and shared message data sets (SMDS), of every other queue manager in the QSG.

This means that each system on which a member of the QSG can run, must meet the requirements for z/OS data set encryption, and all the key labels and encryption keys used to protect the data sets for each queue manager in the QSG must be available on each system.

A queue manager prior to IBM MQ for z/OS 9.1.4 cannot access an encrypted active log data set.

A queue manager prior to IBM MQ for z/OS 9.1.5 cannot access an encrypted SMDS.

Before making use of z/OS data set encryption, you should migrate all queue managers in a QSG to at least IBM MQ for z/OS 9.1.5.

If a queue manager in a QSG is started with any encrypted active log data set, and any other queue manager in the QSG has been started, but was not last started with a version of IBM MQ for z/OS that supports encrypted active logs, the queue manager with the encrypted active log terminates abnormally with abend code 5C6-00F50033.

You can convert a QSG to use encrypted active logs and SMDS without a full outage, by:

1. Migrating each queue manager to at least IBM MQ for z/OS 9.1.5 in turn.
2. Converting active logs to encrypted data sets for each queue manager in turn. This requires the queue manager to be shut down and then restarted.

   At the same time, it is likely that page sets and archive logs would be enabled for encrypted data sets too, but this does not affect QSG migration.

   The procedure for converting each data set is described in “Example of how to encrypt queue manager active logs” on page 438
3. Converting SMDS to encrypted data sets for each individual CF structure in turn by:
   a. Issuing the command RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name) to suspend queue manager access to the SMDS.

      Note that during this time, the data on the shared queues associated with the SMDS is temporarily unavailable.
   b. Converting each data set that makes up the SMDS to encrypted data sets, using the procedure described in “Example of how to encrypt queue manager active logs” on page 438.
   c. Issuing the command RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name) to resume queue manager access to the SMDS.

**Attention:** You should shut the queue manager down cleanly prior to converting the logs, and coupling facility structure recovery might not be possible during the conversion, as the active log data sets will be temporarily unavailable.

## z/OS Backwards migration considerations when using z/OS data set encryption

You need to consider the following when backwards migrating a queue manager, which has one or more encrypted data sets.

z/OS data set encryption is supported on the following IBM MQ for z/OS data sets:

- Active log data sets
- Archive log data sets
- Page sets
- BSDS
- SMDS
- CSQINP* data sets

There are no backwards migration considerations for BSDS, archive log, or CSINP* data sets.

However, there are considerations for

- SMDS
- Page set and
- Active log

data sets, as using these with z/OS data set encryption is not supported in IBM MQ for z/OS 9.1.0, and earlier, long term support releases.

Prior to backwards migration, all encryption policies for SMDS, page set, and active log data sets need to be removed and the data decrypted. This process is described in "Removing data set encryption from a data set" on page 442.

**Attention:** If the queue manager to be backwards migrated is part of a queue sharing group (QSG), read the "Queue sharing group considerations" on page 443 section first.

## Removing data set encryption from a data set

This example describes how to remove data set encryption from the log data set CSQ1.LOGS.LOGCOPY1.DS001. You can use an equivalent process for SMDS and page sets.

The example assumes that:

- RACF is the SAF
- The queue manager that uses the data set has been stopped
- The encryption key label has been associated with the generic RACF profile CSQ1.LOGS.*

Carry out the following procedure:

1. Copy the data from the data set to a back-up data set.
   a. Define a backup data set which is not associated with an encryption key label.

      **Note:** Replace ++EXTDCLASS++ with the name of the extended format data class you want to use for the data set.

```
//DEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DEFINE UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                -
         (NAME(CSQ1.BAK.LOGS.LOGCOPY1.DS001)    -
         LINEAR                                 -
         SHAREOPTIONS(2 3)                      -
         MODEL(CSQ1.LOGS.LOGCOPY1.DS001)        -
         DATACLAS(++EXTDCLASS++))
/*
```

   b. Copy the data from the original data set to the backup. This step decrypts the data.

```
//COPY     EXEC PGM=IDCAMS,REGION=0M
```

```
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* COPY DATA INTO UNENCRYPTED DATA SET                         */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.LOGS.LOGCOPY1.DS001)           -
        OUTDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)
/*
```

c. Delete the original data set

```
//DELETE   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DELETE ORIGINAL                                             */
  /*-------------------------------------------------------------*/
  DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
/*
```

d. Rename the backup to the original data set name. The data remains unencrypted

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  ALTER CSQ1.BAK.LOGS.LOGCOPY1.DS001'                 -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001)
  ALTER 'CSQ1.BAK.LOGS.LOGCOPY1.DS001.*'               -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001.*')
/*
```

2. Optionally, repeat this process for other data sets that have an encryption key label associated with them through the CSQ1.LOGS.* generic profile.
3. Optionally, if all data sets associated with the CSQ1.LOGS.* generic profile have been decrypted, remove the DATAKEY associated with the generic profile by issuing the following command

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

4. Refresh the generic dataset profiles by issuing the following command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Restart the queue manager.
6. If the encryption key is no longer needed, delete it, and delete its associated RACF profile from the CSFKEYS class.

## Queue sharing group considerations

If a queue manager that is part of a queue sharing group is going to be backwards migrated to a version of IBM MQ for z/OS that does not support data set encryption then all of the active log data sets and SMDS of all queue managers in the QSG need to have their data set encryption policies removed, and their data decrypted.

This applies regardless of whether a single member of QSG is backwards migrated, or all members of the QSG.

You can achieve removal of encryption policies, and decryption of data, without a full QSG outage by:

1. Shutting down each queue manager in the QSG in turn, removing the encryption policies and decrypting the data from its active logs, using the process described in "Removing data set encryption from a data set" on page 442.

   If the queue manager is to be backwards migrated, its page set should also be decrypted at this time. Then restart the queue manager.

2. Removing the encryption policies and decrypting the data for the SMDS of each individual CF structure in turn by:

   a. Issuing the command

   ```
   RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
   ```

   to suspend queue manager access to the SMDS. During this time the data on the shared queues associated with the SMDS will be temporarily unavailable.

   b. Following the process in "Removing data set encryption from a data set" on page 442 for each data set which makes up the SMDS.

   c. Issuing the command

   ```
   RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
   ```

   to resume queue manager access to the SMDS.

### Using z/OS data set encryption with a queue manager that does not support it

If you accidentally backwards migrate a queue manager to a version of IBM MQ for z/OS that does not support data set encryption, and forget to remove the encryption policies and decrypt the data you get an error when the queue manager tries to access the data set.

The error depends on the data set type and is shown in the following table.

**Note:** If one or more of these errors occur, you need to follow the processes described in "Removing data set encryption from a data set" on page 442 for the affected data set. These can be performed without changing the version of IBM MQ for z/OS.

| Data set | Error if queue manager does not support z/OS data set encryption |
|---|---|
| Page set 0 | Abend 5C6-00C91400 at queue manager start |
| Page sets 1-99 | MQRC 2193 "Pageset error" when accessing page set, for example, on MQPUT |
| Active log | Abend 5C6-00E80084 at queue manager start |
| SMDS | Message IEC161I-122 logged "The data set has a KEYLABEL, but the user did not specify that the application could handle encryption".<br>SMDS marked AVAIL(ERROR). |

# 메시지의 데이터 무결성

데이터의 무결성을 유지보수하기 위해 다양한 유형의 사용자 엑시트 프로그램을 사용하여 메시지의 메시지 요약 또는 디지털 서명을 제공할 수 있습니다.

## 데이터 무결성

**메시지에서 데이터 무결성 구현**

TLS를 사용할 때 선택한 CipherSpec는 엔터프라이즈에서 데이터 무결성의 레벨을 판별합니다. IBM MQ 고급 메시지 서비스(AMS)를 사용하는 경우에는 고유한 메시지의 무결성을 지정할 수 있습니다.

**메시지 엑시트에서 데이터 무결성 구현**

메시지는 채널의 송신 측에 있는 메시지 엑시트에 의해 디지털로 서명될 수 있습니다. 그런 후, 메시지가 의도적으로 수정되었는지를 감지하기 위해 채널의 수신 측에 있는 메시지 엑시트에 의해 디지털 서명이 검사될 수 있습니다.

디지털 서명 대신에 메시지 요약을 사용하여 일부 보호가 제공될 수 있습니다. 약식 또는 무차별 도용에 대해서는 메시지 요약이 효율적일 수 있으나, 경험이 많은 사람이 메시지를 변경하거나 바꾸고, 그것에 대한 완전

히 새로운 축약을 생성하는 것을 막지는 못합니다. 메시지 요약을 생성하는 데 사용되는 알고리즘이 잘 알려진 것이라면 더욱 그렇습니다.

**송신 및 수신 엑시트에서 데이터 무결성 구현**

메시지 채널에서는 메시지 엑시트가 전체 메시지에 액세스할 수 있기 때문에 이 서비스를 제공하는 데는 메시지 엑시트가 더 적당합니다. MQI 채널에서는 MQI 호출의 매개변수들에 보호되어야 하는 애플리케이션 데이터가 있을 수 있고 송신과 수신 엑시트만이 이 보호를 제공할 수 있습니다.

**API 엑시트 또는 API 교차 엑시트에서 데이터 무결성 구현**

메시지가 송신 애플리케이션에 의해 놓일 때 메시지는 API 또는 API 교차 엑시트에 의해 디지털적으로 서명될 수 있습니다. 그런 다음 디지털 서명은 메시지가 의도적으로 수정되었는지 여부를 감지하기 위해 수신 애플리케이션에 의해 검색될 때 두 번째 엑시트에 의해 검사될 수 있습니다.

디지털 서명 대신에 메시지 요약을 사용하여 일부 보호가 제공될 수 있습니다. 약식 또는 무차별 도용에 대해서는 메시지 요약이 효율적일 수 있으나, 경험이 많은 사람이 메시지를 변경하거나 바꾸고, 그것에 대한 완전히 새로운 축약을 생성하는 것을 막지는 못합니다. 메시지 요약을 생성하는 데 사용되는 알고리즘이 잘 알려진 것이라면 더욱 그렇습니다.

### 추가 정보

데이터 무결성을 확인하는 방법에 대한 자세한 정보는 392 페이지의 『CipherSpec 사용 가능』에 대한 절을 참조하십시오.

**관련 태스크**

TLS를 사용하여 두 개의 큐 관리자 연결
클라이언트를 큐 관리자에 안전하게 연결

## 감사

이벤트 메시지를 사용하여 보안 침입 또는 침입 시도를 확인할 수 있습니다. IBM MQ Explorer를 사용하여 시스템의 보안을 확인할 수도 있습니다.

큐 관리자에 연결 또는 메시지를 큐에 놓기 등과 같은 권한 없는 조치를 수행하려는 시도를 감지하려면 큐 관리자가 생성한 이벤트 메시지, 특히 권한 이벤트 메시지를 검사하십시오. 큐 관리자 이벤트 메시지에 대한 자세한 정보는 큐 관리자 이벤트를, 이벤트 모니터링 일반에 대한 자세한 정보는 이벤트 모니터링을 참조하십시오.

## 클러스터 보안 유지

큐 관리자에게 클러스터를 조인하거나 클러스터 큐에 메시지를 넣도록 권한 부여하거나 금지하십시오. 강제로 큐 관리자가 클러스터를 나가도록 하십시오. 클러스터에 대해 TLS를 구성할 때 몇 가지 추가 고려사항을 고려하십시오.

### 권한 없는 큐 관리자가 메시지를 전송하는 것을 중지

채널 보안 엑시트를 사용하여 권한 없는 큐 관리자가 큐 관리자에 메시지를 전송하는 것을 방지하십시오.

#### 시작하기 전에

클러스터링은 보안 엑시트가 작동하는 방법에는 영향을 미치지 않습니다. 분산 큐잉 환경에서와 같은 방법으로 큐 관리자에 대한 액세스를 제한할 수 있습니다.

#### 이 태스크 정보

선택된 큐 관리자가 큐 관리자에 메시지를 전송하는 것을 방지하십시오.

#### 프로시저

1. CLUSRCVR 채널 정의에 채널 보안 엑시트 프로그램을 정의하십시오.

2. 클러스터 수신자 채널에서 메시지를 전송하려고 시도하는 큐 관리자를 인증하고 권한이 없는 경우 액세스를 거부하는 프로그램을 작성하십시오.

### 다음에 수행할 작업

채널 보안 엑시트 프로그램은 MCA 시작 및 종료 시에 호출됩니다.

## 권한 없는 큐 관리자가 메시지를 사용자의 큐에 넣는 것을 중지

권한 없는 큐 관리자가 메시지를 큐에 넣는 것을 중지하려면 클러스터 수신자 채널에서 채널 넣기 권한 속성을 사용하십시오. 멀티플랫폼의 z/OS의 경우 RACF 또는 OAM을 사용하여 메시지에서 사용자 ID를 검사하여 리모트 큐 관리자에게 권한을 부여하십시오.

### 이 태스크 정보

큐에 대한 액세스를 제어하려면 IBM MQ에서 플랫폼의 보안 기능 및 액세스 제어 메커니즘을 사용하십시오.

### 프로시저

1. 특정 큐 관리자가 메시지를 큐에 넣는 것을 방지하려면 플랫폼에서 사용 가능한 보안 기능을 사용하십시오.

   예를 들면, 다음과 같습니다.

   - z/OS IBM MQ for z/OS 의 RACF 또는 기타 외부 보안 관리자
   - Multi 다른 멀티플랫폼의 오브젝트 권한 관리자 (OAM).
2. CLUSRCVR 채널 정의에서 넣기 권한, PUTAUT 속성을 사용하십시오.

   PUTAUT 속성을 사용하면 메시지를 큐에 넣기 위해 권한을 설정하기 위해 어떤 사용자 ID가 사용되는지를 지정할 수 있습니다.

   PUTAUT 속성의 옵션은 다음과 같습니다.

   **DEF**
   기본 사용자 ID를 사용하십시오.

   z/OS z/OS에서 검사는 네트워크에서 수신한 사용자 ID 및 MCAUSER에서 파생된 사용자 ID 둘 모두를 사용하는 것과 관련될 수 있습니다.

   **CTX**
   메시지와 연관된 컨텍스트 정보에서 사용자 ID를 사용하십시오.

   z/OS z/OS에서 검사는 네트워크에서 수신한 사용자 ID 또는 MCAUSER에서 파생된 사용자 ID 또는 둘 모두의 사용과 관련될 수도 있습니다. 링크가 신뢰되고 인증되는 경우 이 옵션을 사용하십시오.

   z/OS **ONLYMCA ( z/OS 전용)**
   DEF에 대해서는 네트워크에서 수신한 모든 사용자 ID가 사용되지는 않습니다. 링크가 신뢰되지 않은 경우에 이 옵션을 사용하십시오. MCAUSER에 정의된 특정 조치 세트만을 허용하고 싶습니다.

   z/OS **ALTMCA(z/OS에만 해당)**
   CTX에 대해서는 네트워크에서 수신한 모든 사용자 ID가 사용되지는 않습니다.

## 리모트 클러스터 큐에 메시지를 넣는 권한 부여

On z/OS RACF를 사용하여 클러스터 큐에 넣을 수 있는 권한을 설정하십시오. 멀티플랫폼에서 큐 관리자에 연결하고 해당 큐 관리자의 큐에 넣을 수 있는 액세스 권한을 부여하십시오.

### 이 태스크 정보

기본 동작은 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대한 액세스 제어를 수행하는 것입니다. 이 작동은 여러 전송 큐를 사용 중인 경우에도 적용된다는 점을 유의하십시오.

이 주제에 설명된 특정 동작은 보안 스탠자 주제에 설명된 대로 qm.ini 파일에서 **ClusterQueueAccessControl** 속성을 *RQMName*으로 구성하고 큐 관리자를 재시작한 경우에만 적용됩니다.

### 프로시저

- z/OS
  z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- ALW
  AIX, Linux, and Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

- IBM i
  IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

  사용자는 다른 클러스터 큐가 아닌 지정된 클러스터 큐에만 메시지를 넣을 수 있습니다.

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  큐 관리자 이름. z/OS에서 이 값은 큐 공유 그룹의 이름일 수도 있습니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

  **QueueName**
  권한 부여를 변경할 큐 또는 일반 프로파일의 이름입니다.

### 다음에 수행할 작업

클러스터 큐에 메시지를 넣을 때 응답 대상 큐를 지정하면 처리 애플리케이션에 응답을 송신할 수 있는 권한이 있어야 합니다. 370 페이지의 『메시지를 리모트 클러스터 큐에 넣기 위한 권한 부여』의 지시사항에 따라 이 권한을 설정하십시오.

#### 관련 개념

qm.ini의 보안 스탠자

## 큐 관리자가 클러스터에 조인하는 것을 방지

악한 큐 관리자가 클러스터에 조인하면 사용자가 수신하고 싶지 않은 메시지를 큐 관리자가 수신하는 것을 방지하기 어렵게 됩니다.

### 프로시저

특정 권한 있는 큐 관리자만이 클러스터에 조인하게 하려면 다음 세 개의 방법을 선택할 수 있습니다.

- 채널 인증 레코드를 사용하면 원격 시스템이 제공하는 원격 IP 주소, 리모트 큐 관리자 이름 또는 TLS 식별 이름을 기반으로 클러스터 채널 연결을 차단할 수 있습니다.
- 권한 없는 큐 관리자가 SYSTEM.CLUSTER.COMMAND.QUEUE에 쓰는 것을 막는 엑시트 프로그램을 작성하십시오. 큐 관리자가 여기에 쓸 수 없도록 SYSTEM.CLUSTER.COMMAND.QUEUE에 대한 액세스를 제한하지 마십시오. 그렇지 않으면 큐 관리자가 클러스터에 조인하는 것을 막게 됩니다.

- CLUSRCVR 채널 정의에서 보안 엑시트 프로그램.

## 클러스터 채널에서 보안 엑시트

클러스터 채널에서 보안 엑시트를 사용할 때 추가 고려사항

### 이 태스크 정보

클러스터-송신자 채널이 처음 시작될 때 이는 시스템 관리자가 수동으로 정의한 속성을 사용합니다. 채널이 중지된 후 다시 시작되면 이는 해당하는 클러스터-수신자 채널 정의로부터 속성을 선택합니다. 원본 클러스터-송신자 채널 정의는 SecurityExit 속성을 포함하여 새 속성으로 덮어쓰기됩니다.

### 프로시저

1. 채널의 클러스터-송신자 끝 및 클러스터-수신자 끝 둘 모두에서 보안 엑시트를 정의해야 합니다.

   보안 엑시트 이름이 클러스터-수신자 정의로부터 전송되더라도 초기 연결은 보안-엑시트 데이터 교환을 사용하여 작성되어야 합니다.
2. 보안 엑시트에서 MQCXP 구조에서 PartnerName을 유효성 검증하십시오.

   엑시트는 파트너 큐 관리자가 권한 부여된 경우에만 채널이 시작하도록 허용해야 합니다.
3. 클러스터-수신자 정의에서 보안 엑시트가 시작된 수신자가 되도록 설계하십시오.
4. 이를 시작된 송신자로서 설계하는 경우에는 보안 검사가 수행되지 않으므로 보안 엑시트가 없는 권한 없는 큐 관리자는 클러스터에 조인할 수 있습니다.

   채널이 중지되고 다시 시작될 때까지는 SCYEXIT 이름이 클러스터-수신자 정의로부터 전송되고 전체 보안 검사가 수행될 수 없습니다.
5. 현재 사용 중인 클러스터-송신자 채널 정의를 보려면 다음 명령을 사용하십시오.

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   명령은 클러스터-수신자 정의로부터 전송된 속성을 표시합니다.
6. 원본 정의를 보려면 다음 명령을 사용하십시오.

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```
7. 큐 관리자가 다른 플랫폼에 있는 경우 클러스터-송신자 큐 관리자에서 채널 자동 정의 엑시트, CHADEXIT를 정의해야 할 수도 있습니다.

   채널 자동 정의 엑시트를 사용하여 SecurityExit 속성을 대상 플랫폼에 적합한 형식으로 설정하십시오.
8. 보안 엑시트를 배치하고 구성하십시오.

   z/OS **z/OS**

   보안 엑시트 로드 모듈은 채널 시작기 주소 공간 프로시저의 CSQXLIB DD문에 지정된 데이터 세트에 있어야 합니다.

   ALW **AIX, Linux, and Windows 시스템**

   - 보안 엑시트 동적 링크 라이브러리는 채널 정의의 SCYEXIT 속성에 지정된 경로에 있어야 합니다.
   - 채널 자동 정의 엑시트 동적 링크 라이브러리는 큐 관리자 정의의 CHADEXIT 속성에 지정된 경로에 있어야 합니다.

# 필요하지 않은 큐 관리자는 클러스터에서 강제로 제거

전체 저장소 큐 관리자에서 RESET CLUSTER 명령을 실행하여 필요하지 않은 큐 관리자는 클러스터에서 강제로 제거하십시오.

### 이 태스크 정보

필요하지 않은 큐 관리자는 클러스터에서 강제로 제거할 수 있습니다. 예를 들어, 큐 관리자가 삭제되었지만 해당 클러스터 수신자 채널이 여전히 클러스터에 정의되어 있습니다. 이를 정돈하고 싶을 수도 있습니다.

전체 저장소 큐 관리자만이 큐 관리자를 클러스터에서 배출하기 위한 권한이 부여됩니다.

**참고:** RESET CLUSTER 명령을 사용하면 큐 관리자를 강제로 클러스터에서 제거하지만 RESET CLUSTER를 사용하는 것만으로는 큐 관리자가 나중에 클러스터에 다시 조인하는 것을 방지할 수 없습니다. 큐 관리자가 클러스터에 다시 조인하지 않게 하려면 447 페이지의 『큐 관리자가 클러스터에 조인하는 것을 방지』에 설명된 단계를 따르십시오.

큐 관리자 OSLO을 클러스터 NORWAY에서 배출하려면 이 프로시저를 따르십시오.

### 프로시저

1. 전체 저장소 큐 관리자에서 명령을 실행하십시오.

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. 명령에서 QMNAME 대신에 QMID를 사용하십시오.

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

**참고:** QMID 는 문자열이므로 qmid 의 값은 작은따옴표로 묶어야 합니다 (예: QMID('FR01_2019-07-15_14.42.42')).

### 결과

강제로 제거된 큐 관리자는 변경되지 않습니다. 해당 로컬 클러스터 정의는 이것이 클러스터에 있는 것으로 표시합니다. 다른 모든 큐 관리자에서의 정의는 이를 클러스터에 표시하지 않습니다.

## 큐 관리자가 메시지 수신하는 것을 방지

엑시트 프로그램을 사용하여 클러스터 큐 관리자가 수신할 권한이 없는 메시지를 수신하는 것을 막을 수 있습니다.

### 이 태스크 정보

클러스터의 멤버인 큐 관리자가 큐를 정의하지 못하도록 막는 것은 어렵습니다. 악한 큐 관리자가 클러스터에 조인하고 클러스터에 큐 중 하나의 자체 인스턴스를 정의할 위험이 있습니다. 이는 이제 수신할 권한이 없는 메시지를 수신할 수 있습니다. 큐 관리자가 메시지를 수신하는 것을 방지하려면 프로시저에 제공된 다음 옵션 중 하나를 사용하십시오.

### 프로시저

- 각 클러스터 송신자 채널에서 채널 엑시트 프로그램. 엑시트 프로그램은 연결 이름을 사용하여 메시지를 전송할 목적지 큐 관리자의 적합성을 판별합니다.
- 메시지가 전송될 목적지 큐 및 큐 관리자의 적합성을 판별하기 위해 목적지 레코드를 사용하는 클러스터 워크로드 엑시트 프로그램.

## SSL/TLS 및 클러스터

클러스터에 대한 TLS를 구성할 때 CLUSRCVR 채널 정의가 자동 정의된 CLUSSDR 채널로 다른 큐 관리자에 전파됨을 유의하십시오. CLUSRCVR 채널이 TLS를 사용하는 경우 채널을 사용하여 통신하는 모든 큐 관리자에서 TLS를 구성해야 합니다.

TLS에 대한 자세한 정보는 22 페이지의 『IBM MQ의 TLS 보안 프로토콜』의 내용을 참조하십시오. 이 충고는 일반적으로 클러스터 채널에 적용 가능하지만 다음에 일부 특수 고려사항을 제공하고 싶을 수도 있습니다.

IBM MQ 클러스터에서 특별 CLUSRCVR 채널 정의는 종종 자동 정의된 CLUSSDR로 변환되는 다른 많은 큐 관리자로 전파됩니다. 이후에 자동 정의된 CLUSSDR이 채널을 CLUSRCVR에 시작하는 데 사용됩니다. CLUSRCVR이 TLS 연결성을 위해 구성된 경우에는 다음 고려사항이 적용됩니다.

- 이 CLUSRCVR과 통신하려는 모든 큐 관리자에게는 TLS 지원에 대한 액세스가 있어야 합니다. 이 TLS 프로비저닝은 채널의 CipherSpec을 지원해야 합니다.
- 자동 정의된 클러스터 송신자 채널이 전파된 다른 큐 관리자에게는 각각 서로 다른 식별 이름이 연관됩니다. 식별 이름 피어 검사가 CLUSRCVR에서 사용될 예정이면 수신할 수 있는 모든 식별 이름이 성공적으로 일치할 수 있도록 설정되어야 합니다.

  예를 들어, 특정 CLUSRCVR에 연결할 클러스터-송신자 채널을 호스팅하는 모든 큐 관리자에 연관된 인증서가 가정해 보겠습니다. 또한 이러한 모든 인증서에서 식별 이름은 국가를 UK, 조직을 IBM, 조직 단위를 IBM MQ 부서로 정의하고, 모두 DEVT.QMnnn 양식으로 된 공통 이름(여기서, nnn은 숫자)을 가지고 있다고 가정합니다.

  이 경우 CLUSRCVR에서 C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM*의 SSLPEER 값을 통해 필요한 모든 클러스터 송신자 채널은 성공적으로 연결하되 원하지 않는 클러스터 송신자 채널은 연결되지 않도록 할 수 있습니다.
- 사용자 정의 CipherSpec 문자열이 사용된 경우에는 사용자 정의 문자열 형식이 모든 플랫폼에서 허용되지는 않음을 유의하십시오. 예를 들어, CipherSpec string RC4_SHA_US 의 값이 IBM i 에서는 05 이지만 AIX, Linux, and Windows 시스템에서는 올바른 스펙이 아닙니다. 따라서 사용자 정의 SSLCIPH 매개변수가 CLUSRCVR에서 사용되는 경우에는 결과로 나오는 모든 자동 정의된 클러스터 송신자 채널은 기본 TLS 지원이 이 CipherSpec을 구현하고 사용자 정의 값으로 지정될 수 있는 플랫폼에 상주해야 합니다. 클러스터 전체에 걸쳐 이해될 SSLCIPH 매개변수의 값을 선택할 수 없는 경우 이를 이해될 사용 중인 플랫폼으로 변경하기 위해 채널 자동 정의 엑시트가 필요합니다. 가능한 경우 텍스트 CipherSpec 문자열(예: TLS_RSA_WITH_AES_128_CBC_SHA)을 사용하십시오.

SSLCRLNL 매개변수는 개별 큐 관리자에 적용되고 클러스터 내에 다른 큐 관리자에게 전파되지 않습니다.

## 클러스터된 큐 관리자 및 채널을 SSL/TLS로 업그레이드

클러스터 채널을 한 번에 하나씩 업그레이드하여 CLUSSDR 채널 전에 모든 CLUSRCVR 채널을 변경하십시오.

### 시작하기 전에

클러스터를 위한 CipherSpec 선택에 영향을 미칠 수 있으므로 다음 고려사항을 고려하십시오.

- 일부 CipherSpec은 모든 플랫폼에서 사용 가능하지는 않습니다. 클러스터에 있는 모든 큐 관리자에 의해 지원되는 CipherSpec을 주의해서 선택하십시오.
- 일부 CipherSpec은 현재 IBM MQ 릴리스에서는 신규일 수도 있고 이전 릴리스에서는 지원되지 않을 수도 있습니다. 다른 MQ 릴리스에서 실행 중인 큐 관리자를 포함하는 클러스터는 각 릴리스가 지원하는 CipherSpec만을 사용할 수 있습니다.

  클러스터 내에서 새 CipherSpec을 사용하려면 먼저 모든 클러스터 큐 관리자를 현재 릴리스로 마이그레이션해야 합니다.
- 일부 CipherSpec에는 사용할 디지털 인증서의 특정 유형이 필요한데, 특히, 타원 곡선 암호법을 사용하는 유형입니다.

  **주의:** 클러스터의 일부로 함께 조인할 큐 관리자에서 Elliptic Curve 서명 인증서와 RSA 서명 인증서를 혼합해서 사용하는 것은 불가능합니다.

  클러스터의 큐 관리자는 모두 RSA 서명 인증서를 사용하거나 모두 EC 서명 인증서를 사용해야 합니다. 이 둘을 혼합해서 사용할 수는 없습니다.

  자세한 정보는 43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』의 내용을 참조하십시오.

클러스터에 있는 모든 큐 관리자가 아직 IBM MQ V8 이상 레벨에 있지 않으면 업그레이드하십시오. TLS가 이들 각각에서 작동할 수 있도록 인증서 및 키를 분배하십시오.

알리어스 CipherSpecs (ANY_TLS13, ANY_TLS13_OR_HIGHER, ANY_TLS12, ANY_TLS12_OR_HIGHER등)를 업그레이드하거나 사용하려면 먼저 큐 관리자를 업그레이드해야 합니다.

- Multi 클러스터의 모든 IBM MQ for Multiplatforms 큐 관리자를 IBM MQ 9.1.4 이상으로 업그레이드하십시오.
- z/OS 클러스터의 모든 IBM MQ for z/OS 큐 관리자를 IBM MQ for z/OS 9.2.0 이상으로 업그레이드하십시오.

사용자는

### 이 태스크 정보

CLUSSDR 채널 전에 CLUSRCVR 채널을 변경하십시오.

### 프로시저

1. CLUSRCVR 채널을 원하는 순서로 TLS로 전환하여 한 번에 하나의 CLUSRCVR를 변경하고, 다음을 변경하기 전에 변경사항이 클러스터를 통해 플로우할 수 있게 허용하십시오.

   **중요사항:** 현재 채널이 변경사항이 클러스터 전체에 분배될 때까지 역방향 경로를 변경하지 않았는지 확인하십시오.
2. 옵션: 모든 수동 CLUSSDR 채널을 TLS로 전환하십시오.

   이는 REFRESH CLUSTER 명령을 REPOS(YES) 옵션과 함께 사용하지 않는 한 클러스터의 조작에 영향을 미치지 않습니다.

   **참고:** 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중에 클러스터에 지장을 줄 수 있으며, 클러스터 오브젝트가 모든 관련 큐 관리자에 대한 상태 업데이트를 자동으로 송신한 후 27일간격으로 다시 수행할 수 있습니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.
3. DISPLAY CLUSQMGR 명령을 사용하여 새 보안 구성이 클러스터 전체에 걸쳐 전파되었는지 확인하십시오.
4. TLS를 사용하도록 채널을 재시작한 후 REFRESH SECURITY(SSL)를 실행하십시오.

**관련 개념**

392 페이지의 『CipherSpec 사용 가능』
**DEFINE CHANNEL** 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec을 사용 가능으로 설정합니다.

43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』
이 토픽은 IBM MQ에서 CipherSpec과 디지털 인증서 사이의 관계를 소개하여 보안 정책에 대해 적합한 CipherSpec 및 디지털 인증서를 선택하는 방법에 대한 정보를 제공합니다.

**관련 정보**

클러스터링: REFRESH CLUSTER 사용 우수 사례

## 클러스터된 큐 관리자 및 채널에서 SSL/TLS를 사용 안함으로 설정

TLS를 끄려면 SSLCIPH 매개변수를 ' '로 설정하십시오. 클러스터 채널에서 개별적으로 TLS를 사용 안함으로 설정하여 클러스터 송신자 채널 전에 클러스터 수신자 채널을 모두 변경하십시오.

### 이 태스크 정보

한 번에 하나의 클러스터 수신자 채널을 변경하고 다음을 변경하기 전에 변경사항이 클러스터를 통해 플로우하게 허용하십시오.

**중요사항:** 현재 채널이 변경사항이 클러스터 전체에 분배될 때까지 역방향 경로를 변경하지 않았는지 확인하십시오.

### 프로시저

1. SSLCIPH 매개변수의 값을 작은따옴표 ( IBM i 또는 IBM i 의 *NONE ) 안에 있는 비어 있는 문자열인 ' '로 설정하십시오.

   클러스터 수신자 채널에서 원하는 순서대로 TLS를 끌 수 있습니다.

   변경사항이 TLS를 활성인 채로 남겨두는 채널을 통해 반대 방향으로 플로우함을 유의하십시오.
2. **DISPLAY CLUSQMGR(*)** ALL 명령을 사용하여 다른 큐 관리자 모두에서 새 값이 반영되는지 확인하십시오.
3. 모든 매뉴얼 클러스터 송신자 채널에서 TLS를 끄십시오.

   이는 **REFRESH CLUSTER** 명령을 REPOS(YES) 옵션과 함께 사용하지 않는 한 클러스터의 조작에 영향을 미치지 않습니다.

   대형 클러스터의 경우 **REFRESH CLUSTER** 명령 사용은 진행 중에, 그리고 클러스터 오브젝트가 모든 관련 큐 관리자에 대한 상태 업데이트를 자동으로 전송한 후 주기적인 간격으로 클러스터에 지장을 줄 수 있습니다. 자세한 내용은 대형 클러스터의 새로 고침이 클러스터의 성능과 가용성에 영향을 줄 수 있음을 참조하십시오.
4. 클러스터 송신자 채널을 중지하고 다시 시작하십시오.

## 발행/구독 보안

발행/구독에 관련된 컴포넌트 및 상호작용은 다음에 나오는 보다 자세한 설명과 예에 대한 소개로서 설명되어 있습니다.

토픽 발행 및 구독에 관련된 여러 컴포넌트가 있습니다. 이들 간의 일부 보안 관계는 453 페이지의 그림 22에 표시되고 다음 예에 설명됩니다.

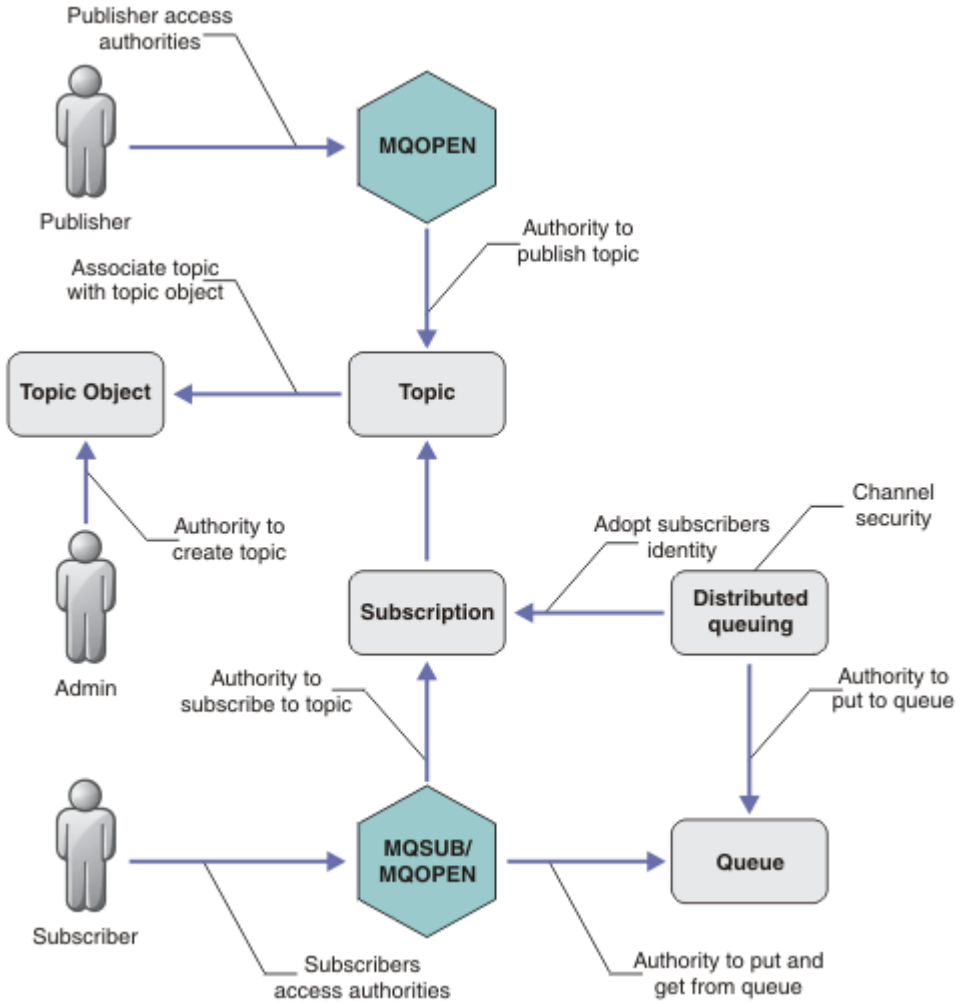



그림 22. 다양한 컴포넌트 간의 발행/구독 보안 관계

**토픽**

토픽은 토픽 문자열로 식별되며, 일반적으로 트리로 구성됩니다(토픽 트리참조). 토픽에 대한 액세스를 제어하려면 토픽을 토픽 오브젝트와 연관시켜야 합니다. 455 페이지의 『토픽 보안 모델』에서는 토픽 오브젝트를 사용하여 토픽을 보안 설정하는 방법을 설명합니다.

**관리 토픽 오브젝트**

관리 토픽 오브젝트 목록과 함께 **setmqaut** 명령을 사용하여 누가 어떤 용도로 토픽에 대한 액세스를 가지는지를 제어할 수 있습니다. 예 459 페이지의 『토픽 구독을 위해 사용자에게 액세스 부여』 및 466 페이지의 『토픽에 발행을 위해 사용자에게 액세스 부여』의 내용을 참조하십시오.

z/OS z/OS에서 토픽 오브젝트에 대한 액세스를 제어하려면 토픽 보안을 위한 프로파일을 참조하십시오.

**구독**

토픽 문자열을 제공하는 구독을 작성하여 하나 이상의 토픽을 구독하십시오. 여기에는 토픽 문자열을 발행물의 토픽 문자열과 일치시키기 위한 와일드카드가 포함될 수 있습니다. 자세한 내용은 다음을 참조하십시오.

**토픽 오브젝트를 사용하여 구독**

456 페이지의 『토픽 오브젝트 이름을 사용하여 구독』

**토픽을 사용하여 구독**

457 페이지의 『토픽 노드가 없는 토픽 문자열을 사용한 구독』

**와일드카드와 함께 토픽을 사용하여 구독**

457 페이지의 『와일드카드 문자가 포함된 토픽 문자열을 사용한 구독』

구독에는 구독자의 ID와 발행물 배치될 목적지 큐의 ID에 대한 정보가 포함됩니다. 또한 목적지 큐에 발행을 배치하는 방법에 대한 정보가 있습니다.

특정 토픽을 구독하는 권한이 있는 구독자를 정의하고 개별 구독자가 사용하는 구독을 제한할 수 있습니다. 또한 발행이 목적지 큐에 있을 때 큐 관리자가 사용하는 구독자 정보를 제어할 수 있습니다. 471 페이지의 『구독 보안』의 내용을 참조하십시오.

**큐**

목적지 큐는 보안을 설정할 중요한 큐입니다. 이 큐는 구독자에 대해 로컬이며 구독과 일치한 발행이 해당 큐에 배치됩니다. 두 가지 관점에서 목적지 큐에 대한 액세스를 고려해야 합니다.

1. 발행을 목적지 큐에 넣습니다.
2. 목적지 큐에서 발행을 가져옵니다.

큐 관리자가 구독자가 제공한 ID를 사용하여 목적지 큐에 발행을 넣습니다. 구독자, 또는 발행을 가져오는 태스크를 위임받은 프로그램이 큐에서 메시지를 가져옵니다. 457 페이지의 『목적지 큐에 대한 권한』의 내용을 참조하십시오.

토픽 오브젝트 알리어스가 없지만 알리어스 큐를 토픽 오브젝트의 알리어스로 사용할 수 있습니다. 사용하는 경우, 큐 관리자는 발행 또는 구독에 토픽을 사용할 수 있는 권한 및 큐 사용 권한을 검사합니다.

**473 페이지의 『큐 관리자 간의 발행/구독 보안』**

토픽 발행 또는 구독 권한은 로컬 ID 및 권한을 사용하여 로컬 큐 관리자에서 검사합니다. 권한은 토픽을 정의하는지의 여부나 정의된 위치에 관계없이 사용합니다. 그 결과 클러스터된 토픽을 사용할 때 클러스터의 모든 큐 관리자에서 토픽 권한을 수행해야 합니다.

**참고:** 토픽의 보안 모델은 큐의 보안 모델과는 다릅니다. 클러스터된 모든 큐에 로컬로 큐 알리어스를 정의하여 큐에 대한 동일한 결과를 얻을 수 있습니다.

큐 관리자는 클러스터에서 구독을 교환합니다. 대부분의 IBM MQ 클러스터 구성에서 채널은 채널 프로세스의 권한을 사용하여 메시지를 대상 큐에 놓기 위해 PUTAUT=DEF로 구성됩니다. PUTAUT=CTX를 사용하여 구독 중인 사용자가 클러스터에 있는 또 다른 큐 관리자에 구독을 전달하는 권한을 갖도록 채널 구성을 수정할 수 있습니다.

473 페이지의 『큐 관리자 간의 발행/구독 보안』에서는 클러스터의 다른 서버에 구독을 전파할 수 있는 사용자를 제어하도록 채널 정의를 변경하는 방법에 대해 설명합니다.

**권한 부여**

큐 및 다른 오브젝트와 같은 토픽 오브젝트에 권한을 적용할 수 있습니다. 토픽에만 적용할 수 있는 세 가지 권한 부여 조작 pub, sub 및 resume이 있습니다. 세부사항은 여러 오브젝트 유형에 권한 지정에 설명되어 있습니다.

**함수 호출**

큐된 프로그램과 같은 발행 및 구독 프로그램에서, 오브젝트를 열거나, 작성하거나, 변경하거나, 삭제할 때 권한 검사가 이루어집니다. 발행을 넣고 가져오도록 MQPUT 또는 MQGET MQI 호출이 이루어지는 경우에는 검사가 수행되지 않습니다.

토픽을 발행하려면 권한 검사를 수행하는 토픽에서 MQOPEN을 수행하십시오. 권한 검사를 수행하지 않는 MQPUT 명령을 사용하여 토픽 핸들에 메시지를 발행합니다.

토픽을 구독하려면 일반적으로 MQSUB 명령을 수행하여 구독을 작성하거나 재개하고, 발행을 수신하기 위해 목적지 큐를 엽니다. 또는 개별 MQOPEN을 수행하여 목적지 큐를 연 다음 MQSUB를 수행하여 구독을 작성하거나 재개합니다.

사용하는 호출에 관계없이 큐 관리자는 사용자가 토픽을 구독할 수 있고 목적지 큐에서 결과 발행을 가져올 수 있음을 확인합니다. 목적지 큐가 관리되지 않는 경우, 큐 관리자가 목적지 큐에 발행을 배치할 수 있는 권한 검사도 수행되며 이 큐 관리자는 일치하는 구독에서 채택한 ID를 사용합니다. 큐 관리자는 항상 관리 대상 목적지 큐에 발행을 배치할 수 있다고 가정합니다.

**역할**

사용자는 실행 중인 발행/구독 애플리케이션의 네 가지 역할에 관련되어 있습니다.

1. 발행자

2. 구독자
3. 토픽 관리자
4. IBM MQ 관리자 - 그룹 mqm의 멤버

발행, 구독 및 토픽 관리 역할에 해당하는 적절한 권한을 가진 그룹을 정의하십시오. 그 다음 특정 발행 및 구독 태스크를 수행하도록 권한을 부여하여 프린시펄을 이러한 그룹에 지정할 수 있습니다.

또한 관리 조작 권한을 발행 및 구독을 이동시키는 큐 및 채널의 관리자로 확장해야 합니다.

## 토픽 보안 모델

정의된 토픽 오브젝트에만 연관된 보안 속성이 있습니다. 토픽 오브젝트에 대한 설명은 관리 토픽 오브젝트를 참조하십시오. 보안 속성은 지정된 사용자 ID 또는 보안 그룹이 각 토픽 오브젝트에서 구독 또는 발행 조작을 수행할 수 있는지의 여부를 지정합니다.

보안 속성은 토픽 트리의 적절한 관리 노드와 연관되어 있습니다. 구독 또는 발행 조작 중에 특정 사용자 ID에 대한 권한 검사가 이루어지는 경우, 부여된 권한은 연관된 토픽 트리 노드의 보안 속성을 기반으로 합니다.

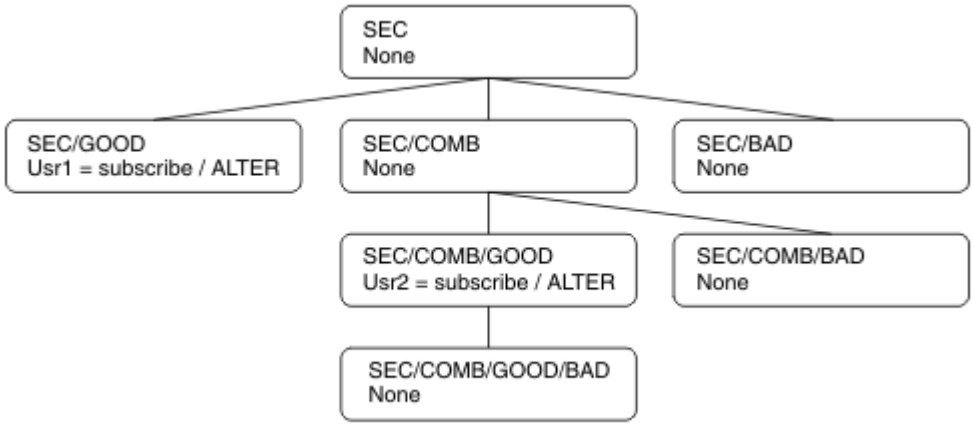
보안 속성은 특정 운영 체제 사용자 ID 또는 보안 그룹이 토픽 오브젝트에 대해 가진 권한을 나타내는 액세스 제어 목록입니다.

표시된 보안 속성 또는 권한으로 토픽 오브젝트를 정의한 다음 예를 고려하십시오.

표 *87*. 토픽 오브젝트 권한의 예

| **토픽 이름** | **토픽 문자열** | **권한-멀티플랫폼** | **z/OS 권한** |
|---|---|---|---|
| SECROOT | SEC | 없음 | 없음 |
| SECGOOD | SEC/GOOD | usr1+subscribe | ALTER<br>HLQ.SUBSCRIBE.SECGOOD |
| SECBAD | SEC/BAD | 없음 | 없음<br>HLQ.SUBSCRIBE.SECBAD |
| SECCOMB | SEC/COMB | 없음 | 없음<br>HLQ.SUBSCRIBE.SECCOMB |
| SECCOMBB | SEC/COMB/<br>GOOD/BAD | 없음 | 없음<br>HLQ.SUBSCRIBE.SECCOMBB |
| SECCOMBG | SEC/COMB/GOOD | usr2+subscribe | ALTER<br>HLQ.SUBSCRIBE.SECCOMBG |
| SECCOMBN | SEC/COMB/BAD | 없음 | 없음<br>HLQ.SUBSCRIBE.SECCOMBN |

각 노드에 있는 연관된 보안 속성을 가진 토픽 트리를 다음과 같이 표시할 수 있습니다.




나열된 예에서는 다음 권한을 제공합니다.

- 트리 /SEC의 루트 노드에서 사용자에게 해당 노드의 권한이 없습니다.
- usr1에는 오브젝트 /SEC/GOOD에 대한 구독 권한이 부여되었습니다.
- usr2에는 오브젝트 /SEC/COMB/GOOD에 대한 구독 권한이 부여되었습니다.

## 토픽 오브젝트 이름을 사용하여 구독

MQCHAR48 이름을 지정하여 토픽 오브젝트를 구독할 때 토픽 트리의 해당 노드를 찾았습니다. 노드와 연관된 보안 속성이 사용자에게 구독 권한이 있음을 나타내면, 액세스가 부여됩니다.

사용자에게 액세스가 부여되지 않은 경우, 트리의 상위 노드가 사용자에게 상위 노드 레벨에서 구독할 수 있는 권한이 있는지의 여부를 판별합니다. 권한이 있는 경우, 액세스가 부여됩니다. 권한이 없는 경우에는 해당 노드의 상위가 고려됩니다. 사용자에게 구독 권한을 부여하는 노드를 찾을 때까지 순환이 계속됩니다. 권한이 부여되지 않고 루트 노드를 고려하는 경우에는 순환이 중지됩니다. 후자의 경우, 액세스가 거부됩니다.

즉, 경로에 있는 임의의 노드가 사용자 또는 애플리케이션에 구독 권한을 부여하는 경우, 구독자는 해당 노드 또는 토픽 트리의 해당 노드 아래에서 구독할 수 있습니다.

예의 루트 노드는 SEC입니다.

액세스 제어 목록이 사용자 ID에 권한이 있거나 사용자 ID가 구성원인 운영 체제 보안 그룹에 권한이 있음을 나타내는 경우 사용자에게 구독 권한이 부여됩니다.

예:

- usr1이 토픽 문자열 SEC/GOOD을 사용하여 구독하려는 경우, 사용자 ID에 해당 토픽과 연관된 노드에 대한 액세스 권한이 있으므로 구독이 허용됩니다. 하지만, usr1이 토픽 문자열 SEC/COMB/GOOD을 사용하여 구독하려는 경우, 사용자 ID에 연관된 노드에 대한 액세스 권한이 없으므로 구독이 허용되지 않습니다.
- usr2가 토픽 문자열 SEC/COMB/GOOD을 사용하여 구독하려는 경우, 사용자 ID에 토픽과 연관된 노드에 대한 액세스 권한이 있으므로 구독이 허용됩니다. 하지만, usr2가 SEC/GOOD을 구독하려는 경우, 사용자 ID에 연관된 노드에 대한 액세스 권한이 없으므로 구독이 허용되지 않습니다.
- usr2가 토픽 문자열 SEC/COMB/GOOD/BAD를 사용하여 구독하려는 경우, 사용자 ID에 상위 노드 SEC/COMB/GOOD에 대한 액세스 권한이 있으므로 구독이 허용됩니다.
- usr1 또는 usr2가 토픽 문자열 /SEC/COMB/BAD를 사용하여 구독하려는 경우, 연관된 토픽 노드 또는 해당 토픽의 상위 노드에 대한 액세스 권한이 없으므로 두 구독 모두 허용되지 않습니다.

존재하지 않는 토픽 오브젝트의 이름을 지정하는 구독 조작으로 인해 MQRC_UNKNOWN_OBJECT_NAME 오류가 발생합니다.

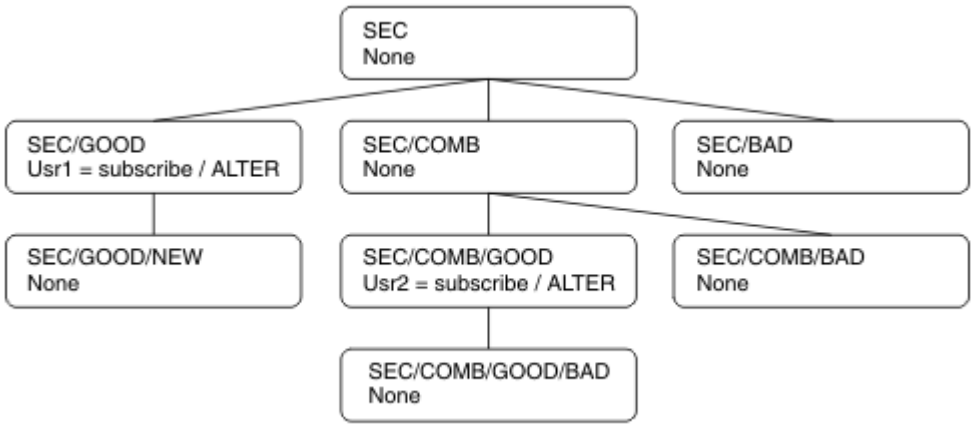
## 토픽 노드가 있는 토픽 문자열을 사용한 구독

작동은 토픽을 MQCHAR48 오브젝트 이름으로 지정할 때와 동일합니다.

## 토픽 노드가 없는 토픽 문자열을 사용한 구독

토픽 트리에 현재 없는 토픽 노드를 나타내는 토픽 문자열을 지정하여 구독하는 애플리케이션의 경우를 고려하십시오. 이전 절에 간단하게 설명된 대로 권한 검사를 수행합니다. 검사는 토픽 문자열로 표시되는 토픽의 상위 노드에서 시작합니다. 권한이 부여되는 경우, 토픽 문자열을 표시하는 새 노드가 토픽 트리에서 작성됩니다.

예를 들어, usr1은 토픽 SEC/GOOD/NEW를 구독하려 합니다. usr1에 상위 노드 SEC/GOOD에 대한 액세스 권한이 있으므로 권한이 부여됩니다. 새 토픽 노드는 다음 다이어그램에 표시된 대로 트리에 작성됩니다. 새 토픽 노드는 보안 속성이 직접 연관되지 않은 토픽 오브젝트가 아닙니다. 속성은 상위에서 상속됩니다.




## 와일드카드 문자가 포함된 토픽 문자열을 사용한 구독

와일드카드 문자가 포함된 토픽 문자열을 사용한 구독의 경우를 고려하십시오. 토픽 문자열의 완전히 규정된 부분과 일치하는 토픽 트리의 노드에서 권한 검사가 수행됩니다.

애플리케이션이 SEC/COMB/GOOD/*를 구독하는 경우, 토픽 트리의 노드 SEC/COMB/GOOD에 있는 두 개의 이전 절에 설명된 대로 권한 검사를 수행합니다.

이와 유사하게 애플리케이션이 SEC/COMB/*/GOOD을 구독해야 하는 경우, 노드 SEC/COMB에서 권한 검사가 수행됩니다.

## 목적지 큐에 대한 권한

토픽을 구독할 때 매개변수 중 하나는 발행물을 수신하기 위해 출력을 위해 열려 있는 큐의 핸들 hobj입니다.

hobj가 지정되지 않았지만 공백인 경우 다음 조건이 적용되면 관리 대상 큐가 작성됩니다.

- MQSO_MANAGED 옵션을 지정했습니다.
- 구독이 없습니다.
- 작성이 지정되어 있습니다.

hobj가 공백이고 기존 구독을 변경하거나 재개하는 경우, 이전에 제공된 목적지 큐는 관리되거나 관리되지 않는 큐일 수 있습니다.

MQSUB 요청을 작성하는 애플리케이션 또는 사용자에게 제공된 목적지 큐에 메시지를 넣을 수 있는 권한(사실상 해당 큐에 넣은 메시지를 발행하는 권한)이 있어야 합니다. 권한 검사는 큐 보안 검사에 대한 기존 규칙을 따릅니다.

보안 검사에는 필요에 따라 대체 사용자 ID 및 컨텍스트 보안 검사가 포함됩니다. ID 컨텍스트 필드 중 하나를 설정할 수 있도록 MQSO_SET_IDENTITY_CONTEXT 옵션 및 MQSO_CREATE 또는 MQSO_ALTER 옵션을 지정해야 합니다. MQSO_RESUME 요청에서 ID 컨텍스트 필드 중 하나를 설정할 수 없습니다.

목적지가 관리 대상 큐인 경우, 관리 대상 목적지에 대해 보안 검사가 수행되지 않습니다. 토픽을 구독할 수 있는 경우 관리 대상 목적지를 사용할 수 있다고 가정합니다.

### 토픽 노드가 있는 토픽 이름 또는 토픽 문자열을 사용한 발행

발행을 위한 보안 모델은 와일드카드 문자를 제외하고는 구독을 위한 보안 모델과 동일합니다. 발행에는 와일드카드 문자가 포함되지 않으므로 와일드카드 문자가 포함된 토픽 문자열을 고려하는 경우는 없습니다.

발행할 권한과 구독할 권한은 별개입니다. 사용자 또는 그룹에는 발행 또는 구독 중 하나를 수행할 수 없어도 다른 하나를 수행할 수 있는 권한이 있습니다.

MQCHAR48 이름 또는 토픽 문자열을 지정하여 토픽 오브젝트를 발행할 때 토픽 트리의 해당 노드를 찾았습니다. 토픽 노드와 연관된 보안 속성이 사용자에게 발행 권한이 있음을 나타내면, 액세스가 부여됩니다.

액세스가 부여되지 않은 경우, 트리의 상위 노드가 사용자에게 해당 레벨에서 발행할 수 있는 권한이 있는지의 여부를 판별합니다. 권한이 있는 경우, 액세스가 부여됩니다. 권한이 없는 경우, 사용자에게 발행 권한을 부여하는 노드를 찾을 때까지 순환이 계속됩니다. 권한이 부여되지 않고 루트 노드를 고려하는 경우에는 순환이 중지됩니다. 후자의 경우, 액세스가 거부됩니다.

즉, 경로에 있는 임의의 노드가 사용자 또는 애플리케이션에 발행할 권한을 부여하는 경우, 발행자는 해당 노드 또는 토픽 트리의 해당 노드 아래에서 발행할 수 있습니다.

### 토픽 노드가 없는 토픽 이름 또는 토픽 문자열을 사용한 발행

구독 조작에서와 같이, 애플리케이션이 토픽 트리에 현재 없는 토픽 노드를 나타내는 토픽 문자열을 지정하여 발행할 때 토픽 문자열로 표시되는 노드의 상위부터 시작하여 보안 검사를 수행합니다. 권한이 부여되는 경우, 토픽 문자열을 표시하는 새 노드가 토픽 트리에서 작성됩니다.

### 토픽 오브젝트로 해석되는 알리어스 큐를 사용한 발행

토픽 오브젝트로 해석되는 알리어스 큐를 사용하여 발행하는 경우, 알리어스 큐 및 이 큐가 해석되는 기본 토픽 모두에서 보안 검사가 수행됩니다.

알리어스 큐에서 보안 검사를 수행하면 사용자에게 해당 알리어스 큐에 메시지를 넣을 수 있는 권한이 있는지 확인 가능하고, 토픽에 대한 보안 검사를 수행하면 사용자가 해당 토픽을 발행할 수 있는지 확인 가능합니다. 알리어스 큐가 다른 큐로 해석되는 경우, 기본 큐에서는 검사를 수행하지 않습니다. 토픽에 대한 권한 검사와 큐에 대한 권한 검사는 서로 다르게 수행됩니다.

### 구독 종료

이 핸들에서 구독을 작성하지 않으면 MQCO_REMOVE_SUB 옵션을 사용하여 구독을 종료하는 경우 추가 보안 검사가 수행됩니다.

조치로 인해 구독이 제거되므로 이를 수행할 수 있는 올바른 권한이 있는지 확인하기 위해 보안 검사를 수행합니다. 토픽 노드와 연관된 보안 속성이 사용자에게 권한이 있음을 나타내는 경우, 액세스가 부여됩니다. 권한이 없는 경우, 트리의 상위 노드가 사용자에게 구독을 종료할 수 있는 권한이 있는지의 여부를 판별합니다. 권한이 부여되거나 루트 노드에 도달할 때까지 순환이 계속됩니다.

### 구독 정의, 대체 및 삭제

MQSUB API 요청을 사용하지 않고 구독이 관리 면에서 작성되는 경우 구독 보안 검사가 수행되지 않습니다. 관리자는 이미 명령을 통해 해당 권한을 받았습니다.

구독과 연관된 목적지 큐에 발행을 넣을 수 있는지 확인하기 위해 보안 검사를 수행합니다. 검사는 MQSUB 요청의 경우와 동일한 방식으로 수행됩니다.

이러한 보안 검사에 사용된 사용자 ID는 발행될 명령에 따라 달라집니다. **SUBUSER** 매개변수를 지정한 경우, 459 페이지의 표 88에 표시된 대로 검사 수행 방식에 영향을 줍니다.

| 표 *88.* 명령에 대한 보안 검사에 사용되는 사용자 *ID* | | | |
|---|---|---|---|
| **명령** | **SUBUSER 지정 및 공백** | **SUBUSER 지정 및 완료** | **SUBUSER가 지정되지 않음** |
| | 관리자 ID 사용 | | LIKE 구독으로부터 사용자 ID 사용 |
| | 관리자 ID 사용 | | SYSTEM.DEFAULT.SUB 구독으로부터 사용자 ID 사용 - 공백이면 관리자 ID 사용 |
| | 관리자 ID 사용 | | 기존 구독에서 사용자 ID 사용 |

DELETE SUB 명령을 사용하여 구독을 삭제할 때 수행된 보안 검사만 명령 보안 검사입니다.

# 발행/구독 보안 설정 예

이 절에서는 보안 제어를 필요에 따라 적용할 수 있는 방식으로 토픽에 대해 액세스 제어가 설정되어 있는 시나리오에 대해 설명합니다.

## 토픽 구독을 위해 사용자에게 액세스 부여

이 주제는 두 명 이상의 사용자에 의해 토픽에 대한 액세스를 부여하는 방법을 설명하는 태스크 목록의 첫 번째 항목입니다.

### 이 태스크 정보

이 태스크는 관리 토픽 오브젝트가 존재하지 않으며 구독 또는 발행용으로 정의된 프로파일도 없다고 가정합니다. 애플리케이션은 기존 구독을 계속하기보다 새 구독을 작성하며 토픽 문자열만을 사용하여 이를 수행합니다.

애플리케이션은 토픽 오브젝트, 토픽 문자열 또는 둘의 조합을 제공하여 구독을 작성할 수 있습니다. 애플리케이션이 선택하는 어떤 방법이든지 그 영향은 토픽 트리의 특정 지점에서 구독을 작성하는 것입니다. 토픽 트리의 이 지점이 관리 토픽 오브젝트에 의해 표시될 경우 보안 프로파일은 해당 토픽 오브젝트의 이름을 기반으로 선택됩니다.

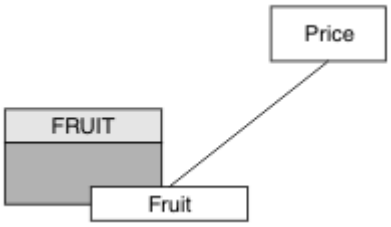



그림 *23.* 토픽 오브젝트 액세스의 예

| 표 *89.* 토픽 오브젝트 액세스의 예 | | |
|---|---|---|
| **토픽** | **필요한 구독 액세스** | **토픽 오브젝트** |
| 가격 | 사용자 없음 | 없음 |
| Price/Fruit | USER1 | 과일 |

다음과 같이 새 토픽 오브젝트를 정의하십시오.

## 프로시저

1. MQSC 명령 DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')를 발행하십시오.
2. 다음과 같이 액세스를 부여하십시오.

- z/OS **z/OS** :

  hlq.SUBSCRIBE.FRUIT 프로파일에 대한 사용자 액세스를 부여하여 "Price/Fruit" 토픽을 구독하기 위해 USER1에 액세스를 부여하십시오. 이를 수행하려면 다음 RACF 명령을 사용하여 다음을 수행하십시오.

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi **멀티플랫폼**:

  FRUIT 오브젝트에 대한 사용자 액세스를 부여하여 "Price/Fruit" 토픽을 구독하기 위해 USER1에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

  ALW **AIX, Linux, and Windows 시스템**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

## 결과

USER1이 토픽 "Price/Fruit"를 구독하려고 시도하면 결과는 성공입니다.

USER2가 토픽 "Price/Fruit"를 구독하려고 시도할 때 결과는 다음 및 MQRC_NOT_AUTHORIZED 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW AIX, Linux, and Windows의 다음 권한 이벤트:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

- IBM i IBMi에서는 다음 권한 부여 이벤트가 제공됩니다.

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  ```

```
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

이는 전체 필드가 아닌 사용자에게 표시되는 그림입니다.

## 트리 내에서 더 깊은 토픽을 구독할 수 있는 액세스를 사용자에게 부여

이 주제는 두 명 이상의 사용자에 의해 토픽에 대한 액세스를 부여하는 방법을 설명하는 태스크 목록의 두 번째 항목입니다.

### 시작하기 전에

이 주제에서는 459 페이지의 『토픽 구독을 위해 사용자에게 액세스 부여』에 설명된 설정을 사용합니다.

### 이 태스크 정보

애플리케이션이 구독을 작성하는 토픽 트리에서의 지점이 관리 토픽 오브젝트에 의해 표시되지 않으면 가장 가까운 상위 관리 토픽 오브젝트를 찾을 때까지 트리 위로 이동하십시오. 보안 프로파일은 해당 토픽 오브젝트의 이름을 기반으로 검사됩니다.

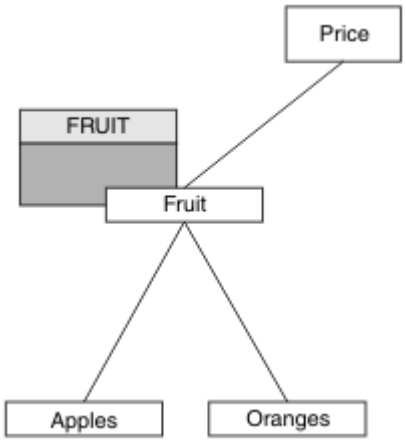



그림 *24.* 토픽 트리 내에서 토픽에 대한 액세스 부여의 예

표 *90.* 예제 토픽 및 토픽 오브젝트에 대한 액세스 요구사항

| 토픽 | 필요한 구독 액세스 | 토픽 오브젝트 |
|---|---|---|
| 가격 | 사용자 없음 | 없음 |
| Price/Fruit | USER1 | 과일 |
| Price/Fruit/Apples | USER1 | |
| Price/Fruit/Oranges | USER1 | |

459 페이지의 『토픽 구독을 위해 사용자에게 액세스 부여』에서 USER1 에는 z/OS 의 `hlq.SUBSCRIBE.FRUIT` 프로파일에 대한 액세스 및 멀티플랫폼의 `FRUIT` 프로파일에 대한 등록 액세스를 부여하여 토픽 `"Price/Fruit"` 에 구독할 수 있는 액세스 권한이 부여되었습니다. 이 단일 프로파일은 또한 `"Price/Fruit/Apples"`, `"Price/Fruit/Oranges"` 및 `"Price/Fruit/#"`를 구독하기 위한 USER1 액세스를 부여합니다.

USER1이 토픽 `"Price/Fruit/Apples"`를 구독하려고 시도하면 결과는 성공입니다.

USER2가 토픽 `"Price/Fruit/Apples"`를 구독하려고 시도할 때 결과는 다음 및 `MQRC_NOT_AUTHORIZED` 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Multi Multiplatforms에서 다음 권한 부여 이벤트가 발생합니다.

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

다음에 유의하십시오.

- z/OS z/OS에서 수신한 메시지는 이전 태스크에서 수신한 메시지와 동일합니다. 동일한 토픽 오브젝트 및 프로파일이 액세스를 제어 중이기 때문입니다.
- Multi 멀티플랫폼에서 수신하는 이벤트 메시지는 이전 태스크에서 수신한 메시지와 유사하지만 실제 토픽 문자열은 다릅니다.

## 트리 내에서 더 깊은 토픽만 구독할 수 있는 또 다른 사용자 액세스 부여

이 주제는 두 명 이상의 사용자가 토픽을 구독할 수 있는 액세스를 부여하는 방법을 설명하는 태스크 목록의 세 번째 항목입니다.

### 시작하기 전에

이 주제에서는 461 페이지의 『트리 내에서 더 깊은 토픽을 구독할 수 있는 액세스를 사용자에게 부여』에 설명된 설정을 사용합니다.

### 이 태스크 정보

461 페이지의 『트리 내에서 더 깊은 토픽을 구독할 수 있는 액세스를 사용자에게 부여』에서 USER2 는 `"Price/Fruit/Apples"`토픽에 대한 액세스가 거부되었습니다. 이 주제에서는 다른 토픽이 아닌 해당 토픽에 대한 액세스를 부여하는 방법을 알려줍니다.

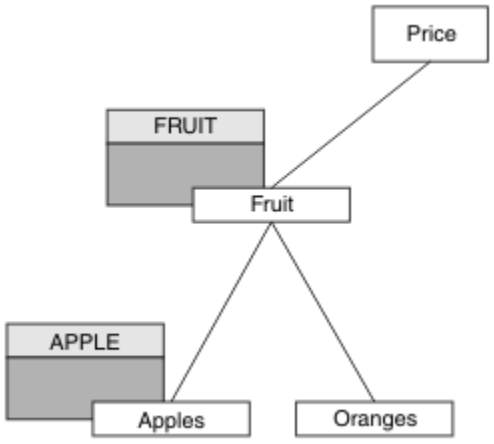



그림 *25.* 토픽 트리 내에서 특정 토픽에 대한 액세스 부여

| 표 91. 예제 토픽 및 토픽 오브젝트에 대한 액세스 요구사항 | | |
|---|---|---|
| **토픽** | **필요한 구독 액세스** | **토픽 오브젝트** |
| 가격 | 사용자 없음 | 없음 |
| Price/Fruit | USER1 | 과일 |
| Price/Fruit/Apples | USER1 및 USER2 | APPLE |
| Price/Fruit/Oranges | USER1 | |

다음과 같이 새 토픽 오브젝트를 정의하십시오.

## 프로시저

1. MQSC 명령 `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`을 실행하십시오.
2. 다음과 같이 액세스를 부여하십시오.

- z/OS **z/OS** :

  461 페이지의 『트리 내에서 더 깊은 토픽을 구독할 수 있는 액세스를 사용자에게 부여』 USER1 에서 `hlq.SUBSCRIBE.FRUIT` 프로파일에 대한 사용자 액세스 권한을 부여하여 `"Price/Fruit/Apples"` 토픽을 구독할 수 있는 액세스 권한이 부여되었습니다.

  이 단일 프로파일은 또한 `"Price/Fruit/Oranges"` `"Price/Fruit/#"` 에 구독할 수 있는 USER1 액세스 권한을 부여했으며 이 액세스 권한은 새 토픽 오브젝트 및 이와 연관된 프로파일이 추가된 경우에도 유지됩니다.

  `hlq.SUBSCRIBE.APPLE` 프로파일에 대한 사용자 액세스를 부여하여 `"Price/Fruit/Apples"` 토픽을 구독하기 위해 USER2에 액세스를 부여하십시오. 이를 수행하려면 다음 RACF 명령을 사용하여 다음을 수행하십시오.

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
  ```

- Multi 멀티플랫폼:

  461 페이지의 『트리 내에서 더 깊은 토픽을 구독할 수 있는 액세스를 사용자에게 부여』 USER1 에서 `FRUIT` 프로파일에 대한 사용자 구독 액세스 권한을 부여하여 토픽 `"Price/Fruit/Apples"` 에 구독하기 위한 액세스 권한을 부여했습니다.

  이 단일 프로파일에는 또한 `"Price/Fruit/Oranges"` 및 `"Price/Fruit/#"`을 구독할 수 있는 USER1 액세스가 부여되고 이 액세스는 새 토픽 오브젝트 및 이와 연관된 프로파일을 추가하는 경우에도 남습니다.

  APPLE 프로파일에 대한 사용자 구독 액세스를 부여하여 `"Price/Fruit/Apples"` 토픽을 구독하기 위해 USER2에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

  ALW **AIX, Linux, and Windows 시스템**

  ```
  setmqaut -t topic -n APPLE -p USER2 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
  ```

### 결과

z/OS z/OS에서, USER1이 토픽 "Price/Fruit/Apples"를 구독하려고 시도할 때 hlq.SUBSCRIBE.APPLE 프로파일에서 첫 번째 보안 검사는 실패하지만 트리 위로 이동할 때 hlq.SUBSCRIBE.FRUIT 프로파일은 USER1이 구독하도록 허용하므로 구독은 성공하고 리턴 코드가 MQSUB에 전송되지 않습니다. 그러나 RACF ICH 메시지가 첫 번째 검사에 대해 생성됩니다.

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

USER2가 토픽 "Price/Fruit/Apples"를 구독하려고 시도할 때 보안 검사가 첫 번째 프로파일에 전달되므로 결과는 성공적입니다.

USER2가 토픽 "Price/Fruit/Oranges"를 구독하려고 시도할 때 결과는 다음 및 MQRC_NOT_AUTHORIZED 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW AIX, Linux, and Windows 플랫폼의 다음 권한 이벤트:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- IBM i IBMi에서는 다음 권한 부여 이벤트가 제공됩니다.

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

z/OS 이 설정의 단점은 z/OS에서는 콘솔에서 추가 ICH 메시지를 받는다는 점입니다. 토픽 트리를 다른 방법으로 보안 설정하면 이를 피할 수 있습니다.

## 추가 메시지를 방지하기 위한 액세스 제어 변경

이 토픽은 둘 이상의 사용자 z/OS 에서 추가 RACF ICH408I 메시지를 방지하기 위해가 토픽을 구독하기 위한 액세스 권한을 부여하는 방법을 설명하는 태스크 목록의 네 번째 토픽입니다.

### 시작하기 전에

이 주제는 추가 오류 메시지를 방지할 수 있도록 462 페이지의 『트리 내에서 더 깊은 토픽만 구독할 수 있는 또 다른 사용자 액세스 부여』에 설명된 설정을 개선합니다.

### 이 태스크 정보

이 주제에서는 트리에서 더 깊은 토픽에 대한 액세스를 부여하는 방법과 이를 필요로 하는 사용자가 없을 때 트리의 더 아래쪽 토픽에 대한 액세스를 제거하는 방법을 알려줍니다.

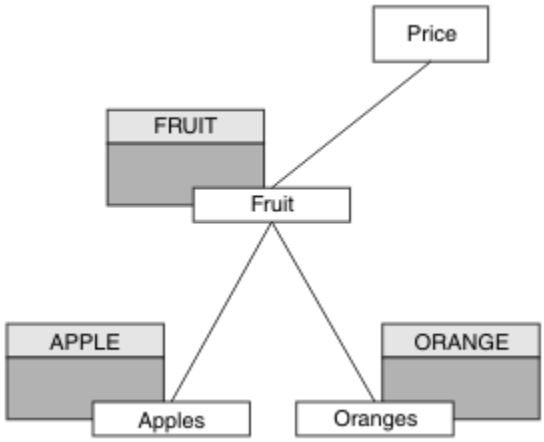



그림 *26.* 추가 메시지를 방지하기 위해 액세스 제어를 부여하는 예

다음과 같이 새 토픽 오브젝트를 정의하십시오.

## 프로시저

1. MQSC 명령 `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`을 실행하십시오.
2. 다음과 같이 액세스를 부여하십시오.

- z/OS **z/OS** :

  새 프로파일을 정의하고 해당 프로파일 및 기존 프로파일에 대한 액세스를 추가하십시오. 이를 수행하려면 다음 RACF 명령을 사용하여 다음을 수행하십시오.

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.ORANGE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.ORANGE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  PERMIT hlq.SUBSCRIBE.APPLE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi 멀티플랫폼:

  플랫폼의 권한 부여 명령을 사용하여 동등한 액세스를 설정하십시오.

  ALW **AIX, Linux, and Windows 시스템**

  ```
  setmqaut -t topic -n ORANGE -p USER1 +sub
  setmqaut -t topic -n APPLE -p USER1 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

## 결과

z/OS z/OS에서 USER1이 토픽 "Price/Fruit/Apples"를 구독하기 위해 시도할 때 `hlq.SUBSCRIBE.APPLE` 프로파일에서 첫 번째 보안 검사가 성공합니다.

USER2가 토픽 "Price/Fruit/Apples"를 구독하려고 시도할 때 보안 검사가 첫 번째 프로파일에 전달되므로 결과는 성공적입니다.

USER2가 토픽 "Price/Fruit/Oranges"를 구독하려고 시도할 때 결과는 다음 및 `MQRC_NOT_AUTHORIZED` 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW AIX, Linux, and Windows의 다음 권한 이벤트:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i IBM i의 다음 권한 이벤트:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

## 토픽에 발행을 위해 사용자에게 액세스 부여

이 주제는 두 명 이상의 사용자가 토픽을 발행할 수 있는 액세스를 부여하는 방법을 설명하는 태스크 목록의 첫 번째 항목입니다.

### 이 태스크 정보

이 태스크는 관리 토픽 오브젝트가 토픽 트리의 오른쪽에 존재하지 않으며 발행용으로 정의된 프로파일도 없다고 가정합니다. 여기서는 발행자가 토픽 문자열만 사용한다는 가정이 사용됩니다.

애플리케이션은 토픽 오브젝트, 토픽 문자열 또는 둘의 조합을 제공하여 토픽을 발행할 수 있습니다. 애플리케이션이 선택하는 어떤 방법이든지 그 영향은 토픽 트리의 특정 지점에서 발행하는 것입니다. 토픽 트리의 이 지점이 관리 토픽 오브젝트에 의해 표시될 경우 보안 프로파일은 해당 토픽 오브젝트의 이름을 기반으로 선택됩니다. 예를 들면, 다음과 같습니다.

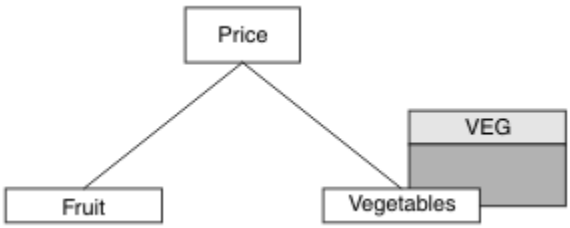



그림 *27.* 토픽에 대한 발행 액세스 권한 부여

표 *92.* 발행 액세스 요구사항의 예

| 토픽 | 필요한 발행 액세스 | 토픽 오브젝트 |
|---|---|---|
| 가격 | 사용자 없음 | 없음 |
| Price/Vegetables | USER1 | VEG |

다음과 같이 새 토픽 오브젝트를 정의하십시오.

## 프로시저

1. MQSC 명령 DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')를 발행하십시오.
2. 다음과 같이 액세스를 부여하십시오.

   - z/OS **z/OS** :

     hlq.PUBLISH.VEG 프로파일에 대한 사용자 액세스를 부여하여 "Price/Vegetables" 토픽을 발행하기 위해 USER1에 액세스를 부여하십시오. 이를 수행하려면 다음 RACF 명령을 사용하여 다음을 수행하십시오.

     ```
     RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
     PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
     ```

   - 기타 플랫폼의 경우:

     VEG 프로파일에 대한 사용자 액세스를 부여하여 "Price/Vegetables" 토픽을 발행하기 위해 USER1에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

     ALW **AIX, Linux, and Windows 시스템**

     ```
     setmqaut -t topic -n VEG -p USER1 +pub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
     ```

## 결과

USER1이 토픽 "Price/Vegetables"을 발행하려고 시도하면 결과는 성공적입니다. 즉, MQOPEN 호출이 성공합니다.

USER2가 토픽 "Price/Vegetables"을 발행하려고 시도할 때 MQOPEN 호출은 다음 및 MQRC_NOT_AUTHORIZED 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

  ```
  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.VEG ...

  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
  ```

- ALW 기타 플랫폼에서는 다음 권한 부여 이벤트가 제공됩니다.

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
  TopicString           "Price/Vegetables"
  ```

- IBM i IBMi에서는 다음 권한 부여 이벤트가 제공됩니다.

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
  TopicString           "Price/Vegetables"
  ```

이는 전체 필드가 아닌 사용자에게 표시되는 그림입니다.

## 트리 내에서 더 깊은 토픽에 발행할 수 있는 액세스를 사용자에게 부여

이 주제는 두 명 이상의 사용자가 토픽을 발행할 수 있는 액세스를 부여하는 방법을 설명하는 태스크 목록의 두 번째 항목입니다.

### 시작하기 전에

이 주제에서는 466 페이지의 『토픽에 발행을 위해 사용자에게 액세스 부여』에 설명된 설정을 사용합니다.

### 이 태스크 정보

애플리케이션이 발행하는 토픽 트리에서의 지점이 관리 토픽 오브젝트에 의해 표시되지 않으면 가장 가까운 상위 관리 토픽 오브젝트를 찾을 때까지 트리 위로 이동하십시오. 보안 프로파일은 해당 토픽 오브젝트의 이름을 기반으로 검사됩니다.

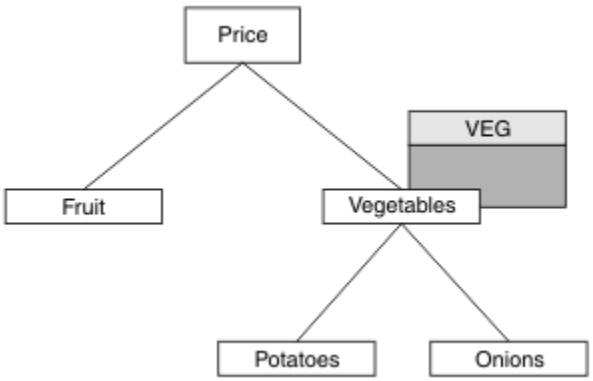



그림 *28.* 토픽 트리 내에서 토픽에 대한 발행 액세스 부여

표 *93.* 발행 액세스 요구사항의 예

| 토픽 | 필요한 구독 액세스 | 토픽 오브젝트 |
|---|---|---|
| 가격 | 사용자 없음 | 없음 |
| Price/Vegetables | USER1 | VEG |
| Price/Vegetables/Potatoes | USER1 | |
| Price/Vegetables/Onions | USER1 | |

이전 태스크에서 USER1 에게는 z/OS 의 `hlq.PUBLISH.VEG` 프로파일에 대한 액세스 또는 멀티플랫폼의 VEG 프로파일에 대한 공개 액세스를 부여하여 토픽 `"Price/Vegetables/Potatoes"` 를 발행하기 위한 액세스 권한이 부여되었습니다. 이 단일 프로파일은 USER1 에 `"Price/Vegetables/Onions"`에서 공개할 수 있는 액세스 권한도 부여합니다.

USER1이 `"Price/Vegetables/Potatoes"` 토픽을 발행하려고 시도하면 결과는 성공적입니다. 즉, MQOPEN 호출이 성공합니다.

USER2가 `"Price/Vegetables/Potatoes"` 토픽을 구독하려고 시도하면 결과는 실패입니다. 즉, MQOPEN 호출은 다음과 함께 `MQRC_NOT_AUTHORIZED` 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Multi Multiplatforms에서 다음 권한 부여 이벤트가 발생합니다.

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

다음에 유의하십시오.

- z/OS z/OS에서 수신한 메시지는 이전 태스크에서 수신한 메시지와 동일합니다. 동일한 토픽 오브젝트 및 프로파일이 액세스를 제어 중이기 때문입니다.
- Multi 멀티플랫폼에서 수신하는 이벤트 메시지는 이전 태스크에서 수신한 메시지와 유사하지만 실제 토픽 문자열은 다릅니다.

## 발행 및 구독을 위한 액세스 부여

이 주제는 두 명 이상의 사용자가 토픽을 발행 및 구독할 수 있는 액세스를 부여하는 방법을 설명하는 태스크 목록의 마지막 항목입니다.

### 시작하기 전에

이 주제에서는 468 페이지의 『트리 내에서 더 깊은 토픽에 발행할 수 있는 액세스를 사용자에게 부여』에 설명된 설정을 사용합니다.

### 이 태스크 정보

이전 태스크에서 USER1에게 토픽 "Price/Fruit"를 구독할 수 있는 액세스가 제공되었습니다. 이 주제에서는 사용자에게 해당 토픽에 발행하기 위한 액세스를 부여하는 방법을 알려줍니다.

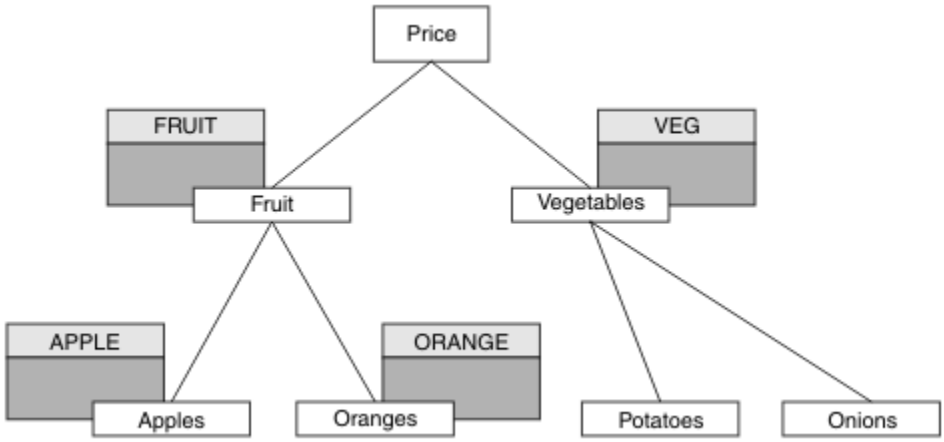



그림 29. 발행 및 구독을 위한 액세스 부여

| 표 94. 액세스 요구사항 발행 및 구독 예 | | | |
|---|---|---|---|
| **토픽** | **필요한 구독 액세스** | **필요한 발행 액세스** | **토픽 오브젝트** |
| 가격 | 사용자 없음 | 사용자 없음 | 없음 |
| Price/Fruit | USER1 | USER1 | 과일 |
| Price/Fruit/Apples | USER1 및 USER2 | | APPLE |
| Price/Fruit/Oranges | USER1 | | ORANGE |

## 프로시저

다음과 같이 액세스를 부여하십시오.

- z/OS **z/OS** :

  이전 태스크에서 USER1는 `hlq.SUBSCRIBE.FRUIT` 프로파일에 대한 사용자 액세스를 부여하여 토픽 `"Price/Fruit"`를 구독할 수 있는 액세스가 부여되었습니다.

  `"Price/Fruit"` 토픽에 발행하기 위해 USER1에게 `hlq.PUBLISH.FRUIT` 프로파일에 대한 액세스를 부여하십시오. 이를 수행하려면 다음 RACF 명령을 사용하여 다음을 수행하십시오.

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi 멀티플랫폼:

  FRUIT 프로파일에 대한 사용자 발행 액세스를 부여하여 `"Price/Fruit"` 토픽을 발행하기 위해 USER1에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

  ALW **AIX, Linux, and Windows 시스템**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
  ```

## 결과

z/OS z/OS에서 USER1이 `"Price/Fruit"` 토픽에 발행하려고 시도하면 MQOPEN 호출에서 보안 검사가 통과됩니다.

USER2가 토픽 `"Price/Fruit"`에서 발행하려고 시도하면 결과는 다음 및 `MQRC_NOT_AUTHORIZED` 메시지와 함께 실패합니다.

- z/OS z/OS에서는 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 다음 메시지가 표시됩니다.

  ```
  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
  ```

- ALW AIX, Linux, and Windows 플랫폼의 다음 권한 이벤트:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i IBM i의 다음 권한 이벤트:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

이러한 태스크의 전체 세트를 뒤따라 USER1 및 USER2에 나열된 토픽을 발행하고 구독하기 위한 다음과 같은 액세스 권한을 제공합니다.

표 *95.* 보안에서 발생하는 액세스 권한의 전체 목록 예

| 토픽 | 필요한 구독 액세스 | 필요한 발행 액세스 | 토픽 오브젝트 |
|---|---|---|---|
| 가격 | 사용자 없음 | 사용자 없음 | 없음 |
| Price/Fruit | USER1 | USER1 | 과일 |
| Price/Fruit/Apples | USER1 및 USER2 | | APPLE |
| Price/Fruit/Oranges | USER1 | | ORANGE |
| Price/Vegetables | | USER1 | VEG |
| Price/Vegetables/Potatoes | | | |
| Price/Vegetables/Onions | | | |

z/OS 토픽 트리 내에서 서로 다른 레벨에 있는 보안 액세스에 대해 다른 요구사항이 있는 경우에는 주의 깊게 계획하면 z/OS 콘솔 로그에서 추가로 보안 경고를 받지 않도록 해줍니다. 트리 내에서 올바른 레벨에 보안을 설정하면 잘못된 보안 메시지를 피할 수 있습니다.

## 구독 보안

### MQSO_ALTERNATE_USER_AUTHORITY

AlternateUserId 필드는 이 MQSUB 호출을 유효성 검증하는 데 사용할 사용자 ID를 포함합니다. 애플리케이션을 실행 중인 사용자 ID가 토픽을 구독하는 권한이 있는지에 관계없이 이 AlternateUserId에 지정된 액세스 옵션을 사용하여 토픽을 구독할 수 있는 권한이 있는 경우에만 호출이 성공할 수 있습니다.

### MQSO_SET_IDENTITY_CONTEXT

구독은 PubAccountingToken 및 PubApplIdentityData 필드에 제공된 계정 토큰 및 애플리케이션 ID 데이터를 사용하는 것입니다.

이 옵션을 지정하면 MQOO_SET_IDENTITY_CONTEXT와 함께 MQOPEN 호출을 사용하여 목적지 큐에 액세스한 경우와 동일한 권한 검사가 수행됩니다(목적지 큐에서 권한 검사를 수행하지 않는 경우 MQSO_MANAGED 옵션을 사용하는 경우 제외).

이 옵션을 지정하지 않으면 다음과 같이 이 구독자에게 전송된 발행에 연관된 기본 컨텍스트 정보가 있습니다.

표 *96.* 기본 발행 컨텍스트 정보

| **MQMD의 필드** | **사용된 값** |
|---|---|
| *UserIdentifier* | 발행이 작성된 시간에 구독과 연관된 사용자 ID(DISPLAY SBSTATUS의 SUBUSER 필드 참조)입니다. |
| *AccountingToken* | 가능한 경우 환경에서 결정됩니다. 그렇지 않으면 MQACT_NONE으로 설정됩니다. |
| *ApplIdentityData* | 공백으로 설정합니다. |

이 옵션은 MQSO_CREATE 및 MQSO_ALTER에서만 유효합니다. MQSO_RESUME와 함께 사용하면 PubAccountingToken 및 PubApplIdentityData 필드가 무시되고 따라서 이 옵션은 적용되지 않습니다.

이전에 구독이 ID 컨텍스트 정보를 제공한 경우 해당 옵션을 사용하지 않고 구독을 대체하면 대체된 구독에 대한 기본 컨텍스트 정보가 생성됩니다.

여러 사용자 ID가 MQSO_ANY_USERID 옵션과 함께 이 옵션을 사용할 수 있게 하는 구독이 다른 사용자 ID에 의해 재개되는 경우 구독을 소유한 새 사용자 ID에 대한 기본 ID 컨텍스트가 생성되고 새 ID 컨텍스트가 포함된 후속 발행물이 전달됩니다.

### AlternateSecurityId

적절한 권한 검사를 수행할 수 있도록 AlternateUserId와 함께 권한 서비스에 전달되는 보안 ID입니다. AlternateSecurityId는 MQSO_ALTERNATE_USER_AUTHORITY가 지정된 경우에만 사용되며, AlternateUserId 필드는 첫 번째 널 문자 또는 필드의 끝까지 전체적으로 비어있지 않습니다.

## MQSO_ANY_USERID subscription 옵션

MQSO_ANY_USERID가 지정된 경우 구독자의 ID는 단일 사용자 ID로 제한되지 않습니다. 이로 인해 적절한 권한이 있을 때 사용자가 구독을 대체하거나 재개할 수 있습니다. 단일 사용자에 한해서 언제든지 구독 가능합니다. 또 다른 애플리케이션이 현재 사용 중인 구독을 재개하기 위한 시도는 호출을 MQRC_SUBSCRIPTION_IN_USE로 실패하게 만듭니다.

이 옵션을 기존 구독에 추가하려면 MQSUB 호출(MQSO_ALTER 사용)은 원본 구독과 같은 사용자 ID로부터 나와야 합니다.

MQSUB 호출이 MQSO_ANY_USERID가 설정된 기존 구독을 참조하고 사용자 ID가 원래 구독과 다른 경우 새 사용자 ID에 토픽을 구독할 수 있는 권한이 있을 때에만 호출에 성공합니다. 성공적인 완료 후에 이 구독자에 대한 추가 발행은 발행에 설정된 새 사용자 ID와 함께 구독자 큐에 놓입니다.

### MQSO_FIXED_USERID

MQSO_FIXED_USERID가 지정될 때 구독은 단일 소유 사용자 ID에 의해서만 변경되거나 재개될 수 있습니다. 이 사용자 ID는 이 옵션을 설정하여 MQSO_ANY_USERID 옵션을 제거한 구독을 변경하기 위한 마지막 사용자 ID이거나, 변경이 발생하지 않으면 구독을 작성한 사용자 ID입니다.

MQSUB 동사가 MQSO_ANY_USERID가 설정된 기존 구독을 가리키고 MQSO_FIXED_USERID 옵션을 사용하기 위해 구독(MQSO_ALTER 사용)을 변경하는 경우, 구독의 사용자 ID는 이제 이 새 사용자 ID로 고정됩니다. 새 사용자 ID에 토픽을 구독할 수 있는 권한이 있는 경우에만 호출에 성공합니다.

구독을 소유한 것으로 기록된 사용자 ID 이외의 사용자 ID가 MQSO_FIXED_USERID 구독을 재개하거나 변경하려고 시도하는 경우 호출은 MQRC_IDENTITY_MISMATCH와 함께 실패합니다. DISPLAY SBSTATUS 명령을 사용하여 구독에에 대한 소유 사용자 ID를 볼 수 있습니다.

MQSO_ANY_USERID 또는 MQSO_FIXED_USERID가 지정되지 않은 경우에는 기본값은 MQSO_FIXED_USERID입니다.

# 큐 관리자 간의 발행/구독 보안

프록시 구독 및 발행물과 같은 발행/구독 내부 메시지는 정상적인 채널 보안 규칙을 사용하여 발행/구독 시스템 큐에 놓입니다. 이 토픽에 있는 정보 및 다이어그램은 메시지 전달에 포함되는 다양한 프로세스 및 사용자 ID를 강조표시합니다.

## 로컬 액세스 제어

발행물 및 구독에 대한 토픽의 액세스는 발행/구독 보안에 설명된 로컬 보안 정의 및 규칙으로 관리됩니다. 액세스 제어를 설정하는 데 로컬 토픽 오브젝트가 필요하지 않습니다. 관리자는 클러스터된 토픽 오브젝트가 클러스터에 아직 존재하는지 여부와 관계없이 이에 대한 액세스 제어를 적용하도록 선택할 수 있습니다.

시스템 관리자는 로컬 시스템에서의 액세스 제어를 담당합니다. 이러한 관리자는 액세스 제어 정책을 담당할 클러스터 집합 또는 계층의 다른 멤버의 관리자를 신뢰해야 합니다. 액세스 제어가 별도의 각 시스템에 대해 정의되어 있으므로 상세 레벨 제어가 필요한 경우 번거로운 일이 될 수 있습니다. 액세스 제어를 부여하는 데 필요하지 않거나 액세스 제어가 토픽 트리의 상위 레벨 오브젝트에서 정의될 수 있습니다. 상세 레벨 액세스 제어는 토픽 네임스페이스의 각 세분화를 위해 정의할 수 있습니다.

## 프록시 구독 작성

조직이 큐 관리자에 해당 큐 관리자를 연결하기 위한 신뢰는 정상 채널 인증 수단으로 확인됩니다. 또한 신뢰되는 조직이 분산 발행/구독을 수행하도록 허용되는 경우 권한 검사가 수행됩니다. 이 검사는 채널이 분산된 발행/구독 큐에 메시지를 넣을 때 수행됩니다. 예를 들어, 메시지를 SYSTEM.INTER.QMGR.CONTROL 큐에 넣는 경우입니다. 큐 권한 검사를 위한 사용자 ID는 수신 채널의 PUTAUT 값에 따라 달라집니다. 예를 들어, 값 및 플랫폼에 따라 채널의 사용자 ID, MCAUSER, 메시지 컨텍스트. 채널 보안에 대한 자세한 정보는 채널 보안을 참조하십시오.

프록시 구독은 리모트 큐 관리자에서 분배된 발행/구독 에이전트의 사용자 ID로 작성됩니다. 예를 들어, 473 페이지의 그림 30의 QM2입니다. 그러면 사용자 ID가 시스템에 정의되어 있고 따라서 도메인 충돌이 없으므로 이 사용자는 쉽게 로컬 토픽 오브젝트 프로파일에 대한 액세스 권한을 부여받게 됩니다.

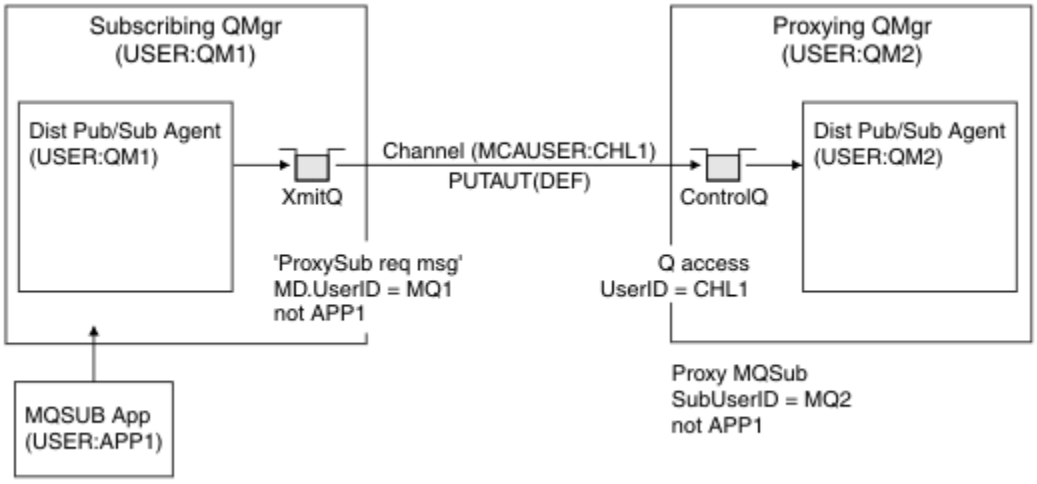



그림 *30.* 프록시 구독 보안, 구독 작성

## 원격 발행물 다시 송신

구독 큐 관리자에서 발행물이 작성되면 발행물의 사본이 프록시 구독용으로 작성됩니다. 복사된 발행물의 컨텍스트에는 구독한 사용자 ID의 컨텍스트를 포함합니다(474 페이지의 그림 31의 QM2). 프록시 구독은 리모트 큐인 목적지 큐와 함께 작성되므로 발행 메시지는 전송 큐에서 해결됩니다.

큐 관리자, QM2을(를) 다른 큐 관리자, QM1에 연결하는 조직에 대한 신뢰는 정상적인 채널 인증 수단을 통해 확인합니다. 신뢰할 수 있는 조직이 분배된 발행/구독을 수행할 수 있는 경우, 채널이 발행 메시지를 분산 발행/구독

발행 큐 SYSTEM.INTER.QMGR.PUBS에 넣을 때 권한 검사가 수행됩니다. 큐 권한 검사에 대한 사용자 ID는 수신 채널의 PUTAUT 값에 의존합니다(예를 들어, 값과 플랫폼에 따라 채널, MCAUSER, 메시지 컨텍스트 등의 사용자 ID). 채널 보안에 대한 자세한 정보는 채널 보안을 참조하십시오.

발행 메시지가 구독 큐 관리자에 도달할 때, 토픽에 대한 다른 MQPUT는 해당 큐 관리자의 권한 하에 수행되고, 메시지가 있는 컨텍스트는 로컬 구독자에게 각각 메시지가 제공되는 대로 각 로컬 구독자의 컨텍스트에 의해 대체됩니다.

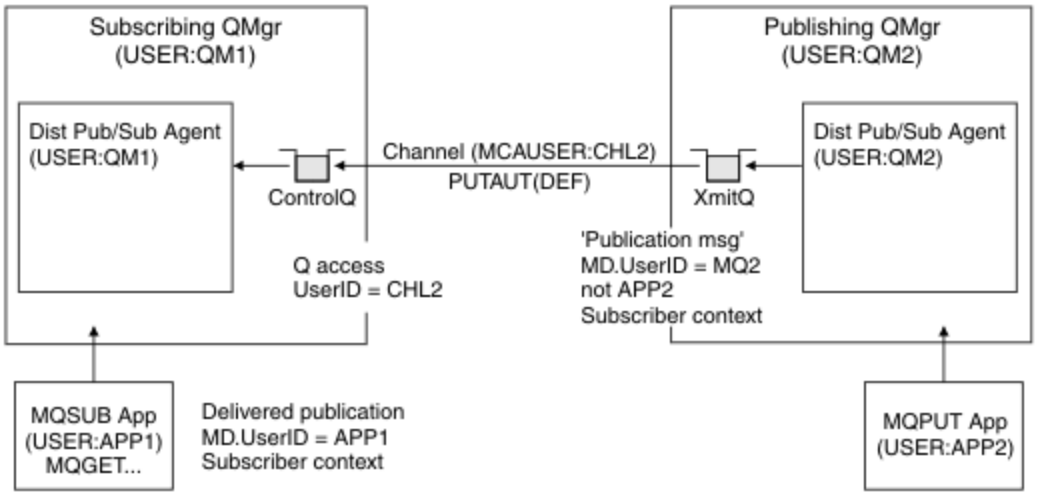



그림 *31.* 프록시 구독 보안, 발행물 전달

보안과 관련하여 거의 고려되지 않은 시스템에서 분산 발행/구독 프로세스는 mqm 그룹의 사용자 ID로 실행될 수 있으며 채널의 MCAUSER 매개변수는 공백(기본값)이고 메시지는 필요에 따라 다양한 시스템 큐로 전달됩니다. 안전하지 않는 시스템은 분배된 발행/구독을 시연하기 위한 개념 증명을 설정하기가 쉽습니다.

보안이 보다 심각하게 고려되는 시스템에서는 이러한 내부 메시지가 채널을 통해 이동하는 메시지와 동일한 보안 제어의 영향을 받습니다.

채널을 비어 있지 않은 MCAUSER 및 PUTAUT 값으로 설정하여 MCAUSER를 확인해야 함을 지정하는 경우 문제의 MCAUSER에 SYSTEM.INTER.QMGR.* 큐에 대한 액세스 권한이 부여되어야 합니다. 다른 MCAUSER ID로 실행되는 채널이 있는 다른 리모트 큐 관리자가 여러 개이면 해당 모든 사용자 ID에 SYSTEM.INTER.QMGR.* 큐에 액세스할 수 있는 권한을 부여해야 합니다. 다른 MCAUSER ID에서 실행 중인 채널은 예를 들어 다중 계층 구조 연결이 단일 큐 관리자에 구성된 경우 발생할 수 있습니다.

메시지의 컨텍스트를 사용하도록 지정하는 PUTAUT 값을 사용하여 채널이 설정된 경우 내부 메시지 내부의 사용자 ID를 기반으로 SYSTEM.INTER.QMGR.* 큐에 대한 액세스 권한이 확인됩니다. 이 메시지 모두 내부 메시지 또는 발행물 메시지를 송신하는 큐 관리자의 분산 발행/구독 에이전트의 사용자 ID로 넣어지기 때문에(474 페이지의 그림 31 참조) 이런 방식으로 분산 발행/구독 보안을 설정하면 사용자 ID 세트가 너무 크지 않기 때문에 다양한 시스템 큐에 액세스를 부여할 수 있습니다(리모트 큐 관리자별로 한 개). 채널 컨텍스트 보안이 항상 가지고 있는 동일한 문제 모두를 계속 가지고 있습니다. 다양한 사용자 ID 도메인의 문제와 메시지의 사용자 ID가 수신 시스템에서 정의되지 않을 수 있다는 문제입니다. 그러나 이는 필요한 경우 완벽히 허용되는 실행 방법입니다.

z/OS 시스템 큐 보안은 분산 발행/구독 환경을 안전하게 설정하기 위해 필요한 큐 및 액세스 목록을 제공합니다. 보안 파기로 인해 내부 메시지 또는 발행물 저장에 실패할 경우 채널은 정상 방법으로 로그에 메시지를 기록하며 이 메시지는 정상 채널 오류 처리에 따라 데드-레터 큐로 송신될 수 있습니다.

분산 발행/구독을 위해 모든 내부 큐 관리자 메시징은 정상 채널 보안을 사용하여 실행됩니다.

토픽 레벨에서 발행물 및 프록시 구독을 제한하는 방법에 대한 정보는 발행/구독 보안을 참조하십시오.

## 큐 관리자 계층으로 기본 사용자 ID 사용

다른 플랫폼에서 실행 중인 큐 관리자의 계층이 있고 기본 사용자 ID를 사용 중인 경우 이러한 기본 사용자 ID는 플랫폼 간에 다를 수 있고 대상 플랫폼에서 인식되지 않을 수 있습니다. 따라서 한 플랫폼에서 실행 중인 큐 관리자는 이유 코드가 MQRC_NOT_AUTHORIZED인 다른 플랫폼의 큐 관리자로부터 수신된 메시지를 거부합니다.

메시지가 거부되지 않도록 하려면 최소한 다른 플랫폼에서 사용되는 기본 사용자 ID에 다음 권한을 추가해야 합니다.

- SYSTEM.BROKER의 *PUT *GET 권한. 큐
- SYSTEM.BROKER의 *PUB *SUB 권한. 토픽
- SYSTEM.BROKER.CONTROL.QUEUE 큐의 *ADMCRT *ADMDLT *ADMCHG 권한.

큐 관리자 계층 구조가 있는 기본 사용자 ID는 다음과 같습니다.

| 플랫폼 | 기본 사용자 ID |
|---|---|
| Windows | mqm |
| AIX and Linux 시스템 | mqm |
| IBM i | QMQM |
| z/OS | 채널 시작기 주소 공간 사용자 ID |

IBM i 이외의 플랫폼에 있는 큐 관리자가 IBM i의 큐 관리자에 계층적으로 첨부된 경우, 'qmqm' 사용자 ID에 대한 액세스를 작성하고 부여하십시오.

IBM i 또는 z/OS 의 큐 관리자가 AIX, Linux, and Windows의 큐 관리자에 계층적으로 첨부된 경우 'mqm' 사용자 ID에 대한 액세스를 작성하고 부여하십시오.

멀티플랫폼 의 큐 관리자가 z/OS의 큐 관리자에 계층적으로 첨부된 경우, z/OS 채널 시작기 주소 공간 사용자 ID에 대한 액세스를 작성하고 부여하십시오.

사용자 ID는 대소문자가 구별될 수 있습니다. 원래 큐 관리자(멀티플랫폼에 있는 경우)에서는 사용자 ID를 모두 대문자로 설정합니다. 수신 큐 관리자(AIX, Linux, and Windows에 있는 경우)에서는 사용자 ID를 모두 소문자로 설정합니다. 따라서 AIX and Linux 시스템에서 작성된 모든 사용자 ID는 소문자 양식으로 작성되어야 합니다. 메시지 엑시트가 설치된 경우 사용자 ID를 대문자 또는 소문자로 강제 실행하는 작업이 발생하지 않습니다. 메시지 엑시트가 사용자 ID를 처리하는 방법을 이해하려면 경우에는 주의하십시오.

사용자 ID의 변환에 관련된 잠재적인 문제점을 방지하려면 다음을 수행하십시오.

- AIX, Linux, and Windows 시스템에서는 사용자 ID가 소문자로 지정되었는지 확인하십시오.
- IBM i 및 z/OS 시스템에서 사용자 ID가 대문자로 지정되었는지 확인하십시오.

# IBM MQ Console 및 REST API 보안

IBM MQ Console 및 REST API에 대한 보안은 `mqwebuser.xml` 파일에서 mqweb 서버 구성을 편집하여 구성됩니다.

## 이 태스크 정보

mqweb 서버의 로그 파일을 검사하여 사용자 조치를 추적하고 IBM MQ Console 및 REST API 사용을 감사할 수 있습니다.

IBM MQ Console 및 REST API의 사용자는 다음을 통해 인증될 수 있습니다.

- 기본 레지스트리
- LDAP 레지스트리
- 로컬 OS 레지스트리
- z/OS의 SAF
- WebSphere Liberty에서 지원하는 다른 레지스트리 유형

역할을 IBM MQ Console 사용자 및 REST API 사용자에게 지정하여 IBM MQ 오브젝트에 대해 해당 사용자에게 부여되는 액세스 권한 레벨을 판별할 수 있습니다. 예를 들어, 메시징을 수행하려면 사용자에게 `MQWebUser` 역할을 지정해야 합니다. 사용 가능한 역할에 대한 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』의 내용을 참조하십시오.

사용자에게 역할이 지정되고 나면 다수의 방법을 사용하여 사용자를 인증할 수 있습니다. IBM MQ Console을 사용하는 경우 사용자는 사용자 이름 및 비밀번호를 사용하여 로그인하거나 클라이언트 인증서 인증을 사용할 수 있습니다. REST API를 사용하는 경우 사용자는 기본 HTTP 인증, 토큰 기반 인증 또는 클라이언트 인증서 인증을 사용할 수 있습니다.

### 프로시저

1. 사용자 레지스트리를 정의하여 사용자를 인증하고 각 사용자 또는 그룹에 역할을 지정하여 사용자 및 그룹에 IBM MQ Console 또는 REST API를 사용할 수 있는 권한을 부여하십시오. 자세한 정보는 다음 항목을 참조하십시오.476 페이지의 『사용자 및 역할 구성』
2. IBM MQ Console의 사용자가 mqweb 서버에 대해 인증되는 방식을 선택하십시오. 모든 사용자에 대해 동일한 방법을 사용하지 않아도 됩니다.
   - 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 IBM MQ Console 로그인 화면에서 사용자 ID 및 비밀번호를 입력합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 인증 옵션을 사용하기 위해 추가 구성은 필요하지 않지만 선택적으로 LTPA 토큰의 만기 시간을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 만기 간격 구성을 참조하십시오.
   - 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 IBM MQ Console에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.
3. REST API의 사용자가 mqweb 서버에 대해 인증되는 방식을 선택하십시오. 모든 사용자에 대해 동일한 방법을 사용하지 않아도 됩니다.
   - 사용자가 HTTP 기본 인증을 사용하여 인증하게 합니다. 이 경우 사용자 이름 및 비밀번호가 인코딩되지만 암호화되지는 않으며 각각의 REST API 요청과 함께 전송되어 해당 요청에 대해 사용자를 인증하고 사용자에게 권한을 부여합니다. 이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 자세한 정보는 493 페이지의 『REST API로 HTTP 기본 인증 사용』의 내용을 참조하십시오.
   - 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API `login` 자원에 제공합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 자세한 정보는 494 페이지의 『REST API로 토큰 기반 인증 사용』의 내용을 참조하십시오.

     이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 하지만 HTTP 연결을 사용하는 경우 HTTP 연결을 위해 발행되는 LTPA 토큰을 HTTP 연결에 사용할 수 있도록 합니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
   - 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 REST API에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.
4. 옵션: REST API에 대한 CORS(Cross Origin Resource Sharing)를 공유하십시오.

   기본적으로 웹 브라우저는 스크립트(예: JavaScript)가 REST API와 동일한 원본에서 제공되지 않는 경우 스크립트가 REST API를 호출하도록 허용하지 않습니다. 즉, 원본 간 요청이 사용으로 설정되지 않습니다. 지정된 URL로부터의 원본 간 요청을 허용하도록 CORS(Cross Origin Resource Sharing)를 구성할 수 있습니다. 자세한 정보는 497 페이지의 『REST API에 대해 CORS 구성』의 내용을 참조하십시오.
5. 옵션: IBM MQ Console 및 REST API에 대한 호스트 헤더 유효성 검증을 구성하십시오.

   호스트 헤더 유효성 검증을 구성하고 호스트 이름 및 포트의 허용 목록을 작성하여 특정 호스트 헤더를 포함하는 요청만 IBM MQ Console 및 REST API에 의해 처리되도록 할 수 있습니다. 자세한 정보는 498 페이지의 『IBM MQ Console 및 REST API에 대한 호스트 헤더 유효성 검증 구성』의 내용을 참조하십시오.

## 사용자 및 역할 구성

IBM MQ Console 또는 REST API를 사용하려면 사용자가 mqweb 서버에 정의된 사용자 레지스트리에 대해 인증해야 합니다.

## 이 태스크 정보

인증된 사용자는 IBM MQ Console 및 REST API의 기능에 대한 액세스 권한을 부여하는 그룹 중 하나의 구성원이어야 합니다. 기본적으로 사용자 레지스트리에는 사용자가 포함되지 않습니다. `mqwebuser.xml` 파일을 편집하여 이를 추가해야 합니다.

사용자 및 그룹을 구성할 때는 먼저 사용자 및 그룹을 인증할 사용자 레지스트리를 구성합니다. 이 사용자 레지스트리는 IBM MQ Console과 REST API 사이에서 공유됩니다. 사용자 및 그룹에 대한 역할을 구성할 때 사용자 및 그룹에 IBM MQ Console, REST API 또는 둘 다에 대한 액세스 권한이 있는지 여부를 제어할 수 있습니다.

사용자 레지스트리를 구성한 후 사용자 및 그룹에 대한 역할을 구성하여 사용자 및 그룹에 권한을 부여합니다. Managed File Transfer용 REST API 사용에 특정한 역할을 포함하여 사용 가능한 여러 역할이 있습니다. 각 역할은 서로 다른 레벨의 액세스 권한을 부여합니다. 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』의 내용을 참조하십시오.

사용자 및 그룹의 구성을 더 단순하게 하기 위해 다수의 샘플 XML 파일이 mqweb 서버와 함께 제공됩니다. WebSphere Liberty(WLP)에서의 보안 구성에 익숙한 사용자는 샘플 사용을 선호하지 않습니다. WLP는 여기서 설명한 기능 외에 다른 권한 부여 기능을 제공합니다.

## 프로시저

- `basic_registry.xml` 파일을 사용하여 기본 레지스트리로 사용자 및 그룹을 구성하십시오.

  레지스트리에 있는 사용자 이름 및 비밀번호는 IBM MQ Console 및 REST API의 사용자를 인증하고 이 사용자에게 권한을 부여하는 데 사용됩니다.

  `basic_registry.xml` 샘플 파일을 사용하여 기본 레지스트리를 구성하려면 478 페이지의 『IBM MQ Console 및 REST API에 대한 기본 레지스트리 구성』의 내용을 참조하십시오.
- `ldap_registry.xml` 파일을 사용하여 LDAP 레지스트리로 사용자 및 그룹을 구성하십시오.

  LDAP 레지스트리에 있는 사용자 이름 및 비밀번호는 IBM MQ Console 및 REST API의 사용자를 인증하고 이 사용자에게 권한을 부여하는 데 사용됩니다.

  `ldap_registry.xml` 샘플 파일을 사용하여 LDAP 레지스트리를 구성하려면 482 페이지의 『IBM MQ Console 및 REST API에 대한 LDAP 레지스트리 구성』의 내용을 참조하십시오.
- ALW

  `local_os_registry.xml` 파일을 사용하여 로컬 운영 체제 레지스트리로 사용자 및 그룹을 구성하십시오.

  운영 체제 레지스트리에 있는 사용자 이름 및 비밀번호는 IBM MQ Console 및 REST API의 사용자를 인증하고 이 사용자에게 권한을 부여하는 데 사용됩니다.

  `local_os_registry.xml` 샘플 파일을 사용하여 로컬 OS 레지스트리를 구성하려면 481 페이지의 『IBM MQ Console 및 REST API에 대한 로컬 OS 레지스트리 구성』의 내용을 참조하십시오.
- z/OS

  `zos_saf_registry.xml` 파일을 사용하여 z/OS 에서 SAF (System Authorization Facility) 인터페이스로 사용자 및 그룹을 구성하십시오.

  RACF 또는 기타 보안 제품의 경우 프로파일을 사용하여 사용자 및 그룹에 역할에 대한 액세스 권한을 부여합니다. RACF 데이터베이스의 사용자 이름 및 비밀번호는 IBM MQ Console 및 REST API의 사용자를 인증하고 권한을 부여하는 데 사용됩니다.

  `zos_saf_registry.xml` 샘플 파일을 사용하여 SAF 인터페이스를 구성하려면 484 페이지의 『Configuring a SAF registry for the IBM MQ Console and REST API』의 내용을 참조하십시오.
- `no_security.xml` 파일을 사용하여 HTTPS를 통해 IBM MQ Console또는 REST API에 액세스하는 기능을 포함하여 보안을 사용 안함으로 설정하십시오.

## 다음에 수행할 작업

사용자 인증 방법을 선택하십시오.

**IBM MQ Console 인증 옵션**

- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 IBM MQ Console 로그인 화면에서 사용자 ID 및 비밀번호를 입력합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 인증 옵션을 사용하기 위해 추가 구성은 필요하지 않지만 선택적으로 LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 만기 간격 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 IBM MQ Console에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

**REST API 인증 옵션**

- 사용자가 HTTP 기본 인증을 사용하여 인증하게 합니다. 이 경우 사용자 이름 및 비밀번호가 인코딩되지만 암호화되지는 않으며 각각의 REST API 요청과 함께 전송되어 해당 요청에 대해 사용자를 인증하고 사용자에게 권한을 부여합니다. 이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 자세한 정보는 493 페이지의 『REST API로 HTTP 기본 인증 사용』의 내용을 참조하십시오.
- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API `login` 자원에 제공합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 자세한 정보는 494 페이지의 『REST API로 토큰 기반 인증 사용』의 내용을 참조하십시오. LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 REST API에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

## IBM MQ Console 및 REST API에 대한 기본 레지스트리 구성

`mqwebuser.xml` 파일 내에서 기본 레지스트리를 구성할 수 있습니다. xml 파일에 있는 사용자 이름, 비밀번호 및 역할은 IBM MQ Console 및 REST API의 사용자를 인증하고 권한 부여하는 데 사용됩니다.

### 시작하기 전에

- 기본 레지스트리 내에 사용자를 구성할 때 각 사용자에게 역할을 지정해야 합니다. 각 역할은 IBM MQ Console 및 REST API에 액세스하는 데 필요한 다양한 레벨의 권한을 제공하고 허용된 조작이 시도될 때 사용되는 보안 컨텍스트를 판별합니다. 기본 레지스트리를 구성하기 전에 이러한 역할을 이해해야 합니다. 각 역할에 대한 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』의 내용을 참조하십시오.
- 이 태스크를 완료하려면 `mqwebuser.xml` 파일을 편집할 수 있는 충분한 권한이 있는 사용자여야 합니다.
  - z/OS z/OS에서는 `mqwebuser.xml` 파일에 대한 쓰기 액세스 권한이 있어야 합니다.
  - Multi 다른 모든 운영 체제에서는 권한이 있는 사용자여야 합니다.
  - Linux V 9.4.0 mqweb 서버가 독립형 IBM MQ Web Server 설치의 일부인 경우 IBM MQ Web Server 데이터 디렉토리의 `mqwebuser.xml` 파일에 대한 쓰기 액세스 권한이 있어야 합니다.

### 프로시저

1. 다음 경로 중 하나에서 샘플 XML 파일 `basic_registry.xml`을 복사하십시오.
   - IBM MQ 설치의 경우:
     - ALW AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS z/OS: `PathPrefix /web/mq/samp/configuration`

       여기서 `PathPrefix`는 IBM MQ for z/OS UNIX System Services Components 설치 경로입니다.

- Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

  여기서 *MQWEB_INSTALLATION_PATH* 는 IBM MQ Web Server 설치 파일의 압축을 푼 디렉토리입니다.

2. 샘플 파일을 적절한 디렉토리에 배치하십시오.
   - IBM MQ 설치의 경우:
     - Linux AIX AIX 또는 Linux의 경우: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb. 여기서 *MQ_DATA_PATH* 는 IBM MQ 데이터 경로입니다. 이 경로는 IBM MQ설치 중에 선택되는 데이터 경로입니다. 기본적으로 이 경로는 C:\ProgramData\IBM\MQ입니다.
     - z/OS z/OS: *WLP_user_directory*/servers/mqweb

       여기서 *WLP_user_directory*는 mqweb 서버 정의를 작성하기 위해 **crtmqweb** 스크립트를 실행할 때 지정된 디렉토리입니다.
   - Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     여기서 *MQ_OVERRIDE_DATA_PATH* 는 **MQ_OVERRIDE_DATA_PATH** 환경 변수가 가리키는 IBM MQ Web Server 데이터 디렉토리입니다.
3. 옵션: mqwebuser.xml에서 구성 설정을 변경한 경우 이를 샘플 파일에 복사하십시오.
4. 기존 mqwebuser.xml 파일을 삭제하고 샘플 파일 이름을 mqwebuser.xml로 변경하십시오.
5. 새 mqwebuser.xml 파일을 편집하여 **basicRegistry** 태그 내에 사용자 및 그룹을 추가하십시오.

   MQWebUser 역할이 지정된 사용자는 큐 관리자에서 수행하도록 사용자 ID가 제공된 조작만 수행할 수 있습니다. 따라서 레지스트리에 정의된 사용자 ID는 IBM MQ가 설치된 시스템의 사용자 ID와 동일해야 합니다. 이러한 사용자 ID는 동일한 대소문자로 되어 있어야 합니다. 그렇지 않을 경우 사용자 ID 간 맵핑이 실패합니다.

   기본 사용자 레지스트리 구성에 대한 자세한 정보는 WebSphere Liberty 문서에서 Liberty에 대한 기본 사용자 레지스트리 구성 을 참조하십시오.
6. mqwebuser.xml 파일을 편집하여 사용자 및 그룹에 역할을 지정하십시오.

   사용자 및 그룹에 IBM MQ Console 및 REST API를 사용할 수 있는 권한을 부여할 수 있는 몇 가지 역할이 있습니다. 각 역할은 서로 다른 레벨의 액세스 권한을 부여합니다. 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』 의 내용을 참조하십시오.
   - IBM MQ Console에 역할을 지정하고 액세스 권한을 부여하려면 **<enterpriseApplication id="com.ibm.mq.console">** 태그 내의 해당 **security-role** 태그 사이에 사용자 및 그룹을 추가하십시오.
   - REST API에 역할을 지정하고 액세스 권한을 부여하려면 **<enterpriseApplication id="com.ibm.mq.rest">** 태그 내의 해당 **security-role** 태그 사이에 사용자 및 그룹을 추가하십시오.

   **security-role** 태그 내에서 사용자 및 그룹 정보의 형식에 대한 도움말은 예를 참조하십시오.
7. mqwebuser.xml에서 사용자에 대한 비밀번호를 제공한 경우 WebSphere Liberty에서 제공하는 **securityUtility encoding** 명령을 사용하여 이러한 비밀번호를 더 안전하게 인코딩해야 합니다. 자세한 정보는 WebSphere Liberty 제품 문서에서 Liberty:securityUtility 명령 을 참조하십시오.

**예**

다음 예제에서 MQWebAdminGroup 그룹에는 MQWebAdmin역할이 있는 IBM MQ Console 에 대한 액세스 권한이 부여됩니다. 사용자 reader에게는 MQWebAdminRO 역할과 함께 액세스 권한이 부여되고 사용자 guest에게는 MQWebUser 역할과 함께 액세스 권한이 부여됩니다.

```
<enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
            <security-role name="MQWebAdmin">
                <group name="MQWebAdminGroup" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebAdminRO">
                <user name="reader" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebUser">
                <user name="guest" realm="defaultRealm"/>
            </security-role>
        </application-bnd>
    </enterpriseApplication>
```

다음 예에서는 사용자 reader 및 guest에게 IBM MQ Console에 대한 액세스 권한이 부여됩니다. 사용자 user에게는 REST API에 대한 액세스 권한이 부여되고 MQAdmin 그룹 내의 모든 사용자에게는 IBM MQ Console 및 REST API에 대한 액세스 권한이 부여됩니다. mftadmin 사용자에게 MFT 용 REST API 에 대한 액세스 권한이 부여됩니다.

```
<enterpriseApplication id="com.ibm.mq.console">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="reader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="guest" realm="defaultRealm"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>

<enterpriseApplication id="com.ibm.mq.rest">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="user" realm="defaultRealm"/>
        </security-role>
        <security-role name="MFTWebAdmin">
            <user name="mftadmin" realm="defaultRealm"/>
        </security-role>
     </application-bnd>
</enterpriseApplication>
```

## 다음에 수행할 작업

사용자 인증 방법을 선택하십시오.

**IBM MQ Console 인증 옵션**

- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 IBM MQ Console 로그인 화면에서 사용자 ID 및 비밀번호를 입력합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 인증 옵션을 사용하기 위해 추가 구성은 필요하지 않지만 선택적으로 LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 만기 간격 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 IBM MQ Console에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

**REST API 인증 옵션**

- 사용자가 HTTP 기본 인증을 사용하여 인증하게 합니다. 이 경우 사용자 이름 및 비밀번호가 인코딩되지만 암호화되지는 않으며 각각의 REST API 요청과 함께 전송되어 해당 요청에 대해 사용자를 인증하고 사용자에게 권한을 부여합니다. 이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 자세한 정보는 493 페이지의 『REST API로 HTTP 기본 인증 사용』의 내용을 참조하십시오.
- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API `login` 자원에 제공합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 자세한 정보는 494 페이지의 『REST API로 토큰 기반 인증 사용』의 내용을 참조하십시오. LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 REST API에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

# ALW IBM MQ Console 및 REST API에 대한 로컬 OS 레지스트리 구성

`mqwebuser.xml` 파일 내에서 로컬 운영 체제 레지스트리를 구성할 수 있습니다. 로컬 운영 체제의 사용자 이름 및 비밀번호는 IBM MQ Console 및 REST API의 사용자를 인증하고 이 사용자에게 권한을 부여하는 데 사용됩니다.

## 시작하기 전에

- 로컬 OS 인증 기능을 사용하는 클라이언트 인증서 인증의 경우, 사용자 ID는 클라이언트 인증서의 식별 이름(DN)에 있는 공통 이름(CN)입니다. 사용자 ID가 운영 체제 사용자로 존재하지 않는 경우 클라이언트 인증서 로그인이 실패하고 비밀번호 기반 인증으로 대체됩니다.
- 이 태스크를 완료하려면 `mqwebuser.xml` 파일을 편집할 수 있는 충분한 권한이 있는 사용자여야 합니다.
  - Linux V 9.4.0 mqweb 서버가 독립형 IBM MQ Web Server 설치의 일부인 경우 IBM MQ Web Server 데이터 디렉토리의 `mqwebuser.xml` 파일에 대한 쓰기 액세스 권한이 있어야 합니다.
  - mqweb 서버가 IBM MQ 설치의 일부인 경우 권한이 있는 사용자여야 합니다.

## 이 태스크 정보

로컬 운영 체제 레지스트리를 사용하면 사용자 및 그룹에 자동으로 역할이 지정됩니다.

- IBM i에서 'mqm' 그룹 또는 'QMQMADM' 그룹의 일부인 모든 사용자에게 `MQWebAdmin` 및 `MFTWebAdmin` 역할이 부여됩니다.
- 다른 모든 사용자에게는 `MQWebUser` 역할이 부여됩니다.

이러한 역할에 대한 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』의 내용을 참조하십시오.

로컬 운영 체제 레지스트리는 AIX, Linux, and Windows에서만 사용할 수 있습니다. z/OS SAF 레지스트리를 구성하여 z/OS 에서 동등한 기능을 제공합니다. 추가 정보는 484 페이지의 『Configuring a SAF registry for the IBM MQ Console and REST API』의 내용을 참조하십시오.

## 프로시저

1. 다음 경로 중 하나에서 샘플 XML 파일 `local_os_registry.xml` 을 복사하십시오.
   - Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     여기서 *MQWEB_INSTALLATION_PATH* 는 IBM MQ Web Server 설치 파일의 압축을 푼 디렉토리입니다.
   - IBM MQ 설치의 경우: *MQ_INSTALLATION_PATH*`/web/mq/samp/configuration`
2. 다음 디렉토리 중 하나에 샘플 파일을 배치하십시오.

- Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

  여기서 *MQ_OVERRIDE_DATA_PATH* 는 **MQ_OVERRIDE_DATA_PATH** 환경 변수가 가리키는 IBM MQ Web Server 데이터 디렉토리입니다.
- IBM MQ 설치의 경우: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

3. 옵션: mqwebuser.xml에서 구성 설정을 변경한 경우 이를 샘플 파일에 복사하십시오.
4. 기존 mqwebuser.xml 파일을 삭제하고 샘플 파일 이름을 mqwebuser.xml로 변경하십시오.

### 다음에 수행할 작업

사용자 인증 방법을 선택하십시오.

**IBM MQ Console 인증 옵션**

- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 IBM MQ Console 로그인 화면에서 사용자 ID 및 비밀번호를 입력합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 인증 옵션을 사용하기 위해 추가 구성은 필요하지 않지만 선택적으로 LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 만기 간격 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 IBM MQ Console에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

**REST API 인증 옵션**

- 사용자가 HTTP 기본 인증을 사용하여 인증하게 합니다. 이 경우 사용자 이름 및 비밀번호가 인코딩되지만 암호화되지는 않으며 각각의 REST API 요청과 함께 전송되어 해당 요청에 대해 사용자를 인증하고 사용자에게 권한을 부여합니다. 이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 자세한 정보는 493 페이지의 『REST API로 HTTP 기본 인증 사용』의 내용을 참조하십시오.
- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API login 자원에 제공합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 자세한 정보는 494 페이지의 『REST API로 토큰 기반 인증 사용』의 내용을 참조하십시오. LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 REST API에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

## IBM MQ Console 및 REST API에 대한 LDAP 레지스트리 구성

mqwebuser.xml 파일 내에서 LDAP 레지스트리를 구성할 수 있습니다. LDAP 레지스트리에 있는 사용자 이름 및 비밀번호는 IBM MQ Console 및 REST API의 사용자를 인증하고 이 사용자에게 권한을 부여하는 데 사용됩니다.

### 시작하기 전에

- LDAP 레지스트리를 구성할 때 각 사용자에게 역할을 지정해야 합니다. 각 역할은 IBM MQ Console 및 REST API에 액세스하는 데 필요한 다양한 레벨의 권한을 제공하고 허용된 조작이 시도될 때 사용되는 보안 컨텍스트를 판별합니다. 레지스트리를 구성하기 전에 이러한 역할을 이해해야 합니다. 각 역할에 대한 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』의 내용을 참조하십시오.

  MQWebUser 역할이 지정된 사용자는 큐 관리자에서 수행하도록 사용자 ID가 제공된 조작만 수행할 수 있습니다. 따라서 LDAP 서버에 정의된 사용자 ID는 IBM MQ가 설치된 시스템의 사용자 ID와 동일해야 합니다. 이러한 사용자 ID는 동일한 대소문자로 되어 있어야 합니다. 그렇지 않을 경우 사용자 ID 간 맵핑이 실패합니다.
- 이 태스크를 완료하려면 mqwebuser.xml 파일을 편집할 수 있는 충분한 권한이 있는 사용자여야 합니다.

- z/OS z/OS에서는 mqwebuser.xml 파일에 대한 쓰기 액세스 권한이 있어야 합니다.
- Multi 다른 모든 운영 체제에서는 권한이 있는 사용자여야 합니다.
- Linux V 9.4.0 mqweb 서버가 독립형 IBM MQ Web Server 설치의 일부인 경우 IBM MQ Web Server 데이터 디렉토리의 mqwebuser.xml 파일에 대한 쓰기 액세스 권한이 있어야 합니다.

## 프로시저

1. 다음 경로 중 하나에서 샘플 XML 파일 ldap_registry.xml을 복사하십시오.
   - IBM MQ 설치의 경우:
     - ALW AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* /web/mq/samp/configuration
     - z/OS z/OS: PathPrefix /web/mq/samp/configuration

       여기서 PathPrefix는 IBM MQ for z/OS UNIX System Services Components 설치 경로입니다.
   - Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

     여기서 *MQWEB_INSTALLATION_PATH* 는 IBM MQ Web Server 설치 파일의 압축을 푼 디렉토리입니다.
2. 샘플 파일을 적절한 디렉토리에 배치하십시오.
   - IBM MQ 설치의 경우:
     - Linux AIX AIX 또는 Linux의 경우: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb. 여기서 *MQ_DATA_PATH* 는 IBM MQ 데이터 경로입니다. 이 경로는 IBM MQ설치 중에 선택되는 데이터 경로입니다. 기본적으로 이 경로는 C:\ProgramData\IBM\MQ입니다.
     - z/OS z/OS: *WLP_user_directory*/servers/mqweb

       여기서 *WLP_user_directory*는 mqweb 서버 정의를 작성하기 위해 **crtmqweb** 스크립트를 실행할 때 지정된 디렉토리입니다.
   - Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     여기서 *MQ_OVERRIDE_DATA_PATH* 는 **MQ_OVERRIDE_DATA_PATH** 환경 변수가 가리키는 IBM MQ Web Server 데이터 디렉토리입니다.
3. 옵션: mqwebuser.xml에서 구성 설정을 변경한 경우 이를 샘플 파일에 복사하십시오.
4. 기존 mqwebuser.xml 파일을 삭제하고 샘플 파일 이름을 mqwebuser.xml로 변경하십시오.
5. 새 mqwebuser.xml 파일을 편집하여 **ldapRegistry** 및 **idsLdapFilterProperties** 태그 내에서 LDAP 레지스트리 설정을 변경하십시오.

   LDAP 레지스트리 구성에 대한 자세한 정보는 WebSphere Liberty 문서의 Liberty에서 LDAP 사용자 레지스트리 구성 을 참조하십시오.
6. mqwebuser.xml 파일을 편집하여 사용자 및 그룹에 역할을 지정하십시오.

   사용자 및 그룹에 IBM MQ Console 및 REST API를 사용할 수 있는 권한을 부여할 수 있는 몇 가지 역할이 있습니다. 각 역할은 서로 다른 레벨의 액세스 권한을 부여합니다. 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』 의 내용을 참조하십시오.

- IBM MQ Console에 역할을 지정하고 액세스 권한을 부여하려면 **<enterpriseApplication id="com.ibm.mq.console">** 태그 내의 해당 **security-role** 태그 사이에 사용자 및 그룹을 추가하십시오.
- REST API에 역할을 지정하고 액세스 권한을 부여하려면 **<enterpriseApplication id="com.ibm.mq.rest">** 태그 내의 해당 **security-role** 태그 사이에 사용자 및 그룹을 추가하십시오.

## 다음에 수행할 작업

사용자 인증 방법을 선택하십시오.

**IBM MQ Console 인증 옵션**

- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 IBM MQ Console 로그인 화면에서 사용자 ID 및 비밀번호를 입력합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 인증 옵션을 사용하기 위해 추가 구성은 필요하지 않지만 선택적으로 LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 만기 간격 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 IBM MQ Console에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

**REST API 인증 옵션**

- 사용자가 HTTP 기본 인증을 사용하여 인증하게 합니다. 이 경우 사용자 이름 및 비밀번호가 인코딩되지만 암호화되지는 않으며 각각의 REST API 요청과 함께 전송되어 해당 요청에 대해 사용자를 인증하고 사용자에게 권한을 부여합니다. 이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 자세한 정보는 493 페이지의 『REST API로 HTTP 기본 인증 사용』의 내용을 참조하십시오.
- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API `login` 자원에 제공합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 자세한 정보는 494 페이지의 『REST API로 토큰 기반 인증 사용』의 내용을 참조하십시오. LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 REST API에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 490 페이지의 『REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성』의 내용을 참조하십시오.

# z/OS Configuring a SAF registry for the IBM MQ Console and REST API

The System Authorization Facility (SAF) interface allows the mqweb server to call the external security manager for authentication and authorization checking. A user can then log in to the IBM MQ Console and REST API with a z/OS user ID and password.

## Before you begin

- When you configure a SAF registry, you must assign users a role. Each role provides different levels of privilege to access the IBM MQ Console and REST API, and determines the security context that is used when an allowed operation is attempted. You need to understand these roles before you configure the registry. For more information about each of the roles, see "IBM MQ Console 및 REST API의 역할" on page 487.
- You need the WebSphere Liberty Angel process running to use the authorized interface to SAF. See Enabling z/OS authorized services on Liberty for z/OS for more information.
- To complete this task, you must have write access to the `mqwebuser.xml` file, and authority to define security manager profiles.

**Note:** From IBM MQ 9.3.5 for Continuous Delivery and from IBM MQ 9.3.0 Fix Pack 20 for Long Term Support, the sample configuration file `zos_saf_registry.xml` is updated to remove a duplicate safAuthorization entry.

This update fixes an issue where an ICH408I error can occur when the IBM MQ Console on z/OS is upgraded to a level that ships WebSphere Liberty Profile 22.0.0.12 or later: that is, from IBM MQ 9.3.0 Fix Pack 2 for Long Term Support and from IBM MQ 9.3.1 CSU 1 and IBM MQ 9.3.2 for Continuous Delivery. Having more than one safAuthorization statement is not supported and might cause an ICH408I error when users who are not in either MQWebAdmin or MQWebAdminRO roles, in the EBJROLE class, try to access a z/OS queue manager through the IBM MQ Console.

The default for **racRouteLog**, which specifies the types of access attempts to log, is NONE. If you require an additional report or record for security auditing, see SAF Authorization (safAuthorization) for more information.

## About this task

The SAF interface allows the mqweb server to call the external security manager for authentication and authorization checking for both the IBM MQ Console and REST API.

## Procedure

1. Follow the steps in Enabling z/OS authorized services on Liberty for z/OS to give your mqweb server access to use z/OS authorized services.

   Sample JCL for starting the angel process is in USS_ROOT/web/templates/zos/procs/bbgzangl.jcl, where USS_ROOT is the path in z/OS UNIX System Services (z/OS UNIX) where z/OS UNIX components are installed.

   In bbgzangl.jcl, change the SET ROOT statement to point to USS_ROOT/web, for example, /usr/lpp/mqm/V9R2M0/web.

   See Administering Liberty on z/OS for further information on stopping and starting the angel process.
2. Follow the steps in Liberty: Setting up the System Authorization Facility (SAF) unauthenticated user to create the unauthenticated user needed by Liberty.
3. Copy the `zos_saf_registry.xml` file from the following path: PathPrefix /web/mq/samp/configuration where PathPrefix is the z/OS UNIX Components installation path.
4. Place the sample file in the *WLP_user_directory*/servers/mqweb directory, where *WLP_user_directory* is the directory that was specified when the **crtmqweb** script ran to create the mqweb server definition.
5. Optional: If you previously changed any configuration settings in `mqwebuser.xml`, copy them into the sample file.
6. Delete the existing `mqwebuser.xml` file and rename the sample file to `mqwebuser.xml`.
7. Customize the **safCredentials** element in `mqwebuser.xml`.
   a. Set **profilePrefix** to a name that is unique to your Liberty server. If you have more than one mqweb server running on a single system, you will need to choose a different name for each server; for example MQWEB920 and MQWEB915.
   b. Set **unauthenticatedUser** to the name of the unauthenticated user created in step “2” on page 485.
8. Define the mqweb server APPLID to RACF.

   The APPLID resource name is the value you specified in the **profilePrefix** attribute in step “7” on page 485. The following example defines the mqweb server APPLID in RACF:

   ```
   RDEFINE APPL profilePrefix UACC(NONE)
   ```
9. Grant all users, or groups, to be authenticated to the IBM MQ Console or REST API READ access to the mqweb server APPLID in the APPL class.

You must also do this for the unauthenticated user defined in step “2” on page 485. The following example grants a user READ access to the mqweb server APPLID in RACF:

```
PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
```

10. Use the **SETROPTS** RACF command to refresh the in-storage RACLISTed APPL class profiles:

    ```
    SETROPTS RACLIST(APPL) REFRESH
    ```

11. Define the profiles in the EJBROLE class needed to give users access to roles in the IBM MQ Console and REST API.

    The following example defines the profiles in RACF, where **profilePrefix** is the value specified for the **profilePrefix** attribute in step “7” on page 485.

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdminRO UACC(NONE)
```

12. Grant users access to roles in the IBM MQ Console and REST API.

    To do this, give users or groups READ access to one or more of the profiles in the EBJROLE class created in step “11” on page 486. For more information about the roles, see “IBM MQ Console 및 REST API의 역할” on page 487.

    The following example gives a user access to the MQWebAdmin role for the REST API in RACF, where **profilePrefix** is the value specified for the **profilePrefix** attribute in step “7” on page 485.

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

## Results

You have set up SAF authentication for the IBM MQ Console and REST API.

## What to do next

사용자 인증 방법을 선택하십시오.

**IBM MQ Console 인증 옵션**

- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 IBM MQ Console 로그인 화면에서 사용자 ID 및 비밀번호를 입력합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 인증 옵션을 사용하기 위해 추가 구성은 필요하지 않지만 선택적으로 LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 만기 간격 구성을 참조하십시오.
- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 IBM MQ Console에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 “REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성” on page 490의 내용을 참조하십시오.

**REST API 인증 옵션**

- 사용자가 HTTP 기본 인증을 사용하여 인증하게 합니다. 이 경우 사용자 이름 및 비밀번호가 인코딩되지만 암호화되지는 않으며 각각의 REST API 요청과 함께 전송되어 해당 요청에 대해 사용자를 인증하고 사용자에게 권한을 부여합니다. 이 인증에 보안을 설정하려면 보안 연결을 사용해야 합니다. 즉, HTTPS를 사용해야 합니다. 자세한 정보는 “REST API로 HTTP 기본 인증 사용” on page 493의 내용을 참조하십시오.
- 사용자가 토큰 인증을 사용하여 인증하게 합니다. 이 경우 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API `login` 자원에 제공합니다. 사용자가 설정된 시간 동안 로그인되어 권한 부여된 상태를 유지할 수 있게 하는 LTPA 토큰이 생성됩니다. 자세한 정보는 “REST API로 토큰 기반 인증 사용” on page 494의 내용을 참조하십시오. LTPA 토큰의 만기 간격을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.

- 사용자가 클라이언트 인증서를 사용하여 인증하게 합니다. 이 경우 사용자는 사용자 ID 또는 비밀번호를 사용하여 REST API에 로그인하지 않고 클라이언트 인증서를 대신 사용합니다. 자세한 정보는 "REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성" on page 490의 내용을 참조하십시오.

## IBM MQ Console 및 REST API의 역할

IBM MQ Console 또는 REST API을(를) 사용하도록 사용자와 그룹에 권한을 부여하는 경우 사용 가능한 **MQWebAdmin**, **MQWebAdminRO**, **MQWebUser**, **MFTWebAdmin** 및 **MFTWebAdminRO** 역할 중 하나를 사용자와 그룹에 지정해야 합니다. 각 역할은 IBM MQ Console 및 REST API에 액세스하는 데 필요한 다양한 레벨의 권한을 제공하고 허용된 조작이 시도될 때 사용되는 보안 컨텍스트를 판별합니다.

**참고: MQWebUser** 역할을 제외하고 사용자 ID는 대소문자를 구분하지 않습니다. 이 역할에 대한 특정 요구사항은 487 페이지의 『MQWebUser』의 내용을 참조하십시오.

**MQWebAdmin**

이 역할이 지정된 사용자 및 그룹은 모든 관리 조작을 수행할 수 있고 mqweb 서버를 시작하는 데 사용된 운영 체제 사용자 ID의 보안 컨텍스트에서 작업합니다.

이 역할의 사용자 또는 그룹에 다음 REST 서비스에 대한 액세스 권한이 없습니다.

- MFT용 REST API . 이러한 서비스를 사용하려면 사용자 또는 그룹에도 **MFTWebAdmin** 또는 **MFTWebAdminRO** 역할이 지정되어야 합니다.
- messaging REST API입니다. messaging REST API을(를) 사용하려면 사용자에게 **MQWebUser** 역할이 지정되어야 합니다.

**MQWebAdminRO**

이 역할은 IBM MQ Console 또는 REST API에 대한 읽기 전용 액세스 권한만 제공합니다. 이 역할이 지정된 사용자 및 그룹은 다음 조작을 수행할 수 있습니다.

- 큐 및 채널과 같은 IBM MQ 오브젝트에서 조작을 표시 및 조회합니다.
- 큐에서 메시지를 찾아봅니다.

이 역할이 지정된 사용자 또는 그룹은 mqweb 서버를 시작하는 데 사용된 운영 체제 사용자 ID의 보안 컨텍스트에서 작업합니다.

이 역할의 사용자 또는 그룹에 다음 REST 서비스에 대한 액세스 권한이 없습니다.

- MFT용 REST API . 이러한 서비스를 사용하려면 사용자 또는 그룹에도 **MFTWebAdmin** 또는 **MFTWebAdminRO** 역할이 지정되어야 합니다.
- messaging REST API입니다. messaging REST API을(를) 사용하려면 사용자에게 **MQWebUser** 역할이 지정되어야 합니다.

**MQWebUser**

이 역할이 지정된 사용자 또는 그룹은 큐 관리자에서 수행하도록 사용자 ID에 부여된 모든 조작을 수행할 수 있습니다. 예를 들면, 다음과 같습니다.

- 채널과 같은 IBM MQ 오브젝트에서 조작을 시작 및 중지합니다.
- 큐 및 채널 같은 IBM MQ 오브젝트에서 조작을 정의 및 설정합니다.
- 큐 및 채널과 같은 IBM MQ 오브젝트에서 조작을 표시 및 조회합니다.
- messaging REST API를 사용하여 메시지를 넣고 가져옵니다.

이 역할이 지정된 사용자 또는 그룹은 프린시펄의 보안 컨텍스트에서 작업하고, 큐 관리자에서 수행하기 위해 사용자 ID가 제공된 조작만 수행할 수 있습니다.

따라서 mqweb 사용자 레지스트리에서 정의되는 사용자 또는 그룹이 조작을 수행하려면 먼저 해당 사용자 또는 그룹에 IBM MQ 내의 권한을 부여해야 합니다. 이 역할을 사용하면 IBM MQ Console 및 REST API를 사용할 때 특정 IBM MQ 자원에 대한 액세스 유형을 갖는 사용자를 세부적으로 제어할 수 있습니다.

**참고:**

- 이 역할이 지정된 사용자 ID의 최대 길이는 12자입니다.

- 사용자 ID의 대소문자는 mqweb 사용자 레지스트리와 IBM MQ 시스템에서 동일해야 합니다. 사용자 ID의 대소문자가 서로 다르면 사용자는 IBM MQ Console 및 REST API에서 인증될 수 있지만 IBM MQ 자원을 사용하는 권한을 부여받지는 않습니다.

**MFTWebAdmin**

이 역할이 지정된 사용자 또는 그룹은 모든 MFT REST 조작을 수행할 수 있으며 mqweb 서버를 시작하는 데 사용되는 운영 체제 사용자 ID의 보안 컨텍스트에서 작동합니다.

이 역할의 사용자 또는 그룹에 IBM MQ REST API 서비스에 대한 액세스 권한이 없습니다. 이러한 서비스를 사용하려면 사용자 또는 그룹에도 **MQWebAdmin**, **MQWebAdminRO** 또는 **MQWebUser** 역할이 지정되어야 합니다.

**MFTWebAdminRO**

이 역할은 MFT 용 REST API 에 대한 읽기 전용 액세스 권한을 제공합니다. 이 역할이 지정된 사용자 또는 그룹은 전송 나열 또는 에이전트 나열과 같이 읽기 전용 조작(GET 요청)만 수행할 수 있습니다.

이 역할이 지정된 사용자 또는 그룹은 mqweb 서버를 시작하는 데 사용된 운영 체제 사용자 ID의 보안 컨텍스트에서 작업합니다.

이 역할의 사용자 또는 그룹에 IBM MQ REST API 서비스에 대한 액세스 권한이 없습니다. 이러한 서비스를 사용하려면 사용자 또는 그룹에도 **MQWebAdmin**, **MQWebAdminRO** 또는 **MQWebUser** 역할이 지정되어야 합니다.

이 역할을 사용하도록 사용자 및 그룹을 구성하는 자세한 정보는 476 페이지의 『사용자 및 역할 구성』의 내용을 참조하십시오.

### 역할 겹침

사용자 또는 그룹에 두 개 이상의 역할이 지정될 수 있습니다. 이런 상황에서 사용자가 조작을 수행하면 조작에 적용되는 가장 높은 권한 역할이 사용됩니다. 예를 들어, 역할이 **MQWebAdminRO**이고 **MQWebUser**에서 조회 큐 조작을 수행하는 경우 **MQWebAdminRO** 역할이 사용되고 웹 서버를 시작한 시스템 사용자 ID의 컨텍스트에서 조작을 시도합니다. 동일한 사용자가 정의 조작을 수행하는 경우 **MQWebUser** 역할이 사용되고 프린시펄의 컨텍스트에서 조작을 시도합니다.

## ALW IBM MQ Console 에서 제공하는 인증서를 브라우저로 변경

인증을 위해 CA 서명 인증서를 제공하도록 IBM MQ Console 를 구성할 수 있습니다. CA 서명 인증서를 제공하도록 IBM MQ Console 를 구성하면 IBM MQ Console 에 액세스할 때 브라우저가 더 이상 자체 서명 인증서 경고를 표시하지 않습니다.

### 이 태스크 정보

IBM MQ Console 에 대한 보안은 IBM MQ Console를 실행하는 mqweb 서버에서 제공됩니다. mqweb 서버가 브라우저에 제공하는 인증서를 변경하려면 먼저 새 인증서를 mqweb 서버 키 저장소에 추가하십시오. 그런 다음 `mqwebuser.xml` 파일에서 보안 구성을 편집하여 서버가 제공하는 인증서를 지정하십시오.

프로시저는 다음과 같이 가정합니다.

- 권한이 있는 사용자입니다.
- AIX, Linux또는 Windows 시스템을 사용 중입니다.
- `mqwebuser.xml` 파일은 `basic_registry.xml`, `local_os_registry.xml`또는 `ldap_registry.xml` 샘플 XML 파일을 기반으로 합니다.

### 프로시저

1. 옵션: **runmqktool** 명령을 사용하여 mqweb 서버 키 저장소 `key.jks` 의 기본 비밀번호를 변경하십시오.

```
runmqktool -storepasswd -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass oldPassword
          -new newPassword
```

***oldPassword***
기존 key.jks 비밀번호를 지정합니다. 기본 비밀번호는 password입니다.

***newPassword***
새 key.jks 비밀번호를 지정합니다.

2. 인증 기관에 전송할 키 쌍 및 인증서 요청을 작성하십시오.

   a) **runmqktool** 명령을 사용하여 키 쌍을 작성하십시오.

```
runmqktool -genkeypair -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password -storetype JKS
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

   **비밀번호**
   key.jks 키 저장소 비밀번호를 지정합니다.

   **레이블**
   인증서 레이블을 지정합니다. 예를 들어, MQWebConsole입니다.

   ***distinguished_name***
   인증서의 X.500 식별 이름을 지정합니다. 식별 이름을 큰따옴표로 묶으십시오.

   예: "cn=MQWebConsole,o=myOrg,c=UK"

   ***signature_algorithm***
   인증서에 서명하는 데 사용할 알고리즘을 지정합니다. 자세한 정보는 서명 알고리즘 를 참조하십시오.

   b) **runmqktool** 명령을 사용하여 인증서 요청을 작성하십시오.

```
runmqktool -certreq -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password -alias label
           -file filename
```

   **비밀번호**
   key.jks 키 저장소 비밀번호를 지정합니다.

   **레이블**
   하위 단계 489 페이지의 『2.a』에서 인증서 레이블을 지정합니다.

   ***filename***
   인증서 요청에 대한 완전한 파일 이름을 지정합니다.

3. 인증서 요청 파일을 인증 기관(CA)에 전송하십시오.
4. CA의 인증서가 있는 경우 **runmqktool** 명령을 사용하여 인증서 및 루트 CA 인증서로 시작하는 인증서 체인의 다른 인증서를 keys.jks 키 저장소로 가져오십시오.

```
runmqktool -importcert -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password
          -alias label -file filename
```

   **비밀번호**
   key.jks 키 저장소 비밀번호를 지정합니다.

   **레이블**
   하위 단계 489 페이지의 『2.a』에서 인증서 레이블을 지정합니다.

   ***filename***
   가져올 인증서의 완전한 파일 이름을 지정합니다.

5. CA 인증서를 표시하도록 mqweb 서버를 구성하십시오.

   a) mqwebuser.xml 파일을 여십시오.

   mqwebuser.xml 파일은 다음 경로에서 찾을 수 있습니다. *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   b) 다음 행을 주석 처리하여 기본 보안 구성을 해제하십시오.

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

클라이언트 인증서 인증을 사용하도록 mqweb 서버를 구성한 경우 xml 파일의 이 행은 이미 주석 처리되어 있습니다.

c) `mqwebuser.xml` 파일에서 사용자 정의 인증서 구성을 사용으로 설정하는 섹션을 주석 해제하십시오. 섹션에는 다음 텍스트가 포함되어 있습니다.

```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
    <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
    <ssl id="thisSSLConfig" clientAuthenticationSupported="true"
keyStoreRef="defaultKeyStore"
          trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"
serverKeyAlias="default"/>
    <sslDefault sslRef="thisSSLConfig"/>
```

클라이언트 인증서 인증을 사용하도록 mqweb 서버를 구성한 경우 xml 파일의 이 섹션은 이미 주석 해제되어 있습니다.

d) 옵션: 488 페이지의 『1』 단계에서 `key.jks` 키 저장소의 비밀번호를 변경한 경우 `defaultKeyStore` 태그에서 **password** 의 값을 설정한 비밀번호의 인코딩된 버전으로 변경하십시오.

i) *MQ_INSTALLATION_PATH*`/web/bin` 디렉토리에서 다음 명령을 입력하십시오.

```
securityUtility encode password
```

ii) `defaultKeyStore`에 대한 **비밀번호** 필드에 이 명령의 출력을 배치하십시오.

e) 클라이언트 인증서 인증을 사용하지 않는 경우 다음 행을 주석 처리하십시오.

```
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
```

f) **serverKeyAlias** 의 값을 `default` 에서 CA 인증서 레이블의 값으로 변경하십시오.

6. **endmqweb** 명령을 사용하여 mqweb 서버를 중지하십시오.
7. **strmqweb** 명령을 사용하여 mqweb 서버를 시작하십시오.

### 결과

웹 서버가 시작되면 IBM MQ Console로 이동하여 새로 고치십시오. CA 인증서가 사용되고 로그인 페이지로 바로 이동됩니다.

## ALW REST API 및 IBM MQ Console 를 사용하여 클라이언트 인증서 인증 구성

클라이언트 인증서를 프린시펄에 맵핑하여 IBM MQ Console 및 REST API 사용자를 인증할 수 있습니다.

### 시작하기 전에

- IBM MQ Console 및 REST API를 사용할 수 있는 권한이 부여되도록 사용자, 그룹 및 역할을 구성하십시오. 자세한 정보는 476 페이지의 『사용자 및 역할 구성』의 내용을 참조하십시오.
- REST API를 사용하는 경우 `login` 자원에서 HTTP GET 메소드를 사용하여 현재 사용자의 신임 정보를 조회하여 클라이언트 인증서를 제공하여 요청을 인증할 수 있습니다. 이 요청은 사용자 이름 및 사용자에게 지정된 역할에 대한 정보를 리턴합니다. 자세한 정보는 GET /login을 참조하십시오.
- 클라이언트 인증서를 프린시펄에 맵핑하여 사용자를 인증할 경우 클라이언트 인증서의 식별 이름을 사용하여 구성된 사용자 레지스트리의 사용자와 비교합니다.
  - 기본 레지스트리의 경우 공통 이름(CN)을 사용자와 비교합니다. 예를 들어, `CN=Fred, O=IBM, C=GB` 는 `Fred`의 사용자 이름과 일치합니다.
  - LDAP 레지스트리의 경우 기본적으로 전체 식별 이름을 LDAP과 비교합니다. 필터 및 맵핑을 설정하여 일치를 사용자 정의할 수 있습니다. 자세한 정보는 WebSphere Liberty 문서의 Liberty :LDAP 인증서 맵 모드 를 참조하십시오.

## 이 태스크 정보

클라이언트 인증서를 사용하여 사용자를 인증한 경우 사용자 이름 및 비밀번호 대신 인증서가 사용됩니다. REST API의 경우, 클라이언트 인증서가 사용자 인증을 위해 각 REST 요청과 함께 제공됩니다. IBM MQ Console의 경우 사용자가 인증서를 사용하여 로그인하면 사용자가 로그아웃할 수 없습니다.

ALW AIX, Linux또는 Windows 시스템에서 프로시저는 다음 정보를 가정합니다.

- `mqwebuser.xml` 파일은 `basic_registry.xml`, `local_os_registry.xml`또는 `ldap_registry.xml` 샘플 XML 파일을 기반으로 합니다.
- 권한이 있는 사용자입니다.

z/OS z/OS 시스템에서 RACF 키 링을 사용하여 클라이언트 인증서 인증을 구성하려면 502 페이지의 『Configuring TLS for the REST API and IBM MQ Console on z/OS』의 프로시저를 따르십시오.

**참고:** 다음 프로시저에서는 IBM MQ Console 및 REST API와 함께 클라이언트 인증서를 사용하기 위해 필요한 단계에 대해 간략하게 설명합니다. 개발자 편의를 위해 단계에서는 자체 서명 인증서를 작성하고 사용하는 방법을 자세히 설명합니다. 하지만 프로덕션의 경우 인증 기관에서 확보한 인증서를 사용하십시오.

## 프로시저

1. **runmqktool** 명령을 사용하여 인증서를 작성하십시오.

```
runmqktool -genkeypair -keystore filename -storepass password -storetype PKCS12
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

***filename***
키 저장소 이름을 지정합니다 (예: `user.p12`). 키 저장소가 없는 경우 명령이 실행될 때 작성됩니다.

**비밀번호**
키 저장소 비밀번호를 지정합니다.

**레이블**
인증서 레이블을 지정합니다. 예를 들어, `user1`입니다.

***distinguished_name***
인증서의 X.500 식별 이름을 지정합니다. 식별 이름을 큰따옴표로 묶으십시오.

기본 사용자 레지스트리를 사용하는 경우 식별 이름의 공통 이름 (CN) 파트에 사용자 레지스트리의 사용자 이름을 입력하십시오. 예를 들어, `mqadmin`사용자의 경우 식별 이름 `"CN=mqadmin"`를 사용하십시오.

로컬 OS 레지스트리를 사용하는 경우, 식별 이름의 공통 이름 (CN) 파트에 로컬 OS 사용자 ID의 이름을 입력하십시오. 예를 들어, `mqadmin`사용자의 경우 식별 이름 `"CN=mqadmin"`를 사용하십시오.

LDAP 사용자 레지스트리를 사용하는 경우 LDAP 레지스트리의 식별 이름과 일치하는 식별 이름을 입력하십시오.

***signature_algorithm***
인증서에 서명하는 데 사용할 알고리즘을 지정합니다. 자세한 정보는 서명 알고리즘 를 참조하십시오.

2. 옵션: 인증 기관 (CA) 에서 인증서를 확보하십시오. 또는 자체 서명된 인증서를 사용하려면 492 페이지의 『3』 단계를 계속하십시오.

a) 인증 기관에서 인증서를 얻으려면 **runmqktool** 명령을 사용하여 인증서 요청을 작성하십시오.

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

***filename***
491 페이지의 『1』 단계에서 키 저장소 이름을 지정합니다.

**비밀번호**
키 저장소 비밀번호를 지정합니다.

**레이블**
491 페이지의 『1』 단계에서 인증서 레이블을 지정합니다.

***filename***
인증서 요청에 대한 완전한 파일 이름을 지정합니다.

b) 인증서 요청 파일을 인증 기관(CA)에 전송하십시오.

c) CA의 인증서가 있는 경우 **runmqktool** 명령을 사용하여 인증서를 키 저장소로 가져오십시오.

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

***filename***
491 페이지의 『1』 단계에서 키 저장소 이름을 지정합니다.

**비밀번호**
키 저장소 비밀번호를 지정합니다.

**레이블**
491 페이지의 『1』 단계에서 인증서 레이블을 지정합니다.

***filename***
CA 인증서의 완전한 파일 이름을 지정합니다.

3. **runmqktool** 명령을 사용하여 인증서의 공용 파트를 추출하십시오.

```
runmqktool -exportcert -keystore filename -storepass password
           -alias label -file filename -rfc
```

***filename***
491 페이지의 『1』 단계에서 키 저장소 이름을 지정합니다.

**비밀번호**
키 저장소 비밀번호를 지정합니다.

**레이블**
491 페이지의 『1』 단계에서 인증서 레이블을 지정합니다.

***filename***
추출된 인증서의 완전한 파일 이름을 지정합니다.

4. 서버가 **runmqktool** 명령을 사용하여 클라이언트 인증서의 유효성을 검증할 수 있도록 인증서의 공용 파트를 서명자 인증서로 mqweb 서버 신뢰 키 저장소로 가져오십시오.

```
runmqktool -importcert -keystore  MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/trust.jks -storepass password
           -alias label -file filename
```

**비밀번호**
`trust.jks` 키 저장소 비밀번호를 지정합니다. 기존 `trust.jks` 키 저장소의 비밀번호를 지정하거나 새 `trust.jks` 키 저장소의 새 비밀번호를 지정할 수 있습니다.

**레이블**
491 페이지의 『1』 단계에서 인증서 레이블을 지정합니다.

***filename***
추출된 인증서의 완전한 파일 이름을 지정합니다.

5. 클라이언트 인증서 인증을 사용하도록 mqweb 서버를 구성하십시오.

a) `mqwebuser.xml` 파일을 여십시오.

`mqwebuser.xml` 파일은 다음 경로에서 찾을 수 있습니다. *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

b) 다음 행을 주석 처리하여 기본 보안 구성을 해제하십시오.

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

브라우저에 CA 인증서를 제공하도록 mqweb 서버를 구성한 경우 이 행은 이미 주석 처리되어 있습니다.

c) 클라이언트 인증서 인증을 사용으로 설정하는 mqwebuser.xml 파일의 섹션을 주석 해제하십시오. 섹션에는 다음 텍스트가 포함되어 있습니다.

```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
    <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
    <ssl id="thisSSLConfig" clientAuthenticationSupported="true"
keyStoreRef="defaultKeyStore"
          trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"
serverKeyAlias="default"/>
    <sslDefault sslRef="thisSSLConfig"/>
```

브라우저에 CA 인증서를 제공하도록 mqweb 서버를 구성한 경우 이 섹션은 이미 주석 해제되어 있습니다. 그러나 **defaultTrustStore** 행의 주석을 해제해야 할 수도 있습니다.

d) trust.jks 키 저장소에 대한 비밀번호와 일치하도록 defaultTrustStore에 대한 **비밀번호**의 값을 변경하십시오.

i) *MQ_INSTALLATION_PATH*/web/bin 디렉토리에서 다음 명령을 입력하십시오.

```
securityUtility encode password
```

ii) defaultTrustStore에 대한 **비밀번호** 필드에 이 명령의 출력을 배치하십시오.

6. **endmqweb** 명령을 사용하여 mqweb 서버를 중지하십시오.
7. **strmqweb** 명령을 사용하여 mqweb 서버를 시작하십시오.
8. 클라이언트 인증서를 사용하여 인증하십시오.

- IBM MQ Console과 함께 클라이언트 인증서를 사용하려면 IBM MQ Console에 액세스하는 데 사용되는 웹 브라우저에 클라이언트 인증서를 설치하십시오.
- REST API에 대해 클라이언트 인증서를 사용하려면 각각의 REST 요청과 함께 클라이언트 인증서를 제공하십시오. HTTP POST, PATCH 또는 DELETE 메소드를 사용하는 경우에는 사이트 간 요청 위조 공격을 방지하기 위해 클라이언트 인증서와 함께 추가 인증을 제공해야 합니다. 즉, 추가 인증은 요청을 인증하는 데 사용 중인 신임 정보가 신임 정보의 소유자에 의해 사용 중인지 확인하는 데 사용됩니다.

  이 추가 인증은 ibm-mq-rest-csrf-token HTTP 헤더가 있어야 제공됩니다. ibm-mq-csrf-token 헤더의 값을 공백을 포함한 임의의 값으로 설정한 후 요청을 제출하십시오.

**예**

**중요사항:** 예제에서 모든 cURL 구현이 자체 서명된 인증서를 지원하는 것은 아니므로 이를 지원하는 cURL 구현을 사용해야 합니다.

다음 cURL 예는 클라이언트 인증서 인증을 사용하여 큐 관리자 QM1에서 새 큐 Q1를 작성하는 방법을 보여줍니다. 이 cURL 명령의 정확한 구성은 cURL 이 빌드된 라이브러리에 따라 다릅니다. 예제는 OpenSSL에 대해 빌드된 cURL 이 있는 Windows 시스템을 기반으로 합니다.

- 클라이언트 인증서로 인증하고 임의 값으로 ibm-mq-rest-csrf-token HTTP 헤더를 포함하여 큐 자원과 함께 HTTP POST 메소드를 사용하십시오. 이 값은 공백을 포함하여 임의의 값이 될 수 있습니다. --cert-type 플래그는 인증서가 PKCS#12 인증서가 되도록 지정합니다. --cert 플래그는 인증서의 위치, 콜론, 인증서의 비밀번호를 차례로 지정합니다.

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -
-cert-type P12 --cert c:\user.p12:password
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

## REST API로 HTTP 기본 인증 사용

REST API의 사용자는 HTTP 헤더에 사용자 ID 및 비밀번호를 제공하여 인증할 수 있습니다. HTTP 메소드(예: POST, PATCH, DELETE)와 함께 이 인증 방법을 사용하려면 사용자 ID 및 비밀번호와 함께 ibm-mq-rest-csrf-token HTTP 헤더도 제공해야 합니다.

### 시작하기 전에

- REST API를 사용할 수 있는 권한이 부여되도록 사용자, 그룹 및 역할을 구성하십시오. 자세한 정보는 476 페이지의 『사용자 및 역할 구성』의 내용을 참조하십시오.
- HTTP 기본 인증이 사용으로 설정되어 있는지 확인하십시오. `mqwebuser.xml` 파일에 다음 XML이 제공되며 주석 처리되어 있지 않은지 확인하십시오. 이 XML은 `<featureManager>` 태그 내에 있어야 합니다.

```
<feature>basicAuthenticationMQ-1.0</feature>
```

z/OS z/OS에서 이 파일을 편집하려면 `mqwebuser.xml`에 대한 쓰기 액세스 권한이 있는 사용자여야 합니다.

Multi 다른 모든 운영 체제에서 `mqwebuser.xml` 파일을 편집하려면 권한이 있는 사용자여야 합니다.

- REST 요청을 전송할 때 보안 연결을 사용하고 있는지 확인하십시오. 사용자 이름 및 비밀번호 조합은 인코딩되지만 암호화되지는 않으므로 REST API에 대해 HTTP 기본 인증을 사용하는 경우 보안 연결(HTTPS)을 사용해야 합니다.
- `login` 자원에 대해 HTTP GET 메소드를 사용하여 현재 사용자의 신임 정보를 조회하여 기본 인증 정보를 제공하여 요청을 인증할 수 있습니다. 이 요청은 사용자 이름 및 사용자에게 지정된 역할에 대한 정보를 리턴합니다. 자세한 정보는 GET /login을 참조하십시오.

### 프로시저

1. 사용자 이름을 콜론 및 비밀번호와 연결하십시오. 사용자 이름은 대소문자를 구분합니다.
   예를 들어, 사용자 이름 admin과 비밀번호 admin은 다음 문자열이 됩니다.

```
admin:admin
```

2. base64 인코딩으로 사용자 이름 및 비밀번호 문자열을 인코딩하십시오.
3. 이 인코딩된 사용자 이름 및 비밀번호를 HTTP `Authorization: Basic` 헤더에 포함하십시오.
   예를 들어, 인코딩된 사용자 이름이 admin이고 비밀번호가 admin이면 다음 헤더가 작성됩니다.

```
Authorization: Basic YWRtaW46YWRtaW4=
```

4. HTTP POST, PATCH 또는 DELETE 메소드를 사용하는 경우에는 사용자 이름 및 비밀번호 외에 추가 인증을 제공해야 합니다.
   이 추가 인증은 `ibm-mq-rest-csrf-token` HTTP 헤더가 있어야 제공됩니다. `ibm-mq-rest-csrf-token` HTTP 헤더가 요청에 있어야 하지만 해당 값은 공백을 포함한 임의 값이 될 수 있습니다.
5. 적절한 헤더를 사용하여 REST 요청을 IBM MQ에 제출하십시오.

#### 예

다음 예는 Windows 시스템에서 기본 인증을 사용하여 큐 관리자 QM1에 새 큐 Q1를 작성하는 방법을 보여줍니다. 예에서는 cURL을 사용합니다.

- 기본 인증으로 인증하고 임의 값으로 `ibm-mq-rest-csrf-token` HTTP 헤더를 포함하여 큐 자원과 함께 HTTP POST 메소드를 사용하십시오. 이 값은 공백을 포함한 임의의 값이 될 수 있습니다.

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST
-u mqadmin:mqadmin
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

## REST API로 토큰 기반 인증 사용

REST API의 사용자는 HTTP POST 메소드를 사용하여 사용자 ID 및 비밀번호를 REST API `login` 자원에 제공하여 인증할 수 있습니다. 이 사용자가 향후 요청을 인증할 수 있게 하는 LTPA 토큰이 생성됩니다. 이 LTPA 토큰

에는 접두부 LtpaToken2가 있습니다. 이 사용자는 HTTP DELETE 메소드를 사용하여 로그아웃할 수 있고 HTTP GET 메소드를 사용하여 현재 사용자의 로그인 정보를 조회할 수 있습니다.

## 시작하기 전에

- REST API를 사용할 수 있는 권한이 부여되도록 사용자, 그룹 및 역할을 구성하십시오. 자세한 정보는 476 페이지의 『사용자 및 역할 구성』의 내용을 참조하십시오.
- 기본적으로 LTPA 토큰이 포함된 쿠키 이름은 LtpaToken2로 시작되며, mqweb 서버가 재시작될 때 변경될 수 있는 접미부를 포함합니다. 이 랜덤화된 쿠키 이름으로 동일한 시스템에서 둘 이상의 mqweb 서버를 실행할 수 있습니다. 하지만 쿠키 이름을 일관된 값으로 유지하려면 **setmqweb** 명령을 사용하여 쿠키의 이름을 지정할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- 기본적으로 LTPA 토큰 쿠키는 120분 후에 만료됩니다. **setmqweb** 명령을 사용하여 LTPA 토큰 쿠키의 만기 시간을 구성할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- REST 요청을 전송할 때 보안 연결을 사용하고 있는지 확인하십시오. login 자원에서 HTTP POST 메소드를 사용하는 경우 요청과 함께 전송되는 사용자 이름 및 비밀번호 조합은 암호화되지 않습니다. 따라서 REST API에 대해 토큰 기반 인증을 사용하는 경우 보안 연결(HTTPS)을 사용해야 합니다. 기본적으로 LTPA 토큰 인증으로 HTTP를 사용할 수 없습니다. **secureLTPA**를 False로 설정하여 비보안 HTTP 연결에서 사용할 LTPA 토큰을 사용으로 설정할 수 있습니다. 자세한 정보는 LTPA 토큰 구성을 참조하십시오.
- login 자원에서 HTTP GET 메소드를 사용하여 현재 사용자의 신임 정보를 조회하며 LTPA 토큰을 제공하여 요청을 인증할 수 있습니다. 이 요청은 사용자 이름 및 사용자에게 지정된 역할에 대한 정보를 리턴합니다. 자세한 정보는 GET /login을 참조하십시오.

## 프로시저

1. 사용자로 로그인하십시오.
   a) login 자원에서 HTTP POST 메소드를 사용하십시오.

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   다음 형식으로 사용자 이름 및 비밀번호를 JSON 요청의 본문에 포함하십시오.

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) 요청에서 리턴되는 LTPA 토큰을 로컬 쿠키 저장소에 저장하십시오. 기본적으로 이 LTPA 토큰에 LtpaToken2 접두부가 있습니다.
2. 모든 요청의 쿠키로서 저장된 LTPA 토큰을 사용하여 REST 요청을 인증하십시오.

   HTTP PUT, PATCH 또는 DELETE 메소드를 사용하는 요청의 경우 ibm-mq-rest-csrf-token 헤더를 포함하십시오. 이 헤더 값은 공백을 포함하여 임의의 값이 될 수 있습니다.
3. 사용자로 로그아웃하십시오.
   a) login 자원에서 HTTP DELETE 메소드를 사용하십시오.

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

   LTPA 토큰을 쿠키로서 제공하여 요청을 인증하고 ibm-mq-rest-csrf-token 헤더를 포함해야 합니다. 이 헤더 값은 공백을 포함하여 임의의 값이 될 수 있습니다.

   b) 지시사항을 처리하여 로컬 쿠키 저장소에서 LTPA 토큰을 삭제하십시오.

   **참고:** 지시사항이 처리되지 않고 LTPA 토큰이 로컬 쿠키 저장소에 남는 경우 LTPA 토큰은 이후 REST 요청을 인증하는 데 사용할 수 있습니다. 즉, 사용자가 세션이 종료된 후에 LTPA 토큰으로 인증하려고 시도하는 경우 기존 토큰을 사용하는 새 세션이 작성됩니다.

**예**

다음 cURL 예는 Windows 시스템에서 토큰 기반 인증을 사용하여 큐 관리자 QM1에 새 큐 Q1를 작성하는 방법을 보여줍니다.

- 로그인하여 접두부가 LtpaToken2인 LTPA 토큰을 로컬 쿠키 저장소에 추가하십시오. 사용자 이름 및 비밀번호 정보가 JSON 본문에 포함됩니다. -c 플래그는 토큰을 저장할 파일의 위치를 지정합니다.

```
curl -k https://localhost:9443/ibmmq/rest/v1/login -X POST
-H "Content-Type: application/json" --data
"{\"username\":\"mqadmin\",\"password\":\"mqadmin\"}"
-c c:\cookiejar.txt
```

- 큐를 작성하십시오. 큐 자원과 함께 HTTP POST 메소드를 사용하여 LTPA 토큰에 대해 인증하십시오. 접두부가 LtpaToken2인 LTPA 토큰은 -b 플래그를 사용하여 cookiejar.txt 파일에서 검색됩니다. CSRF 보호는 ibm-mq-rest-csrf-token HTTP 헤더가 있어야 제공됩니다.

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -b
c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json"
--data "{\"name\":\"Q1\"}"
```

- 로그아웃한 후 로컬 쿠키 저장소에서 LTPA 토큰을 삭제하십시오. LTPA 토큰은 -b 플래그를 사용하여 cookiejar.txt 파일에서 검색됩니다. CSRF 보호는 ibm-mq-rest-csrf-token HTTP 헤더의 존재에 의해 제공됩니다. cookiejar.txt 파일의 위치는 LTPA 토큰이 파일에서 삭제되도록 -c 플래그에 의해 지정됩니다.

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X DELETE
-H "ibm-mq-rest-csrf-token: value" -b c:\cookiejar.txt
-c c:\cookiejar.txt
```

**관련 참조**

POST/login

GET/login

/login 삭제

# IFrame에 IBM MQ Console 임베드

HTML <iframe> 요소는 IFrame(Inline Frame)을 사용하여 한 웹 페이지를 다른 페이지에 임베드할 때 사용할 수 있습니다. 보안 상의 이유로 IBM MQ Console은 기본적으로 IFrame에 임베드할 수 없습니다. 그러나 mqweb 서버에서 **mqConsoleFrameAncestors** 구성 특성을 사용하면 IFrame을 사용으로 설정할 수 있습니다.

## 이 태스크 정보

mqweb 서버는 IFrame을 사용하여 IBM MQ Console을 임베드할 수 있는 웹 페이지의 원본 허용 목록을 유지보수합니다. 원본은 URL 설계, 도메인 및 포트의 결합(예: https://example.com:1234)입니다.

mqweb 서버에서 **mqConsoleFrameAncestors** 구성 특성을 사용하여 목록에서 항목을 지정할 수 있습니다.

기본적으로 **mqConsoleFrameAncestors**는 공백이며, 이는 IBM MQ Console을 IFrame에 임베드할 수 없음을 의미합니다.

## 프로시저

다음 명령을 입력하여 IFrame에 IBM MQ Console을 임베드할 수 있는 웹 페이지의 원본 목록을 지정하십시오.

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

여기서 *allowedOrigins*은 쉼표로 구분되는 원본 목록입니다. 각 원본은 다음으로 구성되어야 합니다.

- 호스트 이름 또는 IP 주소
- 선택적 URL 설계
- 선택적 포트 번호

호스트 이름은 와일드카드 문자(*)로 시작할 수 있고 포트 번호에서도 와일드카드 문자(*)를 사용할 수 있습니다.
예제 원본은 다음과 같습니다.

```
https://example.com:1234
```

여기에서는 `https://example.com:1234`에서 제공되는 웹 페이지에서 IBM MQ Console을 IFrame에 임베드할 수 있습니다.

```
https://*.example.com:*
```

여기에서는 호스트 이름이 `example.com`으로 끝나고 포트를 사용하는 HTTPS 웹 페이지가 IFrame에 IBM MQ Console을 임베드할 수 있습니다.

#### 예

다음 예제에서는 IBM MQ Console이 `https://site2.example.com:1234` 또는 `https://site2.example.com:1235`에서 제공되는 웹 페이지에서 IFrame에 임베드될 수 있습니다.

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

## REST API에 대해 CORS 구성

기본적으로 웹 브라우저는 스크립트(예: JavaScript)가 REST API와 동일한 원본에서 제공되지 않는 경우 스크립트가 REST API를 호출하도록 허용하지 않습니다. 즉, 원본 간 요청이 사용으로 설정되지 않습니다. 지정된 원본으로부터의 원본 간 요청을 허용하도록 CORS(Cross Origin Resource Sharing)를 구성할 수 있습니다.

### 이 태스크 정보

웹 브라우저를 통해(예: 스크립트를 통해) REST API에 액세스할 수 있습니다. 이 요청은 다른 원본으로부터 REST API에 제공되는 원본 간 요청이므로 웹 브라우저는 이 요청을 거부합니다. 도메인, 포트 또는 설계가 동일하지 않은 경우에는 원본이 다릅니다.

예를 들어, `http://localhost:1999/`에서 호스팅되는 스크립트가 있으면 `https://localhost:9443/`에서 호스팅되는 웹 사이트에서 HTTP GET을 발행하는 경우 원본 간 요청을 작성합니다. 포트 번호와 설계(HTTP)가 다르기 때문에 이 요청은 원본 간 요청입니다.

CORS를 구성하고 관리 REST API에 액세스하도록 허용되는 원본을 지정하여 원본 간 요청을 사용으로 설정할 수 있습니다.

CORS에 대한 자세한 정보는 https://www.w3.org/TR/cors/ 및 https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS의 내용을 참조하십시오.

### 프로시저

1. 다음 명령을 입력하여 현재 구성을 보십시오.

   ```
   dspmqweb properties -a
   ```

   `mqRestCorsAllowedOrigins` 입력 항목은 허용되는 원본을 지정합니다.
   `mqRestCorsMaxAgeInSeconds` 입력 항목은 웹 브라우저가 CORS 프리플라이트(pre-flight) 검사의 결과를 캐시할 수 있는 시간(초)을 지정합니다.
2. 다음 명령을 입력하여 REST API에 액세스할 수 있는 원본을 지정하십시오.

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   여기서 *allowedOrigins*는 원본 간 요청을 허용할 원본을 지정합니다. 별표를 큰따옴표로 묶어 모든 원본 간 요청을 허용할 수 있습니다. 둘 이상의 원본을 큰따옴표로 묶어 쉼표로 구분된 목록으로 입력할 수 있습니다. 원본 간 요청을 허용하지 않으려면 *allowedOrigins*의 값으로 비어 있는 따옴표를 입력하십시오.

3. 다음 명령을 입력하여 웹 브라우저에서 CORS 프리플라이트(pre-flight) 검사의 결과를 캐시에 저장할 수 있게 하려는 시간(초)을 지정하십시오.

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

**예**

다음 예에서는 `http://localhost:9883`, `https://localhost:1999` 및 `https://localhost:9663`에 대해 사용으로 설정된 원본 간 요청을 보여줍니다. CORS 프리플라이트(pre-flight) 검사의 캐시에 저장된 결과의 최대 유효 기간은 90초로 설정됩니다.

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

# IBM MQ Console 및 REST API에 대한 호스트 헤더 유효성 검증 구성

지정된 허용 목록과 일치하는 호스트 헤더와 함께 송신되는 요청만 처리되도록 IBM MQ Console 및 REST API에 대한 액세스를 제한하도록 mqweb 서버를 구성할 수 있습니다. 허용 목록에 없는 호스트 헤더 값이 사용되는 경우 오류가 리턴됩니다.

## 이 태스크 정보

mqweb 서버는 허용되는 호스트 헤더의 허용 목록을 정의하는 데 가상 호스트를 사용합니다. 가상 호스트에 관한 자세한 정보는 WebSphere Liberty 문서(https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html)를 참조하십시오.

이 태스크를 완료하려면 `mqwebuser.xml` 파일을 편집할 수 있는 충분한 권한이 있는 사용자여야 합니다.

- z/OS z/OS에서는 `mqwebuser.xml` 파일에 대한 쓰기 액세스 권한이 있어야 합니다.
- Multi 다른 모든 운영 체제에서는 권한이 있는 사용자여야 합니다.
- Linux V 9.4.0 mqweb 서버가 독립형 IBM MQ Web Server 설치의 일부인 경우 IBM MQ Web Server 데이터 디렉토리의 `mqwebuser.xml` 파일에 대한 쓰기 액세스 권한이 있어야 합니다.

## 프로시저

1. `mqwebuser.xml` 파일을 여십시오. 이 파일은 다음 위치 중 하나에 있습니다.
   - IBM MQ 설치의 경우:
     - Linux AIX AIX 또는 Linux의 경우: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`. 여기서 *MQ_DATA_PATH* 는 IBM MQ 데이터 경로입니다. 이 경로는 IBM MQ설치 중에 선택되는 데이터 경로입니다. 기본적으로 이 경로는 `C:\ProgramData\IBM\MQ`입니다.
     - z/OS z/OS: *WLP_user_directory*`/servers/mqweb`

       여기서 *WLP_user_directory* 는 mqweb 서버 정의를 작성하기 위해 **crtmqweb** 명령을 실행할 때 지정된 디렉토리입니다.
   - Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     여기서 *MQ_OVERRIDE_DATA_PATH* 는 **MQ_OVERRIDE_DATA_PATH** 환경 변수가 가리키는 IBM MQ Web Server 데이터 디렉토리입니다.
2. `mqwebuser.xml` 파일에서 다음 코드를 추가하거나 주석 해제하십시오.

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. 허용하려는 호스트 이름 및 포트 조합을 삽입하여 **<hostAlias>** 필드를 편집하십시오.

   이 조합은 mqweb 서버의 구성에서 사용한 호스트 이름 및 포트 조합일 수 있습니다. 예를 들어, `localhost:9443`의 기본 구성을 사용하는 경우 **<hostAlias>** 필드에서 `localhost:9443`을(를) 사용할 수 있습니다.

   필요한 경우 **<virtualHost>** 태그에 여러 **<hostAlias>** 필드를 추가하여 더 많은 호스트 이름 및 포트 조합을 허용할 수 있습니다. 예를 들면, HTTPS 포트를 사용하는 호스트 헤더 외에 HTTP 포트를 사용하는 호스트 헤더 또한 허용하려 하는 경우가 있습니다.

# 감사

IBM MQ Console 및 REST API 에서 수행되는 조작의 감사 레코드는 큐 관리자 명령 및 구성 이벤트를 사용하여 생성할 수 있으며 AIX, Linux, and Windows 에서 중요한 상태 변경사항은 mqweb 서버의 로그 파일에 기록됩니다.

## 중요한 상태 변경

ALW

AIX, Linux, and Windows에서 IBM MQ Console은 중요한 상태 변경을 기록합니다. 각 메시지는 조작을 요청한 인증된 프린시펄 이름을 나타냅니다.

큐 관리자 작성, 시작, 종료 및 삭제 같은 중요한 상태 변경은 mqweb 서버 `messages.log` 및 `console.log` 파일에 로깅됩니다. 각 메시지는 조작을 요청한 인증된 프린시펄 이름을 나타냅니다.

`messages.log` 및 `console.log` 파일은 다음 위치에 있습니다.

- IBM MQ 설치의 경우:
  - Linux AIX AIX 또는 Linux의 경우: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`
  - Windows Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb\logs`. 여기서 *MQ_DATA_PATH* 는 IBM MQ 데이터 경로입니다. 이 경로는 IBM MQ설치 중에 선택되는 데이터 경로입니다. 기본적으로 이 경로는 `C:\ProgramData\IBM\MQ`입니다.
- Linux V 9.4.0 독립형 IBM MQ Web Server 설치의 경우: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb/logs`

  여기서 *MQ_OVERRIDE_DATA_PATH* 는 **MQ_OVERRIDE_DATA_PATH** 환경 변수가 가리키는 IBM MQ Web Server 데이터 디렉토리입니다.

mqweb 서버 로깅 레벨 구성에 대한 자세한 정보는 로깅 구성을 참조하십시오.

## 명령 및 구성 이벤트

선택적으로 큐 관리자에서 명령 및 구성 이벤트를 활성화하여 대부분의 IBM MQ Console 및 REST API 활동에 대한 정보를 제공할 수 있습니다. 예를 들어, 채널 작성 및 큐 조회는 명령 및 구성 이벤트를 생성합니다. 명령 및 구성 이벤트 활성화에 대한 자세한 정보는 구성, 명령 및 로거 이벤트 제어를 참조하십시오.

이러한 명령 및 구성 이벤트 메시지의 경우 **MQIACF_EVENT_ORIGIN** 필드는 `MQEVO_REST` 로 설정되고 **MQCACF_EVENT_APPL_IDENTITY** 필드는 인증된 프린시펄 이름의 처음 32자를 보고합니다. 사용자에게 `MQWebAdmin` 또는 `MQWebAdminRO` 역할이 있는 경우 **MQCACF_EVENT_USER_ID** 필드는 명령을 실행한 프린시펄의 사용자 이름이 아닌 mqweb 서버 사용자 ID를 보고합니다. 그러나 사용자에게 `MQWebUser` 역할이 있는 경우 **MQCACF_EVENT_USER_ID** 는 명령을 실행한 프린시펄의 사용자 이름을 보고합니다.

**관련 개념**
445 페이지의 『감사』
이벤트 메시지를 사용하여 보안 침입 또는 침입 시도를 확인할 수 있습니다. IBM MQ Explorer를 사용하여 시스템의 보안을 확인할 수도 있습니다.

## z/OS z/OS 의 IBM MQ Console 및 REST API 에 대한 보안 고려사항

IBM MQ Console 및 REST API에는 사용자가 명령을 실행하거나 표시하거나 변경할 수 있는지를 제어하는 보안 기능이 있습니다. 그런 다음 명령이 큐 관리자에게 전달되고 큐 관리자 보안을 사용하여 사용자가 해당 특정 큐 관리자에 대해 명령을 실행할 수 있는지를 제어합니다.

### 프로시저

1. mqweb 서버의 시작된 태스크 사용자 ID에 특정 PCF 명령을 실행하고 특정 큐에 액세스하는 권한이 있는지 확인하십시오. 자세한 정보는 500 페이지의 『Authority required by the mqweb server started task user ID』의 내용을 참조하십시오.
2. `MQWebUser` 역할이 부여된 사용자에게 적절한 권한이 있는지 확인하십시오.

   `MQWebUser` 역할에 지정된 IBM MQ Console 및 REST API 사용자는 프린시펄의 보안 컨텍스트에서 작동합니다. 이 사용자 ID는 큐 관리자에 대해 사용자 ID에 수행 권한이 부여된 조작만 수행할 수 있으며 mqweb 서버 주소 공간과 동일한 시스템 큐에 대한 권한이 부여되어야 합니다.

   mqweb 서버의 시작된 태스크 사용자 ID에는 `MQWebUser` 역할에 지정된 모든 사용자에 대한 대체 사용자 액세스 권한이 부여되어야 합니다.

   `MQWebUser` 역할을 가진 사용자에게 적절한 권한을 부여하는 방법에 대한 자세한 정보는 501 페이지의 『IBM MQ Console 또는 REST API 를 사용하는 데 필요한 IBM MQ 자원에 대한 액세스』의 내용을 참조하십시오.
3. 옵션: IBM MQ Console 및 REST API에 대한 TLS를 구성하십시오. 자세한 정보는 502 페이지의 『Configuring TLS for the REST API and IBM MQ Console on z/OS』의 내용을 참조하십시오.

## z/OS Authority required by the mqweb server started task user ID

On z/OS, the mqweb server started task user ID requires certain authorities to issue PCF commands and access system resources.

The mqweb server started task user ID needs:

- A z/OS UNIX user identifier (UID) to be able to use z/OS UNIX System Services.
- Access to the `hlq.SCSQAUTH` and `hlq.SCSQANL*` data sets in the IBM MQ installation.
- Read access to the IBM MQ installation files in z/OS UNIX System Services.
- Read and write access to the Liberty user directory created by the **crtmqweb** script.
- Authority to connect to the queue manager. Grant the mqweb server started task user ID *READ* access to the `hlq.BATCH` profile in the MQCONN class.
- Authority to issue IBM MQ commands and access certain queues. These details are described in "IBM MQ Console - required command security profiles" on page 215, "System queue security" on page 194, and "Profiles for context security" on page 203.
- Authority to subscribe to the `SYSTEM.FTE` topic, in order to use the REST API for MFT. Grant the mqweb server started task user ID *ALTER* access to the `hlq.SUBSCRIBE.SYSTEM.FTE` profile in the MXTOPIC class.
- If you are are configuring a SAF registry, access to various security profiles. See "Configuring a SAF registry for the IBM MQ Console and REST API" on page 484 for more information.

### Connection authentication

If your queue manager has been configured to require that all batch applications provide a valid user ID and password, by setting CHKLOCL(REQUIRED), you must give the mqweb server started task user ID *UPDATE* access to the `hlq.BATCH` profile in the MQCONN class.

This authority causes connection authentication to operate in CHKLOCL(OPTIONAL) mode for the mqweb server started task user ID.

If you have not configured the queue manager to require that all batch applications provide a valid user ID and password, it is sufficient to give the user ID that starts the mqweb server task *READ* access to the `hlq.BATCH` profile in the MQCONN class.

For more information about CHCKLOCL, see "Using CHCKLOCL on locally bound applications" on page 184.

## z/OS IBM MQ Console 또는 REST API 를 사용하는 데 필요한 IBM MQ 자원에 대한 액세스

`MQWebUser` 역할의 사용자가 IBM MQ Console또는 REST API에서 수행하는 조작은 사용자의 보안 컨텍스트에서 발생합니다.

### 이 태스크 정보

IBM MQ Console 및 REST API의 역할에 대한 자세한 정보는 487 페이지의 『IBM MQ Console 및 REST API의 역할』 의 내용을 참조하십시오.

다음 프로시저를 사용하여 `MQWebUser` 역할의 사용자에게 IBM MQ Console 또는 REST API를 사용하는 데 필요한 큐 관리자 자원에 대한 액세스 권한을 부여하십시오.

### 프로시저

1. `mqweb server started task` 사용자 ID에 `MQWebUser` 역할의 각 사용자 ID에 대한 대체 사용자 액세스 권한을 부여하십시오.

   관리자가 IBM MQ Console 또는 REST API를 통해 관리할 모든 큐 관리자에서 이를 수행하십시오.

   다음 샘플 RACF 명령을 사용하여 `mqweb server started task` 사용자 ID에 `MQWebUser` 역할의 사용자에 대한 대체 사용자 액세스 권한을 부여할 수 있습니다.

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   설명:

   **hlq**
   프로파일 접두부이며 큐 관리자 이름 또는 큐 공유 그룹 이름이 될 수 있습니다.

   **userId**
   `MQWebUser` 역할의 사용자입니다.

   **mqwebUserId**
   `mqweb server started task` 사용자 ID입니다.

   **참고:** 대소문자 혼용 보안을 사용하는 경우, MQADMIN 클래스가 아닌 MXADMIN 클래스를 사용하십시오.
2. `MQWebUser` 역할의 각 사용자에게 IBM MQ Console 및 REST API 사용에 필요한 시스템 큐에 대한 액세스 권한을 부여하십시오.

   이를 수행하려면 SYSTEM.ADMIN.COMMAND.QUEUE 및 SYSTEM.REST.REPLY.QUEUE 둘 다에 대해, 각 사용자에게 대소문자 혼합 보안을 사용 중인지 여부에 따라 MQQUEUE 또는 MXQUEUE 클래스에 대한 UPDATE 액세스 권한을 제공하십시오.

   administrative REST API 게이트웨이를 통해 관리되는 리모트 큐 관리자를 포함하여 REST API를 통해 사용자가 관리할 모든 큐 관리자에서 이를 수행해야 합니다.

3. MQWebUser 역할의 사용자가 리모트 큐 관리자를 관리할 수 있게 하려면 리모트 큐 관리자에게 명령을 전송하는 데 사용하는 전송 큐를 보호하여 MQQUEUE 또는 MXQUEUE 클래스의 프로파일에 대한 UPDATE 액세스 권한을 사용자에게 부여하십시오. 참고로 사용자에게 게이트웨이 큐 관리자에서 UPDATE 액세스 권한을 부여해야 합니다.

   리모트 큐 관리자에서는 게이트웨이 큐 관리자로 명령 응답 메시지를 다시 전송하는 데 사용한 전송 큐에 넣을 수 있는 액세스 권한을 동일한 사용자에게 부여하십시오.
4. MQWebUser 역할의 사용자에게 IBM MQ Console 및 REST API에서 지원하는 조작을 수행하는 데 필요한 기타 자원에 대한 액세스 권한을 부여하십시오.

   액세스 권한은 다음을 수행해야 합니다.

   - REST API의 조작 수행은 개별 REST API 자원 의 보안 요구사항 절에 설명되어 있습니다.
   - IBM MQ Console로 명령 실행. 215 페이지의 『IBM MQ Console - required command security profiles』 에 설명되어 있습니다.

## z/OS Configuring TLS for the REST API and IBM MQ Console on z/OS

On z/OS, you can configure the mqweb server to use a RACF key ring to store certificates for secure connections with TLS, and client certificate authentication.

### Before you begin

You must be a user that has write access to the mqwebuser.xml file, and authority to work with SAF key rings, to complete this procedure.

### About this task

The default mqweb server configuration uses Java keystores for the server and trusted certificates. On z/OS, you can configure the mqweb server to use a RACF key ring, instead of the Java keystores. The server can also be configured to allow users to authenticate using a client certificate.

See Liberty: Keystores for information on using RACF key rings in Liberty.

Follow this procedure to configure the mqweb server to use a RACF key ring, and optionally configure client certificate authentication. This procedure describes the steps necessary to create and use certificates signed with your own certificate authority (CA) certificates. For production, you might prefer to use certificates obtained from an external certificate authority.

### Procedure

1. Create a certificate authority (CA) certificate, which will be used to sign the server certificate. For example, enter the following RACF command:

   ```
   RACDCERT GENCERT -
     CERTAUTH -
     SUBJECTSDN(CN('mqweb Certification Authority') -
       O('IBM') -
       OU('MQ')) -
     SIZE(2048) -
     WITHLABEL('mqwebCertauth')
   ```
2. Create a server certificate, signed with the CA certificate created in step 1, by entering the following command:

   ```
   RACDCERT ID(mqwebUserId) GENCERT -
     SUBJECTSDN(CN('hostname') -
      O('IBM') -
      OU('MQ')) -
      SIZE(2048) -
      SIGNWITH (CERTAUTH LABEL('mqwebCertauth')) -
     WITHLABEL('mqwebServerCert')
   ```

   where *mqwebUserId* is the mqweb server started task user ID, and *hostname* is the host name of the mqweb server.

3. Connect the CA certificate and server certificate to a SAF key ring by entering the following commands:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

   where *mqwebUserId* is the mqweb server started task user ID, and *keyring* is the name of the key ring you want to use.
4. Export the CA certificate to a CER file by entering the following command:

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth')) -
        DSN('hlq.CERT.MQWEBCA') -
        FORMAT(CERTDER) -
        PASSWORD('password')
```

5. FTP the exported CA certificate in binary to your workstation, and import it into your browser as a certificate authority certificate.
6. Optional: If you want to configure client certificate authentication, create and export a client certificate.
   a) Create a certificate authority (CA) certificate, which will be used to sign the client certificate. For example, enter the following RACF command:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb User CA') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
  WITHLABEL('mqwebUserCertauth')
```

   b) Connect the CA certificate to a SAF key ring by entering the following command:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

   where *mqwebUserId* is the mqweb server started task user ID, and *keyring* is the name of the key ring you want to use.
   c) Create a client certificate, signed with the CA certificate. For example, enter the following command:

```
RACDCERT ID(clientUserId) GENCERT -
  SUBJECTSDN(CN('clientUserId') -
     O('IBM') -
     OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth')) -
  WITHLABEL('userCertLabel')
```

   where *clientUserId* is the user name.

   The method used to map a certificate to a principal depends on the type of user registry configured:

   - If you are using a basic registry, the Common Name field in the certificate is matched against the user in the registry.
   - If you are using a SAF registry, and the certificate is in the RACF database, the certificate owner, specified with the **ID** parameter when creating the certificate, is used.
   - If you are using an LDAP registry, the full distinguished name in the certificate is matched against the LDAP registry.

   d) Export the client certificate to a PKCS #12 file by entering the following command:

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) -
         PASSWORD('password') DSN('hlq.USER.CERT')
```

   e) FTP the exported certificate in binary to your workstation. To use the client certificate with the IBM MQ Console, import it into the web browser used to access the IBM MQ Console as a personal certificate.

7. Edit the file *WLP_user_directory*/servers/mqweb/mqwebuser.xml, where *WLP_user_directory* is the directory that was specified when the **crtmqweb** script ran to create the mqweb server definition.

   Make the following changes to configure the mqweb server to use a RACF key ring:

   a) Remove, or comment out, the following line:

   ```
   <sslDefault sslRef="mqDefaultSSLConfig"/>
   ```

   b) Add the following statements:

   ```
   <keyStore id="defaultKeyStore" filebased="false"
             location="safkeyring://mqwebUserId/keyring"
             password="password" readOnly="true" type="JCERACFKS" />
   <ssl id="thisSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1.2"
             serverKeyAlias="mqwebServerCert" clientAuthenticationSupported="true" />
   <sslDefault sslRef="thisSSLConfig"/>
   ```

   where:

   - *mqwebUserId* is the mqweb server started task user ID.
   - *keyring* is the name of the RACF key ring.
   - *mqwebServerCert* is the label of the mqweb server certificate.

   **Notes:** The value of **keyStore password** is ignored.

8. Restart the mqweb server by stopping and restarting the mqweb server started task.
9. Optional: Use the client certificate to authenticate:

   - To use the client certificate with the IBM MQ Console, enter the URL for the IBM MQ Console in the web browser where you installed the client certificate.
   - To use the client certificate with the REST API, provide the client certificate with each REST request.

   **Notes:**

   a. If you are using only certificates to authenticate to the IBM MQ Console, the browser might display a list of certificates for you to select from.

   b. If you want to use a different certificate you might need to close and restart your browser.

   c. If you are using client certificates that are not in the RACF database, you can use RACF certificate name filtering, to map certificate attributes to a user ID. For example:

   ```
   RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
   ```

   maps certificates with a subject distinguished name containing OU=DEPT1 and C=US to user ID DEPT3USR.

### Results

You have set up a TLS interface for the IBM MQ Console and REST API.

## ALW AIX, Linux, and Windows에서 키 및 인증서 관리

AIX, Linux, and Windows에서 **runmqakm** 및 V 9.4.0 V 9.4.0 **runmqktool** 명령을 사용하여 키, 인증서 및 인증서 요청을 관리하십시오.

### 이 태스크 정보

**runmqakm** 명령은 **gskitcapicmd**의 기능과 유사한 기능을 제공합니다. V 9.4.0 V 9.4.0 **runmqktool** 명령은 Java **keytool** 인증서 관리 유틸리티의 기능과 유사한 기능을 제공합니다. **runmqakm** 또는 **runmqktool** 명령을 사용하기 전에 **setmqenv** 명령을 실행하여 시스템 환경 변수가 올바르게 구성되었는지 확인하십시오.

**runmqktool** 명령을 사용하려면 IBM MQ JRE 컴포넌트가 설치되어 있어야 합니다. 이 구성요소가 설치되지 않은 경우 대신 **runmqakm** 명령을 사용할 수 있습니다.

FIPS를 준수하는 방법으로 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오. 이는 **runmqakm** 명령이 보다 강력한 암호화를 지원하기 때문입니다.

### 프로시저

- **runmqakm** 및 **runmqktool** 명령을 사용하여 다음 조치를 완료하십시오.
  - IBM MQ 가 지원하는 CMS 및 PKCS #12 키 저장소를 작성하십시오.
  - 인증서 요청을 작성하십시오.
  - 인증서를 내보내십시오.
  - 개인 인증서 및 CA 인증서를 가져오십시오.
  - 자체 서명된 인증서를 관리합니다.
  - 비밀 키를 작성, 추출 및 추가하십시오.

**관련 정보**

키 도구

## ALW AIX, Linux, and Windows 의 runmqakm 및 runmqktool 명령

AIX, Linux, and Windows 시스템에서 **runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 및 인증서를 관리하십시오.

**참고:** V 9.4.0 V 9.4.0

IBM MQ 9.4.0에서 **runmqckm** 및 **strmqikm** 명령이 제거됩니다. **runmqckm** 명령 대신 **runmqktool** 명령을 사용하여 PKCS #12 및 JKS키 저장소를 관리할 수 있습니다. **strmqikm** GUI에 대한 대체가 없습니다.

**runmqckm** 및 **runmqktool** 명령에는 다음과 같은 중요한 차이점이 있습니다.

- **runmqktool** 명령은 키 저장소 비밀번호를 저장하기 위한 스태쉬 파일을 지원하지 않습니다. 키 저장소에 액세스하기 위한 비밀번호는 항상 명령에 대한 매개변수로 실행될 때 또는 명령에 의해 실행되는 프롬프트에 대한 응답으로 **runmqktool** 명령에 제공되어야 합니다.
- **runmqktool** 명령은 CMS 키 저장소를 지원하지 않습니다. 따라서 JKS에서 CMS 키 저장소로 인증서를 내보내려면 다음 단계를 완료해야 합니다.
  1. **runmqktool -importkeystore** 명령을 사용하여 JKS키 저장소에서 중간 PKCS #12 키 저장소로 인증서를 복사하십시오. 인증서 내보내기에 대한 자세한 정보는 514 페이지의 『AIX, Linux, and Windows에서 키 저장소로부터 개인 인증서 내보내기』의 내용을 참조하십시오.
  2. **runmqakm -cert -import** 명령을 사용하여 중간 PKCS #12 키 저장소에서 CMS 키 저장소로 인증서를 가져오십시오. 인증서 가져오기에 대한 자세한 정보는 516 페이지의 『AIX, Linux, and Windows에서 개인 인증서를 키 저장소로 가져오기』의 내용을 참조하십시오.

다음 IBM MQ 명령을 사용하여 키 및 인증서를 관리할 수 있습니다.

**runmqakm**

- **gskitcapicmd**의 기능과 유사한 기능을 제공합니다.
- CMS 및 PKCS #12 키 저장소를 지원합니다.
- 암호화된 키 저장소 비밀번호를 저장할 스태쉬 파일의 작성을 지원합니다.
- FIPS 140-2준수로 인증되며 **-fips** 매개변수를 사용하여 FIPS 준수 방식으로 작동하도록 구성할 수 있습니다.

V 9.4.0 V 9.4.0 **runmqktool**

- Java **keytool** 명령의 기능과 유사한 기능을 제공합니다.

- PKCS #12, JKS및 JCEKS키 저장소를 지원합니다.
- IBM MQ Java runtime environment (JRE) 구성요소가 설치되어 있어야 합니다.

FIPS를 준수하는 방식으로 인증서를 관리해야 하는 경우 **runmqakm** 명령을 사용하십시오.

**runmqakm** 명령에 대한 자세한 정보는 runmqakm을 참조하십시오.

V 9.4.0 V 9.4.0 **runmqktool** 명령에 대한 자세한 정보는 runmqktool을 참조하십시오.

이 절의 주제에는 이러한 명령을 사용하여 공통 인증서 관리 태스크를 완료하는 방법에 대한 예제가 포함되어 있습니다.

## ALW AIX, Linux, and Windows에서 자체 서명된 개인 인증서 작성

키 저장소에서 자체 서명된 개인 인증서를 작성하려면 이 프로시저를 따르십시오.

**참고:** IBM MQ는 SHA-3 또는 SHA-5 알고리즘을 지원하지 않습니다. 디지털 서명 알고리즘 이름 SHA384WithRSA 및 SHA512WithRSA을 사용할 수 있습니다. 두 알고리즘 모두 SHA-2 제품군의 구성원이기 때문입니다.

Deprecated 디지털 서명 알고리즘 이름 SHA3WithRSA 및 SHA5WithRSA는 각각 SHA384WithRSA 및 SHA512WithRSA의 축약된 양식이므로 더 이상 사용되지 않습니다.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 자체 서명된 인증서를 작성할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

자체 서명 인증서를 사용하고 싶은 이유에 대한 자세한 정보는 두 큐 관리자의 상호 인증을 위해 자체 서명 인증 사용을 참조하십시오.

모든 디지털 인증서를 모든 CipherSpec과 함께 사용할 수 있는 것은 아닙니다. 사용하는 CipherSpecs 와 호환 가능한 인증서를 작성해야 합니다. IBM MQ는 세 가지 유형의 CipherSpec을 지원합니다. 자세한 정보는 44 페이지의 『Elliptic Curve 및 RSA CipherSpec의 상호 운용성』 의 내용을 참조하십시오.

유형 1 CipherSpecs (이름이 ECDHE_ECDSA_로 시작함) 를 사용하려면 **runmqakm** 명령을 사용하여 인증서를 작성하고 Elliptic Curve ECDSA 서명 알고리즘 매개변수를 지정해야 합니다. 예를 들어, **-sig_alg** EC_ecdsa_with_SHA384매개변수를 지정합니다.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 자체 서명된 개인 인증서를 작성하십시오.

```
runmqakm -cert -create -db filename -pw password -label label
         -dn distinguished_name -size key_size
         -x509version version -expire days -fips -sig_alg algorithm
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 이미 존재해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
인증서 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

IBM MQ 에서 사용하는 TLS 인증서의 레이블은 **CERTLABL** 속성의 값 (설정된 경우) 또는 큐 관리자의 이름 또는 IBM MQ MQI client 사용자 ID가 추가된 기본 ibmwebspheremq (모두 소문자) 입니다. 자세한 정보는 25 페이지의 『디지털 인증서 레이블, 요구사항 이해』 의 내용을 참조하십시오.

**-dn *distinguished_name***
큰따옴표로 묶인 X.500 식별 이름을 지정합니다. 식별 이름에 하나 이상의 속성이 필요합니다. 여러 OU 및 DC 속성을 제공할 수 있습니다.

**참고:** **runmqakm** 명령은 우편번호 속성을 PC가 아닌 POSTALCODE로 참조합니다. **runmqakm** 명령을 사용하여 우편번호가 있는 인증서를 요청하는 경우 항상 **-dn** 매개변수에 POSTALCODE 를 지정하십시오.

**-size *key_size***
키 크기를 지정합니다. 값은 512, 1024또는 2048입니다.

**-x509version 버전**
작성할 X.509 인증서의 버전입니다. 값은 1, 2 또는 3일 수 있습니다. 기본값은 3입니다.

**-expire *days***
인증서의 만기 시간(일)입니다. 기본값은 인증서의 경우 365일입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS IBM Crypto for C (ICC) 구성요소만 사용되며 이 구성요소는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서 ICC 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

**-sig_alg**
인증서가 작성될 때 사용되는 해싱 알고리즘을 지정합니다. 이 해싱 알고리즘은 인증서와 연관된 서명을 작성하는 데 사용됩니다. 값은 md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384 또는 EC_ecdsa_with_SHA512일 수 있습니다.

기본값은 SHA1WithRSA입니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

## runmqktool 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 자체 서명된 개인 인증서를 작성하십시오.

```
runmqktool -genkeypair -keystore filename -storepass password -storetype store_type
           -alias label -dname distinguished_name -validity days
           -keyalg key_algorithm -keysize key_size -sigalg signature_algorithm
```

설명:

**-keystore 파일 이름**
키 저장소의 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-storepass 암호**
키 저장소 비밀번호를 지정합니다.

**-storetype *store_type* 입니다.**
키 저장소 유형을 지정합니다.

**-alias 레이블**
인증서 레이블을 지정합니다. 인증서 레이블은 소문자로 변환됩니다.

**-dname 구별 이름**
큰따옴표로 묶인 인증서의 X.500 식별 이름을 지정합니다.

**-validity *days***
인증서가 유효한 일 수를 지정합니다.

**-keyalg 키 알고리즘**
키 쌍을 작성하는 데 사용되는 알고리즘을 지정합니다.

**-keysize 키 크기**
키 크기를 지정합니다.

**-sigalg *signature_algorithm***
인증서에 서명하는 데 사용되는 알고리즘을 지정합니다. 지정할 수 있는 서명 알고리즘에 대한 자세한 정보는 서명 알고리즘의 내용을 참조하십시오.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 genkeypair를 참조하십시오.

## ALW AIX, Linux, and Windows에서 개인 인증서 요청

개인 인증서에 대한 요청을 작성하려면 이 프로시저를 따르십시오.

**참고:** IBM MQ는 SHA-3 또는 SHA-5 알고리즘을 지원하지 않습니다. 디지털 서명 알고리즘 이름 SHA384WithRSA 및 SHA512WithRSA을 사용할 수 있습니다. 두 알고리즘 모두 SHA-2 제품군의 구성원이기 때문입니다.

Deprecated 디지털 서명 알고리즘 이름 SHA3WithRSA 및 SHA5WithRSA는 각각 SHA384WithRSA 및 SHA512WithRSA의 축약된 양식이므로 더 이상 사용되지 않습니다.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 개인 인증서를 요청할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

모든 디지털 인증서를 모든 CipherSpec과 함께 사용할 수 있는 것은 아닙니다. 사용하는 CipherSpecs 와 호환 가능한 인증서를 작성해야 합니다. IBM MQ는 세 가지 유형의 CipherSpec을 지원합니다. 자세한 정보는 44 페이지의 『Elliptic Curve 및 RSA CipherSpec의 상호 운용성』의 내용을 참조하십시오.

유형 1 CipherSpecs (이름이 ECDHE_ECDSA_로 시작함) 를 사용하려면 **runmqakm** 명령을 사용하여 인증서를 작성하고 Elliptic Curve ECDSA 서명 알고리즘 매개변수를 지정해야 합니다. 예를 들어, **-sig_alg** EC_ecdsa_with_SHA384매개변수를 지정합니다.

암호화 하드웨어를 사용 중인 경우에는 525 페이지의 『PKCS #11 하드웨어의 개인 인증서 요청』의 내용을 참조하십시오.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 인증서 요청을 작성하십시오.

```
runmqakm -certreq -create -db filename -pw password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 이미 존재해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
인증서 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

IBM MQ 에서 사용하는 TLS 인증서의 레이블은 **CERTLABL** 속성의 값 (설정된 경우) 또는 큐 관리자의 이름 또는 IBM MQ MQI client 사용자 ID가 추가된 기본 ibmwebspheremq (모두 소문자) 입니다. 자세한 정보는 25 페이지의 『디지털 인증서 레이블, 요구사항 이해』의 내용을 참조하십시오.

**-dn *distinguished_name***
큰따옴표로 묶인 X.500 식별 이름을 지정합니다. 식별 이름에 하나 이상의 속성이 필요합니다. 여러 OU 및 DC 속성을 제공할 수 있습니다.

**참고:** **runmqakm** 명령은 우편번호 속성을 PC가 아닌 POSTALCODE로 참조합니다. **runmqakm** 명령을 사용하여 우편번호가 있는 인증서를 요청하는 경우 항상 **-dn** 매개변수에 POSTALCODE 를 지정하십시오.

**-size *key_size***
키 크기를 지정합니다. 값은 512, 1024또는 2048입니다.

**-file *filename***
인증서 요청의 파일 이름을 지정합니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

**-sig_alg**
인증서 요청이 작성될 때 사용되는 해싱 알고리즘을 지정합니다. 이 해싱 알고리즘은 인증서 요청과 연관된 서명을 작성하는 데 사용됩니다. 값은 md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384 또는 EC_ecdsa_with_SHA512일 수 있습니다.

기본값은 SHA1WithRSA입니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -certreq .

## runmqktool 사용

V 9.4.0 V 9.4.0

**runmqktool** 명령을 사용하여 인증서 요청을 작성하려면 먼저 **runmqktool -genkeypair** 명령을 사용하여 키 쌍을 생성해야 합니다. **runmqktool -genkeypair** 명령에 대한 자세한 정보는 506 페이지의 『AIX, Linux, and Windows에서 자체 서명된 개인 인증서 작성』의 내용을 참조하십시오.

다음 명령을 실행하여 **runmqktool** 명령으로 인증서 요청을 작성하십시오.

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

설명:

**-keystore 파일 이름**
키 저장소의 이름을 지정합니다.

**-storepass 암호**
키 저장소 비밀번호를 지정합니다.

**-alias 레이블**
인증서 레이블을 지정합니다. 이는 키 쌍이 생성될 때 지정된 인증서 레이블입니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

**-file *filename***
인증서 요청의 파일 이름을 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 certreq를 참조하십시오.

## 다음에 수행할 작업

인증서 요청을 CA에 제출하십시오. CA에서 서명된 인증서를 수신하면 서명된 인증서를 키 저장소에 추가하십시오. 자세한 정보는 510 페이지의 『AIX, Linux, and Windows에서 키 저장소에 개인 인증서 수신』의 내용을 참조하십시오.

## ALW AIX, Linux, and Windows에서 기존 개인 인증서 갱신

개인 인증서에는 만기 날짜가 있고, 그 후에는 인증서를 더 이상 사용할 수 없습니다. 개인 인증서가 만료되기 전에 갱신하려면 이 프로시저를 따르십시오.

**runmqakm** (GSKCapiCmd) 명령을 사용하여 개인 인증서를 갱신할 수 있습니다.

개인 인증서에 더 큰 키 크기를 사용해야 하는 요구사항이 있는 경우 기존 인증서를 갱신할 수 없습니다. 필요한 키 크기를 사용하는 새 인증서 요청을 작성하려면 508 페이지의 『AIX, Linux, and Windows에서 개인 인증서 요청』에 설명된 단계를 따라서 기존 키를 대체해야 합니다.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 개인 인증서를 갱신하기 위한 인증서 요청을 작성하십시오.

```
runmqakm -certreq -recreate -db filename -pw password
         -label label -target filename
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
인증서 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-target *filename***
인증서 요청의 파일 이름을 지정합니다.

### 다음에 수행할 작업

인증서 요청을 CA에 제출하십시오. CA에서 서명된 인증서를 수신하면 서명된 인증서를 키 저장소에 추가하십시오. 자세한 정보는 510 페이지의 『AIX, Linux, and Windows에서 키 저장소에 개인 인증서 수신』의 내용을 참조하십시오.

## ALW AIX, Linux, and Windows에서 키 저장소에 개인 인증서 수신

키 저장소로 개인 인증서를 수신하려면 이 프로시저를 사용하십시오.

인증 기관 (CA) 이 새 개인 인증서를 전송한 후 새 인증서 요청을 생성한 키 저장소에 추가하십시오. CA가 이메일 메시지의 일부로서 인증서를 송신하면 인증서를 별도의 파일로 복사하십시오.

키 저장소에 CA 서명 개인 인증서를 추가하기 전에 513 페이지의 『AIX, Linux, and Windows 의 키 저장소에 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트 추가』 의 단계를 완료하여 키 저장소에 CA 인증서를 추가하십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소에 개인 인증서를 수신할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

암호화 하드웨어를 사용 중인 경우에는 526 페이지의 『PKCS #11 하드웨어로 개인 인증서 수신』의 내용을 참조하십시오.

### runmqakm 사용

**runmqakm** 명령을 사용하여 키 저장소에 개인 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -receive -file filename -format format
         -db filename -pw password -fips
```

설명:

**-file *filename***
개인 인증서의 완전한 파일 이름을 지정합니다.

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소는 이미 존재해야 하며 인증서 요청을 작성한 저장소와 동일해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-format 형식**
인증서의 형식을 지정합니다. 값은 Base64 인코딩된 ASCII의 경우 `ascii`이거나 2진 DER 데이터의 경우 `binary`입니다. 기본값은 `ascii`입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

### runmqktool 사용

V 9.4.0 V 9.4.0

**runmqktool** 명령을 사용하여 키 저장소에 개인 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

설명:

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소는 이미 존재해야 하며 인증서 요청을 작성한 저장소와 동일해야 합니다.

**-storepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-alias 레이블**
인증서 요청을 작성하는 데 사용된 인증서의 레이블을 지정합니다. 인증서 레이블은 소문자로 변환됩니다.

**-file *filename***
개인 인증서의 완전한 파일 이름을 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 importcert를 참조하십시오.

### 다음에 수행할 작업

인증서가 큐 관리자의 TLS키 저장소에 추가되면 MQSC 명령 **REFRESH SECURITY TYPE(SSL)** 를 실행하여 큐 관리자의 TLS키 저장소 캐시를 새로 고치십시오.

## ALW AIX, Linux, and Windows에서 키 저장소로부터 인증서 추출

키 저장소에서 인증 기관 (CA) 인증서를 추출하려면 이 프로시저를 따르십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소에서 CA 인증서를 추출할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 CA 인증서를 추출하십시오.

```
runmqakm -cert -extract -db filename -pw password -label label
         -target filename -format format -fips
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
CA 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-target *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

**-format 형식**
인증서의 형식을 지정합니다. 값은 Base64 인코딩된 ASCII의 경우 `ascii`이거나 2진 DER 데이터의 경우 `binary`입니다. 기본값은 `ascii`입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -cert .

### `runmqktool` 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 CA 인증서를 추출하십시오.

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

설명:

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다.

**-storepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-alias 레이블**
CA 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

**-file *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

**-rfc**
출력 파일이 인터넷 RFC 1421표준에 정의된 대로 Base64-encoded ASCII 형식으로 되어 있음을 지정합니다. 이 옵션을 지정하지 않으면 출력 파일은 2진 형식입니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 exportcert를 참조하십시오.

## ALW AIX, Linux, and Windows에서 키 저장소로부터 자체 서명 인증서의 공용 부분 추출

키 저장소에서 자체 서명된 인증서의 공용 파트를 추출하려면 이 프로시저를 따르십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소에서 인증서의 공용 파트를 추출할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

### `runmqakm` 사용

다음 명령을 실행하여 **runmqakm** 명령으로 자체 서명 인증서의 공용 파트를 추출하십시오.

```
runmqakm -cert -extract -db filename -pw password -label label
           -target filename -format format -fips
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
CA 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-target *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

**-format 형식**
인증서의 형식을 지정합니다. 값은 Base64-encoded ASCII의 경우 `ascii` 또는 2진 DER 데이터의 경우 `binary`일 수 있습니다. 기본값은 `ascii`입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -cert .

## runmqktool 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 자체 서명 인증서의 공용 파트를 추출하십시오.

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

설명:

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다.

**-storepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-alias 레이블**
CA 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

**-file *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

**-rfc**
출력 파일이 인터넷 RFC 1421표준에 정의된 대로 Base64-encoded ASCII 형식으로 되어 있음을 지정합니다. 이 옵션을 지정하지 않으면 출력 파일은 2진 형식입니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 exportcert를 참조하십시오.

## ALW AIX, Linux, and Windows 의 키 저장소에 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트 추가

CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트를 키 저장소에 추가하려면 이 프로시저를 따르십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트를 키 저장소에 추가할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

추가하려는 인증서가 인증서 체인에 있는 경우 체인에서 위에 있는 모든 인증서 또한 추가해야 합니다. 인증서를 루트에서 시작하여 체인에서 바로 아래에 있는 CA 인증서가 뒤따르는 방식으로 내림차순으로 엄격하게 추가해야 합니다.

**참고:**

- 인증서가 ASCII (UTF-8) 또는 2진 (DER) 인코딩인지 확인하십시오.

- IBM Java 8 **keytool** 명령의 제한사항으로 인해 **runmqktool** 는 파일에 주석이 포함된 경우 인터넷 RFC 1421 에서 정의한 대로 인쇄 가능한 인코딩 형식 ( Base64 인코딩이라고도 함) 으로 인증서를 가져올 수 없습니다. 인쇄 가능한 인코딩 형식으로 인증서를 가져오려면 파일에서 모든 주석을 제거하십시오. 파일은 "----- BEGIN" 으로 시작하는 문자열로 시작하고 "----- END" 로 시작하는 문자열로 끝나야 합니다.

### runmqakm 사용

**runmqakm** 명령을 사용하여 키 저장소에 신뢰할 수 있는 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -add -db filename -pw password -label label
         -file filename -format ascii -fips
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 이미 존재해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
인증서 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-file *filename***
인증서를 포함하는 파일의 이름을 지정합니다.

**-format *ascii***
인증서의 형식을 지정합니다. 값은 Base64-encoded ASCII의 경우 `ascii` 또는 2진 DER 데이터의 경우 `binary`일 수 있습니다. 기본값은 `ascii`입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

### runmqktool 사용

V 9.4.0 V 9.4.0

**runmqktool** 명령을 사용하여 키 저장소에 신뢰할 수 있는 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

설명:

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-storepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-alias 레이블**
인증서 레이블을 지정합니다. 인증서 레이블은 소문자로 변환됩니다.

**-file *filename***
개인 인증서의 완전한 파일 이름을 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 importcert를 참조하십시오.

## ALW AIX, Linux, and Windows에서 키 저장소로부터 개인 인증서 내보내기

이 프로시저에 따라 키 저장소에서 개인 인증서를 내보내십시오.

인증서를 내보내면 인증서 및 연관된 공용 및 개인용 키가 다른 키 저장소로 복사됩니다.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소에서 인증서를 내보낼 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

## runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 인증서를 내보내십시오.

```
runmqakm -cert -export -db filename -pw password -label label
         -target filename -target_pw password -target_type type
         -encryption strength -fips
```

설명:

**-db *filename***
인증서를 포함하는 키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
인증서를 포함하는 키 저장소의 비밀번호를 지정합니다.

**-label *label***
내보낼 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-target *filename***
대상 키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-target_pw 암호**
대상 키 저장소의 비밀번호를 지정합니다.

**-target_type 유형**
대상 키 저장소의 유형을 지정합니다. 값은 cms 또는 pkcs12일 수 있습니다. 기본값은 cms입니다.

**-encryption 강도**
인증서 내보내기 명령에서 사용되는 암호화 강도를 지정합니다. 값은 strong 또는 weak일 수 있습니다. 기본값은 strong입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -cert .

## runmqktool 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 인증서를 내보내십시오.

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

설명:

**-srckeystore 파일 이름**
인증서를 포함하는 키 저장소의 완전한 파일 이름을 지정합니다.

**-srcstorepass 암호**
인증서를 포함하는 키 저장소의 비밀번호를 지정합니다.

**-destkeystore 파일 이름**
대상 키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-deststorepass 암호**
대상 키 저장소의 비밀번호를 지정합니다.

**-destkeypass 비밀번호**
대상 키 저장소에서 키를 보호하기 위한 비밀번호를 지정합니다. 이 매개변수를 지정하지 않으면 키는 소스 키 저장소의 키를 보호하는 데 사용되는 비밀번호로 보호됩니다.

**-deststoretype 유형**
대상 키 저장소의 유형을 지정합니다.

**-srcalias 레이블**
내보낼 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

**-destalias 레이블**
대상 키 저장소에 있는 인증서의 레이블을 지정합니다. 이 매개변수를 지정하지 않으면 소스 키 저장소에서 와 동일한 레이블이 인증서에 지정됩니다.

인증서 레이블은 소문자로 변환됩니다.

**-file *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 importkeystore를 참조하십시오.

## ALW AIX, Linux, and Windows에서 개인 인증서를 키 저장소로 가져오기

개인 인증서를 키 저장소로 가져오려면 이 프로시저를 따르십시오.

인증서를 가져오면 한 키 저장소에서 다른 키 저장소로 인증서 및 연관된 공개 및 개인 키가 복사됩니다.

개인 인증서를 키 저장소로 가져오기 전에 먼저 발행 CA 인증서의 유효한 전체 체인을 키 저장소에 추가해야 합니다. 자세한 정보는 513 페이지의 『AIX, Linux, and Windows 의 키 저장소에 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트 추가』의 내용을 참조하십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소에 인증서를 가져올 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 인증서를 가져오십시오.

```
runmqakm -cert -import -file filename -pw password -type type
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips
```

설명:

**-file *filename***
인증서를 포함하는 키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
인증서를 포함하는 키 저장소의 비밀번호를 지정합니다.

**-type *type***
인증서를 포함하는 키 저장소의 유형을 지정합니다. 값은 cms 또는 pkcs12일 수 있습니다. 기본값은 cms입니다.

**-target *filename***
대상 키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-target_pw 암호**
대상 키 저장소의 비밀번호를 지정합니다.

**-target_type 유형**
대상 키 저장소의 유형을 지정합니다. 값은 cms 또는 pkcs12일 수 있습니다. 기본값은 cms입니다.

**-label *label***
소스 키 저장소에서 가져올 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-new_label 레이블**
대상 키 저장소의 인증서에 지정된 레이블을 지정합니다. 이 매개변수를 지정하지 않으면 소스 키 저장소에서와 동일한 레이블이 인증서에 지정됩니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

## runmqktool 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 인증서를 가져오십시오.

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

설명:

**-srckeystore 파일 이름**
인증서를 포함하는 키 저장소의 완전한 파일 이름을 지정합니다.

**-srcstorepass 암호**
인증서를 포함하는 키 저장소의 비밀번호를 지정합니다.

**-destkeystore 파일 이름**
대상 키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-deststorepass 암호**
대상 키 저장소의 비밀번호를 지정합니다.

**-destkeypass 비밀번호**
대상 키 저장소에서 키를 보호하기 위한 비밀번호를 지정합니다. 이 매개변수를 지정하지 않으면 키는 소스 키 저장소의 키를 보호하는 데 사용되는 비밀번호로 보호됩니다.

**참고:** PKCS #12 키 저장소의 경우, 키는 대상 키 저장소와 동일한 비밀번호로 보호되어야 합니다.

**-deststoretype 유형**
대상 키 저장소의 유형을 지정합니다.

**-srcalias 레이블**
소스 키 저장소에서 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

**-destalias 레이블**
대상 키 저장소에 있는 인증서의 레이블을 지정합니다. 이 매개변수를 지정하지 않으면 소스 키 저장소에서 와 동일한 레이블이 인증서에 지정됩니다.

인증서 레이블은 소문자로 변환됩니다.

**-file *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 importkeystore를 참조하십시오.

## ALW Microsoft.pfx 파일에서 개인 인증서 가져오기

다음 절차에 따라 인증서를 가져오세요.Microsoft .pfx 파일AIX, Linux, and Windows .

.pfx 파일은 동일 키와 관련된 두 개의 인증서를 포함할 수 있습니다. 하나는 공용 및 개인 키를 모두 포함하는 개인 또는 사이트 인증서입니다. 다른 하나는 공개 키만 포함하는 CA (서명자) 인증서입니다. 이러한 인증서는 동일한 CMS 키 저장소에 공존할 수 없으므로 이들 중 하나만 가져올 수 있습니다.

인증서 레이블은 서명자 인증서에만 첨부됩니다. 개인 인증서는 시스템 생성 UUID(Unique User Identifier)에 의해서 식별됩니다. .pfx 파일에서 개인 인증서를 가져오고 개인 인증서 레이블을 .pfx 파일의 CA 인증서에 지정된 레이블로 설정하려면 이 프로시저를 따르십시오. 발행 CA 인증서가 이미 대상 키 데이터베이스에 추가되어 있어야 합니다.

### runmqakm 사용

**runmqakm** 명령을 사용하여 .pfx 파일에서 인증서를 가져오려면 다음 명령을 실행하십시오.

```
runmqakm -cert -import -file filename -pw password -type pkcs12
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips -pfx
```

설명:

**-file *filename***
.pfx 파일의 완전한 이름을 지정합니다.

**-pw *password***
.pfx 파일의 비밀번호를 지정합니다.

**-type pkcs12**
키 저장소의 유형을 지정합니다.

**-target *filename***
대상 키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-target_pw 암호**
대상 키 저장소의 비밀번호를 지정합니다.

**-target_type 유형**
대상 키 저장소의 유형을 지정합니다. 값은 cms 또는 pkcs12일 수 있습니다. 기본값은 cms입니다.

**-label *label***
소스 키 저장소에서 가져올 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-new_label 레이블**
대상 키 저장소의 인증서에 지정된 레이블을 지정합니다. 이 매개변수를 지정하지 않으면 소스 키 저장소에서와 동일한 레이블이 인증서에 지정됩니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

**-pfx**
소스 키 저장소가 PFX 형식을 사용함을 표시합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

## ALW PKCS #7 파일로부터 개인 인증서 가져오기

AIX, Linux, and Windows의 PKCS #7 파일에서 인증서를 가져오려면 이 프로시저를 따르십시오.

**runmqakm** 명령을 사용하여 AIX, Linux, and Windows의 PKCS #7 파일에서 인증서를 가져오십시오.

### CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트 추가

다음 명령을 실행하여 PKCS #7 파일에서 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트를 추가하십시오.

```
runmqakm -cert -add -db filename -pw password -type type
         -label label -file filename
```

설명:

**-db *filename***
키 저장소의 완전한 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-type *type***
키 저장소의 유형을 지정합니다.

**-label *label***
추가할 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

레이블은 추가되는 첫 번째 인증서에 지정됩니다. 있는 경우, 다른 모든 인증서에는 제목 이름이 레이블 지정됩니다.

**-file *filename***
PKCS #7 파일의 완전한 이름을 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

## 개인 인증서 가져오기

다음 명령을 실행하여 PKCS #7 파일에서 개인 인증서를 가져오십시오.

```
runmqakm -cert -import -file filename -pw password -type pkcs7
         -target filename -target_pw password -target_type type
         -label label -new_label label
```

설명:

**-file *filename***
PKCS #7 파일의 완전한 이름을 지정합니다.

**-pw *password***
PKCS #7 파일의 비밀번호를 지정합니다.

**-type pkcs7**
PKCS #7 파일의 유형을 지정합니다.

**-target *filename***
대상 키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 없는 경우 키 저장소가 작성됩니다.

**-target_pw 암호**
대상 키 저장소의 비밀번호를 지정합니다.

**-target_type 유형**
대상 키 저장소의 유형을 지정합니다. 값은 cms 또는 pkcs12일 수 있습니다. 기본값은 cms입니다.

**-label *label***
PKCS #7 파일에서 가져올 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-new_label 레이블**
대상 키 저장소의 인증서에 지정된 레이블을 지정합니다. 이 매개변수를 지정하지 않으면 소스 키 저장소에서와 동일한 레이블이 인증서에 지정됩니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -cert를 참조하십시오.

# ALW AIX, Linux, and Windows 의 키 저장소에 있는 인증서 나열

키 저장소에 있는 인증서를 나열하려면 이 프로시저를 사용하십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소에 있는 인증서에 대한 정보를 표시할 수 있습니다.

## runmqakm 사용

- **runmqakm** 명령을 사용하여 키 저장소에 있는 인증서의 레이블을 나열하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -list -db filename -pw password
```

- **runmqakm** 명령을 사용하여 키 저장소에 있는 인증서의 세부사항을 나열하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -details -showOID -db filename -pw password
         -label label
```

설명:

**-file *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
나열할 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -cert .

### `runmqktool` 사용

V 9.4.0 V 9.4.0

- **`runmqktool`** 명령을 사용하여 키 저장소에 있는 인증서의 레이블을 나열하려면 다음 명령을 실행하십시오.

```
runmqktool -list -keystore filename -storepass password
```

- **`runmqktool`** 명령을 사용하여 키 저장소에 있는 인증서의 세부사항을 나열하려면 다음 명령을 실행하십시오.

```
runmqktool -list -keystore filename -storepass password -alias label -v
```

설명:

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다.

**-storepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-alias 레이블**
나열할 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

**-v**
인증서 세부사항을 포함하는 상세 출력을 요청합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 목록을 참조하십시오.

## ALW AIX, Linux, and Windows에서 키 저장소로부터 인증서 삭제

키 저장소에서 개인 또는 CA 인증서를 삭제하려면 이 프로시저를 사용하십시오.

**`runmqakm`** (GSKCapiCmd) 또는 **`runmqktool`** (keytool) 명령을 사용하여 키 저장소에서 인증서를 삭제할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **`runmqakm`** 명령을 사용하십시오.

### `runmqakm` 사용

다음 명령을 실행하여 **`runmqakm`** 명령으로 인증서를 삭제하십시오.

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

설명:

**-file *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
삭제할 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분합니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -cert .

## runmqktool 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 인증서를 삭제하십시오.

```
runmqktool -delete -keystore filename -storepass password -alias label
```

설명:

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다.

**-storepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-alias 레이블**
삭제할 인증서의 레이블을 지정합니다. 인증서 레이블은 대소문자를 구분하지 않습니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 삭제를 참조하십시오.

## ALW AIX, Linux, and Windows 에서 키 저장소 변환

키 저장소를 다른 유형으로 변환하려면 이 프로시저를 사용하십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소 비밀번호를 다른 유형으로 변환할 수 있습니다.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 키 저장소를 변환하십시오.

```
runmqakm -keydb -convert -db filename -pw password
         -new_db filename -new_pw password
         -old_format type -new_format type
```

설명:

**-file *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-new_db 파일 이름**
새 키 저장소의 완전한 파일 이름을 지정합니다.

**-new_pw 암호**
새 키 저장소의 비밀번호를 지정합니다.

**-old_format 유형**
키 저장소의 현재 유형을 지정합니다. 다음 값을 지정할 수 있습니다.

- pkcs12
- cms

**-new_format 유형**
키 저장소의 새 유형을 지정합니다. 다음 값을 지정할 수 있습니다.

- pkcs12

- cms

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -keydb .

### runmqktool 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **runmqktool** 명령으로 키 저장소를 변환하십시오.

```
runmqktool -importkeystore -srckeystore filename -destkeystore filename
           -srcstoretype type -deststoretype type
           -srcstorepass password -deststorepass password
```

설명:

**-모두**
키 저장소와 동일한 비밀번호로 보호되는 모든 항목에 대한 비밀번호도 변경되도록 지정합니다.

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다.

**-destkeystore 파일 이름**
새 키 저장소의 완전한 파일 이름을 지정합니다.

**-srcstoretype 유형**
키 저장소 유형을 지정합니다.

**-deststoretype 유형**
새 키 저장소 유형을 지정합니다.

**-srcstorepass 암호**
키 저장소의 비밀번호를 지정합니다.

**-deststorepass 암호**
새 키 저장소의 비밀번호를 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 importkeystore를 참조하십시오.

## ALW AIX, Linux, and Windows 에서 키 저장소 비밀번호 변경

키 저장소 비밀번호를 변경하려면 이 프로시저를 사용하십시오.

**runmqakm** (GSKCapiCmd) 또는 **runmqktool** (keytool) 명령을 사용하여 키 저장소 비밀번호를 변경할 수 있습니다.

**참고:**

- V 9.4.0 V 9.4.0 **runmqktool** 명령을 사용하면 개별 개인 및 비밀 키를 보호하는 비밀번호와 별도로 키 저장소 비밀번호를 변경할 수 있습니다. PKCS #12 키 저장소의 경우 키 저장소 비밀번호와 키 저장소의 모든 키를 보호하는 비밀번호가 동일해야 합니다. **runmqktool** 명령을 사용하여 키 저장소 비밀번호를 변경하는 경우 키 비밀번호도 변경되도록 **-all** 매개변수가 지정되었는지 확인하십시오.
- 키 저장소 비밀번호가 스태쉬 파일에 저장되지 않은 경우, 키 저장소에 액세스하는 IBM MQ client 애플리케이션 또는 큐 관리자 구성에 저장된 비밀번호도 변경해야 합니다. 자세한 정보는 282 페이지의 『AIX, Linux, and Windows 에서 큐 관리자의 키 저장소 비밀번호 제공』 및 284 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』의 내용을 참조하십시오.

### runmqakm 사용

다음 명령을 실행하여 **runmqakm** 명령으로 키 저장소 비밀번호를 변경하십시오.

```
runmqakm -keydb -changepw -db filename -pw password -new_pw password -stash
```

설명:

**-file *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 현재 비밀번호를 지정합니다.

**-new_pw 암호**
키 저장소의 새 비밀번호를 지정합니다.

**-stash**
선택사항. 스태쉬 파일에 새 키 저장소 비밀번호를 저장하려면 이 옵션을 지정하십시오. 대신 IBM MQ 비밀번호 보호 시스템을 사용하여 비밀번호를 암호화하는 경우 비밀번호를 스태쉬 파일에 저장할 필요가 없습니다.

이러한 매개변수와 지정할 수 있는 값에 대한 자세한 내용은 다음을 참조하세요. runmqakm -keydb .

## `runmqktool` 사용

V 9.4.0 V 9.4.0

다음 명령을 실행하여 **`runmqktool`** 명령으로 키 저장소 비밀번호를 변경하십시오.

```
runmqktool -storepasswd -all -keystore filename -storepass password
           -new password
```

설명:

**-모두**
키 저장소와 동일한 비밀번호로 보호되는 모든 항목에 대한 비밀번호도 변경되도록 지정합니다.

**-keystore 파일 이름**
키 저장소의 완전한 파일 이름을 지정합니다.

**-storepass 암호**
키 저장소의 현재 비밀번호를 지정합니다.

**-new 암호**
키 저장소의 새 비밀번호를 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 추가 정보는 storepasswd를 참조하십시오.

## ALW 비밀 키 관리AIX, Linux, and Windows

키 저장소에서 비밀 키를 관리하려면 다음 절차를 따르세요.

다음을 사용하여 비밀 키를 관리할 수 있습니다.**`runmqakm`** (GSKCapiCmd ) 명령. 다음을 사용하여 생성된 비밀 키**`runmqktool`** (keytool) 명령은 다음과 함께 사용할 수 없습니다.IBM MQ .

### 비밀 키 만들기

다음 명령을 실행하여 임의의 비밀 키를 생성하십시오.**`runmqakm`** 명령:

```
runmqakm -secretkey -create -db filename -pw password
         -label label -size key_size
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 이미 존재해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
키에 부착된 레이블을 지정합니다.

**-size *key_size***
키 크기를 바이트 단위로 지정합니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -secretkey를 참조하십시오.

### 비밀 키 추출

다음 명령을 실행하여 비밀 키를 추출합니다.**runmqakm** 명령:

```
runmqakm -secretkey -extract -db filename -pw password
         -label label -target filename -format format
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 이미 존재해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
추출할 키의 레이블을 지정합니다.

**-target *filename***
대상 파일의 완전한 파일 이름을 지정합니다.

**-체재 체재**
대상 파일의 키 형식을 지정합니다. 값은 다음과 같습니다.`ascii` ~을 위한Base64-encoded ASCII 또는 `binary` 키의 바이너리 복사본입니다. 기본값은 `ascii`입니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -secretkey를 참조하십시오.

### 비밀 키 추가

다음 명령을 실행하여 비밀 키를 추출합니다.**runmqakm** 명령:

```
runmqakm -secretkey -add -db filename -pw password
         -label label -file filename -format format
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다. 키 저장소가 이미 존재해야 합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-label *label***
키에 부착된 레이블을 지정합니다.

**-file *filename***
키가 포함된 파일의 이름을 지정합니다.

**-체재 체재**
키의 형식을 지정합니다. 값은 다음과 같습니다.`ascii` ~을 위한Base64-encoded ASCII 또는`binary` 바이너리 데이터의 경우. 기본값은 `ascii`입니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -secretkey를 참조하십시오.

## ALW PKCS #11 하드웨어에서 인증서 관리

PKCS #11 인터페이스를 지원하는 암호화 하드웨어에서 디지털 인증서를 관리할 수 있습니다.

인증서를 저장하지 않고 암호화 하드웨어에 모든 인증서를 저장하는 경우에도 IBM MQ 환경을 준비하려면 키 저장소를 작성해야 합니다. 큐 관리자가 해당 **SSLKEYR** 속성에서 참조하거나 클라이언트 애플리케이션이 MQSSLKEYR 환경 변수에서 참조하려면 키 저장소가 필요합니다. 이 키 저장소는 인증서 요청을 작성하는 경우에도 필요합니다.

**runmqakm** (GSKCapiCmd) 명령을 사용하여 키 저장소를 작성하십시오.

다음 명령을 실행하여 **runmqakm** 명령으로 키 저장소를 작성하십시오.

```
runmqakm -keydb -create -db filename -pw password -type type -stash
```

설명:

**-db *filename***
키 저장소의 완전한 파일 이름을 지정합니다.

**-pw *password***
키 저장소의 비밀번호를 지정합니다.

**-type *type***
데이터베이스의 유형을 지정합니다. 값은 IBM MQ에서 사용되는 키 저장소의 경우 cms 또는 pkcs12 여야 합니다.

**-stash**
선택사항. 지정된 경우 암호화된 키 저장소 비밀번호가 파일에 저장됩니다.

## ALW *PKCS #11 하드웨어의 개인 인증서 요청*

이 프로시저를 사용하여 암호화 하드웨어가 있는 IBM MQ MQI client 또는 큐 관리자에 대한 개인 인증서를 요청하십시오.

**참고:** IBM MQ는 SHA-3 또는 SHA-5 알고리즘을 지원하지 않습니다. 디지털 서명 알고리즘 이름 SHA384WithRSA 및 SHA512WithRSA을 사용할 수 있습니다. 두 알고리즘 모두 SHA-2 제품군의 구성원이기 때문입니다.

Deprecated 디지털 서명 알고리즘 이름 SHA3WithRSA 및 SHA5WithRSA는 각각 SHA384WithRSA 및 SHA512WithRSA의 축약된 양식이므로 더 이상 사용되지 않습니다.

암호화 하드웨어에서 인증서 요청을 작성하기 전에 524 페이지의 『PKCS #11 하드웨어에서 인증서 관리』 에 설명된 단계를 완료하여 키 저장소를 작성하십시오.

다음 명령을 실행하여 **runmqakm** (GSKCapiCmd) 명령으로 인증서 요청을 작성하십시오.

```
runmqakm -certreq -create -crypto module_name -tokenlabel hardware_token
         -pw password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

설명:

**-crypto *module_name***
암호화 하드웨어와 함께 제공된 PKCS #11 라이브러리의 완전한 이름을 지정합니다.

**-tokenlabel *hardware_token***
PKCS #11 암호화 디바이스 토큰 레이블을 지정합니다.

**-pw *password***
암호화 하드웨어에 액세스하기 위한 비밀번호를 지정합니다.

**-label *label***
인증서 레이블을 지정합니다.

IBM MQ 에서 사용하는 TLS 인증서의 레이블은 **CERTLABL** 속성의 값 (설정된 경우) 또는 큐 관리자의 이름 또는 IBM MQ MQI client 사용자 ID가 추가된 기본 ibmwebspheremq (모두 소문자) 입니다. 자세한 정보는 25 페이지의 『디지털 인증서 레이블, 요구사항 이해』의 내용을 참조하십시오.

**-dn *distinguished_name***
큰따옴표로 묶인 X.500 식별 이름을 지정합니다. 식별 이름에 하나 이상의 속성이 필요합니다. 여러 OU 및 DC 속성을 제공할 수 있습니다.

**참고:** **runmqakm** 명령은 우편번호 속성을 PC가 아닌 POSTALCODE로 참조합니다. **runmqakm** 명령을 사용하여 우편번호가 있는 인증서를 요청하는 경우 항상 **-dn** 매개변수에 POSTALCODE 를 지정하십시오.

**-size *key_size***
키 크기를 지정합니다. 값은 512, 1024또는 2048입니다.

**-file *filename***
인증서 요청의 파일 이름을 지정합니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

**-sig_alg**
인증서 요청이 작성될 때 사용되는 해싱 알고리즘을 지정합니다. 이 해싱 알고리즘은 인증서 요청과 연관된 서명을 작성하는 데 사용됩니다. 값은 md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384 또는 EC_ecdsa_with_SHA512일 수 있습니다.

기본값은 SHA1WithRSA입니다.

이러한 매개변수 및 지정할 수 있는 값에 대한 자세한 정보는 runmqakm -certreq를 참조하십시오.

## 다음에 수행할 작업

인증서 요청을 CA에 제출하십시오. CA에서 서명된 인증서를 수신하면 서명된 인증서를 키 저장소에 추가하십시오. 자세한 정보는 526 페이지의 『PKCS #11 하드웨어로 개인 인증서 수신』의 내용을 참조하십시오.

## ALW *PKCS #11* 하드웨어로 개인 인증서 수신

큐 관리자 또는 IBM MQ MQI client 에 대한 개인 인증서를 암호화 하드웨어에 수신하려면 이 프로시저를 사용하십시오.

개인 인증서에 서명한 CA의 CA 인증서를 암호화 하드웨어 또는 보조 키 저장소에 추가하십시오. 암호화 하드웨어에 서명된 인증서를 수신하기 전에 이를 수행하십시오. 키 저장소 파일에 CA 인증서를 추가하려면 513 페이지의 『AIX, Linux, and Windows 의 키 저장소에 CA 인증서 또는 신뢰할 수 있는 인증서의 공용 파트 추가』의 프로시저를 따르십시오.

**runmqakm** (GSKCapiCmd) 명령을 사용하여 키 저장소에 개인 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

설명:

**-file *filename***
개인 인증서를 포함하고 있는 파일의 완전한 파일 이름을 지정합니다.

**-crypto *module_name***
암호화 하드웨어와 함께 제공된 PKCS #11 라이브러리의 완전한 이름을 지정합니다.

**-tokenlabel *hardware_token***
PKCS #11 암호화 디바이스 토큰 레이블을 지정합니다.

**-pw *hardware_password***
암호화 하드웨어에 액세스하기 위한 비밀번호를 지정합니다.

**-format *cert_format***
인증서의 형식을 지정합니다. 값은 Base64 인코딩된 ASCII의 경우 ascii이거나 2진 DER 데이터의 경우 binary입니다. 기본값은 ASCII입니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. FIPS 모드에서 IBM Crypto for C (ICC) 구성요소는 FIPS 140-2유효성 검증된 알고리즘을 사용합니다. ICC 구성요소가 FIPS 모드에서 초기화되지 않으면 **runmqakm** 명령이 실패합니다.

**-secondaryDB *filename***
CA 인증서를 저장하는 데 사용되는 키 저장소 파일의 완전한 파일 이름을 지정합니다.

**-secondaryDBpw *password***
CA 인증서를 저장하는 데 사용되는 키 저장소 파일의 비밀번호를 지정합니다.

# IBM MQ 컴포넌트 구성 파일에서 비밀번호 보호

IBM MQ의 특정 기능을 사용하려면 기능에서 사용하는 비밀번호를 제공해야 합니다. IBM MQ 에 제공되는 비밀번호는 비밀번호 보호 시스템을 사용하여 보호할 수 있습니다.

다음 목록은 암호화된 비밀번호를 처리하는 각 구성요소에 사용되는 용어를 설명합니다.

**초기 키**
비밀번호를 보호하는 데 사용되는 암호화 키입니다.

**기본 초기 키**
비밀번호가 암호화될 때 초기 키를 제공하지 않는 경우 사용되는 기본 암호화 키입니다.

**일반 텍스트 문자열**
암호화되는 문자열이며 일반적으로 비밀번호입니다.

**암호화된 비밀번호 문자열**
IBM MQ 가 이해하는 형식으로 암호화된 비밀번호를 포함하는 문자열입니다.

## 초기 키 지정

각 컴포넌트에 대해 비밀번호를 암호화하는 데 사용되는 초기 키를 지정하도록 선택할 수 있습니다.

- 초기 키를 지정하지 않으면 구성요소의 기본 초기 키가 사용됩니다. 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 이는 다른 설치에서 비밀번호를 복호화할 수 있으므로 기본 초기 키로 암호화된 비밀번호가 안전하게 보호되지 않음을 의미합니다.
- 사용자 고유의 초기 키를 제공하는 경우 사용자가 제공하는 초기 키에 대한 액세스 권한이 있는 사용자만 비밀번호를 복호화할 수 있습니다.

**주의:** 저장된 비밀번호에 대해 최상위 레벨의 보안을 제공하려면 각 IBM MQ 구성요소에 대해 고유한 초기 키를 제공하십시오.

사용자 고유의 초기 키를 사용하도록 선택하는 경우 나열되는 각 구성요소에 대해 고유한 초기 키를 지정하십시오. 초기 키는 해당 구성요소의 구성에 저장된 비밀번호를 보호하는 데 사용됩니다. 암호를 해독하려면 구성요소에 동일한 초기 키를 사용할 수 있어야 합니다.

대부분의 구성요소에서는 파일에 초기 키를 제공해야 합니다. 초기 키 파일에 포함된 초기 키는 다음 요구사항을 충족해야 합니다.

- 하나 이상의 문자 길이여야 합니다.
- 단일 텍스트 행이어야 합니다.

초기 키의 최대 길이는 무제한이며 모든 문자를 지정할 수 있습니다. 적절한 보안을 위해 16자이상의 초기 키를 지정하십시오. 예를 들어, 초기 키 파일은 다음 문자열을 포함할 수 있습니다.

```
Th1sIs@n3Ncypt|onK$y
```

초기 키 파일에 대한 액세스는 운영 체제 파일 권한을 사용하여 초기 키에 액세스해야 하는 사용자로만 제한되어야 합니다.

비밀번호 보호의 이점 및 제한사항에 대한 자세한 정보는 533 페이지의 『비밀번호 암호화를 통한 보호 한계』의 내용을 참조하십시오.

## 각 IBM MQ 구성요소에서 비밀번호 보호

여러 IBM MQ 구성요소가 저장된 비밀번호를 보호할 수 있습니다. 컴포넌트에 따라 다음 메커니즘 중 하나를 사용하여 이러한 비밀번호를 제공할 수 있습니다.

- IBM MQ 큐 관리자 또는 IBM MQ client에 직접 제공됩니다.
- 환경 변수에 지정됩니다.
- 구성 파일에 저장됩니다.

각 구성요소는 암호를 암호화하는 방법을 제공합니다. 대부분의 구성요소에서 비밀번호는 IBM MQ 에 제공되거나 구성에 저장되기 전에 암호화되어야 합니다.

**중요사항:** 한 구성요소와 함께 사용하기 위해 생성된 암호화된 비밀번호를 다른 구성요소의 구성 파일에 복사할 수 없습니다. 특정 구성요소에서 사용하도록 암호화된 비밀번호는 동일한 구성요소에서 제공하는 유틸리티를 사용하여 보호되어야 합니다.

비밀번호 보호를 지원하는 각 IBM MQ 구성요소의 비밀번호를 보호하는 방법에 대한 세부사항은 다음 절에 나열되어 있습니다.

- Advanced Message Security
- 529 페이지의 『Managed File Transfer』
- 530 페이지의 『IBM MQ Internet Pass-Thru』
- 530 페이지의 『암호화 하드웨어를 사용하는 IBM MQ clients』
- 531 페이지의 『IBM MQ 큐 관리자』
- 531 페이지의 『IBM MQ C 클라이언트 애플리케이션』
- V 9.4.0 532 페이지의 『기본 HA 구성』
- V 9.4.0 533 페이지의 『IBM MQ 큐 관리자 ( qm.ini 파일의AuthToken 스탠자)』

## Advanced Message Security

Advanced Message Security (AMS) Java 클라이언트는 메시지를 보호하는 데 사용되는 개인 키를 포함하는 키 저장소에 대한 액세스가 필요합니다.

MCA 인터셉션을 수행하도록 구성된 Advanced Message Security (AMS) MQI 클라이언트 또는 큐 관리자에는 PKCS#11 암호화 하드웨어 또는 메시지를 보호하는 데 사용되는 개인 키를 포함하는 PEM 파일에 대한 액세스가 필요할 수 있습니다.

이러한 키 저장소에 액세스하려면 `keystore.conf`라는 AMS 구성 파일에 비밀번호를 제공해야 합니다. **runamscred** 명령을 사용하여 `keystore.conf` 파일에 포함된 민감한 정보를 보호하십시오. 예:

```
runamscred -f <keystore configuration file>
```

**runamscred** 명령은 **-f** 매개변수를 사용하여 지정된 파일 내의 민감한 매개변수를 보호합니다.

IBM MQ 설치에서는 두 개의 **runamscred** 명령을 사용할 수 있습니다.

- `<IBM MQ installation root>/bin` 에 있는 MQI **runamscred** 명령
- `<IBM MQ installation root>/java/bin` 에 있는 Java **runamscred** 명령

**주의:** 호환성을 보장하기 위해

1. Java **runamscred** 명령을 사용하여 Java AMS 클라이언트에서 사용되는 구성 파일을 보호하고 MQI **runamscred** 명령을 사용하여 AMS를 사용하는 IBM MQ MQI clients 의 구성 파일을 보호하십시오.
2. **runamscred** 명령을 실행한 후 필요한 모든 민감한 정보가 보호되는지 확인하십시오.
3. AMS 사용 애플리케이션에 정상적으로 보호된 비밀번호를 포함하는 파일을 제공하십시오.

기본적으로 **runamscred** 명령은 기본 초기 키를 사용하여 구성 파일의 비밀번호를 암호화합니다. 특정 초기 키를 사용하여 비밀번호를 암호화하려면 다음 메커니즘 중 하나를 사용하여 우선순위에 따라 초기 키를 포함하는 파일의 이름을 지정하십시오.

1. **runamscred** 명령에 대한 **-sf** 매개변수.
2. **MQS_AMSCRED_KEYFILE** 환경 변수.
3. keystore.conf 구성 파일의 **amscred.keyfile** 매개변수.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

**runamscred** 명령을 실행하여 AMS 구성에서 비밀번호를 암호화할 때 초기 키 파일을 지정하는 경우, AMS 애플리케이션이 실행될 때 동일한 초기 키 파일도 지정해야 합니다. 다음 메커니즘을 사용하여 초기 키 파일의 이름을 우선순위 순서대로 지정할 수 있습니다.

1. **MQS_AMSCRED_KEYFILE** 환경 변수.
2. keystore.conf 구성 파일의 **amscred.keyfile** 매개변수.

기본적으로 **runamscred** 명령은 IBM MQ 9.2이전의 AMS 버전과 호환되지 않는 보호 시스템으로 신임 정보를 보호합니다. IBM MQ 9.2이전 버전과 호환 가능한 신임 정보 보호 시스템으로 구성 파일을 보호하려면 **runamscred** 명령이 실행될 때 **-sp 0** 매개변수를 지정하십시오.

## Managed File Transfer

Managed File Transfer (MFT) 는 큐 관리자 및 기타 자원에 액세스하는 데 필요한 신임 정보를 다음 XML 특성 파일에 저장합니다.

**MQMFTCredentials.xml**
이 파일에는 다음 신임 정보가 포함되어 있습니다.

- 에이전트, 조정 및 명령 큐 관리자에 연결하는 데 사용되는 신임 정보입니다.
- 보안 통신에 사용되는 키 저장소에 액세스하는 데 사용되는 비밀번호입니다.

**ProtocolBridgeCredentials.xml**
이 파일에는 FTP, SFTP및 FTPS와 같은 프로토콜 서버에 연결하는 데 사용되는 신임 정보가 포함되어 있습니다.

**ConnectDirectCredentials.xml**
이 파일에는 Connect:Direct® 에이전트가 Connect:Direct 노드에 연결하는 데 사용하는 신임 정보가 포함되어 있습니다.

이러한 파일에 저장된 민감한 정보를 보호하려면 fteObfuscate 명령을 사용하십시오. **-f** 플래그를 사용하여 보호할 파일의 이름을 지정하십시오. 예를 들면, 다음과 같습니다.

```
fteObfuscate -f <File to protect>
```

기본적으로 **fteObfuscate** 명령은 기본 초기 키를 사용하여 신임 정보를 보호합니다. 특정 초기 키로 신임 정보를 보호하려면 **-sf** 매개변수를 사용하여 초기 키를 포함하는 파일의 경로를 지정하십시오. 예를 들면, 다음과 같습니다.

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

**주의:**

1. **fteObfuscate**를 실행한 후 모든 민감한 정보가 보호되는지 확인하십시오.
2. 보호된 파일을 MFT에 정상적으로 제공하십시오.

MFT 구성에서 신임 정보를 보호하기 위해 **fteObfuscate** 명령을 실행할 때 초기 키 파일을 지정하는 경우, MFT 가 시작될 때와 동일한 초기 키 파일도 지정해야 합니다. 다음 메커니즘을 사용하여 초기 키 파일의 이름을 우선순위 순서대로 지정할 수 있습니다.

1. **com.ibm.wmqfte.cred.keyfile** Java 시스템 특성입니다.

**참고:** IBM MQ 9.3.1 및 IBM MQ 9.3.0 Fix Pack 10이전에는 이 Java 시스템 특성의 이름 철자가 **com.ibm.wqmfte.cred.keyfile**(으) 로 잘못되었습니다. IBM MQ 9.3.1 및 IBM MQ 9.3.0 Fix Pack 10부터 Managed File Transfer 는 Java 시스템 특성의 두 버전을 모두 사용하여 이전 버전과의 호환성을 유지합니다. 두 Java 시스템 특성이 모두 설정되면 철자가 올바른 특성 **com.ibm.wmqfte.cred.keyfile** 의 값이 사용됩니다.

2. 에이전트, 로거, 명령 및 조정 특성 파일의 특성입니다.
3. `installation.properties` 파일의 **commonCredentialsKeyFile** 특성.

자세한 정보는 535 페이지의 『MFT 에서 저장된 신임 정보 암호화』의 내용을 참조하십시오.

기본적으로 **fteObfuscate** 명령은 IBM MQ 9.2이전의 MFT 버전과 호환되지 않는 보호 시스템으로 신임 정보를 보호합니다. IBM MQ 9.2이전 버전과 호환 가능한 신임 정보 보호 시스템으로 구성 파일을 보호하려면 **fteObfuscate** 명령이 실행될 때 **-sp 0** 매개변수를 지정하십시오.

## IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) 구성 파일은 다양한 자원에 액세스하는 데 사용되는 비밀번호를 포함할 수 있습니다.

**mqiptPW** 명령을 사용하여 MQIPT 구성 파일의 비밀번호를 보호하십시오.

**mqiptPW** 명령은 암호화할 비밀번호를 입력하도록 프롬프트를 표시하고 암호화된 비밀번호를 리턴합니다. 암호화된 비밀번호를 MQIPT 구성 파일에 복사하십시오.

기본적으로 **mqiptPW** 명령은 기본 초기 키를 사용하여 비밀번호를 암호화합니다. 특정 초기 키를 사용하여 비밀번호를 암호화하려면 **-sf** 매개변수를 사용하여 초기 키를 포함하는 파일에 대한 경로를 지정하십시오.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

자세한 정보는 비밀번호 암호화 키 지정을 참조하십시오.

키 저장소 비밀번호를 암호화할 때 초기 키 파일을 지정하는 경우 MQIPT 가 시작될 때 동일한 초기 키 파일도 지정해야 합니다. 다음 메커니즘을 사용하여 초기 키 파일의 이름을 우선순위 순서대로 지정할 수 있습니다.

1. MQIPT를 시작하는 데 사용되는 명령의 **-sf** 매개변수입니다.
2. **MQS_MQIPTCRED_KEYFILE** 환경 변수.
3. **com.ibm.mq.ipt.cred.keyfile** Java 특성.
4. MQIPT 홈 디렉토리에 있는 `mqipt_cred.key` 파일. MQIPT 홈 디렉토리는 MQIPT 구성 파일을 포함하는 디렉토리입니다.

기본적으로 **mqiptPW** 명령은 IBM MQ 9.2이전의 MQIPT 버전과 호환되지 않는 보호 시스템으로 신임 정보를 보호합니다. IBM MQ 9.2이전 버전과 호환 가능한 신임 정보 보호 시스템으로 비밀번호를 보호하려면 IBM MQ 9.2 이전 버전에서 지원되는 **mqiptPW** 명령 구문을 사용하십시오.

## 암호화 하드웨어를 사용하는 IBM MQ clients

TLS 통신에서 사용되는 개인 키 및 인증서를 저장하기 위해 PKCS #11 암호화 하드웨어를 사용하도록 IBM MQ 클라이언트를 구성할 수 있습니다. PKCS #11 디바이스에 액세스하려면 IBM MQ client에 제공되는 구성 문자열의 일부로 비밀번호를 제공해야 합니다.

**중요사항:** MQSCO 구조에서 **CryptoHardware** 필드를 사용하여 제공된 비밀번호 또는 큐 관리자 **SSLCRYP** 속성은 이 메커니즘을 사용하여 보호할 수 없습니다.

IBM MQ 설치 디렉토리의 `bin` 폴더에서 찾을 수 있는 **runp11cred** 명령을 사용하여 이 비밀번호를 보호할 수 있습니다.

**runp11cred** 명령은 암호화할 비밀번호를 입력하도록 프롬프트를 표시하고 암호화된 비밀번호를 리턴합니다. 암호화된 비밀번호를 암호화 하드웨어 구성 문자열에 복사해야 합니다.

예를 들어, 암호화 하드웨어 구성 문자열이 다음과 같은 경우:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

**runp11cred** 명령이 비밀번호를 입력하도록 프롬프트를 표시하면 Passw0rd를 입력하십시오. 명령은 다음과 유사한 문자열을 리턴합니다.

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

암호화된 비밀번호를 포함하는 다음 문자열을 제공하려면 암호화된 하드웨어 구성 문자열의 비밀번호를 **runp11cred** 명령이 리턴하는 문자열로 대체하십시오.

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

IBM MQ client 애플리케이션이 실행될 때 다음 방법 중 하나로 암호화된 비밀번호를 포함하는 암호화 하드웨어 구성 문자열을 지정하십시오.

- 클라이언트 구성 파일의 SSL 스탠자에 있는 **SSLCryptoHardware** 속성.
- **MQSSLCRYP** 환경 변수.

기본적으로 **runp11cred** 명령은 기본 초기 키를 사용하여 비밀번호를 암호화합니다. 사용자 고유의 초기 키로 비밀번호를 보호하려면 다음 메커니즘 중 하나를 사용하여 초기 키를 포함하는 파일의 이름을 우선순위 순서대로 지정하십시오.

1. **runp11cred** 명령에 대한 **-sf** 매개변수.
2. **MQS_SSLCRYP_KEYFILE** 환경 변수.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

키 저장소 비밀번호를 암호화할 때 초기 키 파일을 지정하는 경우 IBM MQ client 가 실행될 때 초기 키를 포함하는 파일의 이름도 지정해야 합니다. 우선순위에 따라 다음 메커니즘 중 하나를 사용하여 초기 키 파일 이름을 지정하십시오.

1. **MQS_SSLCRYP_KEYFILE** 환경 변수.
2. 클라이언트 구성 파일의 **SSL** 스탠자에 있는 **SSLCryptoHardwareKeyFile** 속성.

## IBM MQ 큐 관리자

IBM MQ 큐 관리자는 비밀번호를 내부적으로 여러 속성에 저장합니다. 예를 들어, 큐 관리자 **KEYRPWD** 속성입니다. 큐 관리자는 비밀번호가 디스크의 파일에 저장되기 전에 자동으로 비밀번호를 암호화합니다.

큐 관리자 TLS키 저장소에 대한 비밀번호는 IBM MQ 비밀번호 보호 시스템 또는 키 저장소 스태쉬 파일을 사용하여 보호할 수 있습니다. 이러한 두 가지 방법에 대한 자세한 정보는 279 페이지의 『AIX, Linux, and Windows에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

큐 관리자가 비밀번호를 암호화할 때 사용자 고유의 초기 키를 지정하지 않으면 기본 초기 키가 사용됩니다. 사용자 고유의 초기 키를 사용하려면 암호화된 큐 관리자 속성을 설정하기 전에 큐 관리자 **INITKEY** 속성을 고유하고 강력한 키로 설정하십시오.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

**경고:** 암호화된 속성의 값을 설정한 후 초기 키가 수정되는 경우 암호화된 속성은 새 초기 키로 다시 암호화되지 않습니다. 따라서 키 저장소 비밀번호 문구를 다시 제공하지 않고 초기 키를 변경하면 IBM MQ 가 키 저장소 비밀번호 문구를 복호화할 수 없고 키 저장소에 액세스할 수 없게 됩니다.

자세한 정보는 INITKEY를 참조하십시오.

## IBM MQ C 클라이언트 애플리케이션

IBM MQ C 클라이언트 라이브러리에는 특정 보안 자원에 액세스하기 위한 비밀번호가 필요합니다. 예를 들어, TLS를 사용하여 큐 관리자에 연결하는 애플리케이션의 TLS키 저장소입니다.

키 저장소 비밀번호는 IBM MQ 비밀번호 보호 시스템 또는 키 저장소 스태쉬 파일을 사용하여 보호할 수 있습니다. 이러한 두 가지 방법에 대한 자세한 정보는 279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

IBM MQ 비밀번호 보호 시스템으로 비밀번호를 보호하려면 **runmqicred** 명령을 사용하십시오. 명령은 *MQ_INSTALLATION_PATH*/bin 디렉토리에 있습니다.

**runmqicred** 명령은 암호화할 비밀번호를 입력하도록 프롬프트를 표시하고 암호화된 비밀번호를 리턴합니다. 암호화된 비밀번호는 일반 텍스트 비밀번호 대신 클라이언트 애플리케이션에서 사용할 수 있습니다.

예를 들어, *MQKEYRPWD* 환경 변수를 사용하여 TLS키 저장소 비밀번호를 제공하도록 선택하고 TLS키 저장소 비밀번호가 Passw0rd인 경우입니다. **runmqicred**를 실행할 때 프롬프트가 표시되면 Passw0rd 를 입력하십시오. 명령은 다음과 유사한 문자열을 리턴합니다.

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

이 문자열을 *MQKEYRPWD* 환경 변수의 값으로 설정하십시오.

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

기본적으로 **runmqicred** 명령은 기본 초기 키를 사용하여 비밀번호를 암호화합니다. 사용자의 초기 키를 사용하여 비밀번호를 보호하려면 다음 메커니즘 중 하나를 사용하여 키를 포함하는 파일의 이름을 우선순위 순서대로 지정하십시오.

1. **runmqicred** 명령에 대한 **-sf** 매개변수.
2. *MQS_MQI_KEYFILE* 환경 변수.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

비밀번호를 암호화할 때 초기 키 파일을 지정하는 경우, 클라이언트 애플리케이션이 실행될 때 초기 키를 사용할 수 있도록 해야 합니다.

자세한 정보는 284 페이지의 『AIX, Linux, and Windows 의 IBM MQ MQI client 에 대한 키 저장소 비밀번호 제공』의 내용을 참조하십시오.

## 기본 HA 구성

V 9.4.0

인스턴스 간 기본 HA 로그 복제 트래픽은 TLS를 사용하여 암호화할 수 있습니다. 로그 복제 트래픽을 보호하는 데 사용되는 인증서는 qm.ini 파일의 **NativeHALocalInstance** 스탠자에 지정된 키 저장소에 저장됩니다.

키 저장소 비밀번호는 IBM MQ 비밀번호 보호 시스템 또는 키 저장소 스태쉬 파일을 사용하여 보호할 수 있습니다. 이러한 두 가지 방법에 대한 자세한 정보는 279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

IBM MQ 비밀번호 보호 시스템을 사용하여 고유 HA키 저장소 비밀번호를 보호하려면 **runmqicred** 명령을 사용하십시오.

**runmqicred** 명령은 암호화할 비밀번호를 입력하도록 프롬프트를 표시하고 암호화된 비밀번호를 리턴합니다. 일반 텍스트 비밀번호 대신 암호화된 비밀번호를 사용해야 합니다. qm.ini 파일의 **NativeHALocalInstance** 스탠자에 있는 **KeyRepositoryPassword** 속성의 값을 명령이 리턴하는 암호화된 비밀번호로 설정하십시오.

기본적으로 **runmqicred** 명령은 기본 초기 키를 사용하여 비밀번호를 암호화합니다. 사용자의 초기 키를 사용하여 비밀번호를 보호하려면 다음 메커니즘 중 하나를 사용하여 키를 포함하는 파일의 이름을 우선순위 순서대로 지정하십시오.

1. **runmqicred** 명령에 대한 **-sf** 매개변수.
2. *MQS_MQI_KEYFILE* 환경 변수.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

키 저장소 비밀번호를 암호화할 때 초기 키 파일을 지정하는 경우 `qm.ini` 파일의 **NativeHALocalInstance** 스탠자에서 **InitialKeyFile** 속성을 사용하여 동일한 초기 키 파일도 지정해야 합니다.

자세한 정보는 qm.ini 파일의NativeHALocal인스턴스 스탠자를 참조하십시오.

### IBM MQ 큐 관리자 ( qm.ini 파일의AuthToken 스탠자)

Linux AIX V 9.4.0

IBM MQ 9.3.4부터 AIX 또는 Linux 시스템에서 실행되는 IBM MQ 큐 관리자에 연결하는 IBM MQ MQI clients는 인증 토큰을 사용하여 큐 관리자를 인증할 수 있습니다. 큐 관리자는 인증 토큰을 승인하고 토큰 발행자의 공개 키 인증서 또는 토큰 서명에 사용되는 비밀 키에 액세스할 수 있도록 구성되어야 합니다. 신뢰할 수 있는 발행자의 공개 키 인증서 또는 비밀 키를 포함하는 키 저장소는 비밀번호로 보안됩니다.

키 저장소 비밀번호는 IBM MQ 비밀번호 보호 시스템 또는 키 저장소 스태쉬 파일을 사용하여 보호할 수 있습니다. 이러한 두 가지 방법에 대한 자세한 정보는 279 페이지의 『AIX, Linux, and Windows 에서 키 저장소 비밀번호 암호화』의 내용을 참조하십시오.

IBM MQ 비밀번호 보호 시스템으로 인증 토큰 키 저장소 비밀번호를 보호하려면 **runqmcred** 명령을 사용하여 비밀번호를 암호화하십시오.

**runqmcred** 명령은 암호화할 비밀번호를 입력하도록 프롬프트를 표시하고 암호화된 비밀번호를 리턴합니다. 일반 텍스트 비밀번호 대신 암호화된 비밀번호를 사용해야 합니다. 암호화된 비밀번호를 파일에 복사하고 `qm.ini` 파일에 있는 **AuthToken** 스탠자의 **KeyStorePwdFile** 속성에 파일에 대한 경로를 포함시키십시오.

기본적으로 **runqmcred** 명령은 기본 초기 키를 사용하여 비밀번호를 암호화합니다. 특정 초기 키를 사용하여 비밀번호를 암호화하려면 **-sf** 매개변수를 사용하여 초기 키를 포함하는 파일에 대한 경로를 지정하십시오.

**경고:** 기본 초기 키는 모든 IBM MQ 설치에 대해 동일합니다. 비밀번호를 안전하게 보호하려면 비밀번호를 암호화할 때 설치에 고유한 초기 키를 제공하십시오.

**중요사항:** 비밀번호를 암호화할 때 초기 키를 제공하는 경우 큐 관리자가 비밀번호를 복호화할 수 있도록 큐 관리자 **INITKEY** 속성에 동일한 초기 키를 지정해야 합니다. 큐 관리자 **INITKEY** 속성이 이미 설정된 경우 **runqmcred** 명령을 실행할 때 동일한 초기 키를 사용하십시오. 큐 관리자 **INITKEY** 속성에 대한 자세한 정보는 INITKEY를 참조하십시오.

예를 들어, `/home/initial.key`파일에서 초기 키를 사용하여 인증 토큰 키 저장소 비밀번호를 암호화하려면 다음 명령을 실행하십시오.

```
runqmcred -sf /home/initial.key
```

자세한 정보는 312 페이지의 『로컬 키 저장소를 사용하여 인증 토큰을 승인하도록 큐 관리자 구성』의 내용을 참조하십시오.

## 비밀번호 암호화를 통한 보호 한계

IBM MQ 는 다양한 구성 파일에 저장된 비밀번호에 대해 AES-128 암호화를 지원합니다. 고급 암호화 표준 (AES) 암호화를 사용하여 IBM MQ 구성에서 비밀번호를 보호하는 경우 제공되는 보호에 대한 한계를 이해해야 합니다.

IBM MQ 구성 파일에서 비밀번호를 암호화하는 것이 비밀번호가 안전하거나 보호되는 것을 의미하지는 않습니다. 암호화된 비밀번호에 액세스할 수 있지만 암호화 키를 모르는 사용자가 비밀번호를 쉽게 복구하지 못하게 합니다. IBM MQ 프로세스는 사용할 일반 텍스트 비밀번호를 얻기 위해 암호화된 비밀번호 및 복호화 키 모두에 대한 액세스가 필요합니다. 두 데이터 항목 모두 IBM MQ에 액세스할 수 있는 위치의 파일 시스템에 저장되어야 합니다. 구성 파일에 있는 비밀번호를 암호화하는 사용자는 암호화 키에 대한 액세스 권한도 필요합니다. 공격자가 IBM MQ와 동일한 파일 세트에 액세스할 수 있는 경우, AES 암호화를 비밀번호에 적용하면 최소한의 보호 레벨만 제공됩니다.

그럼에도 불구하고 저장된 비밀번호를 암호화하는 것은 비밀번호가 우연히 노출되는 것을 방지하고 복호화 키도 공유되지 않는 경우 구성 파일을 공유할 수 있도록 하므로 고려해야 합니다.

암호 해독 키를 포함하는 파일이 공유되지 않도록 하는 것 외에도, 파일이 시스템의 다른 사용자로부터 보호되도록 주의해야 합니다. IBM MQ 구성 파일은 모든 사용자가 액세스할 수 있지만 복호화 키를 포함하는 파일에 대한 권한을 필요한 최소한으로 제한하십시오. IBM MQ 프로세스가 실행되는 사용자 ID에는 복호화 키를 포함하는 파일을 읽을 수 있는 액세스 권한이 부여되어야 합니다. 그러나 파일을 그룹 또는 시스템의 모든 사용자에게 읽을 수 있는 액세스 권한을 부여할 필요는 없습니다.

# 데이터베이스 인증의 보호 세부사항

사용자 이름 및 비밀번호 인증을 사용하여 데이터베이스 관리자에 연결하는 경우에는 비밀번호를 `qm.ini` 파일에 일반 텍스트로 저장하지 않도록 MQ XA 인증서 저장소에 저장할 수 있습니다.

## 자원 관리자를 위해 XAOpenString 업데이트

신임 정보 저장소를 사용하려면 `qm.ini` 파일에서 XAOpenString을 수정해야 합니다. 문자열은 데이터베이스 관리자에 연결하는 데 사용됩니다. XAOpenString 문자열 내에서 사용자 이름 및 비밀번호가 대체되는 위치를 식별하기 위해 대체 가능한 필드를 지정합니다.

- +USER+ 필드는 XACredentials 저장소에 저장된 사용자 이름 값으로 대체됩니다.
- +PASSWORD+ 필드는 XACredentials 저장소에 저장된 비밀번호 값으로 대체됩니다.

다음 예는 데이터베이스에 연결하기 위해 신임 정보 파일을 사용하기 위해 XAOpenString을 수정하는 방법을 보여줍니다.

**Db2 데이터베이스에 연결**

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

**Oracle 데이터베이스에 연결**

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

## MQ XA 신임 정보 저장소에 대한 데이터베이스의 신임 정보에 대한 작업

`qm.ini` 파일을 대체 가능한 신임 정보 문자열로 업데이트한 후에는 **setmqxacred** 명령을 사용하여 사용자 이름 및 비밀번호를 MQ 신임 정보 저장소에 추가해야 합니다. **setmqxacred**를 사용하여 기존 신임 정보를 수정하고, 신임 정보를 삭제하거나 신임 정보를 나열할 수도 있습니다. 다음 예는 일부 일반적 유스 케이스를 제공합니다.

**신임 정보 추가**

다음 명령은 자원 `mqdb2`에 대해 큐 관리자 `QM1`의 사용자 이름 및 비밀번호를 안전하게 저장합니다.

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

**신임 정보 업데이트**

데이터베이스에 연결하는 데 사용되는 사용자 이름 및 비밀번호를 업데이트하려면 **setmqxacred** 명령을 새 사용자 이름 및 비밀번호로 재발행하십시오.

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

변경사항을 적용하려면 큐 관리자를 재시작해야 합니다.

**신임 정보 삭제**
다음 명령은 신임 정보를 삭제합니다.

```
setmqxacred -m QM1 -x mydb2 -d
```

**신임 정보 나열**
다음 명령은 신임 정보를 나열합니다.

```
setmqxacred -m QM1 -l
```

**관련 참조**
**setmqxacred**

# Managed File Transfer 보호

설치 직후 수정되지 않은 상태에서 Managed File Transfer는 보호된 환경에서 테스트 또는 평가 목적에 적합할 수 있는 보안 레벨을 가지고 있습니다. 하지만 프로덕션 환경에서는 파일 전송 조작을 시작할 수 있는 사용자, 전송 중인 파일을 읽고 쓸 수 있는 사용자 및 파일의 무결성을 보호하는 방법에 대한 적절한 제어를 고려해야 합니다.

**관련 태스크**
MFT 고유 자원에 대한 그룹 권한 제한
MFT 고유 자원에 대한 권한 관리
596 페이지의 『Managed File Transfer과(와) 함께 Advanced Message Security 사용』
이 시나리오는 Managed File Transfer를 통해 전송되는 데이터에 대한 메시지 개인정보 보호를 제공하도록 Advanced Message Security 를 구성하는 방법을 설명합니다.

**관련 참조**
파일 시스템에 액세스하기 위한 MFT의 권한
commandPath MFT 특성
MFT 에이전트 로그 및 상태 메시지를 발행할 수 있는 권한

## MFT 에서 저장된 신임 정보 암호화

Managed File Transfer (MFT) 에는 두 개의 XML 파일에 저장된 여러 사용자 ID및 신임 정보가 필요하며 **fteObfuscate** 명령을 사용하여 이를 난독화할 수 있습니다.

### 신임 정보 파일

**MQMFTCredentials.xml**
이 파일은 에이전트와 조정 및 명령 큐 관리자에 연결하는 데 필요한 사용자 ID 및 신임 정보를 포함합니다. 큐 관리자에 대한 보안 연결을 위해 키 저장소에 액세스하는 신임 정보도 동일한 파일에 저장됩니다.

MQMFTCredentials.xml 파일의 위치를 정의하는 특성 값에 대한 세부사항은 538 페이지의 『MFT 및 IBM MQ 연결 인증』의 내용을 참조하십시오.

**ProtocolBridgeCredentials.xml**
이 파일은 프로토콜 서버에 연결하는 데 필요한 사용자 ID 및 신임 정보를 포함합니다.

### fteObfuscate 명령을 사용하여 신임 정보 암호화

**fteObfuscate** 명령은 다음 매개변수를 승인합니다.

- **-f** *credentials_file_name* (필수)

  **참고:** Deprecated 이 매개변수는 IBM MQ 9.2.0에서 더 이상 사용되지 않는 **-credentialsFile** 매개변수를 대체합니다.
- **-sp** 보호 모드

- **-sf** 신임 키_파일
- **-o** *output_file_name*

매개변수에 대한 자세한 정보는 **fteObfuscate**를 참조하십시오.

보호 모드 또는 신임 정보 키 파일을 지정하지 않은 경우 명령은 기본 보호 모드를 사용하고, 최신 알고리즘을 사용하지만, 고정 키로 신임 정보를 암호화합니다.

0의 보호 모드를 지정하고 신임 정보 키 파일을 지정하지 않은 경우 명령은 제품의 이전 릴리스에서와 같이 작동합니다. 더 이상 사용되지 않는 보호의 사용을 나타내는 경고 메시지가 콘솔에 표시됩니다.

0의 보호 모드를 지정하고 신임 정보 키 파일을 지정하는 경우 보호 모드 0을 사용할 때 키 파일을 지정하는 것이 올바르지 않음을 나타내는 오류 출력이 콘솔에 표시됩니다.

1의 보호 모드를 지정하고 신임 정보 키 파일을 지정하지 않는 경우 명령은 최신 알고리즘을 사용하지만, 고정 키로 신임 정보를 암호화합니다.

1의 보호 모드를 지정하고 신임 정보 키 파일을 지정하는 경우 명령은 최신 알고리즘으로 신임 정보를 암호화합니다.

1의 보호 모드를 지정하거나 보호 모드를 지정하지 않고 존재하지 않는 신임 정보 키 파일을 지정하는 경우 파일이 없음을 나타내는 오류가 콘솔에 출력됩니다.

1의 보호 모드를 지정하거나 보호 모드를 지정하지 않고 읽을 수 없는 신임 정보 키 파일을 지정하는 경우 파일을 읽을 수 없음을 나타내는 오류가 콘솔에 출력됩니다.

2의 보호 모드를 지정하고 신임 정보 키 파일을 지정하지 않으면 명령은 보호 모드 2를 사용하여 최신 알고리즘 및 암호화할 고정 키를 사용하여 신임 정보를 암호화합니다.

2의 보호 모드를 지정하고 신임 정보 키 파일을 지정하는 경우, 명령은 보호 모드 2를 사용하여 최신 알고리즘 및 암호화할 사용자 지정 키를 사용하여 신임 정보를 암호화합니다.

2의 보호 모드를 지정하거나 보호 모드를 지정하지 않고 존재하지 않는 신임 정보 키 파일을 지정하는 경우 파일이 없음을 나타내는 오류가 콘솔에 출력됩니다.

2의 보호 모드를 지정하거나 보호 모드를 지정하지 않고 읽을 수 없는 신임 정보 키 파일을 지정하는 경우 파일을 읽을 수 없음을 나타내는 오류가 콘솔에 출력됩니다.

## 신임 정보 복호화

다양한 위치에서 초기 키 파일에 대한 경로를 지정할 수 있습니다. 기본값 이외의 초기 키를 사용하여 암호화된 신임 정보를 복호화하려면 초기 키를 포함하는 파일 이름을 다음 방법 중 하나로 MFT에 이 우선 순위로 제공해야 합니다.

1. Java 시스템 특성을 사용하여 다음을 수행하십시오.

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

   **참고:**

   - IBM MQ 9.3.1 및 IBM MQ 9.3.0 Fix Pack 10이전에는 이 Java 시스템 특성의 이름 철자가 제품 코드에서 `com.ibm.wqmfte.cred.keyfile`로 잘못 입력되었습니다. IBM MQ 9.3.1 및 IBM MQ 9.3.0 Fix Pack 10부터 특성 이름의 철자가 `com.ibm.wmqfte.cred.keyfile`로 정정됩니다. Managed File Transfer는 사용자가 신임 정보를 암호화하고 복호화하는 데 사용해야 하는 초기 키를 포함하는 파일을 지정했는지 확인할 때 두 버전의 Java 시스템 특성을 모두 사용합니다. 이를 통해 이전의 철자가 잘못된 이름과의 역호환성을 유지하면서 특성 이름의 올바른 철자를 사용할 수 있습니다. 두 Java 시스템 특성이 모두 설정되면 철자가 올바른 특성 `com.ibm.wmqfte.cred.keyfile` 의 값이 사용됩니다.
   - IBM MQ 9.3.1 및 IBM MQ 9.3.0 Fix Pack 10이전에는 `com.ibm.wqmfte.cred.keyfile`특성을 사용하십시오.

   .
2. 에이전트, 명령, 조정 또는 로거 특성 파일에서 특성을 설정하여. 특성 파일의 이름 및 여기에 설정해야 하는 특성이 다음 표에 표시되어 있습니다.

| 특성 파일 | 특성 이름 |
|---|---|
| agent.properties | agentCredentialsKeyFile |
| command.properties | commandCredentialsKeyFile |
| coordination.properties | coordinationCredentialsKeyFile |
| logger.properties | loggerCredentialsKeyFile |

3. installation.properties 파일에서.

   개별 특성 파일에 특성을 추가하는 대신, 에이전트, 로거 및 명령이 동일한 특성을 사용할 수 있도록 기존 공통 `installation.properties` 파일에 **commonCredentialsKeyFile** 특성을 추가할 수 있습니다.

여러 위치에 다양한 **CredentialsKeyFile** 특성을 정의한 경우 다음을 수행하십시오.

- 에이전트 및 로거에 사용되는 신임 정보 키 파일의 경로는 해당 에이전트 또는 로거의 `output0.log` 파일에 로그됩니다.
- 명령에 사용되는 신임 정보 키 파일의 경로가 콘솔에 표시됩니다.

Java 시스템 특성 **com.ibm.wmqfte.cred.keyfile** 은 다른 모든 특성을 대체합니다. 시스템 특성을 설정하지 않은 경우 에이전트는 먼저 `agent.properties` 파일을 보고 이후 `installation.properties` 파일에서 초기 키 파일을 찾습니다.

초기 키 파일을 계속 찾을 수 없고 **fteObfuscate** 명령에서 보호 모드를 1로 설정한 경우, 에이전트는 `output0.log` 파일에 오류 메시지를 로그합니다.

**fteObfuscate** 명령에서 보호 모드를 0으로 설정한 경우 더 이상 사용하지 않음을 알리는 경고 메시지가 로깅됩니다.

로거 및 명령은 초기 키 파일을 찾는 동일한 단계를 따릅니다.

## 프로토콜 브릿지 및 Connect:Direct 브릿지

프로토콜 브릿지는 FTP, SFTP 및 FTPS 서버에 연결하기 위해 특성 파일 `ProtocolBridgeProperties.xml`을 사용합니다. 이 특성 파일은 이러한 서버에 연결하는 데 필요한 연결 속성을 포함합니다.

`ProtocolBridgeProperties.xml` 파일에서 **credentialsFile** 또는 **credentialsKeyFile** 속성 값을 수정하는 경우 브릿지 에이전트를 재시작해야 합니다.

한 가지 속성은 **credentialsFile**이며, 값은 이러한 서버에 연결하는 데 필요한 UID, PWD 또는 키를 포함하는 XML 파일에 대한 경로를 포함합니다. 속성의 기본값은 *ProtocolBridgeCredentials.xml*이며 `MQMFTCredentials.xml` 파일과 마찬가지로 홈 디렉토리에 있습니다.

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

*MQMFTCredentails.xml*과 마찬가지로, **fteObfuscate** 명령으로 *ProtocolBridgeCredentials.xml*을 암호화할 수 있습니다. 복호화 목적을 위해 다음 텍스트에 표시된 대로, 추가 요소 **credentialsKeyFile**을 사용하여 신임 정보 파일에 대한 필요한 경로를 지정할 수 있습니다. 경로는 환경 변수를 포함할 수 있습니다.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

**참고:** `installation.properties`에서 또는 시스템 특성 **com.ibm.wqmfte.cred.keyfile**를 통해 **agentCredentialsKeyFile** 에이전트 특성, **commonCredentialsKeyFile** 특성의 값을 지정해도 **credentialsKeyFile** 속성에 지정된 값에는 영향을 주지 않습니다.

마찬가지로, Connect:Direct 브릿지는 *ConnectDirectNodeProperties.xml*을 사용하여 Connect:Direct 서버에 연결합니다. XML 파일은 신임 정보 XML 파일에 대한 경로를 정의하는 속성과 함께 필요한 연결 정보를 포함합니다. 이 신임 정보 XML 파일은 Connect:Direct 서버에 연결하는 데 필요한 UID 또는 PWD 및 추가 정보를 포함합니다.

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

*ProtocolBridgeCredentials.xml* 파일과 마찬가지로, **fteObfuscate** 명령으로 *ConnectDirectCredentials.xml*을 암호화할 수 있습니다. 복호화 목적을 위해 다음 텍스트에 표시된 대로, 추가 요소 **credentialsKeyFile**을 사용하여 신임 정보 파일에 대한 필요한 경로를 지정할 수 있습니다. 경로는 환경 변수를 포함할 수 있습니다.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

**참고: agentCredentialsKeyFile** 에이전트 특성, `installation.properties`의 **commonCredentialsKeyFile** 특성 또는 시스템 특성 **com.ibm.wqmfte.cred.keyfile**에 대한 값을 지정해도 **credentialsKeyFile** 속성에 지정된 값에는 영향을 주지 않습니다.

*ProtocolBridgeProperties.xml* 파일에서 **credentialsFile** 요소를 지정하지 않고 **credentialsKeyFile** 요소를 지정할 수 있습니다.

**credentialsFile** 요소를 지정하지 않으면 기본 신임 정보 파일, *ProtocolBridgeCredentials.xml*이 프로토콜 브릿지 에이전트에 의해 사용되고, **credentialsKeyFile** 속성에 지정된 키 파일의 값을 사용하여 신임 정보 파일을 복호화합니다.

마찬가지로, *ConnectDirectNodeProperties.xml* 파일에서 **credentialsFile** 요소를 지정하지 않고 **credentialsKeyFile** 요소를 지정할 수 있습니다.

**credentialsFile** 요소를 지정하지 않으면 기본 신임 정보 파일, *ConnectDirectCredentials.xml*이 Connect:Direct 브릿지에 의해 사용되고, **credentialsKeyFile** 속성에 지정된 키 파일의 값을 사용하여 신임 정보 파일을 복호화합니다.

### z/OS의 데이터 세트에서 키 사용

z/OS

z/OS에서 **MQMFTCredentials**를 지정하고 PDSE를 사용하여 신임 정보 키 파일을 제공할 수 있습니다. 540 페이지의 『Configuring MQMFTCredentials.xml on z/OS』의 내용을 참조하십시오.

**관련 참조**

MFT 명령과 큐 관리자의 연결 조합
MFT 신임 정보 파일 형식
fteObfuscate(민감한 데이터 암호화)

## MFT 및 IBM MQ 연결 인증

연결 인증을 사용하면 큐 관리자가 제공된 사용자 ID 및 비밀번호를 사용하여 애플리케이션을 인증하도록 구성될 수 있습니다. 연관된 큐 관리자에서 보안이 사용되고 신임 정보 세부사항(사용자 ID 및 비밀번호)이 필요한 경우, 큐 관리자에 연결하려면 연결 인증 기능이 사용으로 설정되어야 합니다. 연결 인증은 호환 모드 또는 MQCSP 인증 모드에서 실행할 수 있습니다.

### 신임 정보 세부사항을 제공하는 메소드

많은 Managed File Transfer 명령에서 신임 정보 세부사항을 제공하는 다음 방법을 지원합니다.

**세부사항을 명령행 인수에서 제공**
신임 정보 세부사항은 **-mquserid** 및 **-mqpassword** 매개변수를 사용하여 지정할 수 있습니다. **-mqpassword**가 제공되지 않으면, 사용자는 입력이 표시되지 않는 위치에서 비밀번호를 입력하도록 요청합니다.

**세부사항을 신임 정보 파일 MQMFTCredentials.xml에서 제공**
`MQMFTCredentials.xml` 파일에서 신임 정보 세부사항을 일반 텍스트 또는 난독화된 텍스트로 사전 정의할 수 있습니다.

Multi IBM MQ for Multiplatforms 에서 MQMFTCredentials.xml 파일 설정에 대한 정보는 539 페이지의 『멀티플랫폼에서 MQMFTCredentials.xml 구성』의 내용을 참조하십시오.

z/OS IBM MQ for z/OS 에서 MQMFTCredentials.xml 파일 설정에 대한 정보는 540 페이지의 『Configuring MQMFTCredentials.xml on z/OS』의 내용을 참조하십시오.

### 우선순위

신임 정보 세부사항 판별 우선순위는 다음과 같습니다.

1. 명령행 인수
2. 연관된 큐 관리자 및 명령을 실행하는 사용자별 `MQMFTCredentials.xml` 색인
3. 연관된 큐 관리자별 `MQMFTCredentials.xml` 색인
4. IBM MQ또는 IBM WebSphere MQ 의 이전 릴리스와의 호환성을 허용하기 위해 신임 정보 세부사항이 제공되지 않는 기본 역호환성 모드

**참고:**

- **fteStartAgent** 및 **fteStartLogger** 명령은 명령행 인수 **-mquserid** 또는 **-mqpassword**를 지원하지 않으며 `MQMFTCredentials.xml` 파일을 사용하는 경우에만 신임 정보 세부사항을 지정할 수 있습니다.
- z/OS

  z/OS에서는 사용자의 비밀번호가 소문자를 갖는 경우에도 비밀번호가 대문자여야 합니다. 예를 들어 사용자 비밀번호가 "password"인 경우 "PASSWORD"로 입력해야 합니다.

**관련 참조**

MFT 명령과 큐 관리자의 연결 조합

MFT 신임 정보 파일 형식

## 멀티플랫폼에서 MQMFTCredentials.xml 구성

보안이 사용 가능한 상태로 Managed File Transfer (MFT) 가 구성된 경우 연결 인증을 사용하려면 사용자 ID및 비밀번호 신임 정보를 제공하기 위해 큐 관리자와 연결하는 모든 MFT 명령이 필요합니다. 마찬가지로, MFT 로거는 데이터베이스에 연결할 때 사용자 ID및 비밀번호를 지정해야 할 수 있습니다. 이 신임 정보는 MFT 신임 정보 파일에 저장할 수 있습니다.

### 이 태스크 정보

`MQMFTCredentials.xml` 파일의 요소는 `MQMFTCredentials.xsd` 스키마를 따라야 합니다. `MQMFTCredentials.xml`의 형식에 대한 정보는 MFT 신임 정보 파일 형식을 참조하십시오.

`MQ_INSTALLATION_PATH/mqft/samples/credentials` 디렉토리에서 샘플 신임 정보 파일을 찾을 수 있습니다.

MFT 신임 정보 파일은 조정 큐 관리자, 명령 큐 관리자, 개별 에이전트 및 개별 로거에 대해 각각 하나씩 있을 수 있습니다. 또는 토폴로지의 모든 항목에서 사용되는 하나의 파일을 가질 수 있습니다.

MFT 신임 정보 파일의 기본 위치는 다음과 같습니다.

Linux AIX **AIX and Linux**

`$HOME`

Windows **Windows**

`%USERPROFILE%` 또는 `%HOMEDRIVE%%HOMEPATH%`

신임 정보 파일이 다른 위치에 저장된 경우 다음 특성을 사용하여 명령이 검색해야 하는 위치를 지정할 수 있습니다.

표 *97. :* 다양한 명령에 대한 *MQMFTCredentials.xml* 파일의 위치를 정의하는 특성입니다.

| 명령 유형 | 특성 파일 | 특성 이름 |
|---|---|---|
| 조정 큐 관리자에 연결하는 명령 | coordination.properties | coordinationQMgrAuthenticationCredentialsFile |
| 명령 큐 관리자에 연결하는 명령 | connection.properties | connectionQMgrAuthenticationCredentialsFile |

| 표 97. : 다양한 명령에 대한 *MQMFTCredentials.xml* 파일의 위치를 정의하는 특성입니다. *(계속)* | | |
|---|---|---|
| **명령 유형** | **특성 파일** | **특성 이름** |
| 에이전트 프로세스에 연결하는 명령 | agent.properties | agentQMgrAuthenticationCredentialsFile |
| 로거 프로세스에 연결하는 명령 | logger.properties | loggerQMgrAuthenticationCredentialsFile |

| 표 98. : 에이전트 및 로거 프로세스에 대한 *MQMFTCredentials.xml* 파일의 위치를 정의하는 특성입니다. | | |
|---|---|---|
| **명령 유형** | **특성 파일** | **특성 이름** |
| MFT 에이전트 | agent.properties | agentQMgrAuthenticationCredentialsFile |
| MFT loggers | logger.properties | loggerQMgrAuthenticationCredentialsFile |

어떤 명령 및 프로세스가 어떤 큐 관리자에 연결되는지에 대한 세부사항은 어떤 MFT 명령 및 프로세스가 어떤 큐 관리자에 연결되는지를 참조하십시오.

개별 특성 파일에 특성을 추가하는 대신 에이전트, 로거 및 명령이 동일한 특성을 사용할 수 있도록 기존 공통 `installation.properties` 파일에 **commonCredentialsKeyFile** 특성을 추가할 수 있습니다.

신임 정보 파일에는 사용자 ID및 비밀번호 정보가 포함되어 있으므로 권한이 없는 액세스를 방지하려면 특수 권한이 필요합니다.

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

상속이 사용 가능하지 않은지 확인한 후 신임 정보 파일을 사용할 에이전트 또는 로거를 실행하는 사용자 ID를 제외한 모든 사용자 ID를 제거하십시오.

IBM MQ Explorer Managed File Transfer 플러그인에서 MFT 조정 큐 관리자에 연결하는 데 사용되는 신임 정보 세부사항은 구성 유형에 따라 다릅니다.

**글로벌(로컬 디스크의 구성)**

글로벌 구성은 조정 및 명령 특성에 지정된 신임 정보 파일을 사용합니다.

**로컬(IBM MQ Explorer 내에서 정의됨):**

로컬 구성은 IBM MQ Explorer에 있는 연관 큐 관리자의 연결 세부사항 특성을 사용합니다.

**관련 태스크**

542 페이지의 『MFT의 연결 인증 사용』
조정 큐 관리자 또는 명령 큐 관리자와 연결하는 IBM MQ Explorer MFT 플러그인의 연결 인증, 그리고 조정 큐 관리자 또는 명령 큐 관리자와 연결하는 Managed File Transfer 에이전트의 연결 인증은 호환 모드 또는 MQCSP 인증 모드로 실행될 수 있습니다.

IBM MQ File Transfer 구조 작성

**관련 참조**

MFT 신임 정보 파일 형식

MFT 에서 저장된 신임 정보 암호화

**fteObfuscate**: 민감한 데이터 암호화

## z/OS Configuring `MQMFTCredentials.xml` on z/OS

If Managed File Transfer (MFT) is configured with security enabled, connection authentication requires all MFT agents, and commands that connect to a queue manager, to supply user ID and password credentials.

Similarly, MFT loggers might be required to specify a user ID and password when connecting to a database.

This credential information can be stored in the MFT credentials file. Note that the credentials files are optional, however, it is easier to define the file or files that you require before you customize the environment.

In addition to this, if you have credentials files, you receive fewer warning messages. The warning messages inform you that MFT considers that queue manager security is off, and therefore you are not supplying authentication details.

You can find a sample credentials file in the `MQ_INSTALLATION_PATH/mqft/samples/credentials` directory.

Here is an example of an `MQMFTCredentials.xml`file:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

When a job with userid ADMIN needs to connect to queue manager MQPH, it passes user ID *JOHNDOEH* and uses password *cXXXX*.

If the job is run by any other user ID, and connects MQPH, that job passes user ID *NONEH* and password *yXXXX*.

The default location for the `MQMFTCredentials.xml` file is the user's home directory on z/OS UNIX System Services (USS). It is also possible to store the file in either a different location on USS, or in a member within a partitioned data set.

If the credentials file is stored in a different location, then you can use the following properties to specify where the commands should look for it:

*Table 99. : Properties that define the location of the MQMFTCredentials.xml file for various commands.*

| **Type of command** | **Property file** | **Property name** |
|---|---|---|
| Command which connects to the coordination queue manager | coordination.properties | coordinationQMgrAuthenticationCredentialsFile |
| Command which connects to the command queue manager | connection.properties | connectionQMgrAuthenticationCredentialsFile |
| Command that connects to an agent process | agent.properties | agentQMgrAuthenticationCredentialsFile |
| Command that connects to a logger process | logger.properties | loggerQMgrAuthenticationCredentialsFile |

*Table 100. : Properties that define the location of the MQMFTCredentials.xml file for agents and logger processes.*

| **Type of command** | **Property file** | **Property name** |
|---|---|---|
| MFT agents | agent.properties | agentQMgrAuthenticationCredentialsFile |

| *Table 100. : Properties that define the location of the MQMFTCredentials.xml file for agents and logger processes. (continued)* | | |
|---|---|---|
| **Type of command** | **Property file** | **Property name** |
| MFT loggers | logger.properties | loggerQMgrAuthenticationCredentialsFile |

For details about what commands and processes connect to which queue manager, see Which MFT commands and processes connect to which queue manager.

To create the credentials file within a partitioned data set, carry out the following steps:

- Create a PDSE with format VB and logical record length (Lrecl) 200.
- Create a member within the data set, make a note of the data set and member, and add the following code to the member:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!—-credentials information goes here-->
</tns:mqmftCredentials>
```

You can protect the credentials file using a security product, for example, RACF, but the user IDs running the Managed File Transfer commands, and administering the agent and logger processes, need read access to this file.

You can obscure information in this file using the JCL in member BFGCROBS. This takes the file and encrypts the IBM MQ user ID and password. For example member BFGCROBS takes the line

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

and creates

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

If you want to keep the user ID to IBM MQ user ID mapping, you can add comments to the file. For example

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

These comments are unchanged by the obscuring process.

Note that the content is obscured, not strongly encrypted. You should limit which user IDs have access to the file.

**Related tasks**

"멀티플랫폼에서 MQMFTCredentials.xml 구성" on page 539

보안이 사용 가능한 상태로 Managed File Transfer (MFT) 가 구성된 경우 연결 인증을 사용하려면 사용자 ID및 비밀번호 신임 정보를 제공하기 위해 큐 관리자와 연결하는 모든 MFT 명령이 필요합니다. 마찬가지로, MFT 로거는 데이터베이스에 연결할 때 사용자 ID및 비밀번호를 지정해야 할 수 있습니다. 이 신임 정보는 MFT 신임 정보 파일에 저장할 수 있습니다.

## MFT의 연결 인증 사용

조정 큐 관리자 또는 명령 큐 관리자와 연결하는 IBM MQ Explorer MFT 플러그인의 연결 인증, 그리고 조정 큐 관리자 또는 명령 큐 관리자와 연결하는 Managed File Transfer 에이전트의 연결 인증은 호환 모드 또는 MQCSP 인증 모드로 실행될 수 있습니다.

### 이 태스크 정보

MQCSP 인증 모드가 기본값입니다.

CLIENT 전송을 사용하여 큐 관리자에 연결하는 IBM MQ Explorer Managed File Transfer 플러그인 또는 Managed File Transfer 에이전트의 연결 인증에서, 12자보다 긴 비밀번호는 MQCSP 인증 모드에서만 지원됩니다. 호환 모드를 사용한 권한 부여에서 12자보다 긴 비밀번호를 지정하는 경우에는 오류가 발생하며 큐 관리자에 인증되지 않습니다. 진단 메시지: BFGAG0001 - BFGAG9999에서 BFGAG0187E 메시지를 참조하십시오.

### 프로시저

- IBM MQ Explorer에서 조정 큐 관리자 또는 명령 큐 관리자의 연결 인증 모드를 선택하려면 다음 단계를 완료하십시오.

  a) 연결하려는 큐 관리자를 선택하십시오.

  b) 마우스의 오른쪽 단추를 클릭하고 팝업 메뉴에서 **연결 세부사항->특성**을 선택하십시오.

  c) **사용자 ID** 탭을 클릭하십시오.

  d) 사용할 연결 인증 모드에 대한 선택란이 선택되었는지 확인하십시오.

  - 기본적으로 **사용자 ID 호환 모드** 선택란은 선택되어 있지 않습니다. 이는 **사용자 ID 사용** 선택란이 선택된 경우 IBM MQ Explorer가 큐 관리자에 연결하는 데 MQCSP 인증을 사용함을 의미합니다. IBM MQ Explorer가 MQCSP 인증이 아니라 호환 모드를 사용하여 큐 관리자에 연결해야 하는 경우에는 **사용자 ID 사용** 및 **사용자 ID 호환 모드** 선택란이 둘 다 선택되어 있는지 확인하십시오.

- `MQMFTCredentials.xml` 파일을 사용하여 Managed File Transfer 에이전트에 대해 MQCSP 인증 모드를 사용 또는 사용 안함으로 설정하려면 관련 사용자의 `MQMFTCredentials.xml` 파일에 **useMQCSPAuthentication** 매개변수를 추가하십시오.

  **useMQCSPAuthentication** 매개변수는 다음 값을 갖습니다.

  **true**
  큐 관리자에 사용자를 인증하는 데 MQCSP 인증 모드가 사용됩니다.

  `true` 가 기본값입니다. **useMQCSPAuthentication** 매개변수가 지정되지 않은 경우 이는 기본적으로 `true`로 설정되며 큐 관리자에 사용자를 인증하는 데 MQCSP 인증 모드가 사용됩니다.

  **false**
  큐 관리자에 사용자를 인증하는 데 호환 모드가 사용됩니다.

  다음 예는 `MQMFTCredentials.xml` 파일에서 **useMQCSPAuthentication** 매개변수를 설정하는 방법을 보여줍니다.

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

#### 관련 개념

29 페이지의 『MQCSP 비밀번호 보호』
MQCSP 구조에 지정된 인증 신임 정보는 IBM MQ MQCSP 비밀번호 보호 기능을 사용하여 보호하거나 TLS 암호화를 사용하여 암호화할 수 있습니다.

#### 관련 참조

538 페이지의 『MFT 및 IBM MQ 연결 인증』
연결 인증을 사용하면 큐 관리자가 제공된 사용자 ID 및 비밀번호를 사용하여 애플리케이션을 인증하도록 구성될 수 있습니다. 연관된 큐 관리자에서 보안이 사용되고 신임 정보 세부사항(사용자 ID 및 비밀번호)이 필요한 경우, 큐 관리자에 연결하려면 연결 인증 기능이 사용으로 설정되어야 합니다. 연결 인증은 호환 모드 또는 MQCSP 인증 모드에서 실행할 수 있습니다.

MFT 신임 정보 파일 형식

## MFT 샌드박스

전송 과정에서 에이전트가 액세스할 수 있는 파일 시스템의 영역을 제한할 수 있습니다. 에이전트의 액세스가 제한되는 영역을 샌드박스라고 합니다. 전송을 요청하는 사용자 또는 에이전트에 제한을 적용할 수 있습니다.

에이전트가 프로토콜 브릿지 에이전트이거나 Connect:Direct 브릿지 에이전트인 경우 샌드박스는 지원되지 않습니다. IBM MQ 큐로나 큐로부터 전송해야 할 에이전트에는 에이전트 샌드박싱을 사용할 수 없습니다.

**관련 참조**

544 페이지의 『MFT 에이전트 샌드박스에 대한 작업』
Managed File Transfer에 추가 보안 레벨을 추가하기 위해 에이전트가 액세스할 수 있는 파일 시스템 영역을 제한할 수 있습니다.

545 페이지의 『MFT 사용자 샌드박스에 대한 작업』
전송을 요청한 사용자의 MQMD 사용자 이름에 기반하여 파일을 송수신할 수 있는 파일 시스템의 영역을 제한할 수 있습니다.

## MFT 에이전트 샌드박스에 대한 작업

Managed File Transfer에 추가 보안 레벨을 추가하기 위해 에이전트가 액세스할 수 있는 파일 시스템 영역을 제한할 수 있습니다.

IBM MQ 큐로나 큐로부터 전송하는 에이전트에는 에이전트 샌드박싱을 사용할 수 없습니다. 대신 모든 샌드박싱 요구사항에 대한 권장 솔루션인 사용자 샌드박싱을 사용하여 샌드박싱의 IBM MQ 큐에 대한 액세스 제한을 구현할 수 있습니다. 사용자 샌드박싱에 대한 자세한 정보는 545 페이지의 『MFT 사용자 샌드박스에 대한 작업』의 내용을 참조하십시오.

에이전트 샌드박싱을 사용으로 설정하려면 제한하려는 에이전트의 `agent.properties` 파일에 다음 특성을 추가하십시오.

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

설명:

- *restricted_directory_name*은 허용하거나 거부할 디렉토리 경로입니다.
- `!`는 선택사항이며 *restricted_directory_name*에 대한 다음 값이 거부(제외)되도록 지정합니다. `!`를 지정하지 않으면 *restricted_directory_name*은 허용(포함) 경로입니다.
- *separator*는 플랫폼별 구분 기호입니다.

예를 들어, AGENT1이 보유하는 액세스 권한을 `/tmp` 디렉토리만으로 제한하지만 `private` 서브디렉토리에 대한 액세스는 허용하지 않으려는 경우 AGENT1에 속하는 `agent.properties` 파일에서 `sandboxRoot=/tmp:!/tmp/private`과 같이 특성을 설정하십시오.

sandboxRoot 특성에 대해서는 고급 에이전트 특성에 설명되어 있습니다.

에이전트 및 사용자 샌드박싱은 둘 다 프로토콜 브릿지 에이전트 또는 Connect:Direct 브릿지 에이전트에서 지원되지 않습니다.

### AIX, Linux, and Windows 플랫폼의 샌드박스에서 작업

ALW AIX, Linux, and Windows 플랫폼에서 샌드박싱을 사용하면 Managed File Transfer Agent이(가) 읽고 쓸 수 있는 디렉토리가 제한됩니다. 샌드박싱이 활성화된 경우, Managed File Transfer Agent는 허용된 것으로 지정된 디렉토리와 지정된 디렉토리에 들어 있는 서브디렉토리(서브디렉토리가 sandboxRoot에서 거부된 것으로 지정되지 않은 경우)만 읽고 쓸 수 있습니다. Managed File Transfer 샌드박싱은 운영 체제 보안에서 우선권이 없습니다. Managed File Transfer Agent를 시작한 사용자에게는 디렉토리를 읽고 쓸 수 있는 디렉토리에 대한 적절한 운영 체제 레벨의 액세스 권한이 있어야 합니다. 지정된 sandboxRoot 디렉토리(및 서브디렉토리) 외부로 디렉토리가 링크되면 해당 디렉토리로 링크되는 기호 링크가 작동되지 않습니다.

### z/OS에서의 샌드박스 작업

z/OS z/OS에서 샌드박싱은 Managed File Transfer Agent가 읽고 쓸 수 있는 데이터 세트 이름 규정자를 제한합니다. Managed File Transfer Agent를 시작한 사용자에게는 관련된 모든 데이터 세트에 대한 올바른 운영 체제 권한이 있어야 합니다. sandboxRoot 데이터 세트 이름 규정자 값을 큰따옴표로 묶으면 이 값은 일반적인 z/OS 규칙을 따르며 완전한 값으로 처리됩니다. 큰따옴표를 생략하면 sandboxRoot의 접두부에 현재 사용자 ID가 추가됩니다. 예를 들어, sandboxRoot 특성을 `sandboxRoot=//test`로 설정하면 에이전트가 표준 z/OS 표기법으로 `//`*username*`.test.**` 인 데이터 세트에 액세스할 수 있습니다. 런타임 시 완전히 해석된 데이터 세트 이름의 초기 레벨이 sandboxRoot와 일치하지 않는 경우에는 전송 요청이 거부됩니다.

## IBM i 시스템에서의 샌드박스 작업

IBM i IBM i 시스템에서 통합 파일 시스템에 있는 파일의 경우, 샌드박싱은 Managed File Transfer Agent가 읽고 쓸 수 있는 디렉토리를 제한합니다. 샌드박싱이 활성화된 경우, Managed File Transfer Agent는 허용된 것으로 지정된 디렉토리와 지정된 디렉토리에 들어 있는 서브디렉토리(서브디렉토리가 sandboxRoot에서 거부된 것으로 지정되지 않은 경우)만 읽고 쓸 수 있습니다. Managed File Transfer 샌드박싱은 운영 체제 보안에서 우선권이 없습니다. Managed File Transfer Agent를 시작한 사용자에게는 디렉토리를 읽고 쓸 수 있는 디렉토리에 대한 적절한 운영 체제 레벨의 액세스 권한이 있어야 합니다. 지정된 sandboxRoot 디렉토리(및 서브디렉토리) 외부로 디렉토리가 링크되면 해당 디렉토리로 링크되는 기호 링크가 작동되지 않습니다.

**관련 참조**

548 페이지의 『와일드카드 전송 시의 추가 점검사항』
에이전트가 파일을 주고 받을 수 있는 위치를 제한하기 위해 사용자 또는 에이전트 샌드박스를 사용하여 에이전트를 구성하는 경우 해당 에이전트에 대한 와일드카드 전송 시 추가적인 검사가 이루어지도록 지정할 수 있습니다.

544 페이지의 『MFT 에이전트 샌드박스에 대한 작업』
Managed File Transfer에 추가 보안 레벨을 추가하기 위해 에이전트가 액세스할 수 있는 파일 시스템 영역을 제한할 수 있습니다.

MFTagent.properties 파일

## MFT 사용자 샌드박스에 대한 작업

전송을 요청한 사용자의 MQMD 사용자 이름에 기반하여 파일을 송수신할 수 있는 파일 시스템의 영역을 제한할 수 있습니다.

에이전트가 프로토콜 브릿지 에이전트이거나 Connect:Direct 브릿지 에이전트인 경우에는 사용자 샌드박스가 지원되지 않습니다.

사용자 샌드박싱을 사용으로 설정하려면 제한하려는 에이전트의 agent.properties 파일에 다음 특성을 추가하십시오.

```
userSandboxes=true
```

이 특성이 있고 true로 설정된 경우 에이전트는 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name*/UserSandboxes.xml 파일에 있는 정보를 사용하여 전송을 요청하는 사용자가 액세스할 수 있는 파일 시스템의 부분을 판별합니다.

UserSandboxes.xml XML은 0개 이상의 <sandbox> 요소를 포함하는 <agent> 요소로 구성됩니다. 이러한 요소가 어떤 규칙이 어떤 사용자에게 적용되는지를 기술합니다. <sandbox> 요소의 user 속성은 요청의 MQMD 사용자를 찾는 데 사용되는 패턴입니다.

UserSandboxes.xml 파일은 에이전트가 주기적으로 다시 로드하며 이 파일에 대한 유효한 변경사항은 에이전트의 작동에 영향을 미칩니다. 기본 다시 로드 간격은 30초입니다. 이 간격은 agent.properties 파일에서 xmlConfigReloadInterval 에이전트 특성을 지정하여 변경할 수 있습니다.

userPattern="regex" 속성 또는 값을 지정하면, user 속성은 Java 정규식으로 해석됩니다. 자세한 정보는 MFT에서 사용하는 정규식을 참조하십시오.

userPattern="regex" 속성이나 값을 지정하지 않으면 user 속성은 다음 와일드카드 문자를 가진 패턴으로 해석됩니다.

- 별표(*) - 0개 이상의 문자를 표시
- 물음표(?) - 정확히 하나의 문자를 표시

<sandbox> 요소가 파일에 나열된 순서대로 일치가 수행됩니다. 첫 번째 일치만 사용되며 파일에서 이후 모든 잠재적 일치는 무시됩니다. 파일에 지정된 <sandbox> 요소 중 전송 요청 메시지와 연관된 MQMD 사용자와 일치하는 항목이 없을 경우 전송 시 파일 시스템에 액세스할 수 없습니다. MQMD 사용자 이름과 user 속성 사이에서 일치가 발견되면 해당 일치는 <sandbox> 요소 내에서 전송에 적용되는 규칙 세트를 식별합니다. 이 규칙 세트는 전송 중에 읽거나 쓸 수 있는 파일 또는 데이터 세트 항목을 판별하는 데 사용됩니다.

각 규칙 세트는 읽을 수 있는 파일을 식별하는 <read> 요소와 쓸 수 있는 파일을 식별하는 <write> 요소를 지정할 수 있습니다. 규칙 세트에서 <read> 또는 <write> 요소를 생략하면 해당 규칙 세트와 연관된 사용자는 적절하게 읽기 또는 쓰기를 수행할 수 없는 것으로 간주됩니다.

**참고:** UserSandboxes.xml 파일에서 <read> 요소는 <write> 요소 앞에 있어야 하고 <include> 요소는 <exclude> 요소 앞에 있어야 합니다.

각 <read> 또는 <write> 요소에는 파일이 샌드박스에 있고 전송될 수 있는지 여부를 판별하는 데 사용되는 하나 이상의 패턴이 포함되어 있습니다. <include> 및 <exclude> 요소를 사용하여 이러한 패턴을 지정하십시오. <include> 또는 <exclude> 요소의 name 속성은 일치시킬 패턴을 지정합니다. 선택적인 type 속성은 이름 값이 파일인지 또는 큐 패턴인지 여부를 지정합니다. type 속성이 지정되지 않은 경우에는 에이전트가 패턴을 파일 또는 디렉토리 경로 패턴으로 취급합니다. 예를 들면, 다음과 같습니다.

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

<include> 및 <exclude> name 패턴은 에이전트에서 파일, 데이터 세트 또는 큐를 읽거나 쓸 수 있는지 판별하는 데 사용됩니다. 표준 파일 경로, 데이터 세트 또는 큐 이름이 하나 이상의 포함 패턴과 일치하고 제외 패턴과는 일치하지 않는 경우 조작이 허용됩니다. <include> 및 <exclude> 요소의 name 속성을 사용하여 지정된 패턴은 에이전트가 실행 중인 플랫폼에 적합한 경로 구분 기호 및 규칙을 사용합니다. 상대 파일 경로를 지정하는 경우, 에이전트의 transferRoot 특성과 관련하여 경로가 해석됩니다.

큐 제한을 지정하면 QUEUE@QUEUEMANAGER 구문이 지원되고 다음과 같은 규칙이 적용됩니다.

- at 문자(@)가 항목에서 누락된 경우 패턴을 모든 큐 관리자에서 액세스할 수 있는 큐 이름으로 처리합니다. 예를 들어, 패턴이 name인 경우 이는 name@**와 동일한 방법으로 처리됩니다.
- at 문자(@)가 항목의 첫 번째 문자인 경우 패턴을 큐 관리자 이름으로 처리하며 큐 관리자의 모든 큐에 액세스할 수 있습니다. 예를 들어, 패턴이 @name인 경우 이는 **@name과 동일한 방법으로 처리됩니다.

다음 와일드카드 문자는 <include> 및 <exclude> 요소의 name 속성의 부분으로 지정하는 경우 특별한 의미를 갖습니다.

*****
: 단일 별표는 디렉토리 이름 또는 데이터 세트 이름 또는 큐 이름의 규정자에서 0개이상의 문자와 일치합니다.

**?**
: 물음표는 디렉토리 이름 또는 데이터 세트 이름 또는 큐 이름의 규정자에서 정확히 하나의 문자와 일치합니다.

******
: 두 개의 별표 문자가 데이터 세트 이름 또는 큐 이름에서 0개이상의 디렉토리 이름 또는 0개이상의 규정자와 일치합니다. 또한 경로 구분 기호로 끝나는 경로에는 암시적 "**"가 경로의 끝에 추가됩니다. 따라서 /home/user/는 /home/user/**와 동일합니다.

예를 들면, 다음과 같습니다.

- /**/test/**는 경로에 test 디렉토리가 있는 모든 파일과 일치합니다.
- /test/file?는 file 문자열로 시작하고 뒤에 하나의 문자가 오는 /test 디렉토리 내의 모든 파일과 일치합니다.
- c:\test*.txt는 .txt 확장자가 있는 c:\test 디렉토리 내의 모든 파일과 일치합니다.
- c:\test***.txt는 .txt 확장자가 있는 'c:\test 디렉토리 또는 서브디렉토리 중 하나의 모든 파일과 일치합니다.
- z/OS //'TEST.*.DATA'는 첫 번째 규정자가 TEST이고 임의의 두 번째 규정자와 세 번째 규정자가 DATA인 모든 데이터 세트와 일치합니다.

- *@QM1 는 단일 규정자가 있는 큐 관리자 QM1 의 모든 큐와 일치합니다.
- TEST.*.QUEUE@QM1는 첫 번째 규정자가 TEST이고 임의의 두 번째 규정자와 세 번째 규정자가 QUEUE인 큐 관리자 QM1의 모든 큐와 일치합니다.
- **@QM1 는 큐 관리자 QM1의 모든 큐와 일치합니다.

## 기호 링크

<include> 및 <exclude> 요소에 하드 링크를 지정하여 UserSandboxes.xml 파일의 파일 경로에 사용하는 모든 기호 링크를 완전히 해석해야 합니다. 예를 들어, /var 가 /SYSTEM/var에 맵핑되는 기호 링크가 있는 경우 이 경로를 <tns:include name="/SYSTEM/var"/>로 지정해야 합니다. 그렇지 않으면 의도한 전송이 사용자 샌드박스 보안 오류로 실패합니다.

### 예

이 예는 AGENT_JUPITER의 구성 디렉토리에 있는 UserSandboxes.xml 파일에 다음 <sandbox> 요소를 추가하여 MQMD 사용자 이름이 guest 인 사용자가 AGENT_JUPITER 에이전트가 실행 중인 시스템의 /home/user/public 디렉토리 또는 해당 서브디렉토리에서 파일을 전송할 수 있도록 허용하는 방법을 보여줍니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
        xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="guest">
            <tns:read>
                <tns:include name="/home/user/public/**"/>
             </tns:read>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

### 예

이 예는 MQMD 사용자 이름 account 뒤에 한 자리 숫자가 있는 사용자(예: account4)가 다음 조치를 어떻게 완료하는지 보여줍니다.

- AGENT_SATURN 에이전트가 실행 중인 시스템에서 /home/account/private 디렉토리는 제외하고 /home/account 디렉토리 또는 이의 서브디렉토리에서 파일을 전송합니다.
- AGENT_SATURN 에이전트가 실행 중인 시스템에서 /home/account/output 디렉토리 또는 이의 서브디렉토리로 파일을 전송합니다.
- 로컬 큐 관리자의 큐에서 접두부 ACCOUNT.로 시작하는 메시지를 읽습니다(메시지가 ACCOUNT.PRIVATE.으로 시작하는 경우(즉, 두 번째 레벨에 PRIVATE이 있음) 제외).
- 모든 큐 관리자에서 접두부 ACCOUNT.OUTPUT.으로 시작하는 큐에 데이터를 전송합니다.

MQMD 사용자 이름이 account 인 사용자가 이러한 조치를 완료할 수 있도록 하려면 다음 <sandbox> 요소를 AGENT_SATURN의 구성 디렉토리에 있는 UserSandboxes.xml파일에 추가하십시오.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
            xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="account[0-9]" userPattern="regex">
            <tns:read>
                <tns:include name="/home/account/**"/>
                <tns:include name="ACCOUNT.**" type="queue"/>
                <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
                <tns:exclude name="/home/account/private/**"/>
                                                    </tns:read>
            <tns:write>
                <tns:include name="/home/account/output/**"/>
```

```
                <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
            </tns:write>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

**관련 참조**

548 페이지의 『와일드카드 전송 시의 추가 점검사항』
에이전트가 파일을 주고 받을 수 있는 위치를 제한하기 위해 사용자 또는 에이전트 샌드박스를 사용하여 에이전트를 구성하는 경우 해당 에이전트에 대한 와일드카드 전송 시 추가적인 검사가 이루어지도록 지정할 수 있습니다.

MFTagent.properties 파일

## 와일드카드 전송 시의 추가 점검사항

에이전트가 파일을 주고 받을 수 있는 위치를 제한하기 위해 사용자 또는 에이전트 샌드박스를 사용하여 에이전트를 구성하는 경우 해당 에이전트에 대한 와일드카드 전송 시 추가적인 검사가 이루어지도록 지정할 수 있습니다.

### additionalWildcardSandboxChecking 특성

와일드카드 전송에 대한 추가 검사를 사용으로 설정하려면 검사하려는 에이전트에 대한 `agent.properties` 파일에 다음 특성을 추가하십시오.

```
additionalWildcardSandboxChecking=true
```

이 특성이 true로 설정되어 있는 경우 에이전트가 와일드카드의 일치하는 파일을 찾기 위해 정의된 샌드박스의 외부 위치를 읽으려고 시도하는 전송 요청을 작성하면 전송이 실패합니다. 한 전송 요청에 여러 개의 전송이 포함되어 있고 이들 요청 중 하나가 샌드박스 외부 위치를 읽으려다 실패하는 경우 전체 전송이 실패합니다. 검사에 실패하면 실패 이유가 오류 메시지로 제공됩니다.

additionalWildcardSandboxChecking 특성이 에이전트의 `agent.properties` 파일에서 생략되거나 false로 설정된 경우 해당 에이전트의 와일드카드 전송에 대한 추가 검사가 수행되지 않습니다.

### 와일드카드 검사의 오류 메시지

구성된 샌드박스 위치의 외부로 와일드카드 전송 요청을 보내는 경우 보고되는 메시지는 다음과 같습니다.

다음 메시지는 전송 요청의 와일드카드 파일 경로가 제한된 샌드박스 외부에 있는 경우 발생합니다.

```
BFGSS0077E: 파일 경로 path을(를) 읽으려는 시도가 거부되었습니다.
파일 경로가 제한된 전송 샌드박스 외부에 있습니다.
```

여러 전송 요청 내의 한 전송에 와일드카드 전송 요청이 포함되어 있으며 해당 요청의 경로가 제한된 샌드박스 외부에 있는 경우 다음 메시지가 발생합니다.

```
BFGSS0078E: 파일 경로 읽기 시도: path가 다른 전송으로 무시됨
제한된 전송 샌드박스 외부에서 읽기를 시도했기 때문입니다.
```

다음 메시지는 파일이 제한된 샌드박스 외부에 있는 경우 발생합니다.

```
BFGSS0079E: 파일 file path을(를) 읽으려는 시도가 거부되었습니다.
파일이 제한된 전송 샌드박스 외부에 있습니다.
```

다음 메시지는 다른 와일드카드 전송 요청으로 인해 이 요청이 무시되는 여러 전송 요청에서 발생합니다.

```
BFGSS0080E: 파일 file path을(를) 읽으려는 시도가 거부되었습니다.
제한된 전송 샌드박스 외부에서 읽기를 시도했기 때문입니다.
```

와일드카드를 포함하지 않는 단일 파일 전송에서, 샌드박스 외부에 있는 파일이 전송에 관련되는 경우 보고되는 메시지는 이전 릴리스에서 변경되지 않았습니다.

```
BFGIO0056E: "FILE" 파일을 읽으려는 시도가 거부되었습니다.
파일이 제한된 전송 샌드박스 외부에 있습니다.
```

**관련 참조**

545 페이지의 『MFT 사용자 샌드박스에 대한 작업』

전송을 요청한 사용자의 MQMD 사용자 이름에 기반하여 파일을 송수신할 수 있는 파일 시스템의 영역을 제한할 수 있습니다.

544 페이지의 『MFT 에이전트 샌드박스에 대한 작업』
Managed File Transfer에 추가 보안 레벨을 추가하기 위해 에이전트가 액세스할 수 있는 파일 시스템 영역을 제한할 수 있습니다.

MFTagent.properties 파일

# MFT의 SSL 또는 TLS 암호화 구성

IBM MQ Managed File Transfer 와 함께 SSL 또는 TLS를 사용하여 에이전트와 해당 에이전트 큐 관리자, 명령 및 연결 중인 큐 관리자와 토폴로지 내의 다양한 큐 관리자 대 큐 관리자 연결 간의 통신을 보호할 수 있습니다.

## 시작하기 전에

SSL 또는 TLS 암호화를 사용하여 IBM MQ Managed File Transfer 토폴로지를 통해 플로우되는 메시지를 암호화할 수 있습니다. 다음이 포함됩니다.

- 에이전트와 해당 에이전트 큐 관리자 간에 전달되는 메시지입니다.
- 명령 및 명령이 연결되는 큐 관리자에 대한 메시지입니다.
- 토폴로지 내에서 에이전트 큐 관리자, 명령 큐 관리자 및 조정 큐 관리자 간에 이동하는 내부 메시지입니다.

## 이 태스크 정보

IBM MQ에서 SSL 사용에 대한 일반 정보는 260 페이지의 『SSL/TLS에 대한 작업』의 내용을 참조하십시오. IBM MQ 용어에서 Managed File Transfer는 표준 Java 클라이언트 애플리케이션입니다.

다음 단계를 수행하여 Managed File Transfer에 SSL을 사용하십시오.

## 프로시저

1. 신뢰 저장소 파일을 작성하고 선택적으로 키 저장소 파일을 작성하십시오(이들 파일이 동일할 수 있음). 클라이언트 인증(즉, 채널의 SSLCAUTH=OPTIONAL)이 필요하지 않은 경우 키 저장소를 제공하지 않아도 됩니다. 큐 관리자의 인증서를 인증하기 위해서만 신뢰 저장소가 필요합니다.

   IBM MQ에 대해 작업하려면 신뢰 저장소 및 키 저장소에 대한 인증서를 작성하는 데 사용되는 키 알고리즘이 RSA여야 합니다.
2. SSL을 사용하도록 IBM MQ 큐 관리자를 설정하십시오.

   SSL을 사용하도록 큐 관리자를 설정하는 방법(예: IBM MQ Explorer 사용)에 대한 정보는 큐 관리자의 SSL 구성을 참조하십시오.
3. 신뢰 저장소 파일 및 키 저장소 파일(있을 경우)을 적당한 위치에 저장하십시오. 제안된 위치는 *config_directory*/*coordination_qmgr*/agents/*agent_name* 디렉토리입니다.
4. 해당 Managed File Transfer 특성 파일에서 필요에 따라 각 SSL 사용 큐 관리자에 대한 SSL 특성을 설정하십시오. 하나의 큐 관리자가 두 개 이상의 역할을 수행할 수도 있지만 각 특성 세트는 개별 큐 관리자(에이전트, 조정 및 명령)를 참조합니다.

   **CipherSpec** 또는 **CipherSuite** 특성 중 하나는 필수입니다. 이 특성이 없으면 클라이언트가 SSL 없이 연결을 시도합니다. IBM MQ 및 Java간의 용어 차이로 인해 **CipherSpec** 또는 **CipherSuite** 특성이 둘 다 제공됩니다. Managed File Transfer에서는 특성 및 필수 변환을 승인하므로, 두 특성 모두 설정하지 않아도 됩니다. **CipherSpec** 또는 **CipherSuite** 특성을 모두 지정하는 경우, **CipherSpec**이 사용됩니다.

   **PeerName** 특성은 선택적입니다. 연결할 큐 관리자의 식별 이름에 특성을 설정할 수 있습니다. Managed File Transfer는 일치하지 않는 식별 이름이 있는 올바르지 않은 SSL 서버에 대한 연결을 거부합니다.

   **SslTrustStore** 및 **SslKeyStore** 특성을 신뢰 저장소 및 키 저장소 파일을 가리키는 파일 이름으로 설정하십시오. 이미 실행 중인 에이전트의 특성을 설정하는 경우, 에이전트를 중지시키고 재시작하여 SSL 모드에서 다시 연결하십시오.

   특성 파일에 일반 텍스트 비밀번호가 포함되므로 해당 파일 시스템 권한 설정을 고려하십시오.

SSL 특성에 대한 자세한 정보는 550 페이지의 『MFT에 대한 SSL/TLS 특성』의 내용을 참조하십시오.

5. 에이전트 큐 관리자가 SSL을 사용하는 경우 사용자는 에이전트 작성 시 필수 세부사항을 제공할 수 없습니다. 다음 단계를 수행하여 에이전트를 작성하십시오.
   a) **fteCreateAgent** 명령을 사용하여 에이전트를 작성하십시오. 에이전트의 존재를 조정 큐 관리자에게 발행할 수 없음에 대한 경고가 수신됩니다.
   b) 이전 단계에서 작성된 `agent.properties` 파일을 편집하여 SSL 정보를 추가하십시오. 에이전트가 시작되면 발행이 다시 시도됩니다.
6. `agent.properties` 파일 또는 `coordination.properties` 파일의 SSL 특성이 변경되는 동안 IBM MQ 탐색기의 에이전트 또는 인스턴스가 실행 중인 경우 에이전트 또는 IBM MQ Explorer를 재시작해야 합니다.

**관련 참조**

MFTagent.properties 파일

## MFT에 대한 SSL/TLS 특성

일부 MFT 특성 파일에는 SSL 및 TLS 특성이 포함되어 있습니다. IBM MQ 및 Managed File Transfer에서 SSL 또는 TLS를 사용하여 에이전트와 큐 관리자 사이의 권한이 없는 연결을 방지하고 에이전트와 큐 관리자 간 메시지 트래픽을 암호화할 수 있습니다.

다음 MFT 특성 파일이 SSL 특성을 포함합니다.

- MFT `agent.properties` 파일의 SSL/TLS 특성
- MFT `coordination.properties` 파일의 SSL/TLS 특성
- MFT `command.properties` 파일의 SSL/TLS 특성
- MFT `logger.properties` 파일의 SSL/TLS 특성

Managed File Transfer에서 SSL 또는 TLS 사용에 대한 정보는 549 페이지의 『MFT의 SSL 또는 TLS 암호화 구성』의 내용을 참조하십시오.

IBM WebSphere MQ 7.5에서는 파일 또는 디렉토리 위치를 나타내는 일부 Managed File Transfer 특성에서 환경 변수를 사용할 수 있습니다. 이를 통해 제품 부분 실행 시 사용되는 파일 또는 디렉토리의 위치가 환경 변경사항(예: 프로세스를 실행 중인 사용자)에 따라 달라질 수 있습니다. 자세한 정보는 MFT 특성에서 환경 변수의 사용을 참조하십시오.

**관련 개념**

멀티플랫폼에서의 MFT 구성 옵션

**관련 참조**

MFT 특성에서 환경 변수의 사용

# 채널 인증으로 클라이언트 모드의 큐 관리자에 연결

IBM MQ 는 채널 인증 레코드를 사용하여 채널 레벨에서 보다 정확하게 액세스를 제어합니다. 이는 기본적으로 새로 작성된 큐 관리자가 Managed File Transfer 컴포넌트의 클라이언트 연결을 거부함을 의미합니다.

채널 인증에 대한 자세한 정보는 47 페이지의 『채널 인증 레코드』의 내용을 참조하십시오.

Managed File Transfer에서 사용한 SVRCONN에 대한 채널 인증 구성이 권한 없는 MCAUSER ID를 지정하는 경우, Managed File Transfer Agent 및 명령이 올바르게 작동할 수 있도록 큐 관리자, 큐 및 토픽에 대한 특정 권한 레코드를 부여해야 합니다. MQSC 명령 SET CHLAUTH 또는 PCF 명령 채널 인증 레코드 설정을 사용하여 채널 인증 레코드를 작성, 수정 또는 제거하십시오. IBM MQ 큐 관리자에 연결하려는 모든 Managed File Transfer 에이전트의 경우 모든 에이전트에 사용할 MCAUSER ID를 설정하거나 각 에이전트에 대해 별도의 MCAUSER ID를 설정할 수 있습니다.

각 MCAUSER ID에 다음 권한을 부여하십시오.

- 큐 관리자에 필요한 권한 레코드
  - connect
  - setid

  - inq
- 큐에 필요한 권한 레코드.

  다음 목록에서 *agent_name* 으로 끝나는 모든 에이전트 특정 큐의 경우, 클라이언트 연결을 사용하여 IBM MQ 큐 관리자에 연결하려는 각 에이전트에 대해 이러한 큐 권한 레코드를 작성해야 합니다.

  - put, get, dsp(SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, browse(SYSTEM.FTE.COMMAND.*agent_name*)
  - put, get(SYSTEM.FTE.DATA.*agent_name*)
  - put, get(SYSTEM.FTE.REPLY.*agent_name*)
  - put, get, inq, browse(SYSTEM.FTE.STATE.*agent_name*)
  - put, get, browse(SYSTEM.FTE.EVENT.*agent_name*)
  - put, get(SYSTEM.FTE)
- 토픽에 필요한 권한 레코드.

  - sub, pub(SYSTEM.FTE)
- 파일 전송에 필요한 권한 레코드.

  소스 및 대상 에이전트에 대해 개별 MCAUSER ID가 있는 경우 소스 및 대상 둘 다의 에이전트 큐에서 권한 레코드를 작성하십시오.

  예를 들어, 소스 에이전트의 MCAUSER ID가 **user1**이고 대상 에이전트 MCAUSER ID가 **user2**인 경우 에이전트 사용자에 대해 다음 권한을 설정하십시오.

| 에이전트 사용자 | 큐 | 필요한 권한 |
|---|---|---|
| user1 | SYSTEM.FTE.DATA.*destination_agent_name* | put |
| user1 | SYSTEM.FTE.COMMAND.*destination_agent_name* | put |
| user2 | SYSTEM.FTE.REPLY.*source_agent_name* | put |
| user2 | SYSTEM.FTE.COMMAND.*source_agent_name* | put |

# Connect:Direct 브릿지 에이전트와 Connect:Direct 노드 사이에서 SSL 또는 TLS 구성

Connect:Direct 브릿지 에이전트 특성 파일에서 특성을 설정하고 키 저장소와 신뢰 저장소를 작성하여 SSL 프로토콜을 통해 서로 연결하도록 Connect:Direct 브릿지 에이전트와 Connect:Direct 노드를 구성하십시오.

## 이 태스크 정보

이러한 단계에는 인증 기관에서 서명한 키를 가져오기 위한 지시사항이 포함되어 있습니다. 인증 기관을 사용하지 않는 경우 자체 서명된 인증서를 생성할 수 있습니다. 자체 서명된 인증서 생성에 대한 자세한 정보는 276 페이지의 『AIX, Linux, and Windows에서 SSL/TLS에 대한 작업』의 내용을 참조하십시오.

이러한 단계에는 Connect:Direct 브릿지 에이전트의 새 키 저장소 및 신뢰 저장소 작성에 대한 지시사항이 포함되어 있습니다. Connect:Direct 브릿지 에이전트에 IBM MQ 큐 관리자에 안전하게 연결하는 데 사용하는 키 저장소와 신뢰 저장소가 이미 있는 경우 Connect:Direct 노드에 안전하게 연결할 때 기존 키 저장소와 신뢰 저장소를 사용할 수 있습니다. 자세한 정보는 549 페이지의 『MFT의 SSL 또는 TLS 암호화 구성』의 내용을 참조하십시오.

## 프로시저

Connect:Direct 노드의 경우 다음 단계를 완료하십시오.

1. Connect:Direct 노드에 대해 서명된 인증서와 키를 생성하십시오.

   IBM MQ와 함께 제공되는 IBM 키 관리 도구를 사용하여 이를 수행할 수 있습니다. 자세한 정보는 260 페이지의 『SSL/TLS에 대한 작업』의 내용을 참조하십시오.

2. 서명된 키를 가지려면 인증 기관에 요청을 송신하십시오. 그러면 인증서를 받습니다.
3. 인증 기관의 공개 키를 포함하는 텍스트 파일을 작성하십시오(예: `/test/ssl/certs/CAcert`).
4. Connect:Direct 노드에서 Secure+ Option을 설치하십시오.

   노드가 이미 있는 경우 설치 프로그램을 다시 실행하고 기존 설치의 위치를 지정하고 Secure+ Option만 설치하도록 선택하여 Secure+ Option을 설치할 수 있습니다.
5. 새 텍스트 파일을 작성하십시오(예: `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`).
6. 인증 기관에서 받은 인증서 및 `/test/ssl/cd/privateKeys/`*`node_name`*`.key`에 있는 개인 키를 텍스트 파일로 복사하십시오.

   `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt` 의 컨텐츠는 다음 형식이어야 합니다.

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHQjES
MBAGA1UECBMJSGFtcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xDjAMBgNVBAsTBU1RSVBUMQswCQYDVQQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEwlIYW1wc2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvgP1QIklU9ypSKD1XoODo1yk
EyMFXBOUpZRrDVxjoSEC0vtWNcJ199e+Vc4UpNybDyBu+NkDlMNofX4QxeQcLAFj
WnhakqCiQ+JIAD5Aurhnrwche0MV3kjA84GKH/rOSVqtl984mu/lDyS819XcfSSn
cOOMsK1KbneVSCIV2XECAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0E
HxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFNXMIpSc
csBXUniW4A3UrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj4lVz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7klXa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/vl8+iv01OCL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIiEk9Dlz2U6WO9lICwn
l7PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qHO9ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRUOIQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDWoMNt5fj51v7aPmVeS60bOm+UlGre8B/Ze18JVj2O4K2Uh72rDCXE
5e6eFxSdUM207sQDy20euBVELJtM2kOkL1ROdoQQSlU3XQNgJw/t3ZIx5hPXWEQT
rjRQO64BEhb+PzzxPF8uwzZ9IrUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI8OGvBvWXO4fHyvIX5aslwhBoArXIS1AtNTrptPvoaP1zyIAeZ6OCVo/
SFo+A2UhmtEJeOJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWDyJBVhUOjXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+lYcX8OVZ6
sHFPNlHluCNy/riUcBy9iviVeodX8IomOchSyO5DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uUOAeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM5OoWXa6U5+AYuGUMg
/itPZmUmNrHjTk7ghT6i1IQOaBowXXKJBlMmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lzQ==
-----END RSA PRIVATE KEY-----
```

7. Secure+ Admin 도구를 시작하십시오.
   - AIX and Linux 시스템에서 **spadmin.sh** 명령을 실행하십시오.
   - Windows 시스템에서 **시작** > **프로그램** > **Sterling Commerce Connect:Direct** > **CD Secure+ Admin 도구**를 클릭하십시오.

   CD Secure+ Admin 도구를 시작합니다.
8. CD Secure+ Admin 도구에서 **.Local** 행을 두 번 클릭하여 기본 SSL 또는 TLS 설정을 편집하십시오.
   a) 사용 중인 프로토콜에 따라 **Enable SSL Protocol** 또는 **Enable TLS Protocol**을 선택하십시오.
   b) **대체 사용 불가능**을 선택하십시오.
   c) 최소 하나의 암호 스위트를 선택하십시오.
   d) 양방향 인증을 원하는 경우 **Enable Client Authentication**의 값을 `Yes`로 변경하십시오.
   e) **Trusted Root Certificate** 필드에 인증 기관의 공용 인증서 파일에 대한 경로(`/test/ssl/certs/CAcert`)를 입력하십시오.
   f) **Key Certificate File** 필드에 사용자가 작성한 파일의 경로(`/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`)를 입력하십시오.
9. **.Client** 행을 두 번 클릭하여 기본 SSL 또는 TLS 설정을 편집하십시오.

a) 사용 중인 프로토콜에 따라 **Enable SSL Protocol** 또는 **Enable TLS Protocol**을 선택하십시오.

b) **대체 사용 불가능**을 선택하십시오.

Connect:Direct 브릿지 에이전트의 경우 다음 단계를 수행하십시오.

10. 신뢰 저장소를 작성하십시오. 더미 키를 작성한 후 더미 키를 삭제하여 이를 수행할 수 있습니다.
    다음 명령을 사용할 수 있습니다.

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 인증 기관의 공용 인증서를 신뢰 저장소로 가져오십시오.
    다음 명령을 사용할 수 있습니다.

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. Connect:Direct 브릿지 에이전트 특성 파일을 편집하십시오.
    파일에 다음 행을 포함시키십시오.

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

    이 단계의 예에서 *protocol*은 사용 중인 프로토콜(SSL 또는 TLS)이며 *password*는 신뢰 저장소를 작성할 때 지정한 비밀번호입니다.

13. 양방향 인증을 원하는 경우 Connect:Direct 브릿지 에이전트의 키와 인증서를 작성하십시오.

    a) 키 저장소와 키를 작성하십시오.
    다음 명령을 사용할 수 있습니다.

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

    b) 서명 요청을 생성하십시오.
    다음 명령을 사용할 수 있습니다.

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

    c) 선행 단계에서 수신하는 인증서를 키 저장소로 가져오십시오. 인증서는 x.509 형식이어야 합니다.
    다음 명령을 사용할 수 있습니다.

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

    d) Connect:Direct 브릿지 에이전트 특성 파일을 편집하십시오.
    파일에 다음 행을 포함시키십시오.

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

    이 단계의 예에서 *password*는 키 저장소를 작성할 때 지정한 비밀번호입니다.

**관련 태스크**
Connect:Direct 브릿지 구성

# ALW AMQP 클라이언트 보안 설정

일정 범위의 보안 메커니즘을 사용하여 AMQP 클라이언트에서의 연결에 대해 보안을 설정하고 데이터가 네트워크에서 안전하게 보호되도록 할 수 있습니다. MQ Light 애플리케이션에 보안을 구축할 수 있습니다. 또한 기능이 다른 애플리케이션에 사용되는 것과 동일한 방식으로 AMQP 클라이언트에서 IBM MQ의 기존 보안 기능을 사용할 수도 있습니다.

## 채널 인증 규칙(CHLAUTH)

채널 인증 규칙을 사용하여 큐 관리자에 대한 TCP 연결을 제한할 수 있습니다. AMQP 채널은 사용자 큐 관리자에 대해 구성되는 채널 인증 규칙 사용을 지원합니다. 사용자 큐 관리자에서 AMQP 채널과 일치하는 프로파일로 채널 인증 규칙이 정의되면 이러한 규칙은 해당 채널에 적용됩니다. 기본적으로 채널 인증은 새 IBM MQ 큐 관리자에서 사용으로 설정되므로 AMQP 채널을 사용하기 전에 일부 구성을 완료해야 합니다.

AMQP가 사용자 큐 관리자에 연결할 수 있도록 채널 인증 규칙 구성 방법에 대한 자세한 정보는 AMQP 채널 작성 및 사용을 참조하십시오.

## 연결 인증(CONNAUTH)

연결 인증을 사용하여 큐 관리자에 대한 연결을 인증할 수 있습니다. AMQP 채널은 AMQP 애플리케이션에서 큐 관리자에 대한 액세스를 제어하기 위한 연결 인증 사용을 지원합니다.

AMQP 프로토콜은 SASL(Simple Authentication and Security Layer) 프레임워크를 사용하여 연결 인증 방법을 지정합니다. 여기에는 다양한 SASL 메커니즘이 있고 IBM MQ는 두 가지 SASL 메커니즘(ANONYMOUS 및 PLAIN)을 지원합니다.

ANONYMOUS의 경우 인증을 위해 클라이언트에서 큐 관리자로 신임 정보가 전달되지 않습니다. 큐 관리자 **CONNAUTH** 속성에 지정된 IBM MQ AUTHINFO 오브젝트의 **CHCKCLNT** 값이 REQUIRED 또는 REQDADM (관리 사용자로 연결하는 경우) 인 경우 연결이 거부됩니다. **CHCKCLNT** 의 값이 NONE 또는 OPTIONAL인 경우 연결이 승인됩니다.

PLAIN의 경우 인증을 위해 클라이언트에서 큐 관리자에게 사용자 이름 및 비밀번호가 전달됩니다. 큐 관리자 **CONNAUTH** 속성에 지정된 IBM MQ AUTHINFO 오브젝트의 **CHCKCLNT** 값이 NONE인 경우 연결이 거부됩니다. **CHCKCLNT** 의 값이 OPTIONAL, REQUIRED또는 REQDADM (관리 사용자로 연결하는 경우) 인 경우, 큐 관리자가 사용자 이름 및 비밀번호를 검사합니다. 큐 관리자는 운영 체제 (AUTHINFO 오브젝트가 IDPWOS유형인 경우) 또는 LDAP 저장소 (AUTHINFO 오브젝트가 IDPWLDAP유형인 경우) 를 검사합니다.

다음 테이블은 이 인증 작동을 요약한 것입니다.

표 *101. SASL* 메커니즘 및 연결 인증 요약

| **SASL 메커니즘** | **클라이언트에서 큐 관리자로 신임 정보 전달 여부** | **CHKCLNT 값** |
|---|---|---|
| ANONYMOUS | 아니오 | REQUIRED 또는 REQDADM - 연결 거부됨<br><br>NONE 또는 OPTIONAL - 연결 허용됨 |
| PLAIN | 예, 사용자 이름 및 비밀번호 | REQUIRED, REQDADM 또는 OPTIONAL - 큐 관리자가 사용자 이름 및 비밀번호를 검사함<br><br>NONE - 연결이 거부됨 |

MQ Light 클라이언트를 사용 중인 경우 연결하는 AMQP 주소에 이를 포함시켜 신임 정보를 지정할 수 있습니다. 예:

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

### 채널에서의 MCAUSER 설정

AMQP 채널에는 MCAUSER 속성이 있고, 이는 해당 채널에 대한 모든 연결에 권한을 부여하는 IBM MQ 사용자 ID를 설정할 때 사용할 수 있습니다. AMQP 클라이언트에서 해당 채널로의 모든 연결에서는 사용자가 구성한 MCAUSER ID를 채택합니다. 해당 사용자 ID는 다른 토픽에 대한 메시징의 권한 부여에 사용됩니다.

큐 관리자에 대한 연결의 보안을 설정하려면 채널 인증(CHLAUTH)을 사용하도록 권장합니다. 채널 인증을 사용 중인 경우 MCAUSER 값을 권한이 없는 사용자로 구성하도록 권장합니다. 그러면 채널 연결이 CHLAUTH 규칙에 일치하지 않는 경우 연결에는 큐 관리자에서 메시징을 수행할 권한이 부여되지 않습니다.

### SSL/TLS 지원

AMQP 채널은 사용자 큐 관리자에서 구성된 키 저장소에서 키를 사용한 SSL/TLS 암호화를 지원합니다. SSL/TLS 암호화를 위한 AMQP 채널 구성 옵션에서는 다른 MQ 채널 유형과 동일한 옵션을 지원합니다. 암호 스펙 및 큐 관리자에게 AMQP 클라이언트 연결에서의 인증서가 필요한지 여부를 지정할 수 있습니다.

큐 관리자의 FIPS 속성을 사용하면 SSL/TLS 암호 스위트를 제어할 수 있고, 이는 AMQP 클라이언트로부터의 연결에 대한 보안을 설정할 때 사용할 수 있습니다.

큐 관리자의 키 저장소를 설정하는 방법에 대한 정보는 276 페이지의 『AIX, Linux, and Windows에서 SSL/TLS에 대한 작업』의 내용을 참조하십시오.

AMQP 클라이언트 연결에 대한 SSL/TLS 지원 구성 방법에 대한 정보는 AMQP 채널 작성 및 사용을 참조하십시오.

V 9.4.0 V 9.4.0 IBM MQ 9.4.0부터 AMQP 채널은 더 이상 큐 관리자에서 CMS 키 저장소를 지원하지 않습니다. **runmqakm** 명령을 사용하여 CMS 키 저장소를 지원되는 PKCS #12 형식으로 변환할 수 있습니다. 예를 들어, 다음 명령을 사용하여 `sslTest.kdb` 라는 키 저장소를 CMS 형식에서 PKCS #12 형식으로 변환할 수 있습니다. 새 키 저장소의 이름은 `sslTest.p12`이며 비밀번호 `passw0rd`로 보호됩니다.

```
runmqakm -keydb -convert -type cms -db sslTest.kdb -stashed -new_format pkcs12 -target
sslTest.p12 -new_pw passw0rd
```

### Java 인증 및 권한 서비스 (JAAS)

선택적으로 JAAS 로그인 모듈이 있는 AMQP 채널을 구성할 수 있습니다. 이는 AMQP 클라이언트가 제공하는 사용자 이름 및 비밀번호를 확인할 수 있습니다. 556 페이지의 『AMQP 채널용 JAAS 구성』의 내용을 참조하십시오.

**관련 태스크**

AMQP 클라이언트 애플리케이션 개발

AMQP 채널 작성 및 사용

## ALW AMQP 클라이언트 인계 제한

AMQP 클라이언트 연결이 기존 AMQP 클라이언트 연결과 동일한 클라이언트 ID를 갖도록 하는 경우, 기본적으로 기존 클라이언트의 연결이 끊어집니다. 그러나 특정 조건을 충족할 때에만 인계가 가능하도록 클라이언트 인계 작동을 제한하는 큐 관리자를 구성할 수 있습니다.

예를 들어 기존 클라이언트의 연결을 끊는 것은 다른 팀에서 개발 중인 AMQP 애플리케이션이 있고 여기에서 동일한 클라이언트 ID를 사용하는 경우 적절하지 않습니다. 이 문제를 해결하기 위해 사용 중인 AMQP 채널의 이름, 클라이언트의 IP 주소 및 클라이언트 사용자 ID(SASL 인증이 사용되는 경우)를 기반으로 클라이언트 인계를 제한할 수 있습니다.

다음 테이블에서 자세히 설명한 대로 큐 관리자 속성 **AdoptNewMCA** 및 **AdoptNewMCACheck**의 설정을 사용하여 클라이언트 인계 제한의 필수 레벨을 지정하십시오.

표 *102.* 클라이언트 인계를 제한하는 ***AdoptNewMCA*** 및 ***AdoptNewMCACheck*** 설정

| **AdoptNewMCA** | **AdoptNewMCACheck** | **클라이언트 인계가 허용되기 전 검사하는 기준** |
|---|---|---|
| NO 또는 정의되지 않음 | 적용할 수 없음 | 없음 인증되고 모든 CHLAUTH 규칙을 통과한 모든 클라이언트 연결에 대해 클라이언트 인계가 허용됩니다. |
| ALL(또는 NO 이외의 값) | QM 또는 정의되지 않음 | 없음 인증되고 모든 CHLAUTH 규칙을 통과한 모든 클라이언트 연결에 대해 클라이언트 인계가 허용됩니다. |
| ALL(또는 NO 이외의 값) | 이름 | 사용자 ID(SASL이 사용되는 경우)<br>채널 이름 |
| ALL(또는 NO 이외의 값) | ADDRESS | 사용자 ID(SASL이 사용되는 경우)<br>IP 주소 |
| ALL(또는 NO 이외의 값) | 모두 | 사용자 ID(SASL이 사용되는 경우)<br>채널 이름<br>IP 주소 |

큐 관리자 속성 **AdoptNewMCA** 및 **AdoptNewMCACheck**는 CHANNELS 스탠자에 정의되는 큐 관리자 구성의 일부입니다. Windows용 IBM MQ 및 Linux x86-64 시스템용 IBM MQ 에서는 IBM MQ Explorer를 사용하여 구성 정보를 수정하십시오. 기타 시스템에서는 `qm.ini` 구성 파일을 편집하여 정보를 수정하십시오. 큐 관리자 채널 정보를 수정하는 방법에 대한 정보는 채널의 속성을 참조하십시오.

**관련 태스크**

AMQP 클라이언트 애플리케이션 개발

AMQP 채널 작성 및 사용

## ALW AMQP 채널용 JAAS 구성

JAAS( Java Authentication and Authorization Service) 사용자 정의 모듈은 연결 시 AMQP 클라이언트가 AMQP 채널에 전달한 사용자 이름 및 비밀번호 신임 정보를 인증하는 데 사용할 수 있습니다.

### 이 태스크 정보

다른 Java기반 시스템에서 인증을 위해 이미 JAAS 모듈을 사용하고 MQ에 대한 AMQP 연결을 인증하기 위해 해당 모듈을 재사용하려는 경우 사용자 정의 JAAS 모듈을 사용할 수 있습니다. 또는 MQ에 빌드된 인증 기능이 사용하려는 인증 메커니즘을 지원하지 않는 경우 사용자 정의 JAAS 모듈을 작성하려 할 수 있습니다.

AMQP 채널에 대한 JAAS 모듈 구성은 큐 관리자 레벨에서 수행됩니다. 이는 큐 관리자에 대한 AMQP 연결 인증에 JAAS 모듈이 구성되는 경우, 모듈은 모든 AMQP 채널에 적용됩니다. JAAS 모듈을 호출한 채널의 이름은 모듈로 전달되어 사용자가 다른 채널을 위해 다른 JAAS 로그를 코딩할 수 있습니다.

다음과 같은 다른 정보도 JAAS 모듈로 전달됩니다.

- 인증을 시도하는 AMQP 클라이언트의 클라이언트 ID.
- AMQP 클라이언트의 네트워크 주소.
- JAAS 모듈을 호출한 채널의 이름.

### 프로시저

다음 단계를 완료하여 AMQP 채널의 JAAS 구성 모듈을 구성합니다.

1. 하나 이상의 JAAS 모듈 구성 스탠자를 포함하는 `jaas.config` 파일을 정의하십시오. 스탠자는 JAAS `javax.security.auth.spi.LoginModule` 인터페이스를 구현하는 Java 클래스의 완전한 이름을 지정해야 합니다.
   - 기본 `jaas.config` 파일은 제품과 함께 제공되며 *QM_data_directory*`/amqp/jaas.config`에 있습니다.
   - 사전 구성된 `MQXRConfig` 스탠자는 이미 기본 `jaas.config` 파일에 정의되어 있습니다.
2. AMQP 채널에 사용할 스탠자의 이름을 지정하십시오.
   - Linux AIX `amqp_unix.properties` 파일에 특성을 추가하십시오.
   - Windows `amqp_win.properties` 파일에 특성을 추가하십시오.

   특성의 양식은 다음과 같습니다.

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   for example:

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```
3. 사용자 정의 모듈의 클래스를 포함하도록 큐 관리자 환경을 구성하십시오. AMQP 서비스에는 JAAS 구성 스탠자에 구성된 Java 클래스에 대한 액세스 권한이 있어야 합니다.

   JAAS 클래스에 대한 경로를 MQ `service.env` 파일에 추가하여 이를 수행합니다. MQ 구성 디렉토리(*MQ_config_directory*) 또는 큐 관리자 구성 디렉토리(*QM_config_directory*)에 있는 `service.env` 파일을 편집하여 CLASSPATH 변수를 JAAS 모듈 클래스의 위치로 설정하십시오.

### 다음에 수행할 작업

샘플 JAAS 로그인 모듈은 제품과 함께 *mq_installation_directory*`/amqp/samples` 디렉토리에 제공됩니다. 샘플 JAAS 로그인 모듈은 클라이언트가 연결하는 사용자 이름 또는 비밀번호에 관계없이 모든 클라이언트 연결을 인증합니다.

사용자는 샘플의 소스 코드를 수정하고 이를 다시 컴파일하여 특정 비밀번호를 갖는 특정 사용자만 인증하도록 시도할 수 있습니다. 제품에서 제공하는 샘플 JAAS 로그인 모듈을 사용하도록 UNIX 시스템에서 AMQP 채널을 구성하려면 다음을 수행하십시오.

1. `/var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties` 파일을 편집하고 특성 `com.ibm.mq.MQXR.JAASConfig=MQXRConfig`를 설정하십시오.
2. /var/mqm/service.env 파일을 편집하고 CLASSPATH=*mq_installation_location*`/amqp/samples` 특성을 설정하십시오.

`jaas.config` 파일에는 샘플 클래스 `samples.JAASLoginModule` 를 로그인 모듈 클래스로 지정하는 `MQXRConfig` 라는 스탠자가 이미 포함되어 있습니다. 샘플 모듈을 시도하기 전에 `jaas.config`를 변경할 필요는 없습니다.

**관련 태스크**

AMQP 클라이언트 애플리케이션 개발

AMQP 채널 작성 및 사용

# Advanced Message Security

Advanced Message Security(AMS)는 엔드 애플리케이션에 영향을 미치지 않으면서 IBM MQ 네트워크를 통해 플로우하는 민감한 데이터에 대한 높은 레벨의 보호를 제공하는 IBM MQ의 컴포넌트입니다.

# Advanced Message Security 개요

IBM MQ 애플리케이션은 가치가 높은 금융 거래 및 개인 정보 등과 같은 민감한 데이터를 공개 키 암호화 모델을 사용하여 서로 다른 보호 레벨을 사용하여 전송하기 위해 Advanced Message Security를 사용할 수 있습니다.

**관련 개념**

605 페이지의 『메시지 채널 에이전트(MCA) 인터셉션 및 AMS』

MCA 인터셉션 기능을 사용하면 IBM MQ에서 실행 중인 큐 관리자가 서버 연결 채널에 선택적으로 정책을 적용할 수 있습니다.

**관련 참조**

GSKit는 AMS 메시지에 사용된 코드를 리턴합니다.

## Advanced Message Security의 기능 및 특징

Advanced Message Security는 메시지 레벨에서 데이터 서명 및 암호화를 제공하기 위해 IBM MQ 보안 서비스를 확장합니다. 확장된 서비스는 메시지 데이터가 원래 큐에 놓인 때와 검색된 때 사이에 수정되지 않았음을 보장합니다. 또한, AMS는 메시지 데이터 송신자가 서명된 메시지를 대상 큐에 배치할 권한이 있는지 확인합니다.

AMS는 다음 기능을 제공합니다.

- IBM MQ가 처리하는 민감하거나 가치가 높은 트랜잭션을 보안합니다.
- 악하거나 권한 없는 메시지가 수신 애플리케이션에 의해 처리되기 전에 감지하고 제거합니다.
- 메시지가 큐에서 큐로 이동 중에 수정되지 않았는지 확인합니다.
- 데이터가 네트워크 전체에 걸쳐 흐를 때뿐만 아니라 큐에 놓일 때에도 보호합니다.
- IBM MQ에 대한 기존 독점 및 고객 작성 애플리케이션의 보안을 설정합니다.
- z/OS IBM MQ 9.1.3부터 IBM MQ for z/OS는 네트워크를 통해 전달되는 메시지에서 AMS 보호를 선택적으로 제거 및 추가하는 기능을 제공합니다. 이 기능을 서버 대 서버 메시지 채널 에이전트*(MCA)* 인터셉션이라고 합니다.
- ALW IBM MQ 9.1.4 및 IBM MQ 9.1.0 Fix Pack 4부터는 IBM MQ 라이브러리 코드에 고객의 애플리케이션 프로그램에서 실행되는 검사가 추가되었습니다. 이 검사는 초기화 초기에 실행되어 환경 변수 *AMQ_AMS_FIPS_OFF* 의 값을 읽으며, 임의의 값으로 설정된 경우 IBM Global Security Kit (GSKit) 코드가 해당 애플리케이션에서 비FIPS 모드로 실행됩니다.

## AMS에서 사용 가능한 QoP(Quality of Protection)

Advanced Message Security, `Integrity`, `Privacy` 및 `Confidentiality`에 대한 보호 품질은 세 가지가 있습니다.

`Integrity` 보호는 디지털 서명을 통해 제공되며, 이 서명을 통해 메시지를 작성한 사용자 및 메시지가 변경되거나 변경되지 않았는지 보증합니다.

`Privacy` 보호는 디지털 서명 및 암호화의 조합으로 제공됩니다. 암호화는 의도된 하나 이상의 수신인만 메시지 데이터를 볼 수 있도록 보장합니다. 권한이 없는 수신인이 암호화된 메시지 데이터의 사본을 확보해도 실제 메시지 데이터를 볼 수 없습니다.

`Confidentiality` 보호는 선택적 키 재사용으로만 암호화를 통해 제공됩니다.

### 성능에 미치는 효과

AMS는 대칭 및 비대칭 암호화 루틴의 조합을 사용하여 디지털 서명 및 암호화를 제공합니다. 대칭 키 조작은 비대칭 키 조작에 비해 매우 빠릅니다. 비대칭 키 조작은 CPU 집중적이기 때문에 AMS에서 많은 수의 메시지를 보호하는 비용에 큰 영향을 미칠 수 있습니다.

**비대칭 암호화 루틴**

예를 들어, 서명된 메시지를 넣으면 비대칭 키 조작을 사용하여 메시지 해시에 서명합니다.

서명된 메시지를 가져올 때 서명된 해시를 확인하기 위해 추가적인 비대칭 키 조작을 사용합니다.

따라서 메시지당 메시지 데이터에 서명하고 이를 확인하기 위해 적어도 두 개의 비대칭 키 조작이 필요합니다.

**비대칭 및 대칭 암호화 루틴**

암호화된 메시지를 넣을 때 대칭 키가 생성되고 의도된 각 메시지 수신인에 대해 비대칭 키 조작을 사용하여 해당 대칭 키를 암호화합니다.

그러면 대칭 키를 사용하여 메시지 데이터가 암호화됩니다. 암호화된 메시지를 가져올 때 의도된 수신인은 메시지에서 사용할 대칭 키를 검색하기 위해 비대칭 키 조작을 사용해야 합니다.

따라서 세 가지 QoP(Quality of Protection)는 모두 CPU 집중적인 비대칭 키 조작의 여러 요소를 포함하여, 이로 인해 메시지를 넣고 가져오는 애플리케이션에서 도달 가능한 최대 메시징 비율에 큰 영향을 줍니다.

그러나 `Confidentiality` 정책은 일련의 메시지에 대해 대칭 키 재사용을 허용합니다. 대칭 키 재사용을 통해 `Confidentiality` 정책을 사용하여 CPU 비용을 크게 절감할 수 있습니다. 이 조작 모드는 대칭 암호화 키를 공유하기 위해 PKCS#7 형식을 계속 사용합니다. 하지만 디지털 서명이 없으며, 이로 인해 메시지별 비대칭 키 조작의 일부가 제거됩니다. 대칭 키는 여전히 수신인마다 비대칭 키 조작을 사용하여 암호화되어야 하지만 선택적으로 동일 수신인을 대상으로 하는 여러 메시지에 반복적으로 재사용될 수 있습니다. 키 재사용이 정책에서 허용되면 첫 번째 메시지에만 비대칭 키 조작이 필요합니다. 후속 메시지는 대칭 키 조작만 사용해야 합니다.

## 키 재사용

`Confidentiality` 정책을 사용하면 대칭 키 재사용 방식을 사용하여 동일한 큐에 넣고 동일한 수신인 또는 수신인을 위한 다수의 메시지를 암호화하는 데 관련된 비용을 상당히 줄일 수 있습니다.

예를 들어, 동일한 수신인에 10개의 암호화된 메시지를 넣는 경우 대칭 키가 생성되고 의도된 각 메시지 수신인에 대해 비대칭 키 조작을 사용하여 첫 번째 메시지에 대해 암호화됩니다.

정책에서 제어하는 한계에 기반하여 암호화된 대칭 키는 동일한 수신인에 대해 의도된 후속 메시지에서 재사용할 수 있습니다. 후속 메시지에서 대칭 키를 재사용할 수 있게 하려면 큐에 메시지를 넣은 후 애플리케이션이 큐를 연 상태로 유지해야 합니다. MQPUT1 조작에서 대칭 키를 재사용할 수 없습니다. 암호화된 메시지를 가져오는 애플리케이션은 대칭 키가 변경되지 않아서 대칭 키를 검색하는 비용이 들지 않는 경우 애플리케이션에서 감지할 수 있는 동일한 최적화를 적용할 수 있습니다.

이 예에서 비대칭 키 조작의 90%는 동일한 키를 재사용하여 메시지를 넣고 가져오는 애플리케이션 모두에서 피할 수 있습니다.

키 재사용 방법에 대한 추가적인 정보는 다음을 참조하십시오.

- MQSC 명령 SET POLICY
- 제어 명령 setmqspl
- IBM i IBM i 명령 SETMQMSPL

# AMS의 키 개념

도구가 작동하는 방법 및 이를 효율적으로 사용하는 방법을 이해하려면 Advanced Message Security에서 키 개념에 대해 배우십시오.

## *PKI(Public Key Infrastructure) 및 Advanced Message Security*

PKI(Public Key Infrastructure)는 안전한 통신을 확보하기 위해 공개 키 암호화의 사용을 지원하는 기능, 정책 및 서비스의 시스템입니다.

PKI(Public Key Infrastructure)의 컴포넌트를 정의하는 단일 표준은 없지만 PKI는 일반적으로 공개 키 인증서의 사용과 관련되고 다음 서비스를 제공하는 인증 기관(CA) 및 기타 등록대행 기관(RA)으로 구성됩니다.

- 디지털 인증서 발행
- 디지털 인증서 유효성 검증
- 디지털 인증서 폐기
- 인증서 분배

사용자와 애플리케이션 ID는 서명되고 암호화된 메시지와 연관된 인증서에서 **식별 이름(DN)** 필드로 표시됩니다. Advanced Message Security는 이 ID를 사용하여 사용자 또는 애플리케이션을 나타냅니다. 이 ID를 인증하기 위해 사용자 또는 애플리케이션은 인증서 및 연관된 개인 키가 저장되는 키 저장소에 대한 액세스가 있어야 합니다. 각 인증서는 키 저장소에서 레이블로 표시됩니다.

**관련 개념**

599 페이지의 『AMS과(와) 함께 키스트로크와 인증서 사용』

IBM MQ 애플리케이션에 투명한 암호 보호를 제공하기 위해 Advanced Message Security는 공개 키 인증서 및 개인 키가 저장되는 키 저장소 파일을 사용합니다. z/OS에서 SAF키 링이 키 저장소 파일 대신 사용됩니다.

## *AMS*에서 디지털 인증서

Advanced Message Security는 사용자 및 애플리케이션을 X.509 표준 디지털 인증서와 연관시킵니다. X.509 인증서는 일반적으로 신뢰 인증 기관(CA)에 의해 서명되고 암호화 및 복호화에 사용되는 개인 및 공개 키와 관련됩니다.

디지털 인증서는 공개 키를 해당 소유자가 개인, 큐 관리자 또는 다른 엔티티인지 관계 없이 해당 소유자에게 바인딩하여 위장으로부터 보호를 제공합니다. 비대칭 키 설계를 사용할 때 공개키의 소유권에 대해 보장해주기 때문에, 디지털 인증서를 공개키 인증서라고도 합니다. 이 스키마에서는 애플리케이션을 위해 공개 키 및 개인 키가 생성되어야 합니다. 공개 키로 암호화된 데이터는 해당하는 개인 키를 사용해서만 복호화될 수 있는 반면 개인 키로 암호화된 데이터는 해당하는 공개 키를 사용해서만 복호화될 수 있습니다. 개인 키는 비밀번호 보호된 키 데이터베이스 파일에 저장됩니다. 해당 소유자만이 해당하는 공개 키를 사용하여 암호화되는 메시지를 복호화하는 데 사용되는 개인 키에 대한 액세스가 있습니다.

해당 소유자가 공개 키를 다른 엔티티로 직접 송신하는 경우, 메시지가 인터셉트되고 공개 키는 다른 키로 대체될 위험이 있습니다. 이는 "man-in-the-middle" 공격으로 알려져 있습니다. 해결책은 트러스트되는 Third-Party를 통해 공개키를 교환하는 것이며, 이렇게 하면 공개키가 정말로 사용자가 통신하고 있는 엔티티에 속하는지를 강력하게 보장해줍니다. 공개 키를 직접 송신하는 대신에, 신뢰되는 써드파티에게 그것을 디지털 인증서로 통합하도록 요청합니다. 디지털 인증서를 발행하는 신뢰되는 써드파티는 인증 기관(CA)이라고 불립니다.

디지털 인증서에 대한 자세한 정보는 디지털 인증서의 내용을 참조하십시오.

디지털 인증서에는 엔티티에 대한 공개 키가 있고 공개 키가 그 엔티티에 속한다는 언급이 있습니다.

- 인증서가 개인 엔티티에 대한 것일 때 이는 개인 인증서 또는 사용자 인증서라고 불립니다.
- 인증서가 인증 기관에 대한 것일 때 인증서는 *CA* 인증서 또는 서명자 인증서라고 불립니다.

**참고:** Advanced Message Security는 Java 및 고유 애플리케이션 둘 모두에서 자체 서명 인증서를 지원합니다.

**관련 개념**

10 페이지의 『암호화』

암호화(cryptography)는 읽기 가능한 텍스트인 일반 텍스트와 읽을 수 없는 양식인 암호문 사이에서의 변환 프로세스입니다.

## Multi 오브젝트 권한 관리자 및 *AMS*

멀티플랫폼에서 오브젝트 권한 관리자(OAM)는 IBM MQ 제품과 함께 제공되는 권한 서비스 컴포넌트입니다.

Advanced Message Security 엔티티에 대한 액세스는 IBM MQ 사용자 그룹 및 OAM을 통해 제어됩니다. 관리자는 명령행 인터페이스를 사용하여 필요에 따라 권한 부여를 부여하거나 취소할 수 있습니다. 서로 다른 사용자 그룹은 동일 오브젝트에 대해 서로 다른 종류의 액세스 권한을 가질 수 있습니다. 예를 들어, 한 그룹은 특정 큐에 대해 PUT 및 GET 조작 둘 모두를 수행할 수 있는 반면 다른 그룹은 큐를 찾아보기만이 허용될 수도 있습니다. 마찬가지로, 일부 그룹에는 큐에 대한 GET 및 PUT 권한이 있을 수도 있지만 큐를 변경하거나 삭제는 허용되지 않습니다.

OAM을 통해 다음을 제어할 수 있습니다.

- MQI (Message Queue Interface) 를 통해 Advanced Message Security 오브젝트에 액세스합니다. 애플리케이션 프로그램이 오브젝트에 액세스하려고 시도할 때 OAM은 요청을 하는 사용자 프로파일에 요청된 조작의 권한 부여가 있는지 여부를 검사합니다. 즉, 해당 큐 및 큐의 메시지는 비인가 액세스로부터 보호될 수 있음을 의미합니다.
- PCF 및 MQSC 명령 사용 권한.

**관련 개념**

오브젝트 권한 관리자

MQI(Message Queue Interface) 개요

## Advanced Message Security에 의해서 지원되는 기술

Advanced Message Security는 보안 인프라를 제공하기 위해 몇몇 기술 컴포넌트에 의존합니다.

Advanced Message Security는 다음 IBM MQ API(Application Programming Interface)를 지원합니다.

- 메시지 큐 인터페이스(MQI)
- IBM MQ Java Message Service(JMS) 1.0.2 및 1.1.
- Java용 IBM MQ 기본 클래스
- 비관리 모드에서 .Net을 위한 IBM MQ 클래스

**참고:** Advanced Message Security는 X.509 준수 인증 기관을 지원합니다.

## *AMS*의 알려진 제한사항

지원되지 않거나 Advanced Message Security에 대한 제한사항이 있는 여러 IBM MQ 옵션이 있습니다.

- 다음 IBM MQ 옵션은 지원되지 않거나 제한사항이 있습니다.

**발행/구독**
포인트-투-포인트 모델과 비교하여 발행/구독 메시징 모델의 주요 이점 중 하나는 송신 애플리케이션과 수신 애플리케이션이 데이터 송수신을 위해 서로에 대해 알지 않아도 된다는 점입니다. 그러나 의도한 수신인 또는 권한 있는 서명자를 정의해야 하는 Advanced Message Security 정책의 사용으로 이 이점이 상쇄되었습니다. 애플리케이션은 정책으로 보호되는 알리어스 큐 정의를 통해 토픽에 발행할 수 있고, 구독 애플리케이션은 정책으로 보호된 큐에서 메시지를 가져올 수도 있습니다. 정책은 토픽 문자열에 직접 지정할 수는 없고 큐 정의에만 지정할 수 있습니다.

**채널 데이터 변환**
Advanced Message Security로 보호되는 메시지의 보호된 페이로드는 2진 형식을 사용하여 전송되므로, 애플리케이션 간 채널에서 데이터 변환을 수행해도 메시지 요약이 무효화되지 않습니다. 정책으로 보호된 큐에서 메시지를 검색하는 애플리케이션이 데이터 변환을 요청해야 합니다. 메시지가 성공적으로 확인 및 보호 해제되면 보호된 페이로드의 변환이 시도됩니다.

**분배 목록**
목록에 있는 목적지 큐 각각에 동일한 정책이 정의되어 있는 경우 메시지를 분배 목록에 넣는 애플리케이션을 보호할 때 Advanced Message Security 정책을 사용할 수 있습니다. 애플리케이션이 분배 목록을 열 때 일치하지 않은 정책이 식별될 경우 열기 조작이 실패하며 애플리케이션에 보안 오류가 리턴됩니다.

**애플리케이션 메시지 세그먼트화.**
정책으로 보호된 메시지의 크기는 증가하므로 애플리케이션에서 메시지의 세그먼트 경계를 정확하게 지정할 수 없습니다.

**관리 모드(클라이언트 연결)에서 IBM MQ classes for .NET를 사용하는 애플리케이션**
IBM MQ classes for .NET를 관리 모드(클라이언트 연결)에서 사용하는 애플리케이션은 지원되지 않습니다.

**참고:** MCA 인터셉션은 지원되지 않는 클라이언트가 AMS를 사용할 수 있도록 허용할 수 있습니다.

**관리 모드의 .NET (XMS)용 메시지 서비스 클라이언트 애플리케이션**
.NET(XMS)용 메시지 서비스 클라이언트 애플리케이션은 관리 모드에서 지원되지 않습니다.

**참고:** MCA 인터셉션을 사용하여 지원되지 않는 클라이언트가 AMS를 사용하도록 허용할 수 있습니다.

**IMS 브릿지에서 처리되는 IBM MQ 큐**
IMS 브릿지에서 처리되는 IBM MQ 큐는 지원되지 않습니다.

**참고:** AMS는 CICS 브릿지 큐에서 지원됩니다. CICS 브릿지 큐에서 MQPUT(암호화) 및 MQGET(복호화)에 대해 동일 사용자 ID를 사용해야 합니다.

**대기 중인 Getter에 넣기**
대기 중인 Getter에 넣기는 AMS 정책이 정의되어 있는 큐에 대한 Getter 애플리케이션에서는 지원되지 않습니다.

z/OS **서버 대 서버 MCA 인터셉션**
IBM MQ for z/OS 9.1.3부터 송신자, 서버, 수신자, 및 요청자 채널 유형에 대해서만 서버 대 서버 MCA 인터셉션이 지원됩니다.

- 메시지를 보호할 때 사용할 인증서가 정의되지 않았으므로, 동일한 식별 이름을 사용하는 둘 이상의 인증서를 하나의 키 저장소에 넣으면 안됩니다.
- AMS 는 **WMQ_PROVIDER_VERSION** 특성이 6으로 설정된 경우 JMS 에서 지원되지 않습니다.
- AMS 인터셉터는 AMQP 또는 MQTT 채널을 지원하지 않습니다.

## z/OS Advanced Message Security interception on message channels

On z/OS, Advanced Message Security (AMS) interception provides an additional option of security policy protection (SPLPROT) to sender, server, receiver, and requester channels, allowing you to support AMS and to communicate with business partners who do not support AMS.

Taking the example of a clearing house communicating with a bank, Figure 1 shows that, without AMS interception, both sides of the system need to support AMS.

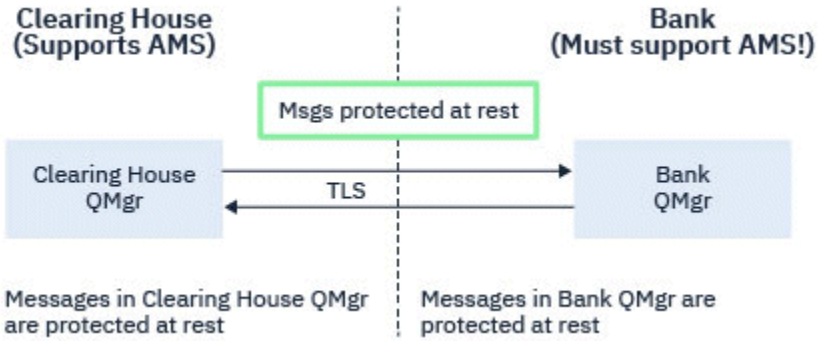



*Figure 32. Usage of AMS without AMS interception*

A key benefit of the AMS interception option is, that if your enterprise has AMS configured, and not all of your business partners support AMS, you can remove protection from outbound messages and protect inbound messages on channels to and from those business partners that do not support AMS.

Using the example of a clearing house and banks, this scenario is shown in Figure 2, where there is a message flow between the clearing house, banks, and business partners where some institutions have AMS, and others do not.

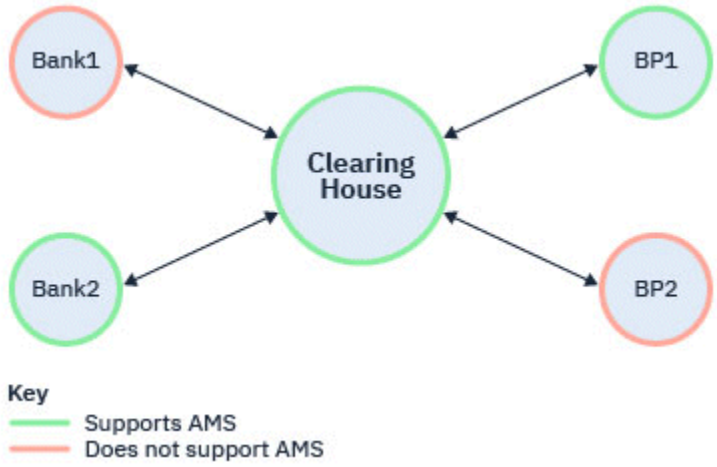



*Figure 33. Some partners support AMS and some do not*

Typically the channels are TLS enabled.

However, there might be a case where some banks and business partners do not support AMS, and there is a requirement to be able to exchange messages between all banks and business partners. This scenario is shown in Figure 3

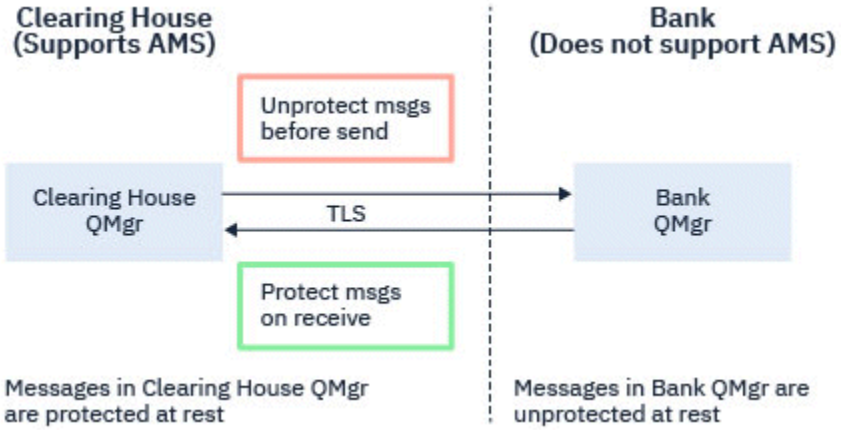



*Figure 34. Message flow between business partners*

**Related tasks**

Server-to-server message channel interception example configurations

## z/OS *AMS interception on server-to-server message channels*

Server-to-server message channel interception provides a means to control if messages should have any applicable Advanced Message Security (AMS) policies applied to them, when sender type message channel agents get messages from transmission queues, and receiver type message channel agents put messages to target queues.

This allows AMS protection to be enabled on a queue manager when communicating, using server-to-server message channels of type sender, server, receiver, and requester, with a queue manager that does not have AMS enabled.

That is, AMS protected messages in AMS enabled queue managers can be unprotected prior to being sent to non-AMS enabled queue managers, and unprotected messages received from non-AMS enabled queue managers can be protected, by applicable AMS policies, on AMS enabled queue managers.

### Configuring server-to-server message channel interception

Server-to-server message channel interception is configured with the SPLPROT attribute on channels with a channel type of sender, server, receiver, or requester. The available options to configure the behavior are dependent on the channel type specified:

**PASSTHRU**

이 채널의 메시지 채널 에이전트에서 송신하거나 수신한 변경되지 않은 모든 메시지를 통과합니다.

이 값은 SDR, SVR, RCVR 또는 RQSTR의 채널 유형(**CHLTYPE**)이 있는 채널에 대해 유효하며 기본값입니다.

**REMOVE**

메시지 채널 에이전트에 의해 전송 큐에서 검색된 메시지에서 모든 AMS 보호를 제거하며 파트너에 메시지를 송신합니다.

메시지 채널 에이전트가 전송 큐에서 메시지를 가져올 때 전송 큐에 대해 AMS 정책이 정의된 경우, 채널을 통해 메시지를 송신하기 전에 메시지에서 AMS 보호를 제거하기 위해 적용됩니다. 전송 큐에 대해 AMS 정책이 정의되지 않은 경우 메시지가 현상태대로 송신됩니다.

이 값은 채널 유형이 SDR 또는 SVR인 채널에 대해서만 유효합니다.

**ASPOLICY**

대상 큐에 대해 정의된 정책을 기반으로 하여 인바운드 메시지를 대상 큐에 넣기 전에 AMS 보호를 적용합니다.

메시지 채널 에이전트가 인바운드 메시지를 수신할 때 대상 큐에 대해 AMS 정책이 정의된 경우, 메시지를 대상 큐에 넣기 전에 메시지에 AMS 보호가 적용됩니다. 대상 큐에 대한 AMS 정책이 정의되지 않은 경우 메시지를 그대로 대상 큐에 넣습니다.

이 값은 채널 유형이 RCVR 또는 RQSTR인 채널에 대해서만 유효합니다.

### User ID for message channel interception

The requirement for user IDs used with server-to-server message channel interception are the same as those for existing AMS enabled applications. For a running channel, the sending message channel agent gets messages from a transmission queue and the receiving message channel agent puts messages to target queues. The message channel agent user ID (MCAUSER) field, set on server to server channels, defines the user ID under which message channel agents perform put and get requests.

With server-to-server message channel interception, AMS functions are performed during get and put requests, as with other AMS enabled applications. Therefore, message channel agent user Ids have the same requirements as those for AMS application user IDs.

The MCAUSER used to perform the put and get is configurable, and dependent on whether it is an outbound or inbound channel. See MCAUSER for details of how the chosen user ID performs actions on the message channel agent. As such, the user ID that the channel initiator is running under is the user ID that is to be used for AMS functions performed during server-to-server message channel interception. Therefore, these user IDs have the same requirements as those for AMS application user IDs.

Authentication is performed using the existing rules for the channel detailed for channels with PUTAUT configuration. See user IDs used by the channel initiator for more information.

**Note:** Server-to-server message channel interception does not take into account the value of the PUTAUT channel attribute.

### Message size and MAXMSGL

Due to AMS protection, the message size of protected messages will be larger than the original message size.

Protected messages are larger than unprotected messages. Therefore, the value of the **MAXMSGL** attribute, on both queues and channels, might need to be altered to take into account the size of protected messages.

**Related reference**

Server-to-server message channel interception example configurations

## AMS의 오류 처리

IBM MQ Advanced Message Security는 오류가 포함된 메시지 또는 보호할 수 없는 메시지를 관리할 오류 핸들링 큐를 정의합니다.

결함 메시지는 예외적인 케이스로 처리됩니다. 수신된 메시지가 현재 있는 큐의 보안 요구사항을 충족하지 않을 경우, 예를 들어 암호화해야 하는 메시지가 서명되었거나 복호화 또는 서명 확인이 실패한 경우 메시지가 오류 핸들링 큐로 송신됩니다. 다음과 같은 이유로 메시지가 오류 핸들링 큐로 송신될 수 있습니다.

- QoP(Quality of Protection) 불일치 - 수신된 메시지와 보안 정책의 QoP 정의 간에 QOP 불일치가 존재합니다.
- 복호화 오류 - 메시지를 복호화할 수 없습니다.
- PDMQ 헤더 오류 - Advanced Message Security(AMS) 메시지 헤더에 액세스할 수 없습니다.
- 크기 불일치 - 복호화 후 메시지 길이가 예상과 다릅니다.
- 암호화 알고리즘 강도 불일치 - 메시지 암호화 알고리즘이 필요한 것보다 약합니다.
- 알 수 없는 오류 - 예상치 못한 오류가 발생했습니다.

AMS 는 SYSTEM.PROTECTION.ERROR.QUEUE 를 오류 핸들링 큐로 QUEUE합니다. IBM MQ AMS에서 SYSTEM.PROTECTION.ERROR.QUEUE 앞에 MQDLH 헤더가 있습니다.

IBM MQ 관리자가 SYSTEM.PROTECTION.ERROR.QUEUE 를 다른 큐를 가리키는 알리어스 큐로 QUEUE합니다.

z/OS IBM MQ for z/OS에서 서버 대 서버 메시지 채널 에이전트 (MCA) 인터셉션이 사용 중인 경우:

- 앞에서 설명한 이유 중 하나로 IBM MQ AMS가 메시지를 전송 큐에서 오류 핸들링 큐로 이동시킨 경우, 송신자 MCA는 사용 가능한 다음 메시지를 전송 큐에서 계속 처리합니다.

- 일반적으로 기존 채널 규칙은 다음 사항에 적용됩니다.
  - 데드-레터 큐에 메시지 넣기
  - 데드-레터 큐에 넣기가 실패할 경우에 수행할 조치

특정 시나리오에 대한 자세한 정보는 565 페이지의 『z/OS 의 AMS 에 대한 전달되지 않은 메시지』의 내용을 참조하십시오.

## z/OS z/OS 의 AMS 에 대한 전달되지 않은 메시지

IBM MQ for z/OS에서 발생하는 서버 대 서버 메시지 채널 에이전트 인터셉션과 관련된 특정 상황입니다.

IBM MQ for z/OS에서 서버 대 서버 메시지 채널 에이전트 (MCA) 인터셉션이 사용 중인 경우:

- 메시지를 받아 보호 해제한 후 예를 들어 메시지가 채널에 너무 큰 이유로 송신자 MCA가 메시지 전달에 실패하고 USEDLQ 송신자 채널 속성이 YES로 설정되면, 송신자 MCA는 메시지를 로컬 DLQ(Dead Letter Queue)로 이동시킵니다.

  로컬 DLQ로서 SYSTEM.DEAD.LETTER.QUEUE를 사용 중인 경우 메시지는 비보호 상태로 배치됩니다.

  **참고:** IBM MQ AMS는 시스템 큐에 넣은 메시지의 보호를 지원하지 않습니다.

  이름 지정된 DLQ가 로컬 DLQ로 사용되는 경우, IBM MQ AMS 정책을 이름 지정된 DLQ와 동일한 이름으로 정의했다면 메시지가 보호된 상태로 배치되고, 적합한 정책을 정의하지 않았다면 비보호 상태로 배치됩니다.
- 어떤 이유로 메시지를 로컬 DLQ에 넣을 수 없는 경우, 채널의 NPMSPEED가 NORMAL로 설정되어 있거나 메시지가 지속 메시지이면 메시지의 현재 배치가 백아웃되고 채널이 RETRY 상태가 됩니다. 그렇지 않은 경우 메시지가 제거되고 송신자 MCA가 계속해서 전송 큐의 다음 메시지를 처리합니다.
- 보안 정책은 SYSTEM.DEAD.LETTER.QUEUE 또는 634 페이지의 『AMS의 시스템 큐 보호』에 나열된 다른 SYSTEM 큐에 영향을 미치지 않으므로, SYSTEM.DEAD.LETTER.QUEUE가 사용 중일 경우, MCA가 이 큐에 넣은 메시지는 현재 상태 그대로 배치됩니다. 즉, 이전에 보호된 메시지는 보호된 상태로 배치되고, 그렇지 않은 메시지는 비보호 상태로 배치됩니다.

  큐 관리자 DEADQ 속성이 대체(비시스템) 데드-레터 큐의 이름으로 설정되고 동일한 이름의 AMS 정책이 없을 경우, MCA가 이 큐에 넣은 메시지는 현재 상태 그대로 배치됩니다. 즉, 이전에 보호된 메시지는 보호된 상태로 배치되고, 그렇지 않은 메시지는 비보호 상태로 배치됩니다.

  큐 관리자 DEADQ 속성이 대체(비시스템) 데드-레터 큐의 이름으로 설정되고 동일한 이름의 AMS 정책이 있을 경우, 해당 정책으로 MCA가 이 큐에 넣은 메시지가 보호됩니다. 이전에 이미 보호된 메시지는 다시 보호되지 않는데, 이는 이중 보호를 방지하기 위한 것입니다. 동일한 이름의 AMS 정책이 없을 경우 메시지는 현재 상태 그대로 배치됩니다.
- setmqspl 명령에 있는 허용 옵션이 오프(**-t** O)로 설정되어 있는 DLQ에 대한 정책이 있는 경우, 메시지가 AMS 보호 상태가 아니면 DLQ에 넣기는 실패하므로 PDMQ 헤더가 없습니다. 이 상황은 메시지가 PDMQ 헤더 없이 수신자에게 도착한 경우에 발생합니다. 즉, 메시지의 원래 putter에는 목적지에 대한 정책이 없으며, 수신자에는 SPLPROT(ASPOLICY)가 설정되어 있지 않습니다.
- DLQ에 대해 정의된 AMS 정책이 채널 시작기를 실행 중인 사용자 ID가 메시지를 보호하는 것을 허용하지 않는 경우에는 MCA가 DLQ에 메시지를 넣는 데 실패할 수 있습니다.
- 수신자 채널은 일반적으로 미전달 메시지를 로컬 DLQ에 배치하지만, 송신자 채널은 일반적으로 메시지가 큐에 비해 너무 크거나 잘못된 MQXQH 헤더 등의 어떤 이유로 보호할 수 없는 메시지를 로컬 DLQ에 배치합니다.
- DLQ 핸들러는 일반적으로 메시지 페이로드(payload) 자체가 아니라 DLQ 헤더(DLH)만 검토합니다. 따라서 핸들러는 메시지 페이로드가 보호된다는 사실만으로 메시지가 DLQ에 배치된 이유를 판별할 수 밖에 없습니다.
- DLQ가 정의되지 않은 경우 채널은 다음 상태가 됩니다.
  - 지속 메시지를 전달할 수 없는 경우 비상적으로 종료되고 재시도 상태가 됩니다.
  - 비지속 미전달 메시지를 제거하고 계속 실행됩니다.

**관련 개념**

564 페이지의 『AMS의 오류 처리』

IBM MQ Advanced Message Security는 오류가 포함된 메시지 또는 보호할 수 없는 메시지를 관리할 오류 핸들링 큐를 정의합니다.

## AMS의 사용자 시나리오

Advanced Message Security로 이룰 수 있는 비즈니스 목적에 대해 이해하려면 가능한 시나리오에 친숙해지십시오.

### Windows *Windows 플랫폼의 AMS 용 빠른 시작 안내서*

이 안내서를 사용하여 Windows 플랫폼에서 메시지 보안을 제공하도록 Advanced Message Security (AMS) 를 빠르게 구성하십시오. 이를 완료할 때쯤 사용자 ID를 확인하기 위한 키 데이터베이스를 작성하게 될 것이고 큐 관리자의 서명/암호화 정책을 정의했을 것입니다.

#### 시작하기 전에

최소한 다음 기능이 시스템에 설치되어 있어야 합니다.

- 서버
- 개발 툴킷(샘플 프로그램용)
- Advanced Message Security (AMS)

자세한 내용은 Windows 시스템의IBM MQ 기능 을 참조하십시오.

**setmqenv** 명령을 사용하여 운영 체제에서 적절한 IBM MQ 명령을 찾아 실행할 수 있도록 현재 환경을 초기화하는 방법에 대한 정보는 setmqenv (set IBM MQ environment)를 참조하십시오.

*1.* 큐 관리자 및 큐 작성

#### 이 태스크 정보

다음 모든 예에서는 애플리케이션 간 메시지를 전달하기 위해 이름이 TEST.Q인 큐를 사용합니다. Advanced Message Security는 표준 IBM MQ 인터페이스를 통해 IBM MQ 인프라에 입력하는 지점에서 메시지에 서명하고 암호화하기 위해 인터셉터를 사용합니다. 이 기본 설정은 IBM MQ에서 수행되고 다음 단계에서 구성됩니다.

IBM MQ Explorer 를 사용하여 모든 기본 마법사 설정을 사용하여 큐 관리자 QM_VERIFY_AMS및 TEST.Q 라는 로컬 큐를 작성하거나 C:\Program Files\IBM\MQ\bin에 있는 명령을 사용할 수 있습니다. 다음 관리 명령을 실행하려면 mqm 사용자 그룹의 구성원이어야 함을 기억하십시오.

#### 프로시저

1. 큐 관리자 작성

```
crtmqm QM_VERIFY_AMS
```

2. 큐 관리자 시작

```
strmqm QM_VERIFY_AMS
```

3. 다음 명령을 큐 관리자 QM_VERIFY_AMS의 **runmqsc**에 입력하여 이름이 TEST.Q인 큐를 작성하십시오.

```
DEFINE QLOCAL(TEST.Q)
```

#### 결과

프로시저가 완료되면 **runmqsc**에 입력된 명령이 TEST.Q에 대한 세부사항을 표시합니다.

```
DISPLAY Q(TEST.Q)
```

*2.* 사용자 작성 및 권한 부여

## 이 태스크 정보

이 예에는 2명의 사용자가 나타납니다. 송신자 `alice`와 수신자 `bob`입니다. 애플리케이션 큐를 사용하려면 이러한 사용자에게는 이를 사용할 권한이 부여되어야 합니다. 또한 우리가 정의한 보호 정책을 성공적으로 사용하기 위해 이러한 사용자에게는 일부 시스템 큐에 대한 액세스가 부여되어야 합니다. **setmqaut** 명령에 대한 자세한 정보는 **setmqaut**를 참조하십시오.

## 프로시저

1. 2명의 사용자를 작성하고 이러한 두 사용자 모두에게 `HOMEPATH` 및 `HOMEDRIVE`가 설정되었는지 확인하십시오.
2. 사용자에게 큐 관리자에 연결하고 큐에 대해 작업할 수 있는 권한을 부여

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
```

```
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
```

```
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get
```

3. 또한 두 사용자가 시스템 정책 큐를 찾아서 메시지를 오류 큐에 넣도록 허용해야 합니다.

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse
```

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put
```

**주의:** IBM MQ 는 SYSTEM.PROTECTION.POLICY.QUEUE 를 QUEUE합니다.

IBM MQ가 사용 가능한 모든 정책을 캐시하지는 않습니다. 정책 수가 많으면 IBM MQ가 제한된 수의 정책을 캐시합니다. 따라서 큐 관리자에 적은 수의 정책이 정의되어 있는 경우 SYSTEM.PROTECTION.POLICY.QUEUE에 찾아보기 옵션을 제공할 필요가 없습니다.

그러나 많은 수의 정책이 정의되어 있거나 이전 클라이언트를 사용 중인 경우 이 큐에 대한 찾아보기 권한을 부여해야 합니다. SYSTEM.PROTECTION.ERROR.QUEUE는 AMS 코드에서 생성된 오류 메시지를 넣는 데 사용됩니다. 이 큐에 대한 넣기(Put) 권한은 오류 메시지를 큐에 넣으려고 할 때만 확인됩니다. AMS 보호 큐에서 메시지를 가져오거나 넣으려고 할 때는 큐에 대한 넣기(Put) 권한이 확인되지 않습니다.

## 결과

그러면 사용자가 작성되고 사용자에게 필요한 권한이 부여됩니다.

## 다음에 수행할 작업

단계가 올바르게 수행되었는지 확인하려면 570 페이지의 『7. 설정 테스트』 절에 설명된 대로 `amqsput` 및 `amqsget` 샘플을 사용하십시오.

*3.* 키 데이터베이스 및 인증서 작성

## 이 태스크 정보

인터셉터에는 메시지를 암호화하기 위해 송신 사용자의 공개 키를 요구합니다. 따라서 공개 및 개인 키에 맵핑된 사용자의 키 데이터베이스가 작성되어야 합니다. 사용자 및 애플리케이션이 여러 컴퓨터에 분산되어 있는 실제 시스템에서는 각 사용자는 고유의 개인용 키 저장소를 가지고 있을 것입니다. 마찬가지로, 이 안내서에서는 `alice`와 `bob`을 위한 키 데이터베이스를 작성하고 이들 간에 사용자 인증서를 공유합니다.

**참고:** 이 안내서에서는 로컬 바인딩을 사용하여 연결하는 C로 작성된 샘플 애플리케이션을 사용합니다. 클라이언트 바인딩을 사용하여 Java 애플리케이션을 사용하려는 경우, Java **keytool** 명령 V 9.4.0 V 9.4.0 또는 IBM MQ **runmqktool** 명령을 사용하여 JKS키 저장소 및 인증서를 작성해야 합니다. 자세한 정보는 587 페이지의 『Java 클라이언트가 있는 AMS 용 빠른 시작 안내서』의 내용을 참조하십시오. 다른 모든 언어 및 로컬 바인딩을 사용하는 Java 애플리케이션의 경우 이 안내서의 단계가 정확합니다.

## 프로시저

1. alice사용자에 대한 새 키 데이터베이스를 작성하십시오.
   예를 들어, 다음 명령을 실행하여 새 키 데이터베이스를 작성하십시오.

   ```
   runmqakm -keydb -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -type cms -pw
   passw0rd -stash
   ```

   **참고:**

   - 강력한 비밀번호를 사용하여 데이터베이스를 보호하십시오.
   - **-stash** 매개변수를 포함하여 암호화된 키 데이터베이스 비밀번호를 파일에 숨기십시오.
2. 새 자체 서명 인증서를 작성하여 암호화에 사용할 alice 사용자를 식별하십시오.
   예를 들어, 다음 명령을 실행하여 새 자체 서명 인증서를 작성하십시오.

   ```
   runmqakm -cert -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -stashed
   -label Alice_Cert -dn "CN=alice, O=IBM, C=GB"
   ```

   **참고:**

   - 이 안내서의 용도상 인증 기관을 사용하지 않고 작성할 수 있는 자체 서명 인증서를 사용 중입니다. 프로덕션 시스템의 경우 인증 기관에서 서명한 인증서를 사용하는 것이 좋습니다.
   - **-label** 매개변수는 인터셉터가 필요한 정보를 수신하기 위해 검색할 인증서의 이름을 지정합니다.
   - **-dn** 매개변수는 인증서에 대한 식별 이름 (DN) 의 세부사항을 지정합니다. 식별 이름은 각 사용자에 대해 고유해야 합니다.
3. bob사용자에 대해 568 페이지의 『1』 및 568 페이지의 『2』 단계를 반복하십시오.

## 결과

2명의 사용자 alice 및 bob은 이제 각각 자체 서명 인증서가 있습니다.

*4. keystore.conf* 작성

## 이 태스크 정보

Advanced Message Security 인터셉터로 키 데이터베이스 및 인증서가 있는 디렉토리를 가리켜야 합니다. 이는 해당 정보를 일반 텍스트 양식으로 보유하는 keystore.conf 파일을 통해 수행됩니다. 각 사용자에게는 .mqs 폴더에 별도의 keystore.conf 파일이 있어야 합니다. alice 및 bob 모두에 대해 이 단계를 완료해야 합니다.

keystore.conf의 컨텐츠를 다음 양식이어야 합니다.

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

### 예

이 시나리오의 경우 keystore.conf의 컨텐츠는 다음과 같습니다.

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

**참고:**

- 키 저장소 파일의 경로는 파일 확장자 없이 제공되어야 합니다.
- 인증서 레이블에는 공백을 포함할 수 있으므로 예를 들어, "Alice_Cert" 및 "Alice_Cert "(끝에 공백 포함)는 두 개의 서로 다른 인증서의 레이블로 인식됩니다. 그러나 혼동을 피하기 위해 레이블의 이름에는 공백을 사용하지 않는 것이 좋습니다.
- CMS(Cryptographic Message Syntax), Java 키 저장소(JKS) 및 JCEKS(Java Cryptographic Extension Keystore)의 키 저장소 형식이 있습니다. 자세한 정보는 600 페이지의 『AMS 에 대한 키 저장소 구성 파일 (keystore.conf) 의 구조』 의 내용을 참조하십시오.
- Advanced Message Security가 `keystore.conf` 파일을 검색하는 기본 위치는 *%HOMEDRIVE%**%HOMEPATH%*\`.mqs\keystore.conf`(예: C:\Documents and Settings\alice\.mqs\keystore.conf)입니다. `keystore.conf`에 대해 기본이 아닌 위치를 사용하는 방법에 대한 정보는 599 페이지의 『AMS과(와) 함께 키스트로크와 인증서 사용』 의 내용을 참조하십시오.
- `.mqs` 디렉토리를 작성하려면 명령 프롬프트를 사용해야 합니다.

*5.* 인증서 공유

## 이 태스크 정보

각 사용자가 서로를 올바르게 식별할 수 있도록 두 키 데이터베이스 사이에서 인증서를 공유하십시오. 각 사용자의 공용 인증서를 파일에 내보내면 이 파일이 다른 사용자의 키 데이터베이스에 추가되는 방식으로 수행됩니다.

**참고:** 신중하게 추출 옵션을 사용하고 내보내기 옵션은 사용하지 마십시오. 추출은 사용자의 공개 키를 가져오지만 내보내기는 공개 키와 개인 키를 모두 가져옵니다. 실수로 추출을 사용하는 경우 개인 키가 누출되어 애플리케이션이 완전히 훼손될 수 있습니다.

## 프로시저

1. `alice`를 식별하는 인증서를 외부 파일로 추출하십시오.

```
runmqakm -cert -extract -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. 인증서를 `bob's` 키 저장소에 추가하십시오.

```
runmqakm -cert -add -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. bob에 대해 단계를 반복하십시오.

```
runmqakm -cert -extract -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm
```

```
runmqakm -cert -add -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Bob_Cert -file bob_public.arm
```

## 결과

두 명의 사용자 `alice` 및 `bob`은 이제 자체 서명 인증서를 작성하고 공유한 서로를 성공적으로 식별할 수 있습니다.

## 다음에 수행할 작업

GUI를 사용하여 찾아보거나 자세한 내용을 출력하는 다음 명령을 실행하여 인증서가 키 저장소에 있는지 확인하십시오.

```
runmqakm -cert -details -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd -label
Alice_Cert
```

```
runmqakm -cert -details -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Bob_Cert
```

6. 큐 정책 정의

## 이 태스크 정보

큐 관리자가 작성되고 인터셉터가 메시지 및 액세스 암호화 키를 방해할 준비가 된 상태에서 `setmqspl` 명령을 사용하여 `QM_VERIFY_AMS`에서 보호 정책 정의를 시작할 수 있습니다. 이 명령에 대한 자세한 정보는 setmqspl 의 내용을 참조하십시오. 각 정책 이름은 적용되는 큐 이름과 동일해야 합니다.

### 예

이는 `TEST.Q` 큐에 정의된 정책의 예입니다. 예에서 메시지는 Deprecated SHA1 알고리즘으로 서명되고 `AES256` 알고리즘으로 암호화됩니다. `alice`는 유일한 유효한 송신자이고 `bob`은 이 큐에서 메시지의 유일한 수신자입니다.

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

**참고:** DN은 키 데이터베이스의 각 사용자 인증서에 지정된 DN과 정확히 일치합니다.

## 다음에 수행할 작업

사용자가 정의한 정책을 확인하려면 다음 명령을 실행하십시오.

```
dspmqspl -m QM_VERIFY_AMS
```

`setmqspl` 명령의 세트로 정책 세부사항을 인쇄하려면 `-export` 플래그를 사용하십시오. 이는 이미 정의된 정책을 저장할 수 있게 해줍니다.

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 설정 테스트

## 이 태스크 정보

다른 사용자 하에서 다른 프로그램을 실행하여 애플리케이션이 제대로 구성되었는지 확인할 수 있습니다.

## 프로시저

1. 사용자 `alice`로서 실행하도록 사용자를 전환하십시오.

   `cmd.exe`를 마우스 오른쪽 단추로 클릭하고 **다음으로 실행...**을 선택하십시오. 메시지가 표시되면 사용자 `alice`로 로그인하십시오.
2. `alice` 사용자가 샘플 애플리케이션을 사용하여 메시지를 넣을 때:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. 메시지의 텍스트를 입력한 다음 Enter를 누르십시오.
4. 사용자 `bob`로서 실행하도록 사용자를 전환하십시오.

   `cmd.exe`를 마우스 오른쪽 단추로 클릭하고 **다음으로 실행...**을 선택하여 다른 창을 여십시오. 메시지가 표시되면 사용자 `bob`로 로그인하십시오.
5. `bob` 사용자가 샘플 애플리케이션을 사용하여 메시지를 가져올 때:

```
amqsget TEST.Q QM_VERIFY_AMS
```

### 결과
애플리케이션이 두 사용자에게 모두 적절하게 구성된 경우 사용자 alice의 메시지가 bob이 가져오기 애플리케이션을 실행할 때 표시됩니다.

*8.* 암호화 테스트

### 이 태스크 정보
암호화가 예상대로 발생했는지 확인하려면 원본 큐 TEST.Q를 참조하는 알리어스 큐를 작성하십시오. 이 알리어스 큐에는 보안 정책이 없고 메시지를 복호화하기 위한 정보를 가진 사용자가 없으므로, 암호화된 데이터가 표시될 것입니다.

### 프로시저
1. 큐 관리자 QM_VERIFY_AMS에 대해 **runmqsc** 명령을 사용하여 알리어스 큐를 작성하십시오.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. 알리어스 큐에서 찾아보기 위해 bob에 액세스를 부여하십시오.

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. alice 사용자로서 방금 전처럼 샘플 애플리케이션을 사용하여 또 다른 메시지를 넣으십시오.

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. bob 사용자로서 이번에는 알리어스 큐를 통해 샘플 애플리케이션을 사용하여 메시지를 찾으십시오.

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. bob 사용자로서 로컬 큐로부터 샘플 애플리케이션을 사용하여 메시지를 가져오십시오.

```
amqsget TEST.Q QM_VERIFY_AMS
```

### 결과
amqsbcg 애플리케이션의 출력에는 메시지가 암호화되었음을 증명하는 큐에 있는 암호화된 데이터가 표시됩니다.

## Linux AIX *AIX and Linux 의 AMS 용 빠른 시작 안내서*
이 안내서를 사용하여 AIX and Linux에서 메시지 보안을 제공하도록 Advanced Message Security 를 빠르게 구성하십시오. 이를 완료할 때쯤 사용자 ID를 확인하기 위한 키 데이터베이스를 작성하게 될 것이고 큐 관리자의 서명/암호화 정책을 정의했을 것입니다.

### 시작하기 전에
최소한 다음 컴포넌트가 시스템에 설치되어 있어야 합니다.

- 런타임
- 서버
- 샘플 프로그램
- IBM Global Security Kit (GSKit)

- Advanced Message Security

각 특정 플랫폼에서 컴포넌트 이름에 대해서는 다음 주제를 참조하십시오.

- Linux Linux 시스템용IBM MQ 구성요소
- AIX AIX 시스템용IBM MQ 구성요소

*1.* 큐 관리자 및 큐 작성

### 이 태스크 정보

다음 모든 예에서는 애플리케이션 간 메시지를 전달하기 위해 이름이 TEST.Q인 큐를 사용합니다. Advanced Message Security는 표준 IBM MQ 인터페이스를 통해 IBM MQ 인프라에 입력하는 지점에서 메시지에 서명하고 암호화하기 위해 인터셉터를 사용합니다. 이 기본 설정은 IBM MQ에서 수행되고 다음 단계에서 구성됩니다.

IBM MQ 탐색기를 사용하여 모든 기본 마법사 설정을 통해 큐 관리자(QM_VERIFY_AMS) 및 해당 로컬 큐(TEST.Q)를 작성하거나 *MQ_INSTALLATION_PATH*/bin에 있는 명령을 사용할 수 있습니다. 다음 관리 명령을 실행하려면 mqm 사용자 그룹의 구성원이어야 함을 기억하십시오.

### 프로시저

1. 큐 관리자 작성

```
crtmqm QM_VERIFY_AMS
```

2. 큐 관리자 시작

```
strmqm QM_VERIFY_AMS
```

3. 다음 명령을 큐 관리자 QM_VERIFY_AMS의 **runmqsc**에 입력하여 이름이 TEST.Q인 큐를 작성하십시오.

```
DEFINE QLOCAL(TEST.Q)
```

### 결과

프로시저가 성공적으로 완료되면 **runmqsc**에 입력된 다음 명령이 TEST.Q에 대한 세부사항을 표시합니다.

```
DISPLAY Q(TEST.Q)
```

*2.* 사용자 작성 및 권한 부여

### 이 태스크 정보

이 예에는 2명의 사용자가 나타납니다. 송신자 alice와 수신자 bob입니다. 애플리케이션 큐를 사용하려면 이러한 사용자에게는 이를 사용할 권한이 부여되어야 합니다. 또한 우리가 정의한 보호 정책을 성공적으로 사용하기 위해 이러한 사용자에게는 일부 시스템 큐에 대한 액세스가 부여되어야 합니다. **setmqaut** 명령에 대한 자세한 정보는 **setmqaut**를 참조하십시오.

### 프로시저

1. 2명의 사용자를 작성하십시오

```
useradd alice
```

```
useradd bob
```

2. 사용자에게 큐 관리자에 연결하고 큐에 대해 작업할 수 있는 권한을 부여

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
```

```
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
```

```
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get
```

3. 또한 두 사용자가 시스템 정책 큐를 찾아서 메시지를 오류 큐에 넣도록 허용해야 합니다.

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse
```

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put
```

**주의:** IBM MQ 는 SYSTEM.PROTECTION.POLICY.QUEUE 를 QUEUE합니다.

IBM MQ가 사용 가능한 모든 정책을 캐시하지는 않습니다. 정책 수가 많으면 IBM MQ가 제한된 수의 정책을 캐시합니다. 따라서 큐 관리자에 적은 수의 정책이 정의되어 있는 경우 SYSTEM.PROTECTION.POLICY.QUEUE에 찾아보기 옵션을 제공할 필요가 없습니다.

그러나 많은 수의 정책이 정의되어 있거나 이전 클라이언트를 사용 중인 경우 이 큐에 대한 찾아보기 권한을 부여해야 합니다. SYSTEM.PROTECTION.ERROR.QUEUE는 AMS 코드에서 생성된 오류 메시지를 넣는 데 사용됩니다. 이 큐에 대한 넣기(Put) 권한은 오류 메시지를 큐에 넣으려고 할 때만 확인됩니다. AMS 보호 큐에서 메시지를 가져오거나 넣으려고 할 때는 큐에 대한 넣기(Put) 권한이 확인되지 않습니다.

## 결과

이제 사용자 그룹이 작성되고 필수 권한이 부여됩니다. 이런 식으로, 사용자 그룹에 지정된 사용자는 큐 관리자에 연결하고 큐에 넣고 큐에서 가져올 권한이 가지게 됩니다.

## 다음에 수행할 작업

단계가 올바르게 수행되었는지 확인하려면 577 페이지의 『8. 암호화 테스트』 절에 설명된 대로 `amqsput` 및 `amqsget` 샘플을 사용하십시오.

*3.* 키 데이터베이스 및 인증서 작성

## 이 태스크 정보

메시지를 암호화하기 위해 인터셉터는 송신 사용자의 개인 키와 수신인의 공개 키가 필요합니다. 따라서 공개 및 개인 키에 맵핑된 사용자의 키 데이터베이스가 작성되어야 합니다. 사용자 및 애플리케이션이 여러 컴퓨터에 분산되어 있는 실제 시스템에서는 각 사용자는 고유의 개인용 키 저장소를 가지고 있을 것입니다. 마찬가지로, 이 안내서에서는 `alice`와 `bob`을 위한 키 데이터베이스를 작성하고 이들 간에 사용자 인증서를 공유합니다.

**참고:** 이 안내서에서는 로컬 바인딩을 사용하여 연결하는 C로 작성된 샘플 애플리케이션을 사용합니다. 클라이언트 바인딩을 사용하여 Java 애플리케이션을 사용할 계획이면 JRE의 일부인 **keytool** 명령을 사용하여 JKS 키 저장소 및 인증서를 작성해야 합니다(세부사항은 587 페이지의 『Java 클라이언트가 있는 AMS 용 빠른 시작 안내서』 의 내용을 참조하십시오.). 다른 모든 언어 및 로컬 바인딩을 사용하는 Java 애플리케이션의 경우 이 안내서의 단계가 정확합니다.

## 프로시저

1. 사용자 `alice`의 새 키 데이터베이스를 작성하십시오.

```
mkdir /home/alice/.mqs -p
```

```
runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
```

**참고:**

- 데이터베이스를 보안 설정하기 위해 강력한 비밀번호를 사용하는 것이 권장됩니다.
- **stash** 매개변수는 비밀번호를 `key.sth` 파일에 저장하고 이는 인터셉터가 데이터베이스를 열기 위해 사용할 수 있습니다.

2. 키 데이터베이스가 읽기 가능한지 확인하십시오.

```
chmod +r /home/alice/.mqs/alicekey.kdb
```

3. 암호화에 사용하기 위해 `alice` 사용자를 식별하는 인증서를 작성하십시오.

```
runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -dn
"cn=alice,O=IBM,c=GB" -default_cert yes
```

**참고:**

- 이 안내서의 용도상 인증 기관을 사용하지 않고 작성할 수 있는 자체 서명 인증서를 사용 중입니다. 프로덕션 시스템의 경우 자체 서명 인증서를 사용하지 않고 대신에 인증 기관이 서명한 인증서에 의존하는 것이 바람직합니다.
- **label** 매개변수는 인터셉터가 필수 정보를 수신하기 위해 찾아볼 인증서의 이름을 지정합니다.
- **DN** 매개변수는 **식별 이름**(DN)의 세부사항을 지정하고 이는 각 사용자에 대해 고유해야 합니다.

4. 이제 키 데이터베이스를 작성했으므로 이에 대한 소유권을 설정하고, 다른 사용자는 이를 읽을 수 없는지 확인하십시오.

```
chown alice /home/alice/.mqs/alicekey.kdb /home/alice/.mqs/alicekey.sth
```

```
chmod 600 /home/alice/.mqs/alicekey.kdb /home/alice/.mqs/alicekey.sth
```

5. 사용자 bob에 대해 1-4단계를 반복하십시오.

## 결과

2명의 사용자 `alice` 및 bob은 이제 각각 자체 서명 인증서가 있습니다.

*4. keystore.conf* 작성

## 이 태스크 정보

Advanced Message Security 인터셉터로 키 데이터베이스 및 인증서가 있는 디렉토리를 가리켜야 합니다. 이는 해당 정보를 일반 텍스트 양식으로 보유하는 `keystore.conf` 파일을 통해 수행됩니다. 각 사용자에게는 `.mqs` 폴더에 별도의 `keystore.conf` 파일이 있어야 합니다. `alice` 및 bob 모두에 대해 이 단계를 완료해야 합니다.

`keystore.conf`의 컨텐츠를 다음 양식이어야 합니다.

```
cms.keystore = dir/keystore_file
```

```
cms.certificate = certificate_label
```

**예**

이 시나리오의 경우 `keystore.conf`의 컨텐츠는 다음과 같습니다.

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

**참고:**

- 키 저장소 파일의 경로는 파일 확장자 없이 제공되어야 합니다.
- CMS(Cryptographic Message Syntax), Java 키 저장소(JKS) 및 JCEKS(Java Cryptographic Extension Keystore)의 키 저장소 형식이 있습니다. 자세한 정보는 600 페이지의 『AMS 에 대한 키 저장소 구성 파일 (keystore.conf) 의 구조』의 내용을 참조하십시오.
- `HOME/.mqs/keystore.conf`는 Advanced Message Security가 `keystore.conf` 파일을 검색하는 기본 위치입니다. `keystore.conf`에 대해 기본이 아닌 위치를 사용하는 방법에 대한 정보는 599 페이지의 『AMS과(와) 함께 키스트로크와 인증서 사용』의 내용을 참조하십시오.

*5.* 인증서 공유

## 이 태스크 정보

각 사용자가 서로를 올바르게 식별할 수 있도록 두 키 데이터베이스 사이에서 인증서를 공유하십시오. 각 사용자의 공용 인증서를 파일에 내보내면 이 파일이 다른 사용자의 키 데이터베이스에 추가되는 방식으로 수행됩니다.

**참고:** 신중하게 추출 옵션을 사용하고 내보내기 옵션은 사용하지 마십시오. 추출은 사용자의 공개 키를 가져오지만 내보내기는 공개 키와 개인 키를 모두 가져옵니다. 실수로 추출을 사용하는 경우 개인 키가 누출되어 애플리케이션이 완전히 훼손될 수 있습니다.

## 프로시저

1. `alice`를 식별하는 인증서를 외부 파일로 추출하십시오.

   ```
   runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
   -target alice_public.arm
   ```

2. 인증서를 `bob's` 키 저장소에 추가하십시오.

   ```
   runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
   alice_public.arm
   ```

3. bob에 대해 단계를 반복하십시오.

   ```
   runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
   bob_public.arm
   ```

4. bob의 인증서를 `alice's` 키 저장소에 추가하십시오.

   ```
   runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
   bob_public.arm
   ```

## 결과

두 명의 사용자 `alice` 및 bob은 이제 자체 서명 인증서를 작성하고 공유한 서로를 성공적으로 식별할 수 있습니다.

## 다음에 수행할 작업

해당 세부사항을 인쇄하는 다음 명령을 실행하여 인증서가 키 저장소에 있는지 확인하십시오.

```
runmqakm -cert -details -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert
```

```
runmqakm -cert -details -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert
```

6. 큐 정책 정의

### 이 태스크 정보
큐 관리자가 작성되고 인터셉터가 메시지 및 액세스 암호화 키를 방해할 준비가 된 상태에서 `setmqspl` 명령을 사용하여 `QM_VERIFY_AMS`에서 보호 정책 정의를 시작할 수 있습니다. 이 명령에 대한 자세한 정보는 setmqspl의 내용을 참조하십시오. 각 정책 이름은 적용되는 큐 이름과 동일해야 합니다.

#### 예

이는 `TEST.Q` 큐에 정의된 정책의 예입니다. 이 예제에서 메시지는 `alice` 사용자가 Deprecated SHA1 알고리즘을 사용하여 서명하고 256비트 AES 알고리즘을 사용하여 암호화합니다. `alice`는 유일한 유효한 송신자이고 `bob`은 이 큐에서 메시지의 유일한 수신자입니다.

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

**참고:** DN은 키 데이터베이스의 각 사용자 인증서에 지정된 DN과 정확히 일치합니다.

### 다음에 수행할 작업
사용자가 정의한 정책을 확인하려면 다음 명령을 실행하십시오.

```
dspmqspl -m QM_VERIFY_AMS
```

`setmqspl` 명령의 세트로 정책 세부사항을 인쇄하려면 `-export` 플래그를 사용하십시오. 이는 이미 정의된 정책을 저장할 수 있게 해줍니다.

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 설정 테스트

### 이 태스크 정보
다른 사용자 하에서 다른 프로그램을 실행하여 애플리케이션이 제대로 구성되었는지 확인할 수 있습니다.

### 프로시저
1. 샘플을 포함하는 디렉토리로 변경하십시오. MQ가 비기본 위치에 설치된 경우에는 이는 다른 위치에 있을 수도 있습니다.

   ```
   cd /opt/mqm/samp/bin
   ```

2. 사용자 `alice`로서 실행하도록 사용자를 전환하십시오.

   ```
   su alice
   ```

3. `alice` 사용자로서, 샘플 애플리케이션을 사용하여 메시지를 넣으십시오.

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. 메시지의 텍스트를 입력한 다음 Enter를 누르십시오.
5. 사용자 alice로서의 실행을 중지하십시오.

```
exit
```

6. 사용자 bob로서 실행하도록 사용자를 전환하십시오.

```
su bob
```

7. bob 사용자로서 샘플 애플리케이션을 사용하여 메시지를 가져오십시오.

```
./amqsget TEST.Q QM_VERIFY_AMS
```

### 결과
애플리케이션이 두 사용자에게 모두 적절하게 구성된 경우 사용자 alice의 메시지가 bob이 가져오기 애플리케이션을 실행할 때 표시됩니다.

*8.* 암호화 테스트

### 이 태스크 정보
암호화가 예상대로 발생했는지 확인하려면 원본 큐 TEST.Q를 참조하는 알리어스 큐를 작성하십시오. 이 알리어스 큐에는 보안 정책이 없고 메시지를 복호화하기 위한 정보를 가진 사용자가 없으므로, 암호화된 데이터가 표시될 것입니다.

### 프로시저
1. 큐 관리자 QM_VERIFY_AMS에 대해 **runmqsc** 명령을 사용하여 알리어스 큐를 작성하십시오.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. 알리어스 큐에서 찾아보기 위해 bob에 액세스를 부여하십시오.

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. alice 사용자로서 방금 전처럼 샘플 애플리케이션을 사용하여 또 다른 메시지를 넣으십시오.

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. bob 사용자로서 이번에는 알리어스 큐를 통해 샘플 애플리케이션을 사용하여 메시지를 찾으십시오.

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. bob 사용자로서 로컬 큐로부터 샘플 애플리케이션을 사용하여 메시지를 가져오십시오.

```
./amqsget TEST.Q QM_VERIFY_AMS
```

### 결과
amqsbcg 애플리케이션의 출력에는 메시지가 암호화되었음을 증명하는 큐에 있는 암호화된 데이터가 표시됩니다.

## z/OS *Example AMS configurations on z/OS*
This section provides example configurations of policies and certificates for Advanced Message Security queuing scenarios on z/OS.

See Configuring Advanced Message Security for z/OS for details on how you configure Advanced Message Security.

The examples cover the Advanced Message Security policies required, and the digital certificates that must exist relative to users and key rings. The examples assume that the users involved in the scenarios have been set up by following the instructions provided in Grant users resource permissions for Advanced Message Security.

See also server-to-server message channel interception examples.

z/OS *Local queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from a queue, local to the putting and getting applications.

The example queue manager and queue are:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

These users are used:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

## Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected messages. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK6.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue a user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security requires:

- The CA certificate (chain).
- The user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6.

When the certificates have been imported on the z/OS system running BNK6, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

In this example, no certificate is required for the recipient user.

## Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6. To create the key rings use the RACDCERT ADDRING commands:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user, WMQBNK6, and a key ring for the sending user, 'TELLER5'. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

When the key rings have been created, the relevant certificates can be connected:

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

```
F BNK6AMSM,REFRESH KEYRING
```

## Create the Advanced Message Security policy

In this example, integrity-protected messages are put to queue FIN.XFER.Q7 by an application running as user 'TELLER5', and retrieved from the same queue by an application running as user 'FINADM2', so only one Advanced Message Security policy is required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

After defining the policy, either restart the BNK6 queue manager, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configuration. For example:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS 의 AMS 에 대한 개인정보 보호된 메시지의 로컬 큐잉*

이 예는 개인정보 보호된 메시지를 넣기 및 가져오기 애플리케이션의 로컬에서 큐에 전송하고 받는 데 필요한 Advanced Message Security 정책 및 인증서를 설명합니다. 개인정보 보호된 메시지는 둘 모두 서명되고 암호화됩니다.

예 큐 관리자 및 로컬 큐는 다음과 같습니다.

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

다음 사용자가 사용됩니다.

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

이 시나리오를 구성하기 위한 단계는 다음과 같습니다.

## 사용자 인증서 작성

이 예에서 두 개의 사용자 인증서가 필요합니다. 이들은 메시지에 서명하기 위해 필요한 송신 사용자의 인증서 및 메시지 데이터를 암호화하고 복호화하기 위해 필요한 수신자의 사용자 인증서입니다. 송신 사용자는 'TELLER5'이고 수신 사용자는 'FINADM2'입니다.

인증 기관(CA) 인증서 또한 필요합니다.CA 인증서는 사용자의 인증서를 발행한 권한의 인증서입니다. 이는 인증서의 체인일 수 있습니다. 그런 경우 체인의 모든 인증서는 Advanced Message Security 태스크 사용자의 키 링에서 필수입니다. 이 경우 사용자 WMQBNK6입니다.

CA 인증서는 RACF RACDCERT 명령을 사용하여 작성할 수 있습니다. 이 인증서는 사용자 인증서를 발행하는 데 사용됩니다. 예를 들면, 다음과 같습니다.

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

이 RACDCERT 명령은 사용자 'TELLER5' 및 'FINADM2'를 위한 사용자 인증서를 발행하는 데 사용할 수 있는 CA 인증서를 작성합니다. 예를 들면, 다음과 같습니다.

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

```
RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

설치에는 CA 인증서를 선택하고 작성하기 위한 프로시저뿐만 아니라 인증서를 발행하고 이를 관련 시스템에 분배하기 위한 프로시저가 있습니다.

이러한 인증서를 내보내고 가져올 때 Advanced Message Security에는 다음이 필요합니다.

- CA 인증서(체인).
- 송신 사용자 인증서 및 해당 개인 키.
- 수신 사용자 인증서 및 해당 개인 키.

RACF를 사용 중인 경우, RACDCERT EXPORT 명령은 인증서를 데이터 세트에 내보내는 데 사용할 수 있고, RACDCERT ADD 명령은 인증서를 데이터 세트로부터 가져오는 데 사용할 수 있습니다. 이러한 RACDCERT 명령 및 기타 RACDCERT 명령에 대한 자세한 정보는 *z/OS: Security Server RACF Command Language Reference*에서 RACDCERT ( RACF 디지털 인증서 관리) 를 참조하십시오.

이 경우 인증서는 큐 관리자 BNK6을 실행 중인 z/OS 시스템에서 필요합니다.

인증서가 BNK6을 실행 중인 z/OS 시스템에 가져오기되면 사용자 인증서에는 TRUST 속성이 필요합니다. RACDCERT ALTER 명령은 TRUST 속성을 인증서에 추가하는 데 사용할 수 있습니다. 예를 들면, 다음과 같습니다.

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

## 인증서를 관련 키 링에 연결

필수 인증서가 작성되거나 가져오기되고, 신뢰되는 것으로 설정될 때 이들은 BNK6을 실행 중인 z/OS 시스템에서 적합한 사용자 키 링에 연결되어야 합니다. 키 링을 작성하려면 RACDCERT ADDRING 명령을 사용하십시오.

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

이는 송산 및 수신 사용자를 위해 Advanced Message Security 태스크 사용자를 위한 키 링을 작성합니다. 키 링 이름 `drq.ams.keyring` 은 필수이며 이름은 대소문자를 구분합니다.

키 링이 작성될 때 관련 인증서를 연결할 수 있습니다.

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

```
RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

송신 및 수신 사용자 인증서는 DEFAULT로서 연결되어야 합니다. 사용자에 해당 drq.ams.keyring에 둘 이상의 인증서가 있는 경우에는 서명 및 복호화 용도를 위해 기본 인증서가 사용됩니다.

수신 사용자의 인증서는 또한 USAGE(SITE)와 함께 Advanced Message Security 태스크 사용자의 키 링에 연결되어야 합니다. 이는 고급 메시지 보안 태스크에 메시지 데이터를 암호화할 때 수신인의 공개 키가 필요하기 때문입니다. USAGE(SITE)는 개인 키가 키 링크에서 액세스 가능하지 못하게 만듭니다.

인증서의 작성 및 수정은 큐 관리자가 중지되고 재시작되거나 z/OS **MODIFY** 명령이 Advanced Message Security 인증서 구성을 새로 고치는 데 사용될 때까지 Advanced Message Security 에서 인식되지 않습니다. 예를 들면, 다음과 같습니다.

```
F BNK6AMSM,REFRESH KEYRING
```

## Advanced Message Security 정책 작성

이 예에서 개인정보 보호된 메시지는 사용자 'TELLER5'로서 실행 중인 애플리케이션에 의해 큐 FIN.XFER.Q8에 놓여지고, 사용자 'FINADM2'로서 실행 중인 애플리케이션에 의해 동일 큐에서 검색되므로, 단 하나의 Advanced Message Security 정책만이 필요합니다.

Advanced Message Security 정책은 메시지 보안 정책 유틸리티(CSQ0UTIL)에 문서화된 CSQ0UTIL 유틸리티를 사용하여 작성됩니다.

다음 명령을 실행하여 CSQ0UTIL 유틸리티를 사용하십시오.

```
setmqspl -m BNK6 -p FIN.XFER.Q8 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

이 정책에서 큐 관리자는 BNK6로 식별됩니다. 정책 이름 및 연관된 큐는 FIN.XFER.Q8입니다. 송신자의 서명을 생성하는 데 사용되는 알고리즘은 Deprecated SHA1이고, 송신 사용자의 식별 이름 (DN) 은 'CN=Teller5,O=BCO,C=US' 이며, 수신 사용자는 'CN=FinAdm2,O=BCO,C=US' 입니다. 메시지 데이터를 암호화하는 데 사용되는 알고리즘은 Deprecated 3DES입니다.

정책을 정의한 후에 BNK6 큐 관리자를 재시작하거나 z/OS **MODIFY** 명령을 사용하여 Advanced Message Security 정책 구성을 새로 고치십시오. 예를 들면, 다음과 같습니다.

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *Remote queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from queues managed by two different queue managers. The two queue managers can be running on the same z/OS system, or on different z/OS systems, or one queue manager can be on a distributed system running Advanced Message Security.

The example queue managers and queues are:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Note: For this example, BNK6 and BNK7 are queue managers running on different z/OS systems.

These users are used:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

The steps to configure this scenario are as follows:

## Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected message. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK7.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security require:

- The CA certificate (chain).
- The sending user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to RACDCERT (Manage RACF digital certificates) in the *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6 and BNK7.

In this example, the sending certificate must be imported on the z/OS system running BNK6, and the CA certificate must be imported on the z/OS system running BNK7. When the certificates have been imported, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example, on BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

## Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6 and BNK7.

To create the key rings use the RACDCERT ADDRING command, on BNK6:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the sending user on BNK6. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

On BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user on BNK7. No user key ring is required for 'TELLER5' on BNK7.

When the key rings have been created, the relevant certificates can be connected.

On BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

On BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

On BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

## Create the Advanced Message Security policies

In this example, integrity-protected messages are put to remote queue FIN.XFER.Q7 on BNK6 by an application running as user 'TELLER5', and retrieved from local queue FIN.RCPT.Q7 on BNK7 by an application running as user 'FINADM2', so two Advanced Message Security policies are required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command to define an integrity policy for the remote queue on BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

Also, use the CSQ0UTIL utility to run the following command to define an integrity policy for the local queue on BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK7. The policy name and associated queue is FIN.RCPT.Q7. The algorithm expected for the sender's signature is MD5, and the distinguished name (DN) of the sending user is expected to be 'CN=Teller5,O=BCO,C=US'.

After defining the two policies, either restart the BNK6 and BNK7 queue managers, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configurations. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

On BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *z/OS 의 AMS 에 대한 개인정보 보호된 메시지의 리모트 큐잉*

이 예는 두 개의 서로 다른 큐 관리자가 관리하는 큐에 개인정보 보호된 메시지를 송신하고 수신하는 데 필요한 Advanced Message Security 정책 및 인증서를 설명합니다. 두 개의 큐 관리자는 동일한 z/OS 시스템 또는 다른 z/OS 시스템에서 실행 중이거나 하나의 큐 관리자가 Advanced Message Security를 실행하는 분산 시스템에 있을 수 있습니다.

예 큐 관리자 및 큐는 다음과 같습니다.

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

참고: 이 예의 경우 BNK6 및 BNK7은 이름이 같은 z/OS 시스템에서 실행 중인 큐 관리자입니다.

다음 사용자가 사용됩니다.

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
```

```
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

이 시나리오를 구성하기 위한 단계는 다음과 같습니다.

## 사용자 인증서 작성

이 예에서 두 개의 사용자 인증서가 필요합니다. 이들은 메시지에 서명하기 위해 필요한 송신 사용자의 인증서 및 메시지 데이터를 암호화하고 복호화하기 위해 필요한 수신자의 사용자 인증서입니다. 송신 사용자는 'TELLER5'이고 수신 사용자는 'FINADM2'입니다.

인증 기관(CA) 인증서 또한 필요합니다.CA 인증서는 사용자의 인증서를 발행한 권한의 인증서입니다. 이는 인증서의 체인일 수 있습니다. 그런 경우 체인의 모든 인증서는 Advanced Message Security 태스크 사용자의 키 링에서 필수입니다. 이 경우 사용자 WMQBNK7입니다.

CA 인증서는 RACF RACDCERT 명령을 사용하여 작성할 수 있습니다. 이 인증서는 사용자 인증서를 발행하는 데 사용됩니다. 예를 들면, 다음과 같습니다.

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

이 RACDCERT 명령은 사용자 'TELLER5' 및 'FINADM2'를 위한 사용자 인증서를 발행하는 데 사용할 수 있는 CA 인증서를 작성합니다. 예를 들면, 다음과 같습니다.

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

```
RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

설치에는 CA 인증서를 선택하고 작성하기 위한 프로시저뿐만 아니라 인증서를 발행하고 이를 관련 시스템에 분배하기 위한 프로시저가 있습니다.

이러한 인증서를 내보내고 가져올 때 Advanced Message Security에는 다음이 필요합니다.

- CA 인증서(체인).
- 송신 사용자 인증서 및 해당 개인 키.
- 수신 사용자 인증서 및 해당 개인 키.

RACF를 사용 중인 경우, RACDCERT EXPORT 명령은 인증서를 데이터 세트에 내보내는 데 사용할 수 있고, RACDCERT ADD 명령은 인증서를 데이터 세트로부터 가져오는 데 사용할 수 있습니다.

이러한 RACDCERT 명령 및 기타 RACDCERT 명령에 대한 자세한 정보는 *z/OS: Security Server RACF Command Language Reference*에서 RACDCERT ( RACF 디지털 인증서 관리) 를 참조하십시오.

이 경우 인증서는 큐 관리자 BNK6 및 BNK7을 실행 중인 z/OS 시스템에서 필요합니다.

이 예에서 송신 및 수신 인증서는 BNK6을 실행 중인 z/OS 시스템에 가져오기되어야 하고 CA 및 수신 인증서는 BNK7을 실행 중인 z/OS 시스템에 가져오기되어야 합니다. 인증서가 가져오기되면 사용자 인증서에는 TRUST 속성이 필요합니다. RACDCERT ALTER 명령은 TRUST 속성을 인증서에 추가하는 데 사용할 수 있습니다. 예를 들면, 다음과 같습니다.

BNK6에서:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

BNK7에서:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

## 인증서를 관련 키 링에 연결

필수 인증서가 작성되거나 가져오기되고, 신뢰되는 것으로 설정될 때 이들은 BNK6 및 BNK7을 실행 중인 z/OS 시스템에서 적합한 사용자 키 링에 연결되어야 합니다.

키 링을 작성하려면 RACDCERT ADDRING 명령을 사용하십시오.

BNK6에서:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

이는 Advanced Message Security 태스크 사용자를 위한 키 링을 작성하고 BNK6에서 송신 사용자를 위한 키 링을 작성합니다. 키 링 이름 drq.ams.keyring은 필수이고 이름은 대소문자를 구분함을 유의하십시오.

BNK7에서:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

이는 Advanced Message Security 태스크 사용자를 위한 키 링을 작성하고 BNK7에서 수신 사용자를 위한 키 링을 작성합니다.

키 링이 작성될 때 관련 인증서를 연결할 수 있습니다.

BNK6에서:

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

BNK7에서:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

송신 및 수신 사용자 인증서는 DEFAULT로서 연결되어야 합니다. 사용자에 해당 drq.ams.keyring에 둘 이상의 인증서가 있는 경우에는 서명 및 암호화/복호화 용도를 위해 기본 인증서가 사용됩니다.

BNK6에서 수신 사용자의 인증서는 또한 USAGE(SITE)와 함께 Advanced Message Security 태스크 사용자의 키 링에 연결되어야 합니다. 이는 고급 메시지 보안 태스크에 메시지 데이터를 암호화할 때 수신인의 공개 키가 필요하기 때문입니다. USAGE(SITE)는 개인 키가 키 링크에서 액세스 가능하지 못하게 만듭니다.

인증서의 작성 및 수정은 큐 관리자가 중지되고 재시작되거나 z/OS **MODIFY** 명령이 Advanced Message Security 인증서 구성을 새로 고치는 데 사용될 때까지 Advanced Message Security 에서 인식되지 않습니다. 예를 들면, 다음과 같습니다.

BNK6에서:

```
F BNK6AMSM,REFRESH,KEYRING
```

BNK7에서:

```
F BNK7AMSM,REFRESH,KEYRING
```

### Advanced Message Security 정책 작성

이 예에서 개인정보 보호된 메시지는 사용자 'TELLER5'로서 실행 중인 애플리케이션에 의해 BNK6에서 리모트 큐 FIN.XFER.Q7에 놓여지고, 사용자 'FINADM2'로서 실행 중인 애플리케이션에 의해 BNK7에서 로컬 큐 FIN.RCPT.Q7에서 검색되므로, 두 개의 Advanced Message Security 정책이 필요합니다.

Advanced Message Security 정책은 메시지 보안 정책 유틸리티(CSQ0UTIL)에 문서화된 CSQ0UTIL 유틸리티를 사용하여 작성됩니다.

BNK6에서 리모트 큐에 대해 개인정보 정책을 정의하려면 CSQ0UTIL 유틸리티를 사용하여 다음 명령을 실행하십시오.

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

이 정책에서 큐 관리자는 BNK6로 식별됩니다. 정책 이름 및 연관된 큐는 FIN.XFER.Q7입니다. 송신자의 서명을 생성하는 데 사용되는 알고리즘은 Deprecated SHA1이고, 송신 사용자의 식별 이름 (DN) 은 'CN=Teller5,O=BCO,C=US' 이며, 수신 사용자는 'CN=FinAdm2,O=BCO,C=US' 입니다. 메시지 데이터를 암호화하는 데 사용되는 알고리즘은 Deprecated 3DES입니다.

또한 BNK7에서 로컬 큐에 대해 개인정보 정책을 정의하려면 CSQ0UTIL 유틸리티를 사용하여 다음 명령을 실행하십시오.

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

이 정책에서 큐 관리자는 BNK7로 식별됩니다. 정책 이름 및 연관된 큐는 FIN.RCPT.Q7입니다. 송신자의 서명에 대해 예상되는 알고리즘은 Deprecated SHA1이고, 송신 사용자의 식별 이름 (DN) 은 'CN=Teller5,O=BCO,C=US' 로 예상되며, 수신 사용자는 'CN=FinAdm2,O=BCO,C=US' 입니다. 메시지 데이터를 복호화하는 데 사용되는 알고리즘은 Deprecated 3DES입니다.

두 개의 정책을 정의한 후에 BNK6 및 BNK7 큐 관리자를 재시작하거나 z/OS **MODIFY** 명령을 사용하여 Advanced Message Security 정책 구성을 새로 고치십시오. 예를 들면, 다음과 같습니다.

BNK6에서:

```
F BNK6AMSM,REFRESH,POLICY
```

BNK7에서:

```
F BNK7AMSM,REFRESH,POLICY
```

## *Java* 클라이언트가 있는 *AMS* 용 빠른 시작 안내서

클라이언트 바인딩을 사용하여 연결하는 Java 애플리케이션에 대한 메시지 보안을 제공하도록 Advanced Message Security 를 빠르게 구성하려면 이 안내서를 사용하십시오. 이를 완료할 때쯤 사용자 ID를 확인하기 위한 키 저장소를 작성하게 될 것이고 큐 관리자의 서명/암호화 정책을 정의했을 것입니다.

### 시작하기 전에

566 페이지의 『Windows 플랫폼의 AMS 용 빠른 시작 안내서』 또는 571 페이지의 『AIX and Linux 의 AMS 용 빠른 시작 안내서』 에 설명된 대로 적절한 구성요소가 설치되어 있는지 확인하십시오.

*1.* 큐 관리자 및 큐 작성

### 이 태스크 정보

다음 모든 예에서는 애플리케이션 간 메시지를 전달하기 위해 이름이 TEST.Q인 큐를 사용합니다. Advanced Message Security는 표준 IBM MQ 인터페이스를 통해 IBM MQ 인프라에 입력하는 지점에서 메시지에 서명하고 암호화하기 위해 인터셉터를 사용합니다. 이 기본 설정은 IBM MQ에서 수행되고 다음 단계에서 구성됩니다.

## 프로시저

1. 큐 관리자 작성

```
crtmqm QM_VERIFY_AMS
```

2. 큐 관리자 시작

```
strmqm QM_VERIFY_AMS
```

3. 다음 명령을 QM_VERIFY_AMS 큐 관리자의 **runmqsc**에 입력하여 리스너를 작성하고 시작하십시오.

```
DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

```
START LISTENER(AMS.LSTR)
```

4. 다음 명령을 QM_VERIFY_AMS 큐 관리자의 **runmqsc**에 입력하여 애플리케이션이 연결할 채널을 작성하십시오.

```
DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
```

5. 다음 명령을 큐 관리자 QM_VERIFY_AMS의 **runmqsc**에 입력하여 이름이 TEST.Q인 큐를 작성하십시오.

```
DEFINE QLOCAL(TEST.Q)
```

## 결과

프로시저가 성공적으로 완료되면 **runmqsc**에 입력된 다음 명령이 TEST.Q에 대한 세부사항을 표시합니다.

```
DISPLAY Q(TEST.Q)
```

*2.* 사용자 작성 및 권한 부여

## 이 태스크 정보

이 시나리오에는 2명의 사용자가 나타납니다. 송신자 alice와 수신자 bob입니다. 애플리케이션 큐를 사용하려면 이러한 사용자에게는 이를 사용할 권한이 부여되어야 합니다. 또한 이 시나리오에 정의된 보호 정책을 성공적으로 사용하기 위해 이러한 사용자에게는 일부 시스템 큐에 대한 액세스가 부여되어야 합니다. **setmqaut** 명령에 대한 자세한 정보는 **setmqaut**를 참조하십시오.

## 프로시저

1. 사용자 플랫폼의 **빠른 시작 안내서**(Windows 또는 AIX and Linux)에 설명된 대로 두 사용자를 작성하십시오.
2. 사용자에게 큐 관리자에 연결하고 큐에 대해 작업할 수 있는 권한을 부여

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
```

```
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
```

```
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get +inq +browse
```

3. 또한 두 사용자가 시스템 정책 큐를 찾아서 메시지를 오류 큐에 넣도록 허용해야 합니다.

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse
```

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put
```

**주의:** IBM MQ 는 SYSTEM.PROTECTION.POLICY.QUEUE 를 QUEUE합니다.

IBM MQ가 사용 가능한 모든 정책을 캐시하지는 않습니다. 정책 수가 많으면 IBM MQ가 제한된 수의 정책을 캐시합니다. 따라서 큐 관리자에 적은 수의 정책이 정의되어 있는 경우 SYSTEM.PROTECTION.POLICY.QUEUE에 찾아보기 옵션을 제공할 필요가 없습니다.

그러나 많은 수의 정책이 정의되어 있거나 이전 클라이언트를 사용 중인 경우 이 큐에 대한 찾아보기 권한을 부여해야 합니다. SYSTEM.PROTECTION.ERROR.QUEUE는 AMS 코드에서 생성된 오류 메시지를 넣는 데 사용됩니다. 이 큐에 대한 넣기(Put) 권한은 오류 메시지를 큐에 넣으려고 할 때만 확인됩니다. AMS 보호 큐에서 메시지를 가져오거나 넣으려고 할 때는 큐에 대한 넣기(Put) 권한이 확인되지 않습니다.

## 결과

그러면 사용자가 작성되고 사용자에게 필요한 권한이 부여됩니다.

## 다음에 수행할 작업

단계가 올바르게 수행되었는지 확인하려면 592 페이지의 『7. 설정 테스트』 섹션에 설명된 대로 `JmsProducer` 및 `JmsConsumer` 샘플을 사용하십시오.

*3.* 키 데이터베이스 및 인증서 작성

## 이 태스크 정보

메시지를 인터셉터로 암호화하려면 송신 사용자의 공개 키가 필요합니다. 따라서 공개 및 개인 키에 맵핑된 사용자의 키 데이터베이스가 작성되어야 합니다. 사용자 및 애플리케이션이 여러 컴퓨터에 분산되어 있는 실제 시스템에서는 각 사용자는 고유의 개인용 키 저장소를 가지고 있을 것입니다. 마찬가지로, 이 안내서에서는 `alice`와 bob을 위한 키 데이터베이스를 작성하고 이들 간에 사용자 인증서를 공유합니다.

**참고:** 이 안내서에서는 클라이언트 바인딩을 사용하여 연결하는 Java로 작성된 샘플 애플리케이션을 사용합니다. 로컬 바인딩을 사용하는 Java 애플리케이션 또는 C 애플리케이션을 사용할 계획이면 **runmqakm** 명령을 사용하여 CMS 키 저장소 및 인증서를 작성해야 합니다. 자세한 정보는 566 페이지의 『Windows 플랫폼의 AMS 용 빠른 시작 안내서』 및 571 페이지의 『AIX and Linux 의 AMS 용 빠른 시작 안내서』의 내용을 참조하십시오.

## 프로시저

1. 키 저장소를 작성할 디렉토리를 작성하십시오(예: `/home/alice/.mqs`). 플랫폼의 빠른 시작 안내서에서 사용한 것과 동일한 디렉토리에 작성할 수 있습니다. 자세한 정보는 566 페이지의 『Windows 플랫폼의 AMS 용 빠른 시작 안내서』 및 571 페이지의 『AIX and Linux 의 AMS 용 빠른 시작 안내서』의 내용을 참조하십시오.

   **참고:** 이 디렉토리는 다음 단계에서 *keystore-dir*로 불립니다.
2. 암호화에 사용하기 위해 `alice` 사용자를 식별하는 새 키 저장소 및 인증서를 작성하십시오.

   **참고:** **keytool** 명령은 JRE의 일부입니다.

```
keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks
-storepass passw0rd
-dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
```

   **참고:**

   - *keystore-dir*에 공백이 포함된 경우에는 키 저장소의 전체 이름 주변에 따옴표를 넣어야 합니다.

- 키 저장소를 보안 설정하기 위해 강력한 비밀번호를 사용하는 것이 권장됩니다.
- 이 안내서의 용도상 인증 기관을 사용하지 않고 작성할 수 있는 자체 서명 인증서를 사용 중입니다. 프로덕션 시스템의 경우 자체 서명 인증서를 사용하지 않고 대신에 인증 기관이 서명한 인증서에 의존하는 것이 바람직합니다.
- **alias** 매개변수는 인터셉터가 필수 정보를 수신하기 위해 찾아볼 인증서의 이름을 지정합니다.
- **dname** 매개변수는 **식별 이름**(DN)의 세부사항을 지정하고 이는 각 사용자에 대해 고유해야 합니다.

3. AIX and Linux에서 키 저장소가 판독 가능한지 확인하십시오.

```
chmod +r keystore-dir/keystore.jks
```

4. 사용자 bob에 대해 1-4단계를 반복하십시오.

## 결과

2명의 사용자 alice 및 bob은 이제 각각 자체 서명 인증서가 있습니다.

*4. keystore.conf* 작성

## 이 태스크 정보

Advanced Message Security 인터셉터로 키 데이터베이스 및 인증서가 있는 디렉토리를 가리켜야 합니다. 이는 해당 정보를 일반 텍스트 양식으로 보유하는 keystore.conf 파일을 통해 수행됩니다. 각 사용자에게는 별도의 keystore.conf 파일이 있어야 합니다. 이 단계는 alice 및 bob 둘 모두에 대해 수행되어야 합니다.

### 예

이 시나리오의 경우 alice의 keystore.conf의 컨텐츠는 다음과 같습니다.

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

이 시나리오의 경우 bob의 keystore.conf의 컨텐츠는 다음과 같습니다.

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

**참고:**

- 키 저장소 파일의 경로는 파일 확장자 없이 제공되어야 합니다.
- 빠른 시작 안내서(Windows 또는 AIX and Linux)의 지침을 따르지 않아 이미 keystore.conf 파일이 있으면 기존 파일을 편집하여 해당 행을 추가할 수 있습니다.
- 자세한 정보는 600 페이지의 『AMS 에 대한 키 저장소 구성 파일 (keystore.conf) 의 구조』 의 내용을 참조하십시오.

*5.* 인증서 공유

## 이 태스크 정보

각 사용자가 서로를 성공적으로 식별할 수 있도록 두 개의 키 저장소 간에 인증서를 공유합니다. 이는 각 사용자의 인증서를 추출한 후 다른 사용자의 키 저장소로 가져오는 방식으로 수행됩니다.

**중요사항:** 추출 및 내보내기 용어는 다른 인증서 관리 명령에 의해 다르게 사용됩니다.

- IBM Global Security Kit (GSKit) **runmqakm** 명령은 추출 이라는 용어를 사용하여 키 저장소에서 인증서의 공용 부분만 복사하는 프로세스를 나타내고, 내보내기 라는 용어를 사용하여 인증서 및 연관된 공용 및 개인용 키를 한 키 저장소에서 다른 키 저장소로 복사하는 프로세스를 나타냅니다.
- Java **keytool** 명령, V 9.4.0 V 9.4.0 및 IBM MQ **runmqktool** 명령은 내보내기 라는 용어를 사용하여 키 저장소에서 인증서의 공용 부분만 복사하는 프로세스를 의미합니다.

내보내기 를 잘못 사용하면 개인 키를 노출하여 애플리케이션이 손상될 수 있으므로 이러한 차이는 중요합니다. 구분이 매우 중요하기 때문에 IBM MQ 문서에서는 이러한 용어를 일관되게 사용합니다. 이러한 이유로 다음 프로시저는 **keytool** 명령에서 `exportcert` 옵션을 사용하여 인증서 추출 을 참조합니다.

## 프로시저

1. `alice`를 식별하는 인증서를 추출하십시오.

```
keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
-alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
```

2. `alice`를 식별하는 인증서를 `bob`이 사용할 키 저장소로 가져오십시오. 프롬프트되면 이 인증서를 신뢰할 것임을 나타내십시오.

```
keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
-keystore bob-keystore-dir/keystore.jks -storepass passw0rd
```

3. `bob`에 대해 단계를 반복하십시오.

## 결과

두 명의 사용자 `alice` 및 `bob`은 이제 자체 서명 인증서를 작성하고 공유한 서로를 성공적으로 식별할 수 있습니다.

## 다음에 수행할 작업

해당 세부사항을 인쇄하는 다음 명령을 실행하여 인증서가 키 저장소에 있는지 확인하십시오.

```
keytool -list -keystore bob-keystore-dir/keystore.jks -storepass passw0rd -alias Alice_Java_Cert
```

```
keytool -list -keystore alice-keystore-dir/keystore.jks -storepass passw0rd -alias Bob_Java_Cert
```

*6.* 큐 정책 정의

## 이 태스크 정보

큐 관리자가 작성되고 인터셉터가 메시지 및 액세스 암호화 키를 방해할 준비가 된 상태에서 `setmqspl` 명령을 사용하여 `QM_VERIFY_AMS`에서 보호 정책 정의를 시작할 수 있습니다. 이 명령에 대한 자세한 정보는 setmqspl 의 내용을 참조하십시오. 각 정책 이름은 적용되는 큐 이름과 동일해야 합니다.

### 예

이것은 사용자 `TEST.Q` 큐에 정의하고, `alice`가 Deprecated SHA1 알고리즘을 사용하여 서명하고, 사용자 `bob`에 대해 256비트 AES 알고리즘을 사용하여 암호화한 정책의 예입니다.

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

**참고:** DN은 키 데이터베이스의 각 사용자 인증서에 지정된 DN과 정확히 일치합니다.

### 다음에 수행할 작업
사용자가 정의한 정책을 확인하려면 다음 명령을 실행하십시오.

```
dspmqspl -m QM_VERIFY_AMS
```

`setmqspl` 명령의 세트로서 정책 세부사항을 인쇄하려면 `-export` 플래그입니다. 이는 이미 정의된 정책을 저장할 수 있게 해줍니다.

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. 설정 테스트

### 시작하기 전에
사용 중인 Java 버전에 제한되지 않은 JCE 정책 파일이 설치되어 있는지 확인하십시오.

**참고:** IBM MQ 설치에 제공된 Java 버전에는 이미 이러한 정책 파일이 있습니다. 이는 *MQ_INSTALLATION_PATH*/java/bin에서 찾을 수 있습니다.

### 이 태스크 정보
다른 사용자 하에서 다른 프로그램을 실행하여 애플리케이션이 제대로 구성되었는지 확인할 수 있습니다. 다른 사용자에서 프로그램을 실행하는 방법에 대한 자세한 정보는 566 페이지의 『Windows 플랫폼의 AMS 용 빠른 시작 안내서』 및 571 페이지의 『AIX and Linux 의 AMS 용 빠른 시작 안내서』의 내용을 참조하십시오.

### 프로시저
1. 이러한 JMS 샘플 애플리케이션을 실행하려면 IBM MQ classes for JMS에서 사용하는 환경 변수에 표시된 대로 사용자 플랫폼에 대한 CLASSPATH 설정을 사용하여 샘플 디렉토리가 포함되도록 하십시오.
2. `alice` 사용자로서 샘플 애플리케이션을 사용하여 클라이언트로서 연결하여 메시지를 넣으십시오.

   ```
   java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

3. `bob` 사용자로서 샘플 애플리케이션을 사용하여 클라이언트로서 연결하여 메시지를 가져오십시오.

   ```
   java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

### 결과
애플리케이션이 두 사용자에게 모두 적절하게 구성된 경우 사용자 `alice`의 메시지가 `bob`이 가져오기 애플리케이션을 실행할 때 표시됩니다.

### *AMS*에서 원격 큐 보호
리모트 큐를 완전히 보호하려면 메시지를 전송하는 리모트 큐 및 로컬 큐에서 정책을 설정해야 합니다.

메시지가 리모트 큐에 놓을 때, Advanced Message Security는 조작을 인터셉트하고 리모트 큐에 설정된 정책에 따라 메시지를 처리합니다. 예를 들어, 암호화 정책의 경우 메시지는 이를 처리하기 위해 IBM MQ에 전달되기 전에 암호화됩니다. Advanced Message Security가 리모트 큐에 놓인 메시지를 처리한 후에 IBM MQ는 이를 연관된 전송 큐에 놓고 이를 대상 큐 관리자 및 대상 큐에 전달합니다.

GET 조작이 로컬 큐에서 수행되면 Advanced Message Security는 로컬 큐에 설정된 저액에 따라 메시지를 디코드하려고 시도합니다. 조작이 성공하려면 메시지를 복호화하는 데 사용된 정책은 이를 암호화하는 데 사용된 것과 동일해야 합니다. 불일치가 있으면 메시지가 거부됩니다.

어떤 이유로든 두 정책이 동시에 설정될 수는 없는 경우에는 스테이지형 롤아웃 지원이 제공됩니다. 정책은 관용 플래그가 있는 로컬 큐에 설정할 수 있고, 이는 큐와 연관된 정책은 큐로부터 메시지를 검색하려는 시도가 보안 정책 세트가 없는 메시지와 관련될 때 무시될 수 있음을 나타냅니다. 이 경우 GET은 메시지를 복호화하려고 시도하지만 암호화되지 않은 메시지가 전달될 수 있도록 허용할 것입니다. 이 방법으로 리모트 큐에 대한 정책은 로컬 큐를 보호하고 테스트한 후에 설정할 수 있습니다

**알아두기:** Advanced Message Security 롤아웃이 완료된 후에 관용 플래그를 제거하십시오.

**관련 참조**
setmqspl(보안 정책 설정)

## *IBM Integration Bus* 를 사용하여 *AMS* 로 보호된 메시지 라우팅
Advanced Message Security는 IBM Integration Bus 또는 WebSphere Message Broker 8.0.0.1 이상이 설치된 인프라에서 메시지를 보호할 수 있습니다. IBM Integration Bus 환경에서 보안을 적용하기 전에 두 제품 모두의 특성을 이해해야 합니다.

### 이 태스크 정보
Advanced Message Security는 메시지 페이로드(payload)의 엔드-투-엔드 보안을 제공합니다. 이는 메시지의 유효한 송신자와 수신자로 지정된 당사자만이 이를 생성하거나 수신할 수 있음을 의미합니다. 이는 IBM Integration Bus를 통해 메시지 플로우를 보안 설정하기 위해서 IBM Integration Bus가 메시지의 컨텐츠를 모른 채로 처리하거나 의미합니다(시나리오 1) 권한 있는 사용자가 메시지를 받거나 전송할 수 있도록 만들 수 있음(시나리오 2)을 의미합니다.

시나리오 *1* - 통합 버스가 메시지 컨텐츠를 볼 수 없음

### 시작하기 전에
IBM Integration Bus가 기존 큐 관리자에 연결되게 해야 합니다. 다음의 명령에서 *QMgrName*을 이 기존 큐 관리자 이름으로 바꾸십시오.

### 이 태스크 정보
이 시나리오에서 Alice는 보호된 메시지를 입력 큐 `QIN`에 넣습니다. 메시지 특성 `routeTo`를 기반으로 메시지가 *bob*(`QBOB`),[1](`QCECIL`) 또는 기본(`QDEF`) 큐로 라우팅됩니다. Advanced Message Security는 메시지 페이로드만을 보호하고 해당 헤더나 특성은 보호하지 않고 보호되지 않은 상태로 남아 있고 IBM Integration Bus에 의해 판독될 수 있으므로 라우팅이 가능합니다. Advanced Message Security는 *alice*, *bob* 및 *cecil*에 의해서만 사용됩니다. IBM Integration Bus에 대해 설치하거나 구성할 필요가 없습니다.

IBM Integration Bus는 메시지를 복호화하려는 시도를 피하기 위해 보호되지 않는 알리어스 큐로부터 보호된 메시지를 받습니다. 보호된 큐를 직접 사용할 계획이었으면 메시지는 복호화 불가능한 것으로서 DEAD LETTER 큐에 놓일 것입니다. 메시지는 IBM Integration Bus에 의해 라우팅되고 대상 큐에 변경되지 않은 상태로 도착합니다. 그러므로 이는 여전히 원본 작성자(*bob* 및 *cecil* 둘 모두만이 *alice*가 전송한 메시지를 승인함)에 의해 서명되고 이전처럼 보호됩니다(*bob* 및 *cecil*만이 이를 읽을 수 있음). IBM Integration Bus는 라우팅된 메시지를 보호되지 않은 알리어스에 넣습니다. 수신인은 AMS가 메시지를 투명하게 복호화하는 보호된 출력 큐로부터 메시지를 검색합니다.

### 프로시저
1. **빠른 시작 안내서** (Windows 또는 AIX) 에 설명된 대로 Advanced Message Security 를 사용하도록 *alice*, *bob* 및 *cecil* 을 구성하십시오.
   다음 단계가 완료되었는지 확인하십시오.
   - 사용자 작성 및 권한 부여
   - 키 데이터베이스 및 인증서 작성
   - keystore.conf 작성
2. *bob* 및 *cecil*이 메시지에서 디지털 서명을 검사할 때 *alice*를 식별할 수 있도록 *alice*의 인증서를 이들에게 제공하십시오.
   *alice*를 식별하는 인증서를 외부 파일로 추출한 다음 추출된 인증서를 *bob* 및 *cecil*의 키 저장소에 추가하여 이를 수행하십시오. 다음에 설명된 방법을 사용하는 것이 중요합니다. **태스크 5. Sharing Certificates** in the **Quick Start Guide** (Windows or AIX).
3. *alice*가 *bob* 및 *cecil*에 대해 암호화된 메시지를 전송할 수 있도록 *bob* 및 *cecil*의 인증서를 *alice*에게 제공하십시오.

[1] cecil

이전 단계에 지정된 방법을 사용하여 이를 수행하십시오.

4. 큐 관리자에서 QIN, QBOB, QCECIL 및 QDEF라는 로컬 큐를 정의하십시오.

```
DEFINE QLOCAL(QIN)
```

5. QIN 큐에 대한 보안 정책을 적합한 구성으로 설정하십시오. QBOB, QCECIL 및 QDEF 큐에 동일한 설정을 사용하십시오.

```
setmqspl -m QMgrName -p QIN -s SHA1 -a "CN=alice,O=IBM,C=GB"
-e AES256 -r "CN=bob,O=IBM,C=GB" -r "CN=cecil,O=IBM,C=GB"
```

이 시나리오는 *alice*가 유일한 권한 있는 사용자이고 *bob*과 *cecil*이 수신인인 보안 정책을 가정합니다.

6. 로컬 큐 QIN, QBOB 및 QCECIL을 각각 참조하는 알리어스 큐 AIN, ABOB 및 ACECIL을 정의하십시오.

```
DEFINE QALIAS(AIN) TARGET(QIN)
```

7. 이전 단계에서 지정된 알리어스의 보안 구성이 존재하지 않는지 확인하십시오. 그렇지 않으면 해당 정책을 NONE으로 설정하십시오.

```
dspmqspl -m QMgrName -p AIN
```

8. IBM Integration Bus는 AIN 알리어스 큐에 도착하는 메시지를 메시지의 routeTo 특성에 따라 BOB, CECIL 또는 DEF 노드에 라우팅하기 위해 메시지 플로우를 작성합니다. 이를 수행하려면, 다음 단계를 따르십시오.
   a) IN이라는 MQInput 노드를 작성하고 AIN 알리어스를 해당 큐 이름으로 지정하십시오.
   b) BOB, CECIL 및 DEF라는 MQOutput 노드를 작성하고 알리어스 큐 ABOB, ACECIL 및 ADEF를 각각의 큐 이름으로 지정하십시오.
   c) 라우트 노드를 작성하고 이를 TEST라고 부르십시오.
   d) IN 노드를 TEST 노드의 입력 터미널에 연결하십시오.
   e) TEST 노드를 위한 bob 및 cecil 출력 터미널을 작성하십시오.
   f) bob 출력 터미널을 BOB 노드에 연결하십시오.
   g) cecil 출력 터미널을 CECIL 노드에 연결하십시오.
   h) DEF 노드를 기본 출력 터미널에 연결하십시오.
   i) 다음 규칙을 적용하십시오.

```
$Root/MQRFH2/usr/routeTo/text()="bob"
```

```
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. 메시지 플로우를 IBM Integration Bus 런타임 컴포넌트에 배치하십시오.
10. Alice 사용자로서 실행하여 bob 또는 cecil 값이 있는 routeTo라는 메시지 특성을 포함하는 메시지를 넣으십시오. **amqsstm** 샘플 애플리케이션을 실행하면 이를 수행할 수 있게 됩니다.

```
Sample AMQSSTMA start
target queue is TEST.Q
Enter property name
routeTo
Enter property value
bob
Enter property name

Enter message text
My Message to Bob
Sample AMQSSTMA end
```

11. *bob* 사용자로서 실행하면 **amqsget** 샘플 애플리케이션을 사용하여 메시지를 큐 QBOB로부터 검색합니다.

## 결과

*alice*가 메시지를 QIN 큐에 넣으면 메시지가 보호됩니다. 이는 IBM Integration Bus에 의해 AIN 알리어스 큐로부터 보호된 양식으로 검색됩니다. IBM Integration Bus는 routeTo 특성을 읽어서(모든 특성처럼 이는 암호화되지 않음) 메시지를 라우트할 위치를 결정합니다. IBM Integration Bus는 메시지를 추가적인 보호를 피하기 위해 적합한 보호되지 않은 알리어스에 놓습니다. 큐로부터 *bob* 또는 *cecil*에 의해 수신되면 메시지를 복호화되고 디지털 서명이 확인됩니다.

시나리오 *2* - 통합 버스가 메시지 컨텐츠를 볼 수 있음

## 이 태스크 정보

이 시나리오에서 개별 그룹은 메시지를 IBM Integration Bus에 보낼 수 있습니다. 또 다른 그룹은 IBM Integration Bus에 의해 작성된 메시지를 받을 권한이 있습니다. 당사자와 IBM Integration Bus 사이의 전송은 도청될 수 없습니다.

IBM Integration Bus는 보호 정책 및 인증서를 큐가 열릴 때에만 판독하므로 보호 정책을 업데이트한 후에 변경사항을 적용하려면 실행 그룹을 다시 로드해야 함을 기억하십시오.

```
mqsireload execution-group-name
```

IBM Integration Bus 가 메시지 페이로드를 읽거나 서명할 수 있는 권한 부여된 당사자로 간주되는 경우 IBM Integration Bus 서비스를 시작하는 사용자에 대해 Advanced Message Security 를 구성해야 합니다. 메시지를 큐에 넣고/가져오는 사용자나 IBM Integration Bus 애플리케이션을 작성하고 배치하는 사용자가 반드시 동일할 필요는 없음을 유의하십시오.

## 프로시저

1. **빠른 시작 안내서** (Windows 또는 AIX) 에 설명된 대로 Advanced Message Security 를 사용하도록 *alice*, *bob*, *cecil* 및 *dave* 및 IBM Integration Bus 서비스 사용자를 구성하십시오.

   다음 단계가 완료되었는지 확인하십시오.

   - 사용자 작성 및 권한 부여
   - 키 데이터베이스 및 인증서 작성
   - keystore.conf 작성
2. *alice*, *bob*, *cecil* 및 *dave* 인증서를 IBM Integration Bus 서비스 사용자에게 제공하십시오.

   *alice*, *bob*, *cecil* 및 *dave*를 식별하는 각 인증서를 외부 파일에 추출한 다음 추출된 인증서를 IBM Integration Bus 키 저장소에 추가하여 이를 수행하십시오. 다음에 설명된 방법을 사용하는 것이 중요합니다. **태스크 5. Sharing Certificates** in the **Quick Start Guide** (Windows or AIX).
3. IBM Integration Bus 서비스 사용자의 인증서를 *alice*, *bob*, *cecil* 및 *dave*에게 제공하십시오.

   이전 단계에 지정된 방법을 사용하여 이를 수행하십시오.

   **참고:** *Alice* 및 *bob*은 메시지를 올바르게 암호화하기 위해서 IBM Integration Bus 서비스 사용자의 인증서가 필요합니다. IBM Integration Bus 서비스 사용자는 메시지의 작성자를 확인하기 위해 *alice* 및 *bob*의 인증서가 필요합니다. IBM Integration Bus 서비스 사용자는 메시지를 암호화하기 위해 *cecil* 및 *dave*의 인증서가 필요합니다. *cecil* 및 *dave*는 메시지가 IBM Integration Bus에서 오는지를 확인하기 위해 IBM Integration Bus 서비스 사용자의 인증서가 필요합니다.
4. 이름이 IN인 로컬 큐를 정의하고 *alice* 및 *bob*이 작성자로 지정되고 IBM Integration Bus의 서비스 사용자가 수신인으로 지정된 보안 정책을 정의하십시오.

   ```
   setmqspl -m QMgrName -p IN -s MD5 -a "CN=alice,O=IBM,C=GB" -a "CN=bob,O=IBM,C=GB"
   -e AES256 -r "CN=broker,O=IBM,C=GB"
   ```
5. 이름이 OUT인 로컬 큐를 정의하고, IBM Integration Bus가 작성자로 지정되고, *cecil* 및 *dave*가 수신인으로 지정된 보안 정책을 정의하십시오.

```
setmqspl -m QMgrName -p OUT -s MD5 -a "CN=broker,O=IBM,C=GB" -e AES256
-r "CN=cecil,O=IBM,C=GB" -r "CN=dave,O=IBM,C=GB"
```

6. IBM Integration Bus에서는 MQInput 및 MQOutput 노드가 있는 메시지 플로우를 작성합니다. IN 큐를 사용하도록 MQInput 노드를 구성하고 OUT 큐를 사용하도록 MQOutput 노드를 구성하십시오.
7. 메시지 플로우를 IBM Integration Bus 런타임 컴포넌트에 배치하십시오.
8. *alice* 또는 *bob* 사용자로서 실행하여 **amqsput** 샘플 애플리케이션을 사용하여 메시지를 IN 큐에 넣으십시오.
9. *cecil* 또는 *dave* 사용자로서 실행하여 **amqsget** 샘플 애플리케이션을 사용하여 OUT 큐로부터 메시지를 검색합니다.

## 결과

*alice* 또는 *bob*이 입력 큐 IN에 전송한 메시지는 암호화되어 IBM Integration Bus만이 이를 읽을 수 있습니다. IBM Integration Bus는 *alice* 및 *bob*으로부터 온 메시지만을 승인하고 나머지는 거부합니다. 승인된 메시지는 적절하게 처리된 다음 출력 큐 OUT에 넣기 전에 *cecil* 및 *dave*의 키로 서명되고 암호화됩니다. *cecil* 및 *dave*만이 이를 읽을 수 있고, IBM Integration Bus에 의해 서명되지 않은 메시지는 거부됩니다.

## *Managed File Transfer*과(와) 함께 *Advanced Message Security* 사용

이 시나리오는 Managed File Transfer를 통해 전송되는 데이터에 대한 메시지 개인정보 보호를 제공하도록 Advanced Message Security 를 구성하는 방법을 설명합니다.

## 시작하기 전에

보호하려는 Managed File Transfer 에서 사용하는 큐를 호스트하는 IBM MQ 설치에 Advanced Message Security 구성요소가 설치되어 있는지 확인하십시오.

Managed File Transfer 에이전트가 바인딩 모드에서 연결 중인 경우 로컬 설치에 IBM Global Security Kit (GSKit) 컴포넌트도 설치되어 있는지 확인하십시오.

## 이 태스크 정보

데이터를 두 개의 Managed File Transfer 에이전트 사이에서 전송할 때 중단되면 기밀 데이터가 전송을 관리하는 데 사용되는 기본 IBM MQ 큐에 보호되지 않은 상태로 남아 있을 수도 있습니다. 이 시나리오에서는 Managed File Transfer 큐에서 이러한 데이터를 보호하기 위해 Advanced Message Security 를 구성하고 사용하는 방법을 설명합니다.

이 시나리오에서는 시나리오 Managed File Transfer 시나리오에 설명된 대로 단일 큐 관리자를 공유하는 두 개의 Managed File Transfer 큐와 두 개의 에이전트 AGENT1 및 AGENT2가 있는 하나의 시스템으로 구성된 단순 토폴로지를 고려합니다. 두 에이전트 모두 바인딩 모드 또는 클라이언트 모드에서 동일한 방법으로 연결됩니다.

*1.* 인증서 작성

## 시작하기 전에

이 시나리오에서는 FTAGENTS 그룹의 사용자 ftagent를 사용하여 Managed File Transfer Agent 프로세스를 실행하는 단순한 모형을 사용합니다. 사용자 고유의 사용자 이름과 그룹 이름을 사용하는 경우 그에 맞게 명령을 변경하십시오.

## 이 태스크 정보

Advanced Message Security는 보호된 큐에서 메시지를 서명 및/또는 암호화하기 위해 공개 키 암호화를 사용합니다.

**참고:**

- Managed File Transfer 에이전트가 바인딩 모드에서 실행 중인 경우에는 CMS(Cryptographic Message Syntax) 키 저장소를 작성하는 데 사용하는 명령이 플랫폼별 **빠른 시작 안내서**(Windows 또는 AIX)에 설명되어 있습니다.

- Managed File Transfer 에이전트가 클라이언트 모드에서 실행 중이면 JKS(Java 키 저장소)를 작성하는 데 필요한 명령은 587 페이지의 『Java 클라이언트가 있는 AMS 용 빠른 시작 안내서』에 설명되어 있습니다.

### 프로시저

1. 적합한 빠른 시작 안내서에 설명된 대로 사용자 `ftagent`를 식별하기 위해 자체 서명 인증서를 작성하십시오.
   다음과 같이 식별 이름(DN)을 사용하십시오.

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. 적합한 빠른 시작 안내서에 설명된 대로 키 저장소의 위치와 그 안의 인증서의 위치를 식별하기 위해 `keystore.conf` 파일을 작성하십시오.

*2.* 메시지 보호 구성

### 이 태스크 정보

**setmqspl** 명령을 사용하여 AGENT2가 사용하는 데이터 큐의 보안 정책을 정의해야 합니다. 이 시나리오에서 동일한 사용자가 두 에이전트를 모두 시작하는 데 사용되므로 서명자와 수신자 DN은 동일하고 우리가 생성한 인증서와 일치합니다.

### 프로시저

1. **fteStopAgent** 명령을 사용하여 보호하기 위한 준비로 Managed File Transfer 에이전트를 종료하십시오.
2. SYSTEM.FTE.DATA.AGENT2 큐를 보호하기 위해 보안 정책을 작성하십시오.

   ```
   setmqspl -m hubQM -p SYSTEM.FTE.DATA.AGENT2 -s SHA1 -a "CN=ftagent, OU=MFT,
   O=<organisation>, L=<location>, ST=<state>, C=<country>"
   -e AES128 -r "CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>"
   ```

3. Managed File Transfer Agent 프로세스를 실행 중인 사용자에게 시스템 정책 큐를 찾아보고 메시지를 오류 큐에 넣기 위해 필요한 액세스 권한이 있는지 확인하십시오.

   ```
   setmqaut -m hubQM -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p ftagent +browse
   ```

   ```
   setmqaut -m hubQM -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p ftagent +put
   ```

4. **fteStartAgent** 명령을 사용하여 Managed File Transfer 에이전트를 다시 시작하십시오.
5. **fteListAgents** 명령을 사용하고 에이전트가 READY 상태에 있는지 확인하여 에이전트가 성공적으로 재시작되었는지 확인하십시오.

### 결과

이제 전송을 AGENT1에서 AGENT2로 제출할 수 있고, 파일 컨텐츠는 두 에이전트 간에 안전하게 전송됩니다.

## Advanced Message Security 설치 개요

다양한 플랫폼에 Advanced Message Security 컴포넌트를 설치합니다.

### 프로시저

- Multi
  멀티플랫폼에서 Advanced Message Security를 설치하십시오.
- z/OS
  IBM MQ Advanced for z/OS 설치.

- z/OS
  IBM MQ Advanced for z/OS Value Unit Edition 설치.

**관련 태스크**

Advanced Message Security 설치 제거

## z/OS Auditing for AMS on z/OS

Advanced Message Security (AMS) for z/OS provides a means for optional auditing of operations by applications on policy protected queues. When enabled, IBM System Management Facility (SMF) audit records are generated for the success and failure of these operations on policy-protected queues. Operations audited include MQPUT, MQPUT1, and MQGET.

Auditing is disabled by default, however, you can activate auditing by configuring _AMS_SMF_TYPE and _AMS_SMF_AUDIT in the configured Language Environment® _CEE_ENVFILE file for the AMS address space. For more information, see Create procedures for Advanced Message Security. The _AMS_SMF_TYPE variable is used to designate the SMF record type and is a number between 128 and 255. A SMF record type of 180 is usual, however is not mandatory. Auditing is disabled by specifying a value of 0. The _AMS_SMF_AUDIT variable configures whether audit records are created for operations that are successful, operations that fail, or both. The auditing options can also be dynamically changed while AMS is active using operator commands. For more information, see Operating Advanced Message Security.

The SMF record is defined using subtypes, with subtype 1 being a general auditing event. The SMF record contains all data relevant to the request being processed.

The SMF record is mapped by the CSQ0KSMF macro (note the zero in the macro name), which is provided in the target library SCSQMACS. If you are writing data-reduction programs for SMF data, you can include this mapping macro to aid in the development and customization of SMF post-processing routines.

In the SMF records produced by Advanced Message Security for z/OS, the data is organized into sections. The record consists of:

- a standard SMF header
- a header extension defined by Advanced Message Security for z/OS
- a product section
- a data section

The product section of the SMF record is always present in the records produced by Advanced Message Security for z/OS. The data section varies based on subtype. Currently, one subtype is defined and therefore a single data section is used.

SMF is described in the z/OS System Management Facilities manual (SA22-7630). Valid record types are described in the SMFPRMxx member of your system PARMLIB data set. See SMF documentation for more information.

### Advanced Message Security audit report generator (CSQ0USMF)

Advanced Message Security for z/OS provides an audit report generator tool called CSQ0USMF which is provided in the installation SCSQAUTH library. Sample JCL to run the CSQ0USMF utility called CSQ40RSM is provided in the installation library SCSQPROC.

Before running the CSQ0USMF utility, the SMF type 180 records must be dumped from the system SMF data sets to a sequential data set. As an example, this JCL dumps SMF type 180 records from an SMF data set, and transfers them to a target data set:

```
//IFAUDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYSn.MANn.syst,DISP=SHR
//OUTDD1 DD DSN=your.target.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
```

```
OUTDD(OUTDD1,TYPE(180))
/*
```

You must verify the actual SMF data set names used by your installation. The target data set for the dumped records must have a record format of VBS, and a record length of 32760.

**Note:** If SMF logstreams are being used, you must use program IFASMFDL to dump a logstream out to a sequential dataset. See Processing type 116 SMF records for an example of the JCL used.

The target data set can then be used as input to the CSQ0USMF utility to produce an AMS audit report. For example:

```
//STEP1 EXEC PGM=CSQ0USMF,
// PARM=('/ -SMFTYPE 180 -M qmgr')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//    DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SMFIN DD DSN=your.target.dataset,DISP=SHR
//
```

The CSQ0USMF program accepts two optional parameters, which are listed in Table 103 on page 599:

*Table 103. CSQ0USMF optional parameters*

| Parameter | Value | Description |
|---|---|---|
| SMFTYPE | nnn | The SMF record type applicable to the audit report. The CSQ0USMF program uses only SMF records that match the SMFTYPE value when generating the report. If you do not specify SMFTYPE, a default value of 180 is used. |
| M | qmgr | The IBM MQ queue manager name applicable to the audit report. If you do not specify the -M parameter, the audit report will include all audit records for all queue managers represented in the SMFIN data set. |

## AMS과(와) 함께 키스트로크와 인증서 사용

IBM MQ 애플리케이션에 투명한 암호 보호를 제공하기 위해 Advanced Message Security는 공개 키 인증서 및 개인 키가 저장되는 키 저장소 파일을 사용합니다. z/OS에서 SAF키 링이 키 저장소 파일 대신 사용됩니다.

Advanced Message Security에서 사용자 및 애플리케이션은 PKI(Public Key Infrastructure) ID로 표시됩니다. 이 ID 유형은 메시지를 서명하고 암호화하는 데 사용됩니다. PKI ID는 서명되고 암호화된 메시지와 연관된 인증서에 제목의 **식별 이름(DN)** 필드에 나타납니다. 사용자나 애플리케이션이 이들의 메시지를 암호화하기 위해서는 인증서 및 연관된 개인 및 공개 키가 저장되는 키 저장소 파일에 대한 액세스가 필요합니다.

ALW AIX, Linux, and Windows에서 키 저장소의 위치는 기본적으로 `keystore.conf`인 키 저장소 구성 파일에 제공됩니다. 각 Advanced Message Security 사용자에게는 키 저장소 파일을 가리키는 키 저장소 구성 파일이 있어야 합니다. Advanced Message Security에서는 `.kdb`, `.jceks`, `.jks` 형식의 키 저장소 파일 형식을 승인합니다.

`keystore.conf` 파일의 기본 위치는 다음과 같습니다.

- Linux IBM i AIX IBM i, AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows Windows: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

지정된 키 저장소 파일 이름 및 위치를 사용하는 경우 다음 예제 명령에 표시된 대로 **MQS_KEYSTORE_CONF** 환경 변수를 사용하여 이를 지정해야 합니다.

- Java의 경우: `java -DMQS_KEYSTORE_CONF=path/filename app_name`
- C 클라이언트 및 서버의 경우:
  - Linux AIX AIX and Linux: `export MQS_KEYSTORE_CONF=path/filename`
  - Windows Windows: `set MQS_KEYSTORE_CONF=path\filename`

    **참고:** Windows의 경로는 둘 이상의 드라이브 문자가 사용 가능한 경우 드라이브 이름을 지정할 수 있습니다.

## keystore.conf 파일에서 민감한 정보 보호

비밀번호와 같은 키 저장소 파일의 민감한 정보에 액세스하려면 IBM MQ Advanced Message Security(AMS)에서 키 저장소에 액세스하고 메시지에 서명하며 메시지를 암호화할 수 있도록 토큰을 제공해야 합니다.

AMS와 함께 제공되는 **runamscred** 명령을 사용하여 키 저장소 구성 파일에 포함된 민감한 정보를 보호해야 합니다. 구성 파일을 보호하는 방법에 대한 자세한 정보는 617 페이지의 『구성 파일에 대한 AMS 비밀번호 보호 설정』의 내용을 참조하십시오.

비밀번호를 보호하는 경우 강력한 사용자 정의 암호화 키를 사용해야 합니다. 런타임 중에 비밀번호에 액세스하려면 이 암호화 키를 AMS에 제공해야 합니다.

암호화 키 파일 위치를 제공하는 다음과 같은 두 가지 방법이 있습니다.

- keystore.conf 파일의 **amscred.keyfile** 구성 특성
- **MQS_AMSCRED_KEYFILE** 환경 변수

우선순위의 순서는 **MQS_AMSCRED_KEYFILE**, **amscred.keyfile**, 기본 키 순입니다.

**관련 개념**

626 페이지의 『AMS의 송신자 식별 이름』
송신자 식별 이름(DN)은 메시지를 큐에 배치할 수 있는 권한을 가진 사용자를 식별합니다. 송신자는 메시지를 큐에 배치하기 전에 인증서를 사용하여 메시지에 서명합니다.

627 페이지의 『AMS의 수신인 식별 이름』
수신자 식별 이름(DN)은 큐로부터 메시지를 수신할 권한이 있는 사용자를 식별합니다.

## AMS 에 대한 키 저장소 구성 파일 (keystore.conf) 의 구조

키 저장소 구성 파일 (keystore.conf) 은 Advanced Message Security 에서 적절한 키 저장소의 위치를 가리킵니다.

다음 구성 파일 유형 각각에는 접두부가 있습니다.

**AMSCRED**
비밀번호 보호 시스템과 관련된 매개변수입니다.

**CMS**
인증서 관리 시스템(CMS), 구성 항목의 접두부는 cms. 입니다.

**PKCS#11**
공개 키 암호 표준(PKCS) #11, 구성 항목의 접두부는 pkcs11. 입니다.

IBM i **PEM**
PEM(Privacy Enhanced Mail) 형식, 구성 항목의 접두부는 pem. 입니다.

**JKS**
Java 키 저장소, 구성 항목의 접두부는 jks. 입니다.

**JCEKS**
Java 암호 암호화 키 저장소, 구성 항목의 접두부는 jceks. 입니다.

z/OS MQ Adv. VUE **JCERACFKS**

Java 암호 암호화 RACF 키 링 키 저장소, 구성 항목의 접두부는 `jceracfks`입니다.

**중요사항:** IBM MQ 9.0에서 `JCEKS.provider` 및 `JKS.provider` 값이 무시됩니다. Bouncy Castle 제공자는 사용 중 JRE에서 제공하는 JCE/JCE 프로비전과 함께 사용됩니다. 자세한 정보는 604 페이지의 『AMS 에서 비 IBM JRE에 대한 지원』의 내용을 참조하십시오.

키 저장소의 구조 예:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.encrypted = no
```

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
pem.encrypted = no
```

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
```

Java JCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
```

Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

Java PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.secondary_keystore_pass = password
pkcs11.encrypted = no
```

표 *104.* 각 구성 파일 유형에 필요한 매개변수의 요약

| 매개변수 | 필수 | 구성 파일 유형 | | | | |
|---|---|---|---|---|---|---|
| | | **Java (PKCS#11, JKS, JCEKS 및 JCERACFKS)** | **IBM i PEM** | **PKCS#11** | **CMS** | **AMSCRED** |
| keystore | ✓ | ✓ | | | ✓ | |
| IBM i private | ✓ | | IBM i ✓ | | | |
| IBM i public | ✓ | | IBM i ✓ | | | |
| IBM i password | ✓ | | IBM i ✓ | | | |
| library | ✓ | ✓ | | ✓ | | |
| certificate | ✓ | ✓ | | ✓ | ✓ | |
| token | ✓ | ✓ | | ✓ | | |
| token_pin | ✓ | ✓ | | ✓ | | |
| secondary_keystore | ✓ | ✓ | | ✓ | | |
| secondary_keystore_password | ✓ | ✓ | | | | |
| encrypted | | ✓ | IBM i ✓ | ✓ | | |
| keystore_pass | ✓ | ✓ | | | | |
| key_pass | | ✓ | | | | |
| provider | | ✓ | | | | |
| keyfile | | | | | | ✓ |

# 기호를 사용하여 주석을 추가할 수 있습니다.

구성 파일 매개변수는 다음과 같이 정의됩니다.

**keystore**

CMS 및 Java 구성만.

CMS, JKS 및 JCEKS 구성에 대한 키 저장소 파일의 경로입니다.

z/OS MQ Adv. VUE JCERACFKS 구성을 위한 RACF 키 링에 대한 URI입니다.

**중요사항:**

- 키 저장소 파일에 대한 경로는 파일 확장자를 포함하지 않아야 합니다.

- z/OS MQ Adv. VUE RACF 키 링에 대한 URI는 다음 양식이어야 합니다.

  ```
  safkeyring://user/keyring
  ```

  설명:

  - *user*은(는) 키 링을 소유하는 사용자 ID입니다.
  - *keyring*은(는) 키링 이름입니다.

IBM i **private**

PEM 구성만.

PEM 형식에 개인 키 및 인증서를 포함하는 파일의 파일 이름.

IBM i **public**

PEM 구성만.

PEM 형식에 신뢰되는 공개 인증서를 포함하는 파일의 파일 이름.

IBM i **password**

PEM 구성만.

암호화된 개인 키를 복호화하는 데 사용되는 비밀번호입니다.

고유 AMS 비밀번호 보호 도구를 사용하여 이 필드를 보호해야 합니다. 604 페이지의 『비밀번호 보호』의 내용을 참조하십시오.

**library**

PKCS#11만.

PKCS#11 라이브러리의 경로 이름.

**certificate**

CMS, PKCS#11 및 Java 구성만.

인증서 레이블.

**token**

PKCS#11만.

토큰 레이블.

**token_pin**

PKCS#11만.

토큰을 잠금 해제하기 위한 PIN.

Java 조작 전용으로, Java AMS 비밀번호 보호 도구를 사용하여 이 필드를 보호해야 합니다(604 페이지의 『비밀번호 보호』 참조).

고유 조작의 경우에 한해, 고유 AMS 비밀번호 보호 도구를 사용하여 이 필드를 보호해야 합니다. 604 페이지의 『비밀번호 보호』의 내용을 참조하십시오.

**secondary_keystore**

PKCS#11만.

PKCS #11 토큰에 저장된 인증서에 필요한 앵커 인증서(루트 인증서)를 포함하는 CMS 키 저장소의 경로 이름(.kdb 확장자 없이 제공됨). 보조 키 저장소에는 또한 신뢰 체인에서 중간에 있는 인증서뿐만 아니라 개인 정보 보안 정책에 정의된 수신인 인증서를 포함할 수 있습니다. 이 CMS 키 저장소는 보조 키 저장소와 같은 디렉토리에 있어야 하는 스태쉬 파일을 동반해야 합니다.

Java 환경의 경우 JKS 키 저장소가 필요하며 **secondary_keystore_password**을(를) 제공해야 합니다.

**secondary_keystore_password**

Java PKCS#11만 해당합니다.

JKS 키 저장소의 비밀번호는 secondary_keystore 특성을 통해 제공됩니다. Java AMS 비밀번호 보호 도구를 사용하여 이 필드를 보호해야 합니다. 604 페이지의 『비밀번호 보호』의 내용을 참조하십시오.

**encrypted**

Java 및 IBM MQ 9.3.0에서는 PKCS#11 및 IBM i PEM 전용입니다.

비밀번호의 상태입니다.

**keystore_pass**
Java 구성만.

키 저장소 파일의 비밀번호.

Java 조작만 해당합니다. Java AMS 비밀번호 보호 도구를 사용하여 이 필드를 보호해야 합니다. 604 페이지의 『비밀번호 보호』의 내용을 참조하십시오.

**key_pass**
Java 구성만.

사용자 개인 키의 비밀번호입니다.

Java 조작 전용으로, Java AMS 비밀번호 보호 도구를 사용하여 이 필드를 보호해야 합니다(604 페이지의 『비밀번호 보호』 참조).

**keyfile**
이 구성 파일에 포함된 비밀번호를 보호하거나 복호화할 때 사용할 초기 키 위치를 제공합니다(604 페이지의 『비밀번호 보호』 참조).

**provider**
Java 구성만.

키 저장소 인증서가 필요로 하는 암호화 알고리즘을 구현하는 Java 보안 제공자.

**중요사항:** 키 저장소에 저장되는 정보는 IBM MQ를 사용하여 전송되는 데이터의 안전한 플로우를 위해 중요합니다. 보안 관리자는 이들에게 파일 권한을 지정할 때 특히 주의를 기울여야 합니다.

### 비밀번호 보호

`keystore.conf` 파일에 포함된 비밀번호 및 기타 민감한 정보를 보호해야 합니다. 자세한 정보는 **runamscred**의 내용을 참조하십시오.

`keystore.conf` 파일의 예:

```
# Native AMS application configuration
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

# Java AMS application configuration
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

**관련 태스크**
617 페이지의 『구성 파일에 대한 AMS 비밀번호 보호 설정』
일반 텍스트로 키 저장소 및 개인 키 비밀번호를 저장하면 보안 위험에 노출되므로, Advanced Message Security에서는 사용자 키를 사용하여 이러한 비밀번호를 스크램블할 수 있는 도구를 제공합니다.

## AMS 에서 비IBM JRE에 대한 지원

IBM MQ classes for Java 및 IBM MQ classes for JMS 는 비IBM JRE를 사용하여 실행할 때 Advanced Message Security 조작을 지원합니다.

Advanced Message Security(AMS)는 CMS(Cryptographic Message Syntax)를 구현합니다. CMS 구문은 임의의 메시지 컨텐츠에 대한 디지털 서명, 요약, 인증 또는 암호화에 사용됩니다.

IBM MQ 9.0부터 IBM MQ classes for Java 및 IBM MQ classes for JMS 의 Advanced Message Security 지원은 오픈 소스 Bouncy Castle 패키지를 사용하여 CMS를 지원합니다. 이는 이러한 클래스가 비IBM JRE를 사용하여 실행할 때 Advanced Message Security 조작을 지원할 수 있음을 의미합니다.

IBM MQ 9.0이전에는 Java 클라이언트의 비IBM JRE에서 Advanced Message Security 가 지원되지 않았습니다. IBM MQ classes for Java 및 IBM MQ classes for JMS 의 Advanced Message Security 지원은 JCE ( Java Cryptography Extensions) 의 IBM 구현에서 특별히 제공하는 CMS 지원에 종속되어 있습니다. 이 제한사항으로

인해 Java JCE 제공자가 포함된 JRE ( Java runtime environment ) 를 사용하는 경우에만 기능을 사용할 수 있습니다.

## Bouncy Castle JAR 파일에 대한 위치 및 버전 번호 지정

비IBM JRE에 대한 지원에 필요한 Bouncy Castle JAR 파일은 IBM MQ classes for Java 및 IBM MQ classes for JMS 설치 패키지의 부분으로 포함됩니다.

사용되는 Bouncy Castle JAR 파일은 다음과 같습니다.

**Bouncy Castle 조작의 핵심인 제공자 JAR 파일.**

V 9.4.0 IBM MQ 9.4.0부터 이 JAR 파일을 `bcprov-jdk18on.jar`라고 합니다.

**Advanced Message Security에서 사용하는 CMS 조작 지원을 포함하는 "PKIX" JAR 파일.**

V 9.4.0 IBM MQ 9.4.0부터 이 JAR 파일을 `bcpkix-jdk18on.jar`라고 합니다.

**다른 Bouncy Castle JAR 파일에서 사용하는 클래스를 포함하는 "util" JAR 파일.**

V 9.4.0 IBM MQ 9.4.0부터 이 JAR 파일을 `bcutil-jdk18on.jar`라고 합니다.

## 종속 항목

IBM MQ 9.1 이상 클래스는 IBM JRE 및 Oracle JRE로 테스트되었습니다. 또한 J2SE 준수 JRE에서 성공적으로 실행될 수 있습니다. 그러나 다음 종속성에 주의해야 합니다.

- Advanced Message Security 구성에 대한 변경사항이 없습니다.
- Bouncy Castle 클래스는 CMS 조작에 대해서만 사용됩니다. 다른 모든 보안 관련 조작(예: 키 저장소 액세스), 실제 데이터 암호화, 서명 체크섬 계산은 JRE에서 제공하는 기능을 사용합니다.

  **중요사항:** 이러한 이유로, 사용되는 JRE는 JCE 제공자 구현을 포함해야 합니다.
- 일부 강력한 암호화 알고리즘을 사용하려면 JRE의 JCE 구현에 대해 제한 없는 정책 파일을 설치해야 합니다.

  자세한 정보는 JRE 문서를 참조하십시오.
- Java 보안을 사용하는 경우:
  - `java.security.SecurityPermissioninsertProvider.BC`를 애플리케이션에 추가하십시오. 그러면 Bouncy Castle 클래스를 보안 제공자로 사용할 수 있습니다.
  - `java.security.AllPermission` 를 Bouncy Castle JAR 파일에 부여하십시오.

    V 9.4.0 IBM MQ 9.4.0에서 이러한 파일은 다음과 같습니다.

    *mq_install_dir*`/java/lib/bcutil-jdk18on.jar`
    *mq_install_dir*`/java/lib/bcpkix-jdk18on.jar`
    *mq_install_dir*`/java/lib/bcprov-jdk18on.jar`

**관련 개념**

JMS용 IBM MQ 클래스에 대해 설치되는 항목
Java 용 IBM MQ 클래스에 대해 설치되는 항목

## Multi 메시지 채널 에이전트(MCA) 인터셉션 및 AMS

MCA 인터셉션 기능을 사용하면 IBM MQ에서 실행 중인 큐 관리자가 서버 연결 채널에 선택적으로 정책을 적용할 수 있습니다.

AMS 외부에 남아 있는 클라이언트는 MCA 인터셉션을 통해 큐 관리자에 계속 연결되어 있고 메시지를 암호화하고 복호화할 수 있습니다.

MCA 인터셉션은 클라이언트에서 AMS를 사용할 수 없을 때 AMS 기능을 제공하기 위한 것입니다. MCA 인터셉션과 AMS 가능 클라이언트를 사용하면 메시지가 이중으로 보호되어 애플리케이션 수신에 장애가 될 수 있습니다. 자세한 정보는 608 페이지의 『클라이언트에서 Advanced Message Security 사용 안함』의 내용을 참조하십시오.

**참고:** AMQP 또는 MQTT 채널의 경우 MCA 인터셉터가 지원되지 않습니다.

### 키 저장소 구성 파일

기본적으로 MCA 인터셉션의 키 저장소 구성 파일은 `keystore.conf`이고 큐 관리자 또는 리스너를 시작한 사용자의 HOME 디렉토리 경로에 `.mqs` 디렉토리에 있습니다. 키 저장소는 MQS_KEYSTORE_CONF 환경 변수를 사용하여 구성될 수도 있습니다. AMS 키 저장소 구성에 대한 자세한 정보는 599 페이지의 『AMS과(와) 함께 키 스트로크와 인증서 사용』을 참조하십시오.

MCA 인터셉션을 사용으로 설정하려면 키 저장소 구성 파일에서 사용하고 싶은 채널의 이름을 제공해야 합니다. MCA 인터셉션의 경우 CMS 키 저장소 유형만 사용할 수 있습니다.

606 페이지의 『AMS의 MCA 인터셉션 예』에서 MCA 인터셉션 설정 예를 확인하십시오.

**주의:** 권한 있는 클라이언트만이 이 기능에 연결하여 사용할 수 있게 하려면 SSL 및 SSLPEER 또는 CHLAUTH TYPE(SSLPEERMAP)을 사용하여 선택된 채널에서 클라이언트 인증 및 암호화를 완료해야 합니다.

IBM i

엔터프라이즈에서 IBM i를 사용하고 인증서에 서명하기 위해 상용 인증 기관 (CA) 을 선택한 경우 디지털 Certificate Manager 는 PEM (Privacy-Enhanced Mail) 형식으로 인증서 요청을 작성합니다. 요청을 선택한 CA에 전달해야 합니다.

이를 수행하려면 `channelname`에 지정된 채널의 올바른 인증서를 선택하기 위해 다음 명령을 사용해야 합니다.

```
pem.certificate.channel.channelname
```

## *AMS*의 *MCA* 인터셉션 예

AMS MCA 인터셉션 설정 방법에 대한 태스크 예입니다.

### 시작하기 전에

**주의:** 권한 있는 클라이언트만이 이 기능에 연결하여 사용할 수 있게 하려면 SSL 및 SSLPEER 또는 CHLAUTH TYPE(SSLPEERMAP)을 사용하여 선택된 채널에서 클라이언트 인증 및 암호화를 완료해야 합니다.

엔터프라이즈에서 IBM i를 사용하고 인증서에 서명하기 위해 상용 인증 기관 (CA) 을 선택한 경우 디지털 Certificate Manager 는 PEM (Privacy-Enhanced Mail) 형식으로 인증서 요청을 작성합니다. 요청을 선택한 CA에 전달해야 합니다.

### 이 태스크 정보

이 태스크는 MCA 인터셉션을 사용하도록 시스템을 설정하고 설정을 확인하는 프로세스를 설명합니다.

**참고:** IBM MQ는 AMS 인터셉터를 포함하며 MQ 클라이언트 및 서버 런타임 환경에서 동적으로 사용으로 설정합니다.

**주의:**

- 코드에서 `userID`를 사용자의 ID로 바꾸십시오.
- 다음 프로시저는 클라이언트에서 AMS 인터셉션이 비활성화되지 않으면 IBM MQ 에서 예상대로 작동하지 않습니다.

### 프로시저

1. 쉘 스크립트를 작성하려면 다음 명령을 사용하여 키 데이터베이스 및 인증서를 작성하십시오.

또한 **INSTLOC** 및 **KEYSTORELOC**를 변경하거나 필수 명령을 실행하십시오. bob의 인증서를 작성할 필요가 없을 수도 있음을 유의하십시오.

```
INSTLOC=/opt/mqm
KEYSTORELOC=/home/userID/var/mqm
mkdir -p $KEYSTORELOC
chmod -R 777 $KEYSTORELOC
chown -R mqm:mqm $KEYSTORELOC
export PATH=$PATH:$INSTLOC/gskit8/bin
echo "PATH = $PATH"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTLOC/gskit8/lib64

runmqakm -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
runmqakm -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
runmqakm -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd \
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
runmqakm -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd \
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. 각 사용자가 서로를 올바르게 식별할 수 있도록 두 키 데이터베이스 사이에서 인증서를 공유하십시오.

   엔터프라이즈에서 사용하는 플랫폼에 대해 빠른 시작 안내서에 설명된 인증서 공유 방법을 사용하는 것이 중요합니다.

   **Windows**
   태스크 5인증서 공유

   **AIX and Linux**
   태스크 5인증서 공유

   **Java 클라이언트**
   태스크 5인증서 공유
3. 다음 구성으로 `keystore.conf`를 작성하십시오. `Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

**주의:**

a. 키 저장소는 큐 관리자가 있는 시스템에 있어야 합니다.

b. MCA 개입을 사용으로 설정하려면 `cms.certificate` 에 대한 특정 채널을 지정해야 합니다. 그러면 큐 관리자가 해당 채널을 통해 정책이 설정된 큐에 연결하는 애플리케이션에서 AMS 조작을 수행합니다.

4. AMSQMGR1 큐 관리자를 작성하고 시작하십시오.
5. QMGR 제어 하에서 사용 가능한 포트 번호를 사용하여 TCP 리스너를 정의하십시오.
   예를 들면, 다음과 같습니다.

```
DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
```

6. 리스너를 시작하고 올바르게 시작되었는지 확인하십시오.
   예를 들면, 다음과 같습니다.

```
START  LISTENER(MY.LISTENER)
DISPLAY LSSTATUS(MY.LISTENER) PORT
```

7. 큐 관리자를 중지합니다.
8. 키 저장소를 설정하십시오.

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. MQS_KEYSTORE_CONF 환경 변수가 큐 관리자에 사용 가능하도록 동일한 쉘에서 큐 관리자를 시작하십시오.
10. 보안 정책을 설정하고 다음을 확인하십시오.

```
setmqspl -m AMSQMGR1 -s SHA256 -e AES256 -p TESTQ -a "CN=alice,O=IBM,C=IN" \
-r "CN=alice,O=IBM,C=IN"
dspmqspl -m AMSQMGR1
```

자세한 정보는 setmqspl 및 dspmqspl의 내용을 참조하십시오.

11. *MQSERVER* 환경 변수를 설정하십시오.

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. 보안 정책을 제거하고 결과를 확인하십시오.

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

13. IBM MQ 9.4 설치에서 큐를 찾아보십시오.

```
/opt/mq93/samp/bin/amqsbcg TESTQ AMSQMGR1
```

찾아보기 출력은 암호화된 형식으로 메시지를 표시합니다.

14. 보안 정책을 설정하고 결과를 확인하십시오.

```
setmqspl -m AMSQMGR1 -s SHA256 -e AES256 -p TESTQ -a "CN=alice,O=IBM,C=IN"
-r "CN=alice,O=IBM,C=IN"
dspmqspl -m AMSQMGR1
```

15. IBM MQ 9.4 설치에서 **amqsgetc** 를 실행하십시오.

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

**관련 개념**

600 페이지의 『AMS 에 대한 키 저장소 구성 파일 (keystore.conf) 의 구조』

키 저장소 구성 파일 (`keystore.conf`) 은 Advanced Message Security 에서 적절한 키 저장소의 위치를 가리킵니다.

**관련 참조**

561 페이지의 『AMS의 알려진 제한사항』

지원되지 않거나 Advanced Message Security에 대한 제한사항이 있는 여러 IBM MQ 옵션이 있습니다.

## 클라이언트에서 Advanced Message Security 사용 안함

IBM MQ 클라이언트를 사용하여 이전 버전의 제품에서 큐 관리자에 연결하고 2085 (MQRC_UNKNOWN_OBJECT_NAME) 오류가 보고되는 경우 IBM MQ Advanced Message Security (AMS) 를 사용 안함으로 설정해야 합니다.

### 이 태스크 정보

IBM MQ Advanced Message Security (AMS) 는 IBM MQ 클라이언트에서 자동으로 사용으로 설정되므로 기본적으로 클라이언트는 큐 관리자에서 오브젝트에 대한 보안 정책을 검사하려고 시도합니다.

이 오류가 보고되면 제품의 이전 버전에서 큐 관리자에 연결하려고 할 때 다음과 같이 AMS를 사용 안함으로 설정할 수 있습니다.

- Java 클라이언트의 경우, 다음 방법 중 하나를 사용하십시오.
  - 환경 변수 **AMQ_DISABLE_CLIENT_AMS**를 설정합니다.
  - Java 시스템 특성 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 설정.
  - **DisableClientAMS** 특성을 사용하여 `mqclient.ini` 파일의 보안 스탠자 아래에서.
- C 클라이언트의 경우 환경 변수 **MQS_DISABLE_ALL_INTERCEPT**를 설정합니다.

**참고:** C 클라이언트에는 **AMQ_DISABLE_CLIENT_AMS** 환경 변수를 사용할 수 없습니다. 대신 **MQS_DISABLE_ALL_INTERCEPT** 환경 변수를 사용해야 합니다.

## 프로시저

- 클라이언트에서 AMS를 사용 안함으로 설정하려면, 다음 옵션 중 하나를 사용하십시오.

  **AMQ_DISABLE_CLIENT_AMS 환경 변수**
  다음의 경우에 이 변수를 설정해야 합니다.

  - IBM Java runtime environment (JRE) 이외의 Java runtime environment (JRE) 를 사용하는 경우
  - IBM MQ IBM MQ classes for JMS 또는 IBM MQ classes for Java 클라이언트를 사용하는 경우.

  **AMQ_DISABLE_CLIENT_AMS** 환경 변수를 작성하고 애플리케이션이 실행 중인 환경에서 TRUE 로 설정하십시오. 예를 들면, 다음과 같습니다.

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  **Java 시스템 특성 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS**
  IBM MQ classes for JMS 및 IBM MQ classes for Java 클라이언트의 경우 Java 애플리케이션에 대해 Java 시스템 특성 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 를 TRUE 값으로 설정할 수 있습니다.

  예를 들어, Java 명령이 호출될 때 Java 시스템 특성을 -D 옵션으로 설정할 수 있습니다.

  ```
  JM 3.0 java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/
  java/lib/com.ibm.mq.jakarta.client.jar  my.java.applicationClass
  ```

  ```
  JMS 2.0 java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/
  java/lib/com.ibm.mq.allclient.jar  my.java.applicationClass
  ```

  또는 애플리케이션이 이 파일을 사용하는 경우 JMS 구성 파일 jms.config 내에 Java 시스템 특성을 지정할 수 있습니다.

  **MQS_DISABLE_ALL_INTERCEPT 환경 변수**
  원시 클라이언트에서 IBM MQ 를 사용 중이고 클라이언트에서 AMS 를 사용 안함으로 설정해야 하는 경우 이 환경 변수를 설정해야 합니다.

  환경 변수 **MQS_DISABLE_ALL_INTERCEPT** 를 작성하고 클라이언트가 실행 중인 환경에서 TRUE 로 설정하십시오. 예를 들면, 다음과 같습니다.

  ```
  export MQS_DISABLE_ALL_INTERCEPT =TRUE
  ```

  C 클라이언트의 경우에만 **MQS_DISABLE_ALL_INTERCEPT** 환경 변수를 사용할 수 있습니다. Java 클라이언트의 경우 **AMQ_DISABLE_CLIENT_AMS** 환경 변수를 대신 사용해야 합니다.

  **mqclient.ini 파일의 DisableClientAMS 특성**
  이 옵션은 IBM MQ classes for JMS 및 IBM MQ classes for Java 클라이언트 그리고 C 클라이언트에 사용할 수 있습니다.

  다음 예제에서 표시된 대로 mqclient.ini 파일의 **Security** 스탠자 아래에 특성 이름 DisableClientAMS를 추가하십시오.

  ```
  Security:
  DisableClientAMS=Yes
  ```

  다음 예제에 표시된 대로 AMS를 사용으로 설정할 수도 있습니다.

  ```
  Security:
  DisableClientAMS=No
  ```

## 다음에 수행할 작업

AMS 보호 큐를 여는 문제점에 대한 자세한 정보는 JMS와 함께 AMS 를 사용할 때 보호 큐를 여는 문제점을 참조하십시오.

**관련 개념**
605 페이지의 『메시지 채널 에이전트(MCA) 인터셉션 및 AMS』

MCA 인터셉션 기능을 사용하면 IBM MQ에서 실행 중인 큐 관리자가 서버 연결 채널에 선택적으로 정책을 적용할 수 있습니다.

**관련 태스크**

IBM MQ MQI client 구성 파일, `mqclient.ini`

**관련 참조**

IBM MQ classes for JMS 구성 파일

## AMS의 인증서 요구사항

Advanced Message Security에서 사용하려면 인증서에 RSA 공개 키가 있어야 합니다.

여러 공개 키 유형 및 공개 키를 작성하는 방법에 대한 자세한 정보는 43 페이지의 『IBM MQ에서 디지털 인증서와 CipherSpec의 호환성』의 내용을 참조하십시오.

### 키 사용 확장

키 사용 확장은 인증서를 사용할 수 있는 방법에 추가 제한사항을 둡니다.

Advanced Message Security에서, X.509 v3 인증서의 키 사용은 RFC 5280 스펙에 따라 설정되어야 합니다.

QoP(Quality of Protection) 무결성을 위해서는, 인증서 키 사용 확장이 설정된 경우 이 설정이 다음 두 항목 중 하나 이상을 포함해야 합니다.

- **nonRepudiation**
- **digitalSignature**

QoP(Quality of Protection) 개인정보 보호를 위해서는, 인증서 키 사용 확장이 설정된 경우 이 설정이 다음 항목을 포함해야 합니다.

- **keyEncipherment**

QoP(Quality of Protection) 기밀성을 위해서는, 인증서 키 사용 확장이 설정된 경우 이 설정이 다음 항목을 포함해야 합니다.

- **dataEncipherment**

확장 키 사용은 키 사용 확장을 좀 더 세분화합니다. 모든 QoP(Quality of Protection)를 위해서는, 인증서 확장 키 사용이 설정된 경우 이 설정이 다음 항목을 포함해야 합니다.

- **emailProtection**

**관련 개념**

629 페이지의 『AMS의 보호 품질』

Advanced Message Security 데이터 보호 정책은 QoP(Quality of Protection)를 의미합니다.

## AMS에서 인증서 유효성 검증 메소드

Advanced Message Security를 사용하여 큐의 메시지가 보안 표준을 충족하지 않는 인증서를 사용하여 보호되지 않도록 폐기된 인증서를 감지하고 거부할 수 있습니다.

AMS를 사용하면 온라인 인증서 상태 프로토콜(OCSP) 또는 인증서 폐기 목록(CRL)을 사용하여 인증서 유효성을 확인할 수 있습니다.

AMS는 OCSP 또는 CRL 검사 또는 둘 모두에 대해 구성될 수 있습니다. 두 메소드 모두가 사용으로 설정되면 성능상의 이유로 AMS는 먼저 폐기 상태를 위해 OCSP를 사용합니다. 인증서의 폐기 상태가 OCSP 검사 후에도 미해결이면 AMS는 CRL 검사를 사용합니다.

참고로, 기본적으로 OCSP 및 CRL 검사를 모두 사용할 수 있습니다.

**관련 개념**

611 페이지의 『AMS의 온라인 인증서 상태 프로토콜(OCSP)』

온라인 인증서 상태 프로토콜(OCSP)은 인증서가 폐기되었는지 여부를 판별하므로 인증서를 신뢰할 수 있는지 여부를 판별하는 데 도움을 줍니다. OCSP는 기본적으로 사용됩니다.

613 페이지의 『AMS의 인증서 폐기 목록(CRL)』
CRL은 예를 들어, 개인 키가 분실되었거나 손상되었다는 등의 다양한 이유로 더 이상 신뢰되지 않으므로 인증 기관(CA)에 의해 표시된 인증서 목록을 보유합니다.

## *AMS*의 온라인 인증서 상태 프로토콜*(OCSP)*

온라인 인증서 상태 프로토콜(OCSP)은 인증서가 폐기되었는지 여부를 판별하므로 인증서를 신뢰할 수 있는지 여부를 판별하는 데 도움을 줍니다. OCSP는 기본적으로 사용됩니다.

OCSP는 IBM i 시스템에서는 지원되지 않습니다.

*Advanced Message Security*의 기본 인터셉터에서 *OCSP* 검사 사용

Advanced Message Security의 OCSP(Online Certificate Status Protocol) 검사는 사용되는 인증서의 정보를 기반으로 하여 기본적으로 사용됩니다.

### 프로시저

키 저장소 구성 파일에 다음 옵션을 추가하십시오.

**참고:** 모든 OCSP 스탠자는 선택사항이고 독립적으로 지정될 수 있습니다.

| 옵션 | 설명 |
|---|---|
| `ocsp.enable=off` | 검사 중인 인증서에 OCSP 응답자가 있는 URI를 포함하는 PKIX_AD_OCSP 액세스 메소드와 함께 권한 정보 액세스(AIA) 확장이 있는 경우 OCSP 검사를 사용으로 설정하십시오.<br>가능한 값: on 또는 `off`. |
| `ocsp.url=`*`responder_URL`* | OCSP 응답자의 URL 주소입니다. 이 옵션이 생략되면 비AIA OCSP 검사가 사용 안함으로 설정됩니다. |
| `ocsp.http.proxy.host=`*`OCSP_proxy`* | OCSP 프록시 서버의 URL 주소입니다. 이 옵션이 생략되면 프록시는 비AIA 온라인 인증서 검사에 사용되지 않습니다. |
| `ocsp.http.proxy.port=`*`port_number`* | OCSP 프록시 서버의 포트 번호입니다. 이 옵션이 생략되면 기본 포트 8080이 사용됩니다. |
| `ocsp.nonce.generation=on/off` | OCSP를 조회할 때 임시값을 생성합니다.<br>기본값은 `off`입니다. |
| `ocsp.nonce.check=on/off` | OCSP로부터 응답을 받은 후 임시값을 검사합니다.<br>기본값은 `off`입니다. |
| `ocsp.nonce.size=8` | 임시값 크기(바이트)입니다. |
| `ocsp.http.get=on/off` | 요청 메소드로서 HTTP GET을 지정합니다. 이 옵션이 `off`로 설정되면 HTTP POST가 사용됩니다. 기본값은 `off`입니다. |
| `ocsp.max_response_size=20480` | 제공되는 OCSP 응답자로부터의 최대 응답 크기(바이트)입니다. |
| `ocsp.cache_size=100` | 내부 OCSP 응답 캐싱을 사용으로 설정하고 캐시 항목 수에 대한 한계를 설정합니다. |
| `ocsp.timeout=30` | Advanced Message Security 시간 종료 후 서버 응답을 기다리는 시간(초)입니다. |

| 옵션 | 설명 |
| --- | --- |
| ocsp.unknown=ACCEPT | OCSP 서버가 제한시간 기간 내에 도달할 수 없을 때의 동작을 정의합니다. 가능한 값:<br>• ACCEPT 인증서를 허용합니다.<br>• WARN 인증서를 허용하고 경고를 로그합니다.<br>• REJECT 인증서가 사용되는 것을 막고 오류를 로그합니다. |

*AMS* 의 *Java* 에서 *OCSP* 검사 사용
Advanced Message Security에서 Java에 대한 OCSP 검사를 사용으로 설정하려면 java.security 파일 또는 키 저장소 구성 파일을 수정하십시오.

## 이 태스크 정보
Advanced Message Security에서 OCSP 검사를 사용으로 설정하는 데에는 두 가지 방법이 있습니다.

*java.security* 사용
인증서에 권한 정보 액세스(AIA) 인증서 확장자가 포함되는지 여부를 확인하십시오.

## 프로시저
1. AIA가 설정되지 않았거나 인증서를 대체하려는 경우 다음 특성으로 $JAVA_HOME/lib/security/java.security 파일을 편집하십시오.

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   그리고 $JAVA_HOME/lib/security/java.security 파일을 다음 행으로 편집하여 OCSP 검사를 사용으로 설정하십시오.

   ```
   ocsp.enable=true
   ```

2. AIA가 설정된 경우 $JAVA_HOME/lib/security/java.security 파일을 다음 행으로 편집하여 OCSP 검사를 사용으로 설정하십시오.

   ```
   ocsp.enable=true
   ```

## 다음에 수행할 작업
Java 보안 관리자를 사용 중인 경우에도 구성을 완료하고 다음 Java 권한을 lib/security/java.policy에 추가하십시오.

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

*keystore.conf* 사용

## 프로시저
다음 속성을 구성 파일에 추가하십시오.

```
ocsp.enable=true
```

**중요사항:** 구성 파일에서 이 속성을 설정하면 java.security 설정이 대체됩니다.

## 다음에 수행할 작업

구성을 완료하려면 다음 Java 권한을 `lib/security/java.policy`에 추가하십시오.

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

## *AMS*의 인증서 폐기 목록*(CRL)*

CRL은 예를 들어, 개인 키가 분실되었거나 손상되었다는 등의 다양한 이유로 더 이상 신뢰되지 않으므로 인증 기관(CA)에 의해 표시된 인증서 목록을 보유합니다.

인증서를 유효성 검증하기 위해 Advanced Message Security는 서명자의 인증서 및 신뢰 앵커까지 인증 기관(CA)의 인증서 체인으로 구성된 인증서 체인을 구성합니다. 신뢰 앵커는 인증서의 신뢰를 주장하는 데 사용되는 신뢰되는 인증서 또는 신뢰되는 루트 인증서를 포함하는 신뢰되는 키 저장소 파일입니다. AMS는 PKIX 유효성 검증 알고리즘을 사용하여 인증서 경로를 확인합니다. 체인이 작성되고 확인되면, AMS는 인증서 유효성 검사를 완료합니다. 여기에는 체인에서 각 인증서의 발행 및 만기 날짜를 현재 날짜와 비교하여 유효성 검증하고, 키 사용법 확장이 종료 엔티티 인증서에 있는지 여부에 대한 검사가 포함됩니다. 확장이 인증서에 추가된 경우에는 AMS는 **digitalSignature** 또는 **nonRepudiation** 또한 설정되었는지 여부를 확인합니다. 이들이 설정되지 않은 경우에는 `MQRC_SECURITY_ERROR`가 보고되고 로그됩니다. 그런 다음 AMS는 구성 파일에 지정된 내용에 따라서 파일 또는 LDAP에서 CRL을 다운로드합니다. AMS에서는 DER 형식으로 인코딩된 CRL만이 지원됩니다. 키 저장소 구성 파일에 CRL 관련 정보를 찾을 수 없는 경우 AMS는 CRL 유효성 검증 검사를 수행하지 않습니다. 각 CA 인증서의 경우 AMS는 해당 CRL을 찾기 위해 CA의 식별 이름을 사용하여 LDAP에서 CRL을 조회합니다. 다음 속성이 LDAP 조회에 포함됩니다.

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

**참고:** `deltaRevocationList`는 분배 지점으로 지정된 경우에만 지원됩니다.

고유 인터셉터에서 인증서 유효성 검증 및 인증서 폐기 목록 지원 사용으로 설정
Advanced Message Security가 LDAP( Lightweight Directory Access Protocol) 서버로부터 CRL을 다운로드할 수 있도록 키 저장소 구성 파일을 수정해야 합니다.

## 이 태스크 정보

IBM i 기본 인터셉터에서 인증서 유효성 검증 및 인증서 폐기 목록 지원을 사용으로 설정하는 것은 IBM i 의 Advanced Message Security 에 대해 지원되지 않습니다.

## 프로시저

다음 옵션을 구성 파일에 추가하십시오.

**참고:** 모든 CRL 스탠자는 선택사항이고 독립적으로 지정될 수 있습니다.

| 옵션 | 설명 |
|---|---|
| `crl.ldap.host=`*`host_name`* | LDAP 서버 호스트 이름 |
| `crl.ldap.port=`*`port_number`* | LDAP 서버 포트 번호.<br>최대 11개의 서버를 지정할 수 있습니다. 다중 LDAP 호스트는 LDAP 연결 실패 시에 투명한 장애 복구를 보증하는 데 사용됩니다. 모든 LDAP 서버가 복제본이고 동일 데이터를 포함할 것으로 예상됩니다. AMS Java 인터셉터가 LDAP 서버에 성공적으로 연결하면 이는 제공된 나머지 서버에서 CRL을 다운로드하려고 시도하지 않습니다. |

| 옵션 | 설명 |
| --- | --- |
| `crl.cdp=off` | 이 옵션을 사용하여 인증서에서 CRLDistributionPoints 확장자를 검사하거나 사용하십시오. |
| `crl.ldap.version=3` | LDAP 프로토콜 버전 번호. 가능한 값: 2 또는 3. |
| `crl.ldap.user=cn=`*`username`* | LDAP 서버에 로그인하십시오. 이 값이 지정되지 않으면 LDAP의 CRL은 읽기 쉬워야 합니다. |
| `crl.ldap.pass=`*`password`* | LDAP 서버에 대한 비밀번호. |
| `crl.ldap.encrypted=`*`no/yes`* | `clr.ldap.pass`의 암호화 여부입니다. 자세한 정보는 AMS 구성 파일에서 비밀번호 보호를 참조하십시오. |
| `crl.ldap.cache_lifetime=0` | LDAP 캐시 수명(초 단위). 가능한 값: 0-86400. |
| `crl.ldap.cache_size=50` | LDAP 캐시 크기. 이 옵션은 `crl.ldap.cache_lifetime` 값이 `0`보다 큰 경우에만 지정할 수 있습니다. |
| `crl.http.proxy.host=some.host.com` | CDP CRL 검색을 위한 Http 프록시 서버 포트. |
| `crl.http.proxy.port=8080` | Http 프록시 서버 포트 번호. |
| `crl.http.max_response_size=204800` | IBM Global Security Kit (GSKit)에서 허용하는 HTTP 서버에서 검색할 수 있는 CRL의 최대 크기 (바이트) 입니다. |
| `crl.http.timeout=30` | AMS 시간 종료 후 서버 응답을 기다리는 시간(초)입니다. |
| `crl.http.cache_size=0` | HTTP 캐시 크기(바이트 단위). |
| `crl.unknown=ACCEPT` | CRL 서버가 제한시간 기간 내에 도달할 수 없을 때의 동작을 정의합니다. 가능한 값:<br>• `ACCEPT` 인증서를 허용합니다.<br>• `WARN` 인증서를 허용하고 경고를 로그합니다.<br>• `REJECT` 인증서가 사용되는 것을 막고 오류를 로그합니다. |

*AMS* 의 *Java* 에서 인증서 폐기 목록 지원 사용
Advanced Message Security에서 CRL 지원을 사용으로 설정하려면 AMS가 LDAP(Lightweight Directory Access Protocol) 서버에서 CRL을 다운로드하고 java.security 파일을 구성할 수 있도록 키 저장소 구성 파일을 수정해야 합니다.

## 프로시저

1. 다음 옵션을 구성 파일에 추가하십시오.

| 헤더 | 설명 |
| --- | --- |
| `crl.ldap.host=`*`host_name`* | LDAP 호스트 이름. |

| 헤더 | 설명 |
| --- | --- |
| crl.ldap.port=*port_number* | LDAP 서버 포트 번호.<br>최대 11개의 서버를 지정할 수 있습니다. 다중 LDAP 호스트는 LDAP 연결 실패 시에 투명한 장애 복구를 보증하는 데 사용됩니다. 모든 LDAP 서버가 복제본이고 동일 데이터를 포함할 것으로 예상됩니다. AMS Java 인터셉터가 LDAP 서버에 성공적으로 연결하면 이는 제공된 나머지 서버에서 CRL을 다운로드하려고 시도하지 않습니다.<br>Java는 crl.ldap.user 및 crl.ldaworldp.pass 값을 사용하지 않습니다. 이는 LDAP 서버에 연결할 때 사용자 및 비밀번호를 사용하지 않습니다. 따라서 LDAP의 CRL 속성은 읽기 쉬워야 합니다. |
| crl.cdp=on/off | 이 옵션을 사용하여 인증서에서 CRLDistributionPoints 확장자를 검사하거나 사용하십시오. |

2. 다음 특성으로 JRE/lib/security/java.security 파일을 수정하십시오.

| 특성 이름 | 설명 |
| --- | --- |
| com.ibm.security.enableCRLDP | 이 특성은 다음 값 true, false를 사용합니다.<br>true로 설정되면 인증서 폐기 검사를 수행할 때 CRL은 인증서의 CRL 분배 지점 확장으로부터 URL을 사용하여 찾습니다.<br>false로 설정되거나 설정되지 않으면 CRL 분배 지점 확장을 사용하는 CRL 검사는 사용 불가능으로 설정됩니다. |
| ibm.security.certpath.ldap.cache.lifetime | 이 특성은 LDAP CertStore의 메모리 캐시에서 항목의 수명을 초 단위의 값으로 설정하는 데 사용할 수 있습니다. 값 0은 캐시를 사용 불가능으로 설정하고 1은 무제한의 수명을 의미합니다. 설정되지 않으면 기본 수명은 30초입니다. |
| com.ibm.security.enableAIAEXT | 이 특성은 다음 값 true, false를 사용합니다.<br>true로 설정되면 빌드 중인 인증서 경로의 인증서 내에서 발견된 권한 정보 액세스 확장은 LDAP URI를 포함하는지 여부를 판별하기 위해 검사됩니다. 발견된 각 LDAP URI에 대해 LDAPCertStore 오브젝트가 작성되고 인증서 경로를 빌드하는 데 필요한 다른 인증서를 찾는 데 사용되는 CertStore의 콜렉션에 추가됩니다.<br>false로 설정되거나 설정되지 않으면 추가 LDAPCertStore 오브젝트가 작성되지 않습니다. |

z/OS *z/OS*에서 인증서 폐기 목록*(CRL)* 사용으로 설정

Advanced Message Security는 데이터 메시지를 보호하는 데 사용되는 디지털 인증서의 인증서 폐기 목록(CRL) 검사를 지원합니다.

## 이 태스크 정보

사용 가능으로 설정되면, Advanced Message Security는 메시지가 개인정보 보호된 큐에 배치될 때 수신인 인증서를 유효성 검증하고 메시지가 보호된 큐로부터 검색될 때(무결성 또는 개인정보) 송신자 인증서를 유효성 검증합니다. 이 경우 유효성 검증에는 관련 인증서가 관련 CRL에 등록되지 않았다는 확인을 포함합니다.

Advanced Message Security는 IBM 시스템 SSL 서비스를 사용하여 송신자 및 수신인 인증서를 유효성 검증합니다. z/OS Cryptographic Services System Secure Sockets Layer Programming 매뉴얼에서 시스템 SSL 인증서 유효성 검증과 관련된 자세한 문서를 찾을 수 있습니다.

사용 가능해진 CRL 검사에 AMS 주소 공간의 시작된 태스크 JCL에서 CRLFILE DD를 통해 CRL 구성 파일의 위치를 지정합니다. 사용자 정의할 수 있는 샘플 CRL 구성 파일은 *thlqual*.SCSQPROC(CSQ40CRL)에 제공됩니다. 이 파일에 허용된 설정은 다음과 같습니다.

표 *105. Advanced Message Security CRL* 구성 변수

| **변수** | **올바른 값** | **설명** |
|---|---|---|
| crl.ldap.host[.n] | *hostname -or- hostname:port* | 발행인 인증서의 CRL을 호스트하는 LDAP 서버의 ipaddr/hostname입니다. LDAP 서버의 포트 번호를 지정하지 않는 경우에는 crl.ldap.port에 의해 지정된 포트 번호가 사용됩니다. |
| crl.ldap.port | 포트 | LDAP 서버의 TCP/IP 포트 번호입니다. |
| crl.ldap.user | *ldap_user* | LDAP 서버에 연결할 때 사용할 LDAP 사용자 이름입니다. |
| crl.ldap.pass | *ldap_password* | crl.ldap.user와 연관된 LDAP 비밀번호입니다. |

다음과 같이 여러 LDAP 서버 호스트 이름 및 포트를 지정할 수 있습니다.

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

최대 10개의 호스트 이름을 지정할 수 있습니다. LDAP 서버의 포트 번호를 지정하지 않는 경우에는 crl.ldap.port에 의해 지정된 포트 번호가 사용됩니다. 각 LDAP 서버는 액세스를 위해 동일한 crl.ldap.user/password 결합을 사용해야 합니다.

CRLFILE DD가 지정되면 구성은 Advanced Message Security 주소 공간의 초기화 동안에 로드되고 CRL 검사가 사용 가능으로 설정됩니다. CRLFILE DD가 지정되지 않거나 CRL 구성 파일을 사용할 수 없거나 올바르지 않은 경우 CRL 검사가 사용 안함으로 설정됩니다.

AMS는 IBM 시스템 SSL 인증서 유효성 검증 서비스를 사용하여 다음과 같이 CRL 검사를 수행합니다.

표 *106. Advanced Message Security CRL* 검사

| **Operation** | **보호 품질** | **검사한 인증서** |
|---|---|---|
| PUT | 개인정보 | 수신자 |
| GET | 무결성/개인정보 | 송신자 |

메시지 조작이 CRL은 검사에 실패하면 Advanced Message Security는 다음 조치를 수행합니다.

| 표 *107. Advanced Message Security CRL* 검사 실패 동작 | |
|---|---|
| **Operation** | **CRL 검사 실패** |
| PUT | 메시지가 대상 큐에 놓이지 않습니다. MQCC_FAILED의 완료 코드 및 MQRC_SECURITY_ERROR의 이유 코드가 애플리케이션으로 돌아갑니다. |
| GET | 메시지가 대상 큐에서 제거되고 시스템 보호 오류 큐로 이동됩니다. MQCC_FAILED의 완료 코드 및 MQRC_SECURITY_ERROR의 이유 코드가 애플리케이션으로 돌아갑니다. |

z/OS 용 AMS 는 IBM 시스템 SSL 서비스를 사용하여 CRL및 신뢰 검사를 포함하는 인증서의 유효성을 검증합니다.

IBM MQ 는 인증서 유효성 검증에서 LDAP 서버에 접속할 수 있어야 하지만 CRL을 정의할 필요가 없는 보안 설정을 사용합니다.

**참고:** 관련 LDAP 서비스가 사용 가능한지 확인하고 관련 인증 기관을 위해 CRL 항목을 유지보수하는 것은 관리자의 책임입니다.

## 구성 파일에 대한 AMS 비밀번호 보호 설정

일반 텍스트로 키 저장소 및 개인 키 비밀번호를 저장하면 보안 위험에 노출되므로, Advanced Message Security에서는 사용자 키를 사용하여 이러한 비밀번호를 스크램블할 수 있는 도구를 제공합니다.

### 시작하기 전에

`keystore.conf` 파일 소유자는 파일 소유자에게만 파일을 읽고 쓸 수 있는 권한이 있는지 확인해야 합니다. 이 주제에서 설명된 비밀번호 보호는 추가 보호 수단일 뿐입니다. 또한 보안 시스템에서 이 프로시저를 수행해야 합니다.

구성 파일을 읽을 AMS 클라이언트 유형에 대해 올바른 **runamscred** 변형을 사용하는지 확인하십시오. AMS 클라이언트에 따라 다음이 해당합니다.

- Java 클라이언트의 경우 Java **runamscred** 명령을 사용해야 합니다. 이 명령은 `<IBM MQ installation root>/java/bin`에 있습니다.
- MQI 클라이언트의 경우 `<IBM MQ installation root>/bin` 에 있는 MQI **runmqascred** 명령을 사용해야 합니다.

### 프로시저

1. 보호해야 하는 비밀번호를 비롯한 모든 필수 정보를 포함하도록 `keystore.conf` 파일을 편집하십시오.

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

2. `keystore.conf` 파일을 보호하는 사용자가 액세스할 수 있는 파일 내에 비밀번호를 암호화하기 위한 암호화 키를 배치하십시오.

   이 키는 나중에 AMS 클라이언트에서 사용할 키와 같아야 합니다.

```
ThisIsAnExampleEncryptionKey
```

3. **runamscred** 명령을 실행하여 암호화 키 파일을 제공하는 `keystore.conf` 파일을 보호하십시오.

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. `keystore.conf` 파일이 보호되었고 암호화된 비밀번호를 포함하는지 확인하십시오.

**예**

다음 예제는 보호된 `keystore.conf` 파일의 다음과 같은 내용을 보여줍니다.

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = yes
jceks.keystore_pass =
<AMS>1!62K/a4RinT+bks4RjFWx4A==!Vhi/RjIN2FH5qStUJ/0hsgKyn2IdMuhanemRRDrJq
HM=
jceks.key_pass =
<AMS>1!qmnxY++rsOUtZfDSgwcR1g==!VmWVREdVkNp1xYJstvuW64ph5vxxf7SPoqtsXxYh2
Tk=
jceks.provider = IBMJCE
```

**관련 정보**

runamscred: protect AMS 키워드

# z/OS Using certificates with AMS on z/OS

## About this task

Advanced Message Security implements three levels of protection: integrity, confidentiality, and privacy.

With an integrity policy, messages are signed using the private key of the originator (the application doing the MQPUT). Integrity provides detection of message modification, but the message text itself is not encrypted.

With a confidentiality policy, the message is encrypted when it is put to the queue. The message is encrypted using a symmetric key and an algorithm specified in the relevant Advanced Message Security policy. The symmetric key itself is encrypted with the public key of each recipient (the application doing the MQGET). Public keys are associated with certificates stored in key rings.

With a privacy policy, messages are both signed and encrypted.

When a message that is protected with privacy is dequeued by a recipient application doing an MQGET, the message must be decrypted. Because it was encrypted using the recipient's public key, it must be decrypted using the recipient's private key found in a key ring.

### z/OS *Use of SAF key rings with AMS on z/OS*

Advanced Message Security (AMS) makes use of z/OS SAF key ring services to define and manage the certificates needed for signing and encryption. Security products that are functionally equivalent to RACF may be used instead of RACF if they provide the same level of support.

Efficient use of key rings can reduce the administration needed to manage the certificates.

After a certificate is generated (or imported), it must be connected to a key ring to become accessible. The same certificate can be connected to more than one key ring.

Advanced Message Security uses two sets of key rings. One set consists of key rings owned by the individual user IDs that originate or receive messages. Each key ring contains the private key associated with the certificate of the owning user ID. The private key of each certificate is used to sign messages for integrity protected or privacy protected queues. It is also used to decrypt messages from privacy protected or confidentiality protected queues when receiving messages.

The other set is a single key ring owned by the AMS address space user. It contains the chain of signing CA certificates necessary to validate the certificates of the message originator and recipients.

When privacy or confidentiality protection is used, the key ring owned by the AMS address space user also contains the certificates of the message recipients. The public keys in these certificates are used to encrypt the symmetric key that was used to encrypt the message data when the message was put to the protected queue. When these messages are retrieved, the private key of relevant recipients is used to decrypt the symmetric key which is then used to decrypt the message data.

Advanced Message Security uses a key ring name of **drq.ams.keyring** when searching for certificates and private keys. This is the case for both the user and the AMS address space key rings.

For an illustration and further explanation of certificates and key ring, and their role in data protection, refer to Summary of the certificate-related operations.

The private key used for signing can have any label but must be connected as the default certificate. The private key or keys used for decryption can have any label, and must be connected to the key ring, but are no longer required to be connected as the default certificate.

Digital certificates and key rings are managed in RACF primarily by using the RACDCERT command.

For more information about certificates, labels, and the RACDCERT command, see the *z/OS: Security Server RACF Command Language Reference* and the *z/OS: Security Server RACF Security Administrator's Guide*.

z/OS *Replacing certificates*

When a certificate is renewed or replaced (for example, when the existing certificate is approaching its expiry date), it is not always possible to remove the protection from existing messages that are already on queues protected by confidentiality or privacy policies.

This can occur when the certificate was:

- Renewed with the same private key, and the reissued certificate has replaced the original certificate
- Re-keyed with a new private key and the RACDCERT ROLLOVER command has deleted the original private key

Messages will be decrypted, provided the necessary certificate is connected to the keyring of the user; it is no longer required to be connected as the default. This allows messages already on the queue, when the new certificate is connected, to be successfully decrypted.

The following example shows how a new certificate can be generated based on the existing certificate:

- A new certificate is created based on the existing certificate, with new public/private key pair.
- The new certificate is signed by the issuing authority.
- The public key of the old certificate is removed from the keyring of the AMS address space, and the public key of the new certificate is added.
- The new certificate and private key is added to the keyring of the user, in addition to the old certificate.

```
RACDCERT ID(user1) REKEY(LABEL('user1'))          -
         WITHLABEL('user1new')

RACDCERT GENREQ(LABEL('user1new')) ID(user1)      -
         DSN(output_data_set_name)

RACDCERT GENCERT(output_data_set_name) ID(user1)  -
         SIGNWITH(CERTAUTH LABEL('AMSCA'))

RACDCERT ID(user1) ALTER (LABEL('user1new'))      -
         TRUST

RACDCERT ID(WMQAMSD) REMOVE(ID(user1)             -
         LABEL('user1')                           -
         RING(drq.ams.keyring)  )

RACDCERT ID(WMQAMSD) CONNECT(ID(user1)            -
         LABEL('user1new') USAGE(SITE)            -
         RING(drq.ams.keyring)  )

RACDCERT ID(user1) CONNECT(ID(user1)              -
         LABEL('user1new') USAGE(PERSONAL)        -
         RING(drq.ams.keyring) DEFAULT )
```

For more information about certificates, labels, and the RACDCERT command, see the *z/OS: Security Server RACF Command Language Reference* and the *z/OS: Security Server RACF Security Administrator's Guide*.

### z/OS *Authorizing access to the RACDCERT command for AMS on z/OS*

Authorization to use the RACDCERT command is a post-installation task that should have been completed by your z/OS system programmer. This task involves granting relevant permissions to the Advanced Message Security security administrator.

As a summary, these commands are needed to allow access to the RACF RACDCERT command:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

In this example, *admin* specifies the user ID of your security administrator, or any user you want to use the RACDCERT command.

### z/OS *Creating the certificates and key rings for AMS users on z/OS*

This section documents the steps required to create the certificates and key rings necessary for z/OS users of Advanced Message Security (AMS), using a RACF Certificate Authority (CA).

## Resolving problems with certificates when using Advanced Message Security on z/OS

If you are having problems with certificates and missing entries in key stores you can enable a IBM Global Security Kit (GSKit) trace.

In the file referenced by the ENVARS DD in the AMS started task procedure, add:

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

See Environment variables for more information.

For every access to the keystore, data is written to the trace file specified in `GSK_TRACE_FILE`.

To format the trace file use the command:

```
gsktrace inputtrace file > output_file
```

## Scenario

A scenario of a sending application and a receiving application is used to explain the required steps.

In the examples that follow, `user1` is the originator of a message and `user2` is the recipient. The user ID of the Advanced Message Security address space is `WMQAMSD`.

All of the commands in the examples shown here are issued from ISPF option 6 by the administrative user ID `admin`.

### z/OS *Defining a local Certificate Authority certificate for AMS on z/OS*

If you are using RACF as your CA, you must create a certificate authority certificate, if you have not already done so. The command shown here creates a certificate authority (or signer) certificate. This example creates a certificate called AMSCA to be used when creating subsequent certificates that reflect the identity of Advanced Message Security users and applications.

This command may be modified, specifically SUBJECTSDN, to reflect the naming structure and conventions used at your installation:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

**Note:** Certificates signed with this local certificate authority certificate show an issuer of CN=AMSCA,O=ibm,C=us when listed with the RACDCERT LIST command.

z/OS *Creating a digital certificate with a private key for AMS on z/OS*

A digital certificate with a private key must be generated for each Advanced Message Security user. In the example shown here, RACDCERT commands are used to generate certificates for user1 and user2, which are signed with the local CA certificate identified by the label AMSCA.

```
RACDCERT ID(user1) GENCERT SUBJECTSDN(CN('user1') O('ibm') C('us'))
WITHLABEL('user1') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user2) GENCERT SUBJECTSDN(CN('user2') O('ibm') C('us'))
WITHLABEL('user2') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user1) ALTER (LABEL('user1')) TRUST
RACDCERT ID(user2) ALTER (LABEL('user2')) TRUST
```

The RACDCERT ALTER command is required to add the TRUST attribute to the certificate. When a certificate is first created using this procedure, it has a different valid date range than the signing certificate. As a result, RACF marks it as NOTRUST, which means that the certificate is not to be used. Use the RACDCERT ALTER command to set the TRUST attribute.

The KEYUSAGE attributes HANDSHAKE, DATAENCRYPT and DOCSIGN must be specified for certificates used by Advanced Message Security.

*Table 108. RACDCERT KEYUSAGE values and indicators*

| **KEYUSAGE Value** | **Indicators Set** |
|---|---|
| HANDSHAKE | digitalSignature and keyEncipherment |
| DATAENCRYPT | dataEncipherment |
| DOCSIGN | nonRepudiation |
| CERTSIGN | keyCertSign and cRLSign |

z/OS *Creating the RACF key rings for AMS on z/OS*

The commands shown here create a key ring for RACF-defined user IDs user1, user2, and the Advanced Message Security address space task user WMQAMSD. The key ring name is fixed by Advanced Message Security and must be coded as shown, without quotes. The name is case-sensitive.

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS *Connecting the certificates to the key rings for AMS on z/OS*

Connect the user and CA certificates to the key rings:

```
RACDCERT ID(WMQAMSD) CONNECT(CERTAUTH LABEL('AMSCA')
RING(drq.ams.keyring))
RACDCERT ID(user1) CONNECT(ID(user1) LABEL('user1')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(user2) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(WMQAMSD) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) USAGE(SITE))
```

Any certificates containing the private key or keys used for decryption must be connected to the key ring of the user, however they are no longer required to be connected as the default certificate.

The RACDCERT USAGE(SITE) attribute prevents the private key from being accessible in the key ring, while the RACDCERT USAGE(PERSONAL) attribute allows the private key to be used, if it exists. User2's certificate must be connected to the Advanced Message Security address space key ring because its public key is needed to encrypt messages as they are put to the queue. USAGE(SITE) limits exposure of user2's private key.

The CERTAUTH certificate with label AMSCA must be connected to the Advanced Message Security address space key ring because it was used to sign the certificate of user1, who is the message originator. It is used to validate user1's signing certificate.

z/OS *Key ring verification for AMS on z/OS*

The key ring should appear as shown here, after all commands have been entered:

```
RACDCERT ID(user1) LISTRING(drq.ams.keyring)
Digital ring information for user USER1:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user1                          ID(USER1)    PERSONAL YES

RACDCERT ID(user2) LISTRING(drq.ams.keyring)
Digital ring information for user USER2:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user2                          ID(USER2)    PERSONAL YES

RACDCERT ID(WMQAMSD) LISTRING(drq.ams.keyring)
Digital ring information for user WMQAMSD:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
AMSCA                          CERTAUTH     CERTAUTH NO
user2                          ID(USER2)    SITE     NO
```

Listing the individual certificates also shows the ring association.

```
RACDCERT ID(user2) LIST(label('user2'))
Digital certificate information for user USER2:

***
Label: user2
Certificate ID: 2QfH8Pny9/LzpKKFmfFA
Status: TRUST
Start Date: 2010/05/03 22:59:53
End Date:  2011/05/04 22:59:52
Serial Number:>15<:
Issuer's Name:>OU=AMSCA.O=ibm.C=us<:
Subject's Name:>CN=user2.O=ibm.C=us<:
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: USER2
Ring:>drq.ams.keyring<:
Ring Owner: WMQAMSD
Ring:>drq.ams.keyring<:
```

To improve performance, the contents of the drq.ams.keyring associated with the AMS address space is cached for the life of the address space. Changes to that key ring do not become effective automatically. The administrator can refresh the cache by either:

- Stopping and restarting the queue manager.

- Using the z/OS MODIFY command:

```
F qmgrAMSM,REFRESH KEYRING
```

**Related tasks**

Operating Advanced Message Security

## z/OS *Summary of the certificate-related operations for AMS on z/OS*

Figure 35 on page 623 illustrates the relationships between sending and receiving applications and relevant certificates. The scenario illustrated involves remote queuing between two z/OS queue managers using a data-protection policy of privacy. In Figure 35 on page 623, "AMS" indicates " Advanced Message Security".

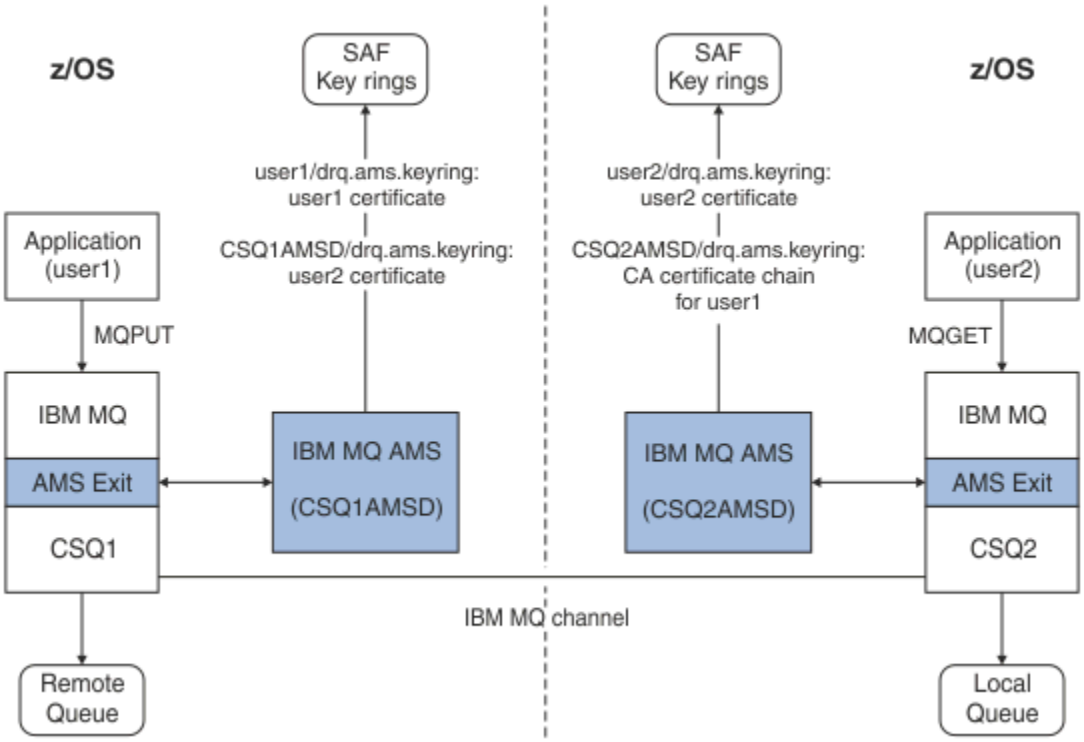



*Figure 35. Application and certificate relationships*

In this diagram, an application running as 'user1' puts a message to a remote queue managed by queue manager CSQ1, intended to be retrieved by an application running as 'user2' from a local queue managed by queue manager CSQ2. The diagram assumes an Advanced Message Security policy of privacy, which means the message is both signed and encrypted.

Advanced Message Security intercepts the message when a put occurs and uses user2's certificate (stored in the AMS address space user's key ring) to encrypt a symmetric key used to encrypt the message data.

Note that user2's certificate is connected to the AMS address space user key ring with option USAGE(SITE). This means the AMS address space user can access the certificate and public key, but not the private key.

On the receiving end, Advanced Message Security intercepts the get issued by user2, and uses user2's certificate to decrypt the symmetric key so that it can decrypt the message data. It then validates user1's signature using the CA certificate chain of user1's certificate stored in the AMS address space user's key ring.

Given this scenario, but with a data-protection policy of integrity, certificates for user2 would not be required.

To use Advanced Message Security to enqueue messages on IBM MQ-protected queues having a message protection policy of privacy or integrity, Advanced Message Security must have access to these data items:

- The X.509 V2 or V3 certificate and private key for the user enqueuing the message.
- The chain of certificates used to sign the digital certificates of all message signers.
- If the data protection policy is privacy, the X.509 V2 or V3 certificate of the intended recipients. The intended recipients are listed in the Advanced Message Security policy associated with the queue.

For processes and applications that run on z/OS, Advanced Message Security must have certificates in two places:

- In a SAF-managed key ring associated with the RACF identity of the sending application (the application that enqueues the protected message) or receiving application (if using privacy).

  The certificate that Advanced Message Security locates is the default certificate, and must include the private key. Advanced Message Security assumes the z/OS user identity of the sending application. That is, it acts as a surrogate, so it can access the user's private key.
- In a SAF-managed key ring associated with the AMS address space user.

  When sending messages protected with privacy, this key ring contains the public key certificates of the message recipients. When receiving messages, it contains the chain of Certificate Authority certificates needed to validate the message sender's signature.

The earlier examples shown have used RACF as the local CA. However, you may use another PKI provider (Certificate Authority) at your installation. If you intend to use another PKI product, remember that the private key and the certificate must be imported into a key ring associated with the z/OS RACF user IDs that originate IBM MQ messages protected by Advanced Message Security.

You can use the RACF RACDCERT command as the mechanism to generate certificate requests, which can be exported and sent to the PKI provider of your choice to be issued.

Here is a summary of the certificate-related steps:

1. Request the creation of a CA certificate, one in which RACF is the local CA. Omit this step if you are using another PKI provider.
2. Generate user certificates signed by the CA.
3. Create the key rings for the users and the Advanced Message Security AMS address space ID.
4. Connect the user certificate to the user key ring with the default attribute.
5. Connect the recipients certificates to the Advanced Message Security AMS address space user key ring using the usage(site) attribute (This step is necessary only for user certificates that will ultimately be the recipients of privacy-protected messages).
6. Connect the CA certificate chains for message senders to the Advanced Message Security AMS address space user key ring. (This step is necessary only for AMS tasks that will be verifying sender signatures.)

### z/OS *Configuring a non-z/OS resident PKI for AMS*

Advanced Message Security for z/OS, uses X.509 V3 digital certificates in the protection-processing of messages placed on or received from IBM MQ queues. Advanced Message Security itself does not create or manage the life cycle of these certificates; that function is provided by a public key infrastructure (PKI). The examples in this publication that illustrate the use of certificates use z/OS Security Server RACF to fill certificate requests.

Whether or not a z/OS or non-z/OS resident PKI is used, AMS for z/OS uses only key rings that are managed by RACF or its equivalent. These key rings are based on Security Authorization Facility (SAF) and are the repository used by AMS for z/OS to retrieve certificates for originators and recipients of messages placed on or received from IBM MQ queues.

For messages that are originated from z/OS, which are protected by either integrity or encryption policy, the certificate and private key of the originating user ID must be stored in an SAF-managed key ring that is associated with the z/OS user ID of the message originator.

RACF includes the capability for importing certificates and private keys into RACF-managed key rings. See the z/OS Security Server RACF publications for the details and examples of how to load certificates to RACF managed key rings.

If your installation is using one of the supported PKI products, refer to the publications that accompany the product for information on how to deploy it.

# Advanced Message Security 보안 정책 관리

Advanced Message Security는 보안 정책을 사용하여 큐를 통해 플로우하는 메시지를 암호화 및 인증하기 위한 암호화 및 서명 알고리즘을 지정합니다.

## AMS에 대한 보안 정책 개요

Advanced Message Security 보안 정책은 메시지가 암호학적으로 암호화되고 서명되는 방법을 설명하는 개념적 오브젝트입니다.

보안 정책 속성의 세부사항은 다음 하위 주제를 참조하십시오.

**관련 개념**

629 페이지의 『AMS의 보호 품질』
Advanced Message Security 데이터 보호 정책은 QoP(Quality of Protection)를 의미합니다.

628 페이지의 『AMS의 보안 정책 속성』
Advanced Message Security를 사용하여 데이터를 보호하기 위한 특정 알고리즘 또는 방법을 선택할 수 있습니다.

### *AMS*의 정책 이름

정책 이름은 특정 Advanced Message Security 정책 및 이를 적용하는 큐를 식별하는 고유 이름입니다.

정책 이름은 정책이 적용되는 큐 이름과 같아야 합니다. Advanced Message Security(AMS) 정책 및 큐 간의 일대일 맵핑이 있습니다.

큐와 이름이 같은 정책을 작성함으로써 해당 큐의 정책을 활성화합니다. 일치하는 정책 이름이 없는 큐는 AMS에 의해 보호되지 않습니다.

정책의 범위는 로컬 큐 관리자 및 해당 큐와 관련됩니다. 리모트 큐 관리자에는 이들이 관리하는 큐에 로컬에 정의된 자체 정책이 있어야 합니다..

### *AMS*에서 서명 알고리즘

서명 알고리즘은 데이터 메시지에 서명할 때 사용되어야 하는 알고리즘을 나타냅니다.

유효한 값은 다음을 포함합니다.

- MD5
- SHA-1
- SHA-2 제품군:
  - SHA256
  - SHA384(허용 가능한 최소 키 길이 - 768비트)
  - SHA512(허용 가능한 최소 키 길이 - 768비트)

서명 알고리즘을 지정하지 않거나 NONE의 알고리즘을 지정하는 정책은 정책과 연관된 큐에 배치된 메시지가 서명되지 않음을 암시합니다.

**참고:** 메시지 넣기 및 가져오기 함수에 사용되는 보호의 품질은 일치해야 합니다. 큐와 큐에 있는 메시지 간의 정책 보호 품질이 불일치하면 메시지는 허용되지 않고 오류 처리 큐로 전송됩니다. 이 규칙은 로컬 및 리모트 큐 둘 모두에 적용됩니다.

### *AMS*에서 암호화 알고리즘

암호화 알고리즘은 암호화 데이터 메시지가 정책과 연관된 큐에 배치될 때 사용되어야 하는 알고리즘을 나타냅니다.

유효한 값은 다음을 포함합니다.

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

암호화 알고리즘을 지정하지 않거나 NONE의 알고리즘을 지정하는 정책은 정책과 연관된 큐에 배치된 메시지가 암호화되지 않음을 암시합니다.

NONE이 아닌 암호화 알고리즘을 지정하는 정책은 또한 하나 이상의 수신인 DN 및 서명 알고리즘을 지정해야 합니다. Advanced Message Security 암호화된 메시지 또한 서명되기 때문입니다.

**중요사항:** 메시지 넣기 및 가져오기 함수에 사용되는 보호의 품질은 일치해야 합니다. 큐와 큐에 있는 메시지 간의 정책 보호 품질이 불일치하면 메시지는 허용되지 않고 오류 처리 큐로 전송됩니다. 이 규칙은 로컬 및 리모트 큐 둘 모두에 적용됩니다.

### *AMS*의 허용

허용 속성은 Advanced Message Security가 보안 정책이 지정되지 않은 메시지를 승인할 수 있는지 여부를 나타냅니다.

메시지를 암호화하기 위한 정책이 있는 큐로부터 메시지를 검색할 때 메시지가 암호화되지 않은 경우에는 이는 호출 애플리케이션으로 돌아갑니다. 유효한 값은 다음을 포함합니다.

**0**
아니오(**기본값** ).

**1**
예

허용 값을 지정하지 않거나 0을 지정하는 정책은 정책과 연관된 큐에 배치되는 메시지가 정책 규칙과 일치해야 함을 암시합니다.

허용은 선택사항이고 정책이 큐에 적용되었지만 이러한 큐에 이미 보안 정책이 지정되지 않음 메시지가 포함되어 있는 경우에 구성 공개를 쉽게 하기 위해 존재합니다.

### *AMS*의 송신자 식별 이름

송신자 식별 이름(DN)은 메시지를 큐에 배치할 수 있는 권한을 가진 사용자를 식별합니다. 송신자는 메시지를 큐에 배치하기 전에 인증서를 사용하여 메시지에 서명합니다.

Advanced Message Security(AMS)는 큐가 검색될 때까지 메시지가 올바른 사용자에 의해 데이터 보호된 큐에 배치되었는지 여부를 검사하지 않습니다. 이때 정책이 하나 이상의 올바른 송신자를 규정하고 큐에 메시지를 배치한 사용자가 올바른 송신자 목록에 없는 경우, AMS 는 수신 애플리케이션에 오류를 리턴하고 메시지를 AMS 오류 큐에 배치합니다.

정책에는 0개 이상의 송신자 DN이 지정될 수 있습니다. 정책에 송신자 DN이 지정되지 않은 경우 송신자는 송신자의 인증서가 신뢰되는 경우 데이터 보호 메시지를 큐에 넣을 수 있습니다. 송신자의 인증서는 수신 애플리케이션에 사용 가능한 키 저장소에 공용 인증서를 추가하여 신뢰할 수 있습니다.

송신자 식별 이름의 형식은 다음과 같습니다.

```
CN=Common Name,O=Organization,C=Country
```

**중요사항:**

- 모든 DN 구성요소 이름은 대문자여야 합니다. DN의 모든 구성요소 이름 ID는 다음 표에 표시된 순서대로 지정되어야 합니다.

| 컴포넌트 이름 | 값 |
|---|---|
| CN | 이 DN의 오브젝트의 공통 이름(예: 전체 이름 또는 디바이스의 의도된 용도). |
| OU | DN의 오브젝트가 가입된 조직 내의 단위(예: 회사 부서 또는 제품 이름). |
| O | DN의 오브젝트가 가입된 조직(예: 회사). |
| L | DN의 오브젝트가 위치한 로컬성(시 또는 지자체). |
| ST | DN의 오브젝트가 위치한 시/도 이름. |
| C | 식별 이름(DN)의 오브젝트가 있는 국가. |

- 하나 이상의 송신자 DN이 정책에 지정된 경우 해당 사용자만 정책과 연관된 큐에 메시지를 배치할 수 있습니다.
- 송신자 DN(지정된 경우)은 메시지를 배치하는 사용자와 연관된 디지털 인증서와 정확하게 일치해야 합니다.
- AMS는 라틴-1 문자 세트의 값만 있는 DN을 지원합니다. 세트의 문자로 DN을 작성하려면 먼저 UTF-8 코딩이 켜진 AIX and Linux 을 사용하여 UTF-8 코딩으로 작성된 DN으로 인증서를 작성해야 합니다. 그런 다음 UTF-8 코딩이 켜진 Linux 또는 AIX 플랫폼에서 정책을 작성하거나 IBM MQ에 AMS 플러그인을 사용해야 합니다.
- AMS가 송신자 이름을 x.509 형식에서 DN 형식으로 변환하기 위해 사용하는 방법은 항상 시/도 값에 ST=를 사용합니다.
- 다음 특수 문자에는 이스케이프 문자가 필요합니다.

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- 식별 이름에 임베드된 공백이 포함된 경우 DN을 큰따옴표로 묶어야 합니다.

**관련 개념**

627 페이지의 『AMS의 수신인 식별 이름』
수신자 식별 이름(DN)은 큐로부터 메시지를 수신할 권한이 있는 사용자를 식별합니다.

## *AMS*의 수신인 식별 이름

수신자 식별 이름(DN)은 큐로부터 메시지를 수신할 권한이 있는 사용자를 식별합니다.

정책에는 수신인 DN이 0개 이상 지정될 수 있습니다. 수신자 식별 이름 다음 양식을 가지고 있습니다.

```
CN=Common Name,O=Organization,C=Country
```

**중요사항:**

- 모든 DN 구성요소 이름은 대문자여야 합니다. DN의 모든 구성요소 이름 ID는 다음 표에 표시된 순서대로 지정되어야 합니다.

| 컴포넌트 이름 | 값 |
|---|---|
| CN | 이 DN의 오브젝트의 공통 이름(예: 전체 이름 또는 디바이스의 의도된 용도). |
| OU | DN의 오브젝트가 가입된 조직 내의 단위(예: 회사 부서 또는 제품 이름). |

| 컴포넌트 이름 | 값 |
|---|---|
| O | DN의 오브젝트가 가입된 조직(예: 회사). |
| L | DN의 오브젝트가 위치한 로컬성(시 또는 지자체). |
| ST | DN의 오브젝트가 위치한 시/도 이름. |
| C | 식별 이름(DN)의 오브젝트가 있는 국가. |

- 정책에 수신인 DN이 지정되지 않으면 어느 사용자나 정책과 연관된 큐에서 메시지를 가져올 수 있습니다.
- 하나 이상의 수신자 DN이 정책에 지정된 경우 해당 사용자만 정책과 연관된 큐에서 메시지를 가져올 수 있습니다.
- 수신자 DN(지정된 경우)은 메시지를 가져오는 사용자와 연관된 디지털 인증서와 정확하게 일치해야 합니다.
- Advanced Message Security는 라틴-1 문자 세트의 값만 있는 DN을 지원합니다. 세트의 문자로 DN을 작성하려면 먼저 UTF-8 코딩이 켜진 AIX 또는 Linux 를 사용하여 UTF-8 코딩으로 작성된 DN으로 인증서를 작성해야 합니다. 그런 다음 UTF-8 코딩이 켜진 Linux 또는 AIX 플랫폼에서 정책을 작성하거나 IBM MQ에 대한 Advanced Message Security 플러그인을 사용해야 합니다.

**관련 개념**

626 페이지의 『AMS의 송신자 식별 이름』

송신자 식별 이름(DN)은 메시지를 큐에 배치할 수 있는 권한을 가진 사용자를 식별합니다. 송신자는 메시지를 큐에 배치하기 전에 인증서를 사용하여 메시지에 서명합니다.

## *AMS*의 보안 정책 속성

Advanced Message Security를 사용하여 데이터를 보호하기 위한 특정 알고리즘 또는 방법을 선택할 수 있습니다.

보안 정책은 메시지가 암호학적으로 암호화되고 서명되는 방법을 설명하는 개념적 오브젝트입니다.

표 *109. AMS*의 보안 정책 속성

| 속성 | 설명 |
|---|---|
| 정책 이름 | 큐 관리자의 정책의 고유 이름. |
| 서명 알고리즘 | 전송하기 전에 메시지를 서명하는 데 사용되는 암호화 알고리즘. |
| 암호화 알고리즘 | 전송하기 전에 메시지를 암호화하는 데 사용되는 암호화 알고리즘. |
| 수신인 목록 | 메시지의 잠재적인 수신자의 인증서 식별 이름(DN) 목록. |
| 서명 DN 체크리스트 | 메시지 검색 중에 유효성을 검증할 서명 DN 목록. |

Advanced Message Security에서 메시지는 대칭 키로 암호화되고, 대칭 키는 수신인의 공개 키로 암호화됩니다. 공개 키는 유효 길이가 최대 2048비트인 키를 사용하여 RSA 알고리즘으로 암호화됩니다. 실제 비대칭 키 암호화는 인증서 키 길이에 따라 다릅니다.

지원되는 대칭-키 알고리즘은 다음과 같습니다.

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Advanced Message Security는 또한 다음과 같은 암호화 해시 함수를 지원합니다.

- Deprecated MD5
- Deprecated SHA-1
- SHA-2 제품군:
  - SHA256
  - SHA384(허용 가능한 최소 키 길이 - 768비트)
  - SHA512(허용 가능한 최소 키 길이 - 768비트)

**참고:** 메시지 넣기 및 가져오기 함수에 사용되는 보호의 품질은 일치해야 합니다. 큐와 큐에 있는 메시지 간의 정책 보호 품질이 불일치하면 메시지는 허용되지 않고 오류 처리 큐로 전송됩니다. 이 규칙은 로컬 및 리모트 큐 둘 모두에 적용됩니다.

### *AMS*의 보호 품질

Advanced Message Security 데이터 보호 정책은 QoP(Quality of Protection)를 의미합니다.

Advanced Message Security의 세 개의 QoP(Quality of Protection) 레벨은 IBM MQ 9.0이상에서 네 번째 레벨에 의해 보충되며 모두 메시지 서명 및 암호화에 사용되는 암호화 알고리즘에 따라 다릅니다.

- 개인정보 보호 - 큐에 배치된 메시지는 서명되고 암호화되어야 합니다.
- 무결성 - 큐에 배치된 메시지는 서명자에 의해 서명되어야 합니다.
- 기밀성 - 큐에 배치된 메시지는 암호화되어야 합니다. 자세한 정보는 558 페이지의 『AMS에서 사용 가능한 QoP(Quality of Protection)』의 내용을 참조하십시오.
- 없음 - 데이터 보호가 적용 가능하지 않습니다.

메시지는 큐에 배치될 때 서명되어야 함을 규정하는 정책에는 INTEGRITY의 QOP가 있습니다. INTEGRITY의 QOP는 정책이 서명 알고리즘을 규정하지만 암호화 알고리즘을 규정하지는 않음을 의미합니다. 무결성 보호된 메시지는 또한 "SIGNED"라고도 불립니다.

메시지는 큐에 배치될 때 서명되고 암호화되어야 함을 규정하는 정책에는 PRIVACY의 QOP가 있습니다. PRIVACY의 QOP는 정책이 서명 알고리즘 및 암호화 알고리즘을 규정하는 경우를 의미합니다. 개인정보 보호된 메시지는 또한 "SEALED"라고도 불립니다.

큐에 배치할 때 메시지를 암호화하도록 규정하는 정책에는 CONFIDENTIALITY의 QOP가 있습니다. 기밀성(CONFIDENTIALITY)의 QOP는 정책이 암호화 알고리즘을 규정함을 의미합니다.

서명 알고리즘 또는 암호화 알고리즘을 규정하지 않는 정책에는 NONE의 QOP가 있습니다. Advanced Message Security에서는 NONE의 QOP가 있는 정책이 있는 큐에 대해 데이터 보호를 제공하지 않습니다.

## AMS의 보안 정책 관리

보안 정책은 메시지가 암호학적으로 암호화되고 서명되는 방법을 설명하는 개념적 오브젝트입니다.

보안 정책과 관련된 모든 관리 태스크를 실행하는 위치는 사용 중인 플랫폼 위치에 따라 다릅니다.

- ALW AIX, Linux, and Windows에서 DELETE POLICY, DISPLAY POLICY 및 SET POLICY(또는 동등한 PCF) 명령을 사용하여 보안 정책을 관리합니다.
  - Linux AIX AIX and Linux에서 관리 태스크는 *MQ_INSTALLATION_PATH*/bin에서 실행될 수 있습니다.
  - Windows Windows 플랫폼에서 관리 태스크는 PATH 환경 변수가 설치 시 업데이트되므로 임의 위치에서 실행될 수 있습니다.
- IBM i IBM i에서 DSPMQMSPL, SETMQMSPL 및 WRKMQMSPL 명령은 IBM MQ가 설치될 때 시스템의 기본 언어의 QSYS 시스템 라이브러리에 설치됩니다.

  추가로 자국어 버전이 언어 피처 로드에 따라 QSYS29xx 라이브러리에 설치됩니다. 예를 들어, 미국 영어를 기본 언어로 사용하고 한국어를 2차 언어로 사용하는 시스템에는 QSYS에 미국 영어 명령이 설치되고, 2962가 한국어의 언어 로드이므로 한국어가 QSYS2962에 로드되어 있습니다.

- z/OS z/OS에서 관리 명령은 메시지 보안 정책 유틸리티(CSQ0UTIL)를 사용하여 실행됩니다. z/OS에서 정책이 작성, 수정 또는 삭제되면 큐 관리자가 중지되고 재시작되거나 z/OS MODIFY 명령을 사용하여 Advanced Message Security 정책 구성을 새로 고칠 때까지 Advanced Message Security 에서 변경사항을 인식하지 못합니다. 예를 들면, 다음과 같습니다.

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

**관련 태스크**

630 페이지의 『AMS에서 보안 정책 작성』
보안 정책은 메시지를 놓일 때 메시지가 보호되는 방법 또는 메시지를 수신할 때 메시지를 보호해야 하는 방법을 정의합니다.

631 페이지의 『AMS에서 보안 정책 변경』
Advanced Message Security를 사용하여 이미 정의한 보안 정책의 세부사항을 변경할 수 있습니다.

631 페이지의 『AMS의 보안 정책 표시 및 덤프』
**dspmqspl** 명령을 사용하여 사용자가 제공하는 명령행 매개변수에 따라 모든 보안 정책 목록 또는 이름 지정된 정책의 세부사항을 표시할 수 있습니다.

633 페이지의 『AMS에서 보안 정책 제거』
Advanced Message Security에서 보안 정책을 제거하려면 `setmqspl` 명령을 사용해야 합니다.

Advanced Message Security 작동

**관련 참조**

메시지 보안 정책 유틸리티(CSQ0UTIL)

## *AMS*에서 보안 정책 작성

보안 정책은 메시지를 놓일 때 메시지가 보호되는 방법 또는 메시지를 수신할 때 메시지를 보호해야 하는 방법을 정의합니다.

### 시작하기 전에

보안 정책을 작성할 때 준수해야 하는 몇 가지 초기 조건이 있습니다.

- 큐 관리자가 실행 중이어야 합니다.
- 보안 정책의 이름은 IBM MQ 오브젝트 이름 지정 규칙을 따라야 합니다.
- 큐 관리자에 연결하고 보안 정책을 작성하기 위해서는 필수 권한이 있어야 합니다:
  - z/OS z/OS에서 메시지 보안 정책 유틸리티(CSQ0UTIL)에 문서화된 권한을 부여하십시오.
  - Multi 멀티플랫폼에서는 **setmqaut** 명령을 사용하여 필요한 `+connect`, `+inq` 및 `+chg` 권한을 부여해야 합니다.

  보안 구성에 대한 자세한 정보는 121 페이지의 『보안 설정』의 내용을 참조하십시오.
- z/OS z/OS에서 필수 시스템 오브젝트가 CSQ4INSM에서 정의에 따라 정의되었는지 확인하십시오.

### 예

다음은 QMGR 큐 관리자에서 정책 작성의 예입니다. 이 정책은 SHA256 알고리즘을 사용하여 메시지에 서명하고 DN: CN=joe, O=IBM, C=US및 DN: CN=jane, O=IBM, C = US인 인증서에 대해 AES256 알고리즘을 사용하여 암호화하도록 지정합니다. 이 정책은 MY.QUEUE에 첨부됩니다.

```
setmqspl -m QMGR -p MY.QUEUE -s SHA256 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

다음은 QMGR 큐 관리자에서 정책 작성의 예입니다. 정책은 DN이 CN=john, O=IBM, C=US및 CN=jeff, O=IBM, C=US인 인증서에 대해 3DES 알고리즘을 사용하여 메시지를 암호화하고 DN이 CN=phil, O=IBM, C=US인 인증서에 대해 SHA256 알고리즘으로 서명하도록 지정합니다.

```
setmqspl -m QMGR -p MY.OTHER.QUEUE -s SHA256 -e 3DES -r CN=john,O=IBM,C=US -r
CN=jeff,O=IBM,C=US -a CN=phil,O=IBM,C=US
```

**참고:**

- 메시지 넣기 및 가져오기에 사용되는 보호의 품질은 일치해야 합니다. 메시지에 정의된 정책 보호 품질은 큐에 정의된 것보다 약하고, 메시지는 오류 처리 큐로 전송됩니다. 이 정책은 로컬 및 리모트 큐 둘 모두에 대해 유효합니다.

**관련 참조**

setmqspl 명령 속성의 전체 목록

## *AMS*에서 보안 정책 변경

Advanced Message Security를 사용하여 이미 정의한 보안 정책의 세부사항을 변경할 수 있습니다.

### 시작하기 전에

- 운영하려는 큐 관리자가 실행 중이어야 합니다.
- 큐 관리자에 연결하고 보안 정책을 작성하기 위해서는 필수 권한이 있어야 합니다.
  - z/OS z/OS에서 메시지 보안 정책 유틸리티(CSQ0UTIL)에 문서화된 권한을 부여하십시오.
  - Multi 멀티플랫폼에서는 **setmqaut** 명령을 사용하여 필요한 +connect, +inq 및 +chg 권한을 부여해야 합니다.

  보안 구성에 대한 자세한 정보는 121 페이지의 『보안 설정』의 내용을 참조하십시오.

### 이 태스크 정보

보안 정책을 변경하려면 setmqspl 명령을 이미 존재하는 정책에 적용하여 새 속성을 제공하십시오.

**예**

다음은 식별 이름 (DN) 이 CN=alice, O=IBM, C=US인 인증서가 있고 DN이 CN=jeff, O=IBM, C = US인 인증서가 있는 수신자 (**-r**) 에 대해 SHA256 알고리즘으로 서명된 작성자 (**-a**) 에 대한 3DES 알고리즘을 사용하여 메시지가 암호화되도록 지정하는 QMGR라는 큐 관리자에서 MYQUEUE 라는 정책을 작성하는 예입니다.

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

이 정책을 변경하려면 setmqspl 명령을 예외 모든 속성과 함께 실행하여 수정하려는 값만을 변경하십시오. 이 예에서 이전에 작성된 정책이 새 큐에 첨부되고 해당 암호화 알고리즘이 AES256으로 변경됩니다.

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

**관련 참조**

setmqspl(보안 정책 설정)

## *AMS*의 보안 정책 표시 및 덤프

**dspmqspl** 명령을 사용하여 사용자가 제공하는 명령행 매개변수에 따라 모든 보안 정책 목록 또는 이름 지정된 정책의 세부사항을 표시할 수 있습니다.

### 시작하기 전에

- 보안 정책 세부사항을 표시하기 위해서는 큐 관리자가 존재하고 실행 중이어야 합니다.
- 큐 관리자에 연결하고 보안 정책을 작성하기 위해서는 필수 권한이 있어야 합니다.

- z/OS z/OS에서 메시지 보안 정책 유틸리티(CSQ0UTIL)에 문서화된 권한을 부여하십시오.
- Multi 멀티플랫폼에서는 **setmqaut** 명령을 사용하여 필요한 +connect, +inq 및 +chg 권한을 부여해야 합니다.

보안 구성에 대한 자세한 정보는 121 페이지의 『보안 설정』의 내용을 참조하십시오.

## 이 태스크 정보

다음은 **dspmqspl** 명령 플래그의 목록입니다.

표 *110.* ***dspmqspl*** 명령 플래그.

| 명령 플래그 | 설명 |
|---|---|
| **-m** | 큐 관리자 이름(필수). |
| **-p** | 정책 이름. |
| **-export** | 이 플래그를 추가하면 다른 큐 관리자에 쉽게 적용할 수 있는 출력이 생성됩니다. |

### 예

다음 예에서는 venus.queue.manager를 위해 두 개의 보안 정책을 작성하는 방법을 보여줍니다.

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s sha256 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s sha256 -a "CN=another signer,O=IBM,C=US"
-e NONE
```

이 예는 venus.queue.manager에 대해 정의된 모든 정책 및 생성되는 출력에 대한 세부사항을 표시하는 명령을 보여줍니다.

```
dspmqspl -m venus.queue.manager

Policy Details:
Policy name: AMS_POL_04_ONE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

이 예는 venus.queue.manager에 대해 정의된 선택된 보안 정책 및 생성되는 출력에 대한 세부사항을 표시하는 명령을 보여줍니다.

```
dspmqspl -m venus.queue.manager -p AMS_POL_06_THREE

Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
```

```
Recipient DNs: -
Toleration: 0
```

다음 예에서는 먼저, 보안 정책을 작성한 다음에 **-export** 플래그를 사용하여 정책을 내보냅니다.

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s SHA256 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export
```

z/OS z/OS에서 내보내기된 정책 정보는 CSQ0UTIL에 의해 EXPORT DD에 작성됩니다.

Multi 멀티플랫폼에서 출력을 파일로 경로 재지정하십시오. 예를 들어, 다음과 같습니다.

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

보안 정책을 가져오려면:

- Linux AIX AIX and Linux에서:
  1. mqm IBM MQ 관리 그룹에 속하는 사용자로서 로그온하십시오.
  2. . policies.sh를 실행하십시오.
- Windows Windows에서 policies.bat을(를) 실행하십시오.
- z/OS z/OS에서 CSQ0UTIL 유틸리티를 사용하여 내보낸 정책 정보를 포함하는 데이터 세트를 SYSIN에 지정하십시오.

**관련 참조**
dspmqspl 명령 속성의 전체 목록

## *AMS*에서 보안 정책 제거

Advanced Message Security에서 보안 정책을 제거하려면 setmqspl 명령을 사용해야 합니다.

### 시작하기 전에

보안 정책을 관리할 때 준수해야 하는 몇 가지 초기 조건이 있습니다.

- 큐 관리자가 실행 중이어야 합니다.
- 큐 관리자에 연결하고 보안 정책을 작성하기 위해서는 필수 권한이 있어야 합니다.
  - z/OS z/OS에서 메시지 보안 정책 유틸리티(CSQ0UTIL)에 문서화된 권한을 부여하십시오.
  - Multi 멀티플랫폼에서는 **setmqaut** 명령을 사용하여 필요한 +connect, +inq 및 +chg 권한을 부여해야 합니다.

  보안 구성에 대한 자세한 정보는 121 페이지의 『보안 설정』의 내용을 참조하십시오.

### 이 태스크 정보

**setmqspl** 명령을 **-remove** 옵션과 함께 사용하십시오.

#### 예

다음은 정책 제거의 예입니다.

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

**관련 참조**
setmqspl 명령 속성의 전체 목록

## AMS의 시스템 큐 보호

시스템 큐를 사용하면 IBM MQ와 해당 보조 애플리케이션 사이의 통신이 가능합니다. 큐 관리자가 작성될 때마다 시스템 큐 또한 IBM MQ 내부 메시지 및 데이터를 저장하기 위해 작성됩니다. 권한 있는 사용자만이 액세스하거나 복호화할 수 있도록 시스템 큐를 Advanced Message Security로 보호할 수 있습니다.

시스템 큐 보호는 일반 큐의 보호와 같은 패턴을 따릅니다. 630 페이지의 『AMS에서 보안 정책 작성』의 내용을 참조하십시오.

Windows Windows에서 시스템 큐 보호를 사용하려면 `keystore.conf` 파일을 다음 디렉토리로 복사하십시오.

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS z/OS에서 `SYSTEM.ADMIN.COMMAND.QUEUE`에 대한 보호를 제공하려면 명령 서버가 키 및 구성을 포함하는 `keystore` 및 `keystore.conf`에 대한 액세스 권한이 있어야 명령 서버가 키 및 인증서에 액세스할 수 있습니다. `SYSTEM.ADMIN.COMMAND.QUEUE`의 보안 정책에 작성된 모든 변경사항을 적용하려면 명령 서버를 재시작해야 합니다.

명령 큐로부터 전송되고 수신된 모든 메시지는 정책 설정에 따라 서명되거나 서명된 후 암호화됩니다. 관리자가 권한 있는 서명자를 정의하는 경우, 서명자 식별 이름(DN) 검사를 통과하지 못한 명령 메시지는 명령 서버가 실행하지 않고 Advanced Message Security 오류 처리 큐로 라우트되지 않습니다. IBM MQ 탐색기 임시 동적 큐에 회신으로 전송된 메시지는 AMS에 의해 보호되지 않습니다.

보안 정책에는 다음 SYSTEM 큐에 영향을 미치지 않습니다.

- SYSTEM.ADMIN.ACCOUNTING.QUEUE
- SYSTEM.ADMIN.ACTIVITY.QUEUE
- SYSTEM.ADMIN.CHANNEL.EVENT
- SYSTEM.ADMIN.COMMAND.EVENT
- z/OS SYSTEM.ADMIN.COMMAND.QUEUE
- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- z/OS SYSTEM.BROKER.CLIENTS.DATA
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- z/OS SYSTEM.BROKER.SUBSCRIPTIONS.DATA
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- z/OS SYSTEM.CHLAUTH.DATA.QUEUE
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE

- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- z/OS SYSTEM.COMMAND.INPUT
- z/OS SYSTEM.DDELAY.LOCAL.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL
- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- z/OS SYSTEM.JMS.PS.STATUS.QUEUE
- z/OS SYSTEM.JMS.REPORT.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- z/OS SYSTEM.QSG.CHANNEL.SYNCQ
- z/OS SYSTEM.QSG.TRANSMIT.QUEUE
- z/OS SYSTEM.QSG.UR.RESOLUTION.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- z/OS SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

## Multi 큐와 AMS 스트리밍

중복 Advanced Message Security(AMS) 보호 메시지를 스트리밍할 수 있습니다.

큐에 넣은 메시지를 서명 및/또는 암호화하는 AMS 정책이 해당 큐에 정의되어 있으면 보호된 각 메시지의 사본을 두 번째 큐에 넣도록 큐의 **STREAMQ** 속성도 구성할 수 있습니다. 스트리밍된 중복 메시지는 원본 큐에 구성된 정책을 사용하여 서명 및/또는 암호화합니다.

다음 예에서는 QUEUE1 및 QUEUE2라는 두 개의 큐를 구성합니다. QUEUE1에는 **STREAMQ** 속성이 구성되어 있어 스트리밍된 메시지를 QUEUE2에 넣습니다.

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

AMS 보호 메시지는 CN=bob,O=IBM,C=GB 인증서가 있는 사용자가 QUEUE1에 넣습니다.

CN=alice,O=IBM,C=GB 인증서가 있는 애플리케이션이 QUEUE1의 메시지를 사용합니다.
CN=fred,O=IBM,C=GB 인증서가 있는 개별 애플리케이션에서 QUEUE2의 메시지를 사용합니다.

QUEUE1에는 다음 AMS 개인보호 정책이 적용되어 있습니다.

```
SET POLICY(QUEUE1) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=alice,O=IBM,C=GB') RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

QUEUE1의 정책에 암호화 알고리즘이 구성되어 있으면 정책에 나열된 수신자가 QUEUE1의 원본 메시지 수신자와 QUEUE2의 중복 메시지를 사용할 수신자를 모두 포함해야 합니다.

애플리케이션이 QUEUE2의 메시지를 사용하려고 할 때 무결성 검사를 수행하고 QUEUE2에 설정된 정책에 따라 메시지의 암호를 해독합니다. 애플리케이션이 QUEUE2에서 스트리밍된 메시지를 사용하려는 경우 메시지의 무결성을 확인하고 올바르게 암호 해독할 수 있도록 QUEUE2에 적절한 정책을 설정해야 합니다.

특히 서명 알고리즘, 서명자, 암호화 알고리즘은 QUEUE1에 적용된 정책과 같아야 합니다. QUEUE2의 정책 수신자는 QUEUE2의 메시지를 사용하는 수신자의 ID를 포함해야 합니다.

**참고:** QUEUE2에 적용된 정책에는 QUEUE1에 설정된 정책에 있는 모든 수신자가 나열되지 않아도 됩니다.

예를 들어, 다음 정책을 QUEUE2에 설정하면 인증서 식별 이름이 `CN=fred,O=IBM,C=GB`인 애플리케이션이 AMS 보호된 메시지를 읽을 수 있습니다.

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

**관련 개념**

스트리밍 큐

## AMS에서 OAM 권한 부여

파일 권한은 모든 사용자가 `setmqspl` 및 `dspmqspl` 명령을 실행하기 위한 권한을 제공합니다. 그러나 Advanced Message Security는 오브젝트 권한 관리자(OAM)에 의존하고 IBM MQ 관리 그룹인 mqm 그룹에 속하지 않는 사용자 또는 부여된 보안 정책 설정을 읽을 권한이 없는 사용자가 이러한 명령을 실행하려고 시도할 때마다 오류가 발생합니다.

### 프로시저

사용자에게 필수 권한을 부여하려면 다음을 실행하십시오.

```
setmqaut -m SOME.QUEUE.MANAGER -t qmgr -p SOME.USER +connect +inq
setmqaut -m SOME.QUEUE.MANAGER -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p SOME.USER +browse
+put
setmqaut -m SOME.QUEUE.MANAGER -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p SOME.USER +put
```

**참고:** Advanced Message Security 7.0.1을 사용하여 클라이언트를 큐 관리자에 연결하려면 이러한 OAM 권한을 설정하기만 하면 됩니다.

**주의:** SYSTEM.PROTECTION.POLICY.QUEUE에 대한 찾아보기 권한은 모든 상황에서 필수는 아닙니다. IBM MQ 는 SYSTEM.PROTECTION.POLICY.QUEUE 를 QUEUE합니다.

IBM MQ가 사용 가능한 모든 정책을 캐시하지는 않습니다. 정책 수가 많으면 IBM MQ가 제한된 수의 정책을 캐시합니다. 따라서 큐 관리자에 적은 수의 정책이 정의되어 있는 경우 SYSTEM.PROTECTION.POLICY.QUEUE에 찾아보기 옵션을 제공할 필요가 없습니다.

그러나 많은 수의 정책이 정의되어 있거나 이전 클라이언트를 사용 중인 경우 이 큐에 대한 찾아보기 권한을 부여해야 합니다. SYSTEM.PROTECTION.ERROR.QUEUE는 AMS 코드에서 생성된 오류 메시지를 넣는 데 사용됩니다. 이 큐에 대한 넣기(Put) 권한은 오류 메시지를 큐에 넣으려고 할 때만 확인됩니다. AMS 보호 큐에서 메시지를 가져오거나 넣으려고 할 때는 큐에 대한 넣기(Put) 권한이 확인되지 않습니다.

## AMS에서 보안 권한 부여

명령 자원 보안을 사용할 때 Advanced Message Security가 작동할 수 있도록 권한을 설정해야 합니다. 이 토픽은 예에서 RACF 명령을 사용합니다. 사용자의 엔터프라이즈가 다른 외부 보안 관리자(ESM)를 사용하는 경우에는 해당 ESM에 동등한 명령을 사용해야 합니다.

보안 권한을 부여하는 데에는 세 가지 측면이 있습니다.

- 637 페이지의 『AMSM 주소 공간』
- 637 페이지의 『CSQ0UTIL』
- 637 페이지의 『Advanced Message Security 정책이 정의된 큐 사용』

**참고:** 예 명령은 다음 변수를 사용합니다.

1. *QMgrName* - 큐 관리자의 이름.

   z/OS z/OS에서 이 값은 또한 큐 공유 그룹의 이름일 수도 있습니다.

2. *username* - 이는 그룹 이름일 수 있습니다.

3. 예는 MQQUEUE 클래스를 보여줍니다. z/OS 이는 또한 MXQUEUE, GMQQUEUE 또는 GMXQUEUE 일 수도 있습니다. 자세한 정보는 188 페이지의 『Profiles for queue security』의 내용을 참조하십시오.

게다가, 프로파일이 이미 있는 경우 RDEFINE 명령을 필요로 하지 않습니다.

### AMSM 주소 공간

일부 IBM MQ 보안을 Advanced Message Security 주소 공간을 실행 중인 사용자 이름에 실행해야 합니다.

- 큐 관리자에 대한 배치 연결을 위해 다음을 실행하십시오.

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
           PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.POLICY.QUEUE에 대한 액세스의 경우 다음을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

### CSQOUTIL

사용자가 **setmqspl** 및 **dspmqspl** 명령을 실행하도록 허용하는 유틸리티에는 다음 권한이 필요합니다. 여기서 사용자 이름은 작업 사용자 ID입니다.

- 큐 관리자에 대한 배치 연결을 위해 다음을 실행하십시오.

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- **setmqpol** 명령에 필요한 SYSTEM.PROTECTION.POLICY.QUEUE에 대한 액세스를 위해 다음을 실행하십시오.

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
             PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- **dspmqpol** 명령에 필요한 SYSTEM.PROTECTION.POLICY.QUEUE에 대한 액세스를 위해 다음을 실행하십시오.

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
               PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

### Advanced Message Security 정책이 정의된 큐 사용

애플리케이션이 정책이 정의되어 있는 큐에 대해 작업할 때 해당 애플리케이션에는 Advanced Message Security가 메시지를 보호할 수 이도록 추가 권한이 필요합니다.

애플리케이션에는 다음이 필요합니다.

- SYSTEM.PROTECTION.POLICY.QUEUE에 대한 읽기 액세스. 다음을 실행하여 이를 수행하십시오.

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.ERROR.QUEUE에 대한 넣기 액세스. 다음을 실행하여 이를 수행하십시오.

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

## IBM i IBM i 에서 AMS 에 대한 인증서 및 키 저장소 구성 파일 설정

Advanced Message Security 보호를 설정할 때 첫 번째 태스크는 인증서를 작성하고, 이를 사용자의 환경과 연관시키는 것입니다. 이 연관은 통합 파일 시스템(IFS)에 보관된 파일을 통해 구성됩니다.

### 프로시저

1. IBM i와 함께 제공된 OpenSSL 도구를 사용하여 자체 서명 인증서를 작성하려면 QShell에서 다음 명령을 실행하십시오.

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   명령은 다음을 포함하여 새 자체 서명 인증서의 다양한 식별 이름 속성에 프롬프트를 표시합니다.

   - 공용 이름(CN=)
   - 조직(O=)
   - 국가(C=)

   이는 암호화되지 않은 개인 키 및 일치하는 인증서를 둘 모두 PEM(Privacy Enhanced Mail) 형식으로 작성합니다.

   단순성을 위해 공용 이름, 조직 및 국가에 값을 입력하십시오. 이러한 속성 및 값은 정책을 작성할 때 중요합니다.

   명령행에서 **-config** 매개변수를 사용해 사용자 정의 openssl 구성을 지정하여 추가 프롬프트 및 속성을 사용자 정의할 수 있습니다. 구성 파일 구문에 대한 자세한 정보는 OpenSSL 문서를 참조하십시오.

   예를 들면, 다음 명령은 추가 X.509 v3 인증서 확장을 추가합니다.

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
   -keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
   ```

   여기서 myconfig.cnf는 다음 항목을 포함하는 ASCII 스트림 파일입니다.

   ```
   [req]
   distinguished_name = req_distinguished_name
   x509_extensions = myextensions

   [req_distinguished_name]
   countryName = Country Name (2 letter code)
   countryName_default = GB
   stateOrProvinceName = State or Province Name (full name)
   stateOrProvinceName_default = Hants
   localityName = Locality Name (eg, city)
   localityName_default = Hursley
   organizationName = Organization Name (eg, company)
   organizationName_default = IBM United Kingdom
   organizationalUnitName = Organizational Unit Name (eg, department)
   organizationalUnitName_default = IBM MQ Development
   commonName = Common Name (eg, Your Name)

   [myextensions]
   keyUsage = digitalSignature,nonRepudiation,dataEncipherment,keyEncipherment
   extendedKeyUsage = emailProtection
   ```

2. AMS에는 인증서 및 개인 키 둘 모두가 동일 파일에 보관되어야 합니다. 이를 달성하려면 다음 명령을 실행하십시오.

   ```
   cat $HOME/mycert.pem >> $HOME/private.pem
   ```

$HOME의 private.pem 파일에는 이제 일치하는 개인 키와 인증서가 포함되는 반면, mycert.pem 파일에는 메시지를 암호화하고 서명을 유효성 검증할 수 있는 공개 인증서가 모두 포함됩니다.

기본 위치에 키 저장소 구성 파일(keystore.conf)을 작성하여 두 파일을 사용자 환경과 연관시켜야 합니다.

기본적으로 AMS는 홈 디렉토리의 .mqs 서브디렉토리에서 키 저장소 구성을 찾습니다.

3. QShell에서 keystore.conf 파일을 작성하십시오.

```
mkdir -p $HOME/.mqs
echo "pem.private = $HOME/private.pem" > $HOME/.mqs/keystore.conf
echo "pem.public = $HOME/mycert.pem" >> $HOME/.mqs/keystore.conf
echo "pem.password = unused" >> $HOME/.mqs/keystore.conf
```

## IBM i *IBM i 에서 AMS 에 대한 정책 작성*

정책을 작성하기 전에 보호된 메시지를 보관하기 위해 큐를 작성해야 합니다.

### 프로시저

1. 명령행 프롬프트에 다음을 입력하십시오.

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

여기서 mqmname은 큐 관리자의 이름입니다.

DSPMQM 명령을 사용하여 큐 관리자가 보안 정책 사용이 가능한지를 확인하십시오. **Security Policy Capability** 에 **YES*가 표시되는지 확인하십시오.

정의할 수 있는 가장 단순한 정책은 무결성 정책이고 이는 암호화 알고리즘이 아니라 디지털 서명 알고리즘을 사용하여 정책을 작성하여 달성됩니다.

메시지는 서명되지만 암호화되지 않습니다. 메시지가 암호화될 예정이면 암호화 알고리즘과 하나 이상의 의도된 메시지 수신인을 지정해야 합니다.

의도된 메시지 수신인의 공개 키 저장소의 인증서는 식별 이름을 통해 식별됩니다.

2. QShell에서 다음 명령을 사용하여 $HOME에서 인증서의 식별 이름을 공개 키 저장소 mycert.pem을 표시하십시오.

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

식별 이름을 의도된 수신인으로 입력해야 하고, 정책 이름은 보호될 큐 이름과 일치해야 합니다.

3. CL 명령 프롬프트에서 예를 들어 다음을 입력하십시오.

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

여기서 mqmname은 큐 관리자의 이름입니다.

정책이 작성되면 해당 큐 이름을 통해 놓여지고, 찾거나 파괴적으로 제거된 메시지는 AMS 정책을 따릅니다.

**관련 참조**

메시지 큐 관리자 표시(DSPMQM)
MQM 보안 정책 설정(SETMQMSPL)

## IBM i *IBM i 에서 AMS 에 대한 정책 테스트*

제품과 함께 제공된 샘플 애플리케이션을 사용하여 보안 정책을 테스트하십시오.

### 이 태스크 정보

IBM MQ와 함께 제공된 샘플 애플리케이션(예: AMQSPUT4, AMQSGET4, AMQSGBR4 및 WRKMQMMSG 등과 같은 도구)을 사용하여 PROTECTED 큐 이름을 사용하여 메시지를 넣고, 찾고, 가져올 수 있습니다.

모든 것이 올바르게 구성된 경우 애플리케이션의 작동에는 이 사용자의 보호되지 않은 큐의 애플리케이션의 작동과 차이가 있어서는 안됩니다.

그러나 Advanced Message Security에 대해 설정되지 않은 사용자 또는 메시지를 복호화하는 데 필요한 개인 키가 없는 사용자는 메시지를 볼 수 없습니다. 사용자는 MQCC_FAILED (2)와 동등한 RCFAIL 완료 코드 및 RC2063(MQRC_SECURITY_ERROR) 이유 코드를 수신합니다.

AMS 보호가 유효한지 확인하려면 예를 들어, AMQSPUT0를 사용하여 PROTECTED 큐에 일부 테스트 메시지를 두십시오. 그런 다음 알리어스 큐를 작성하여 쉬는 중인 원시 보호된 데이터를 찾아볼 수 있습니다.

### 프로시저

사용자에게 필수 권한을 부여하려면 다음을 실행하십시오.

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

예를 들어, AMQSBCG4 또는 WRKMQMMSG를 사용하여 ALIAS 큐 이름 찾아보기는 PROTECTED 큐의 찾아보기가 일반 텍스트 메시지를 표시하는 것보다 더 큰 `scrambled` 메시지를 표시해야 합니다.

스크램블된 메시지는 표시 가능하지만 이 이름과 일치를 강제 실행하는 AMS에 대한 정책이 없으므로 원본 일반 텍스트는 ALIAS 큐를 사용하여 해독할 수 없습니다. 따라서 원시 보호된 데이터가 리턴됩니다.

**관련 참조**

MQM 보안 정책 설정(SETMQMSPL)
MQ 메시지에 대한 작업(WRKMQMMSG)

## AMS의 명령 및 구성 이벤트

Advanced Message Security를 사용하면 로그되고 감사를 위한 정책 변경사항의 레코드로서 사용될 수 있는 명령 및 구성 이벤트 메시지를 생성할 수 있습니다.

IBM MQ에 의해 생성된 명령 및 구성 이벤트는 이벤트가 발생하는 큐 관리자에서 전용 큐로 전송된 PCF 형식의 메시지입니다.

구성 이벤트 메시지는 SYSTEM.ADMIN.CONFIG.EVENT 큐에 전송됩니다.

명령 이벤트 메시지는 SYSTEM.ADMIN.COMMAND.EVENT 큐에 전송됩니다.

이벤트는 Advanced Message Security 보안 정책을 관리하기 위해 사용 중인 도구와 관계 없이 생성됩니다.

Advanced Message Security에는 보안 정책에서 다른 조치에 의해 생성되는 네 가지 유형의 이벤트가 있습니다.

- 630 페이지의 『AMS에서 보안 정책 작성』 - 두 개의 IBM MQ 이벤트 메시지를 생성:
  - 구성 이벤트
  - 명령 이벤트
- 631 페이지의 『AMS에서 보안 정책 변경』 - 세 개의 IBM MQ 이벤트 메시지를 생성:
  - 오래된 보안 구성 값을 포함하는 구성 이벤트
  - 새 보안 정책 값을 포함하는 구성 이벤트
  - 명령 이벤트
- 631 페이지의 『AMS의 보안 정책 표시 및 덤프』 - 한 개의 IBM MQ 이벤트 메시지를 생성:
  - 명령 이벤트
- 633 페이지의 『AMS에서 보안 정책 제거』 - 두 개의 IBM MQ 이벤트 메시지를 생성:
  - 구성 이벤트
  - 명령 이벤트

## *AMS*의 이벤트 로깅 사용 및 사용 안함

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

## 프로시저

**구성 이벤트**

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

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

**관련 태스크**

IBM MQ에서 구성, 명령, 로거 이벤트 제어

## *AMS*의 명령 이벤트 메시지 형식

명령 이벤트 메시지는 MQCFH 구조와 이를 뒤따르는 PCF 매개변수로 구성됩니다.

다음은 선택된 MQCFH 값입니다.

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

**참고:** ParameterCount 값은 2입니다. 항상 MQCFGR 유형(그룹)의 두 개의 매개변수가 있기 때문입니다. 각 그룹은 적합한 매개변수로 구성됩니다. 이벤트 데이터는 두 개의 그룹, CommandContext 및 CommandData로 구성됩니다.

CommandContext에는 다음이 포함됩니다.

**EventUserID**

| | |
|---|---|
| 설명: | 이벤트를 생성한 명령 또는 호출을 실행한 사용자 ID입니다. (이는 명령 또는 호출을 실행하기 위한 권한을 검사하는 데 사용된 것과 같은 사용자 ID입니다. 큐로부터 수신한 명령의 경우 이는 또한 명령 메시지의 MD로부터의 사용자 ID(UserIdentifier)입니다). |
| ID: | MQCACF_EVENT_USER_ID. |
| 데이터 유형: | MQCFST. |
| 최대 길이: | MQ_USER_ID_LENGTH. |
| 리턴됨: | 항상. |

**EventOrigin**

설명: 이벤트를 초래한 조치의 원본.

ID: MQIACF_EVENT_ORIGIN.

데이터 유형: MQCFIN.

값:

**MQEVO_CONSOLE**
콘솔 명령 - 명령행.

**MQEVO_MSG**
IBM MQ Explorer 플러그인으로부터의 명령 메시지.

리턴됨: 항상.

**EventQMgr**

설명: 명령 또는 호출이 입력된 큐 관리자. (명령이 실행되고 이벤트를 생성한 큐 관리자는 이벤트 메시지의 MD에 있습니다.)

ID: MQCACF_EVENT_Q_MGR.

데이터 유형: MQCFST.

최대 길이: MQ_Q_MGR_NAME_LENGTH.

리턴됨: 항상.

**EventAccountingToken**

설명: 메시지(MQEVO_MSG)로서 수신된 명령의 경우 명령 메시지의 MD로부터의 회계 토큰(AccountingToken).

ID: MQBACF_EVENT_ACCOUNTING_TOKEN.

데이터 유형: MQCFBS.

최대 길이: MQ_ACCOUNTING_TOKEN_LENGTH.

리턴됨: EventOrigin이 MQEVO_MSG인 경우에만.

**EventIdentityData**

설명: 메시지(MQEVO_MSG)로서 수신된 명령의 경우 명령 메시지의 MD로부터의 애플리케이션 ID 데이터(ApplIdentityData).

ID: MQCACF_EVENT_APPL_IDENTITY.

데이터 유형: MQCFST.

최대 길이: MQ_APPL_IDENTITY_DATA_LENGTH.

리턴됨: EventOrigin이 MQEVO_MSG인 경우에만.

**EventApplType**

설명: 메시지(MQEVO_MSG)로서 수신된 명령의 경우 명령 메시지의 MD로부터의 애플리케이션 유형(PutApplType).

ID: MQIACF_EVENT_APPL_TYPE.

데이터 유형: MQCFIN.

리턴됨: EventOrigin이 MQEVO_MSG인 경우에만.

**EventApplName**

| | |
|---|---|
| 설명: | 메시지(MQEVO_MSG)로서 수신된 명령의 경우 명령 메시지의 MD로부터의 애플리케이션 이름(PutApplName). |
| ID: | MQCACF_EVENT_APPL_NAME. |
| 데이터 유형: | MQCFST. |
| 최대 길이: | MQ_APPL_NAME_LENGTH. |
| 리턴됨: | EventOrigin이 MQEVO_MSG인 경우에만. |

**EventApplOrigin**

| | |
|---|---|
| 설명: | 메시지(MQEVO_MSG)로서 수신된 명령의 경우 명령 메시지의 MD로부터의 애플리케이션 원본 데이터(ApplOriginData). |
| ID: | MQCACF_EVENT_APPL_ORIGIN. |
| 데이터 유형: | MQCFST. |
| 최대 길이: | MQ_APPL_ORIGIN_DATA_LENGTH. |
| 리턴됨: | EventOrigin이 MQEVO_MSG인 경우에만. |

**명령**

| | |
|---|---|
| 설명: | 명령 코드. |
| ID: | MQIACF_COMMAND. |
| 데이터 유형: | MQCFIN. |
| 값: | **MQCMD_INQUIRE_PROT_POLICY 숫자 값 205**<br>**MQCMD_CREATE_PROT_POLICY 숫자 값 206**<br>**MQCMD_DELETE_PROT_POLICY 숫자 값 207**<br>**MQCMD_CHANGE_PROT_POLICY 숫자 값 208**<br>이는 IBM MQ 8.0 `cmqcfc.h`에 정의되어 있습니다. |
| 리턴됨: | 항상. |

CommandData에는 PCF 명령을 구성하는 PCF 요소가 포함됩니다.

## *AMS*의 구성 이벤트 메시지 형식

구성 이벤트는 표준 Advanced Message Security 형식의 PCF 메시지입니다.

MQMD 메시지 디스크립터의 가능한 값은 이벤트 메시지 MQMD(메시지 디스크립터)에서 볼 수 있습니다.

다음은 선택된 MQMD 값입니다.

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

메시지 버퍼는 MQCFH 구조 및 이를 뒤따르는 매개변수 구조로 구성됩니다. 가능한 MQCFH 값은 이벤트 메시지 MQCFH(PCF 헤더)에서 볼 수 있습니다.

다음은 선택된 MQCFH 값입니다.

```
Type = MQCFT_EVENT
Command = MQCMD_CONFIG_EVENT
MsgSeqNumber = 1 or 2          // 2 will be in case of Change Object event
Control = MQCFC_LAST or MQCFC_NOT_LAST     //MQCFC_NOT_LAST will be in case of 1 Change Object
event
```

```
ParameterCount = reflects number of PCF parameters following MQCFH
CompCode = MQCC_WARNING
Reason = one of {MQRC_CONFIG_CREATE_OBJECT, MQRC_CONFIG_CHANGE_OBJECT,
MQRC_CONFIG_DELETE_OBJECT}
```

MQCFH를 뒤따르는 매개변수는 다음과 같습니다.

***EventUserID***

설명: 이벤트를 생성한 명령 또는 호출을 실행한 사용자 ID입니다. (이는 명령 또는 호출을 실행하기 위한 권한을 검사하는 데 사용된 것과 같은 사용자 ID입니다. 큐로부터 수신한 명령의 경우 이는 또한 명령 메시지의 MD로부터의 사용자 ID(UserIdentifier)입니다).

ID: **MQCACF_EVENT_USER_ID**

데이터 유형: MQCFST.

최대 길이: MQ_USER_ID_LENGTH.

리턴됨: 항상.

***SecurityId***

설명: 명령 서버 메시지 또는 로컬 명령의 Windows SID의 경우에 MQMD.AccountingToken의 값.

ID: **MQBACF_EVENT_SECURITY_ID**

데이터 유형: MQCBS.

최대 길이: MQ_SECURITY_ID_LENGTH.

리턴됨: 항상.

***EventOrigin***

설명: 이벤트를 초래한 조치의 원본.

ID: **MQIACF_EVENT_ORIGIN**

데이터 유형: MQCFIN.

값:

**MQEVO_CONSOLE**
콘솔 명령 - 명령행.

**MQEVO_MSG**
IBM MQ 탐색기 플러그인으로부터의 명령 메시지.

리턴됨: 항상.

***EventQMgr***

설명: 명령 또는 호출이 입력된 큐 관리자. (명령이 실행되고 이벤트를 생성한 큐 관리자는 이벤트 메시지의 MD에 있습니다.)

ID: **MQCACF_EVENT_Q_MGR**

데이터 유형: MQCFST

최대 길이: MQ_Q_MGR_NAME_LENGTH

리턴됨: 항상.

***ObjectType***

설명: 오브젝트 유형.

ID: **MQIACF_OBJECT_TYPE**

데이터 유형: MQCFIN

값: **MQOT_PROT_POLICY**
Advanced Message Security 보호 정책. **1019** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

리턴됨: 항상.

***PolicyName***

설명: Advanced Message Security 정책 이름.

ID: **MQCA_POLICY_NAME**.

데이터 유형: MQCFST.

값: **2112** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

최대 길이: MQ_OBJECT_NAME_LENGTH.

리턴됨: 항상.

***PolicyVersion***

설명: Advanced Message Security 정책 버전.

ID: **MQIA_POLICY_VERSION**

데이터 유형: MQCFIN

값 **238** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

리턴됨: 항상

***TolerateFlag***

설명: Advanced Message Security 정책 관용 플래그.

ID: **MQIA_TOLERATE_UNPROTECTED**

데이터 유형: MQCFIN

값 **235** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

리턴됨: 항상.

***SignatureAlgorithm***

설명: Advanced Message Security 정책 서명 알고리즘.

ID: **MQIA_SIGNATURE_ALGORITHM**

데이터 유형: MQCFIN

값: **236** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

리턴됨: Advanced Message Security 정책에 정의된 서명 알고리즘이 있을 때마다

***EncryptionAlgorithm***

설명: Advanced Message Security 정책 암호화 알고리즘.

ID: **MQIA_ENCRYPTION_ALGORITHM**

데이터 유형: MQCFIN

값: **237** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

리턴됨: IBM MQ 정책에 정의된 암호화 알고리즘이 있을 때마다

***SignerDNs***

설명: 허용된 서명자의 제목 식별 이름.

ID: **MQCA_SIGNER_DN**

데이터 유형: MQCFSL

값: **2113** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

최대 길이: 정책에서 가장 긴 서명자 DN이지만 MQ_DISTINGUISHED_NAME_LENGTH보다 길지는 않음

리턴됨: IBM MQ 정책에 정의될 때마다.

***RecipientDNs***

설명: 허용된 서명자의 제목 식별 이름.

ID: **MQCA_RECIPIENT_DN**

데이터 유형: MQCFSL

값: **2114** - IBM MQ 8.0 또는 cmqc.h 파일에 정의된 숫자 값.

최대 길이: 정책에서 가장 긴 수신인 DN이지만 MQ_DISTINGUISHED_NAME_LENGTH보다 길지는 않음

리턴됨: IBM MQ 정책에 정의될 때마다.

# 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 라이센스까지 부여하는 것은 아닙니다. 라이센스에 대한 의문사항은 다음으로 문의하십시오.

07326
서울특별시 영등포구
국제금융로 10, 3IFC
한국 아이.비.엠 주식회사
U.S.A.

2바이트(DBCS) 정보에 관한 라이센스 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 라이센스 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 라이센스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이센스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이센스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을

© Copyright IBM Corp. 2007년, 2024

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 라이센스:

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 소프트카피로 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: