7.5

# *IBM WebSphere MQ* 보안

**참고**

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 299 페이지의 『주의사항』에 있는 정보를 확인하십시오.

This edition applies to version 7 release 5 of IBM® WebSphere® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

# 목차

# 보안

보안은 IBM WebSphere MQ 애플리케이션 개발자 및 IBM WebSphere MQ 권한을 구성하는 시스템 관리자 모두에 대해 중요한 고려사항입니다.

# 보안 개요

이 토픽 콜렉션은 IBM WebSphere 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 WebSphere MQ에서 구현됩니다.

## 보안 개념 및 메커니즘

이 주제 모음에서는 IBM WebSphere MQ 설치에서 고려해야 하는 보안 측면에 대해 설명합니다.

공통적으로 허용되는 보안 관점은 다음과 같습니다.

- 5 페이지의 『식별 및 인증』
- 6 페이지의 『권한 부여』
- 6 페이지의 『감사』
- 6 페이지의 『기밀성』
- 7 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.

- 7 페이지의 『암호화』
- 9 페이지의 『메시지 요약 및 디지털 서명』
- 9 페이지의 『디지털 인증서』
- 13 페이지의 『공개 키 기반 구조(PKI)』

IBM WebSphere MQ 구현을 계획하는 경우, 중요한 보안 측면을 구현하는 데 필요한 보안 메커니즘에 대해 고려하십시오. 이러한 토픽을 읽은 후에 고려해야 할 사항에 대한 자세한 정보는 43 페이지의 『보안 요구사항 계획』의 내용을 참조하십시오.

**관련 개념**

188 페이지의 『SSL 또는 TLS를 사용하여 두 개의 큐 관리자 연결』
SSL 또는 TLS 암호화 보안 프로토콜을 사용하는 보안 통신은 통신 채널 설정 및 인증하는 데 사용할 디지털 인증서 관리를 수반합니다.

101 페이지의 『SSL 또는 TLS에 대한 작업』
이러한 주제는 IBM WebSphere MQ을 사용하여 SSL 또는 TLS 사용과 관련된 단일 태스크를 수행하기 위한 지시사항을 제공합니다.

### 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.

## 부인 방지

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 결국 특정 메시지가 특정 개인과 연관되는지를 증명하기 위해서 사용됩니다.

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

IBM WebSphere MQ Advanced Message Security을 포함한 IBM WebSphere MQ에서는 부인 방지 서비스를 기본 기능의 일부로서 제공하지 않습니다. 그러나 이 제품 문서에는 자체 엑시트 프로그램을 작성함으로써 WebSphere MQ에서 자체 부인 방지 서비스를 제공할 수 있는 방법을 제안합니다.

**관련 개념**

20 페이지의 『IBM WebSphere MQ의 식별 및 인증』
IBM WebSphere MQ에서 메시지 컨텍스트 정보 및 상호 인증을 사용하여 식별 및 인증을 구현할 수 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.

**관련 개념**

20 페이지의 『IBM WebSphere MQ 의 권한』
권한 부여를 사용하여 IBM WebSphere MQ 환경에서 수행할 수 있는 특정 개인 또는 응용프로그램을 제한할 수 있습니다.

## 감사

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

권한 설정 방법에 대한 자세한 정보는 45 페이지의 『권한 부여 계획』 및 관련 하위 주제를 참조하십시오.

**관련 개념**

21 페이지의 『IBM WebSphere MQ에서 감사』
IBM WebSphere MQ는 일반적이지 않은 활동이 발생하는 레코드에 이벤트 메시지를 발행할 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.

## 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.

## 암호화 개념

이 주제 콜렉션에서는 WebSphere MQ에 적용 가능한 암호화 개념을 설명합니다.

엔티티라는 용어는 큐 관리자, WebSphere MQ MQI 클라이언트, 개별 사용자 또는 메시지 교환이 가능한 기타 시스템 기능을 나타냅니다.

**관련 개념**

22 페이지의 『IBM WebSphere MQ 의 암호화』

IBM WebSphere MQ는 SSL(Secure Sockets Layer) 및 TLS(Transport Security Layer) 프로토콜을 사용하여 암호를 제공합니다.

## 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. 프로세스의 이 부분을 암호화(때로는 *encipherment*)라 합니다.
2. 암호문은 수신자에게 전송됩니다.
3. 수신자는 암호문 메시지를 다시 일반 텍스트 양식으로 변환합니다. 프로세스의 이 부분을 암호 해독(때로는 *decipherment*)이라 합니다.

암호화 정의는 용어집을 참조하십시오.

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 18 페이지의 『SSL 및 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.8 페이지의 그림 1 는 대칭 키 암호화를 나타낸다.
- 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.8 페이지의 그림 2 공용 키 암호화라고도 하는 비대칭 키 암호화를 설명합니다.

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

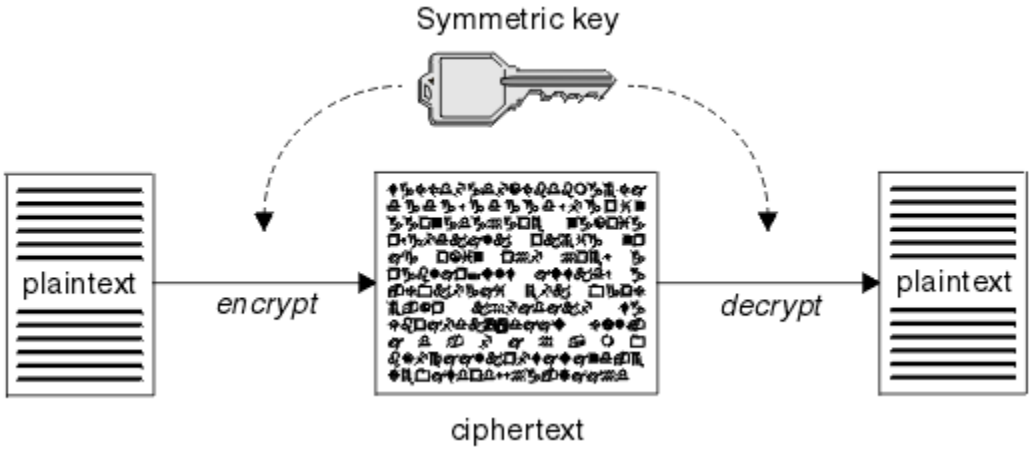



그림 *1.* 대칭 키 암호화

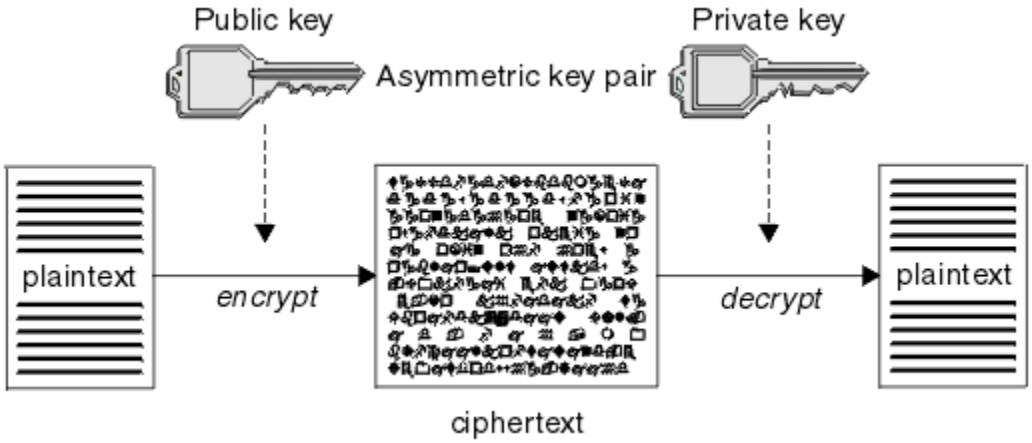



그림 *2.* 비대칭 키 암호화

8 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. 이것을 키 분배 문제점이라 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 축약은 메시지 컨텐츠를 고정된 크기의 숫자로 표현한 것으로 해시 기능으로 계산됩니다. 메시지 축약을 암호화하여 디지털 서명을 만들 수 있습니다.

메시지의 크기는 원래 다양합니다. 메시지 요약은 메시지 컨텐츠에 대한 고정된 크기의 숫자 표시입니다. 메시지 축약은 다음 두 가지 기준을 충족시키는 변환인 해시 기능에 의해 처리됩니다.

- 해시 기능은 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**

18 페이지의 『SSL 및 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 10 페이지의 『인증 기관』에서 설명하는 것처럼 인증 기관(CA)이라고 합니다.

디지털 인증서의 내용

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

WebSphere MQ가 사용하는 디지털 인증서는 X.509 표준을 준수하며, 필요한 정보와 정보를 송신하기 위한 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 않으면 시스템이 이를 인식하지 않는 경우에 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

WebSphere MQ는 X.509 표준을 준수하는 디지털 인증서를 지원합니다. 이 경우 클라이언트 인증 옵션이 필요합니다.

IBM WebSphere MQ가 피어투피어 시스템이기 때문에 SSL 용어에서 클라이언트 인증으로 표시됩니다. 따라서, SSL 인증에 사용되는 모든 개인 인증서는 클라이언트 인증의 주요 사용법을 허용해야 합니다. 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 WebSphere MQ에서의 인증서 유효성 검증 정책에 대한 자세한 정보는 32 페이지의 『IBM WebSphere MQ 의 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. 추가 정보는 137 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.

다음 속성 유형이 DN 안에서 흔히 발견됩니다.

| | |
|---|---|
| SERIALNUMBER | 인증서 일련 번호 |
| 메일 | 이메일 주소 |
| 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 WebSphere MQ에서는 생성 결함이 있는 특정 DN가 허용됩니다. 자세한 정보는 SSLPEER 값에 대한 WebSphere MQ 규칙을 참조하십시오.

**관련 개념**

9 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. 인증 요청은 일반적으로 시스템에서 사용하는 인증 관리 도구에 의해 생성됩니다 (예: UNIX, Linux®및 Windows 시스템의 iKeyman 도구 및 z/OS®의 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.

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

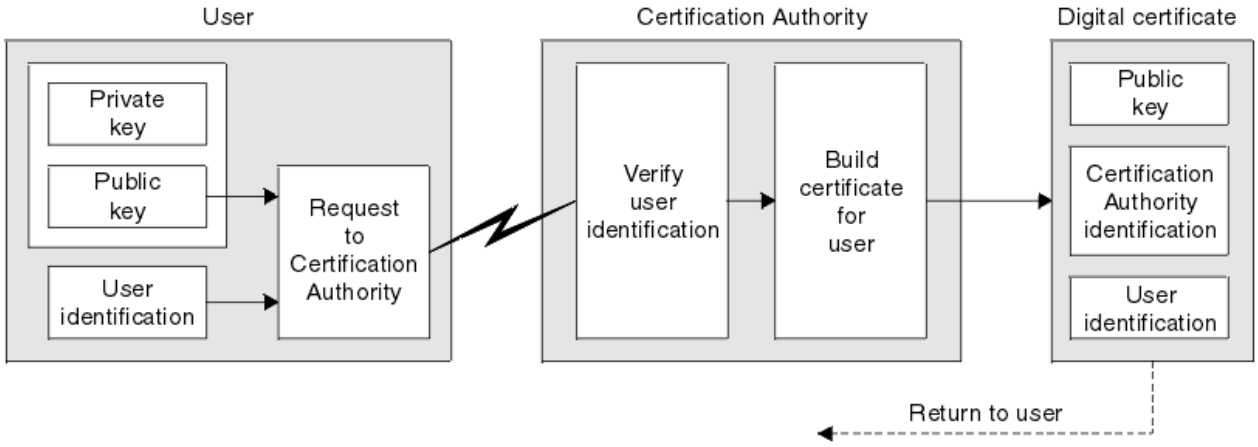



그림 *3.* 디지털 인증서 확보

다이어그램에서:

- "사용자 ID"에는 해당하는 식별 이름이 포함됩니다.
- "인증 기관 ID"에는 인증서를 발급하는 CA의 식별 이름이 포함됩니다.
-

디지털 인증서에는 다이어그램에 표시되는 필드 이외에도 추가 필드가 있습니다. 디지털 인증서의 다른 필드에 대한 자세한 정보는 9 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.

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

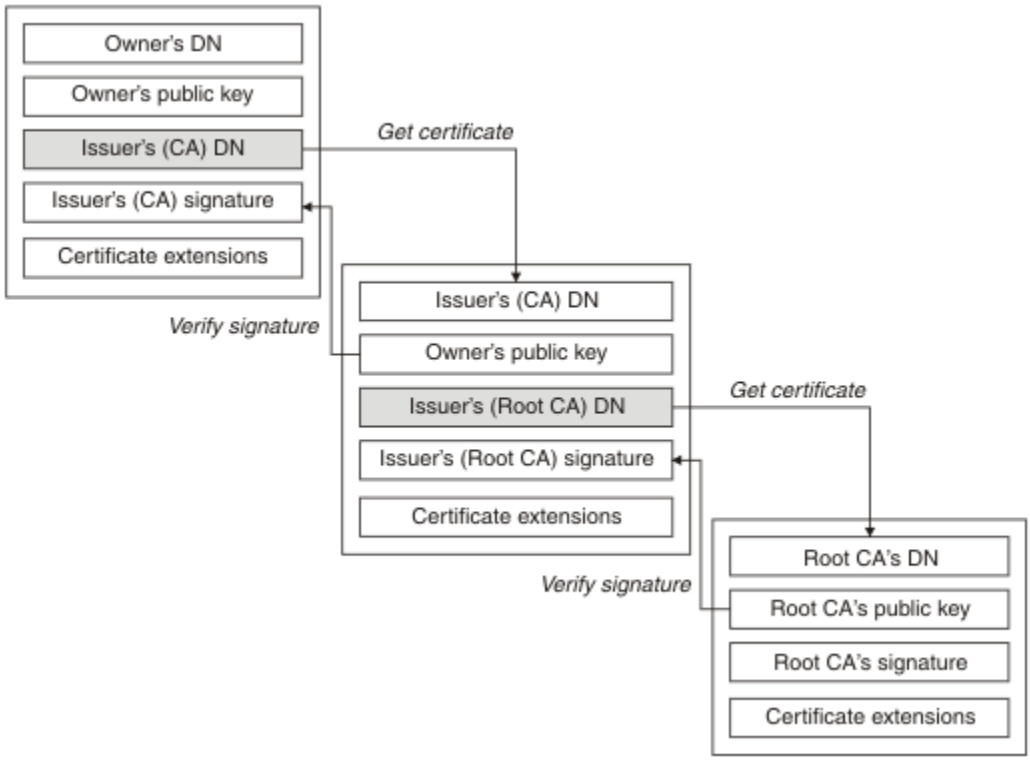



그림 *4.* 신뢰 체인

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.

인증서 만기 정의에 대해서는 용어집을 참조하십시오.

인증서는 다음을 비롯하여 다양한 이유로 폐기될 수 있습니다.

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

WebSphere MQ는 OCSP(Online Certificate Status Protoco) 응답자(UNIX, Linux 및 Windows 시스템에서만) 요청을 보내어 인증서가 폐기되었는지 여부를 점검할 수 있습니다. LDAP 서버의 CRL에 액세스할 수도 있습니다. OCSP 폐기 및 CRL 정보는 인증 기관이 발행합니다. 추가 정보는 137 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 9 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: SSL 및 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 WebSphere MQ는 SSL과 TLS를 모두 지원합니다.

두 프로토콜의 주요 목표는 기밀성(때로는 사생활 보호라고도 함), 데이터 무결성, ID 및 디지털 인증서를 사용한 인증을 제공하는 것입니다.

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

**관련 개념**

22 페이지의 『IBM WebSphere MQ 의 보안 프로토콜』
IBM WebSphere MQ는 메시지 채널 및 MQI 채널에 대한 링크 레벨의 보안을 제공하기 위해 TLS(Transport Layer Security) 및 SSL(Secure Sockets Layer) 프로토콜 모두를 지원합니다.

### *SSL(Secure Socket Layer)* 및 *TLS(Transport Layer Security)* 개념

SSL 및 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.

SSL 및 TLS 프로토콜은 인터넷을 통한 통신 보안을 제공하며 클라이언트/서버 애플리케이션이 비밀이 유지되는 안정적인 방식으로 통신할 수 있도록 해줍니다. 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.

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

IBM WebSphere MQ가 지원하는 모든 CipherSpec 목록은 199 페이지의 『CipherSpec 지정』에 있습니다.

SSL 프로토콜에 대한 자세한 정보는 https://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt에서 제공되는 정보를 참조하십시오. TLS 프로토콜에 대한 자세한 정보는 IETF(Internet Engineering Task Force) 웹 사이트 (https://www.ietf.org)에서 TLS 작업 그룹이 제공하는 정보를 참조하십시오.

### *SSL* 또는 *TLS* 데이터 교환의 개요

SSL 또는 TLS 데이터 교환을 사용하면 통신하는 SSL 또는 TLS 클라이언트와 서버가 비밀 키를 설정할 수 있습니다.

이 절에서는 SSL 또는 TLS 클라이언트와 서버가 서로 통신할 수 있도록 하는 단계에 대한 요약을 제공한다.

- 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 SSL 또는 TLS는 메시지에 대해 비대칭 암호화보다 빠른 대칭 암호화의 공유 키를 사용합니다.

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

대략적으로, SSL 데이터 교환에 연관된 단계는 다음과 같습니다.

1. SSL 또는 TLS 클라이언트는 SSL 또는 TLS 버전과 같은 암호화 정보 및 클라이언트에서 지원되는 CipherSuite를 클라이언트의 환경 설정 순으로 나열하는 "client hello" 메시지를 송신합니다. 메시지에는 나

중의 처리에서 사용되는 무작위 바이트 문자열도 포함됩니다. 프로토콜은 "client hello"가 클라이언트에서 지원되는 데이터 압축 메소드도 포함하도록 허용합니다.

2. SSL 또는 TLS 서버는 클라이언트, 세션 ID, 기타 무작위 바이트 문자열로 제공되는 목록에서 서버가 선택한 CipherSuite가 포함되는 "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. SSL 또는 TLS 클라이언트는 서버의 디지털 인증서를 확인합니다. 자세한 정보는 16 페이지의 『SSL 및 TLS가 ID, 인증, 기밀성, 무결성을 제공하는 방법』의 내용을 참조하십시오.
4. SSL 또는 TLS 클라이언트는 클라이언트와 서버 모두가 후속 메시지 데이터 암호화에 사용되는 비밀 키를 처리할 수 있도록 하는 무작위 바이트 문자열을 송신합니다. 무작위 바이트 문자열 자체는 서버의 공개 키로 암호화됩니다.
5. SSL 또는 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. SSL 또는 TLS 서버는 클라이언트 인증서를 확인합니다. 자세한 정보는 16 페이지의 『SSL 및 TLS가 ID, 인증, 기밀성, 무결성을 제공하는 방법』의 내용을 참조하십시오.
7. SSL 또는 TLS 클라이언트는 데이터 교환의 클라이언트 측이 완료되었음을 표시하는 비밀 키와 같이 암호화되는 "완료됨" 메시지를 서버에 송신합니다.
8. SSL 또는 TLS 서버는 데이터 교환의 서버 측이 완료되었음을 표시하는 비밀 키와 같이 암호화되는 "완료됨" 메시지를 클라이언트에 송신합니다.
9. 이제 SSL 또는 TLS 세션 동안 서버와 클라이언트는 공유 비밀 키로 대칭적으로 암호화되는 메시지를 교환할 수 있습니다.

16 페이지의 그림 5에서는 SSL 또는 TLS 데이터 교환에 대해 설명합니다.

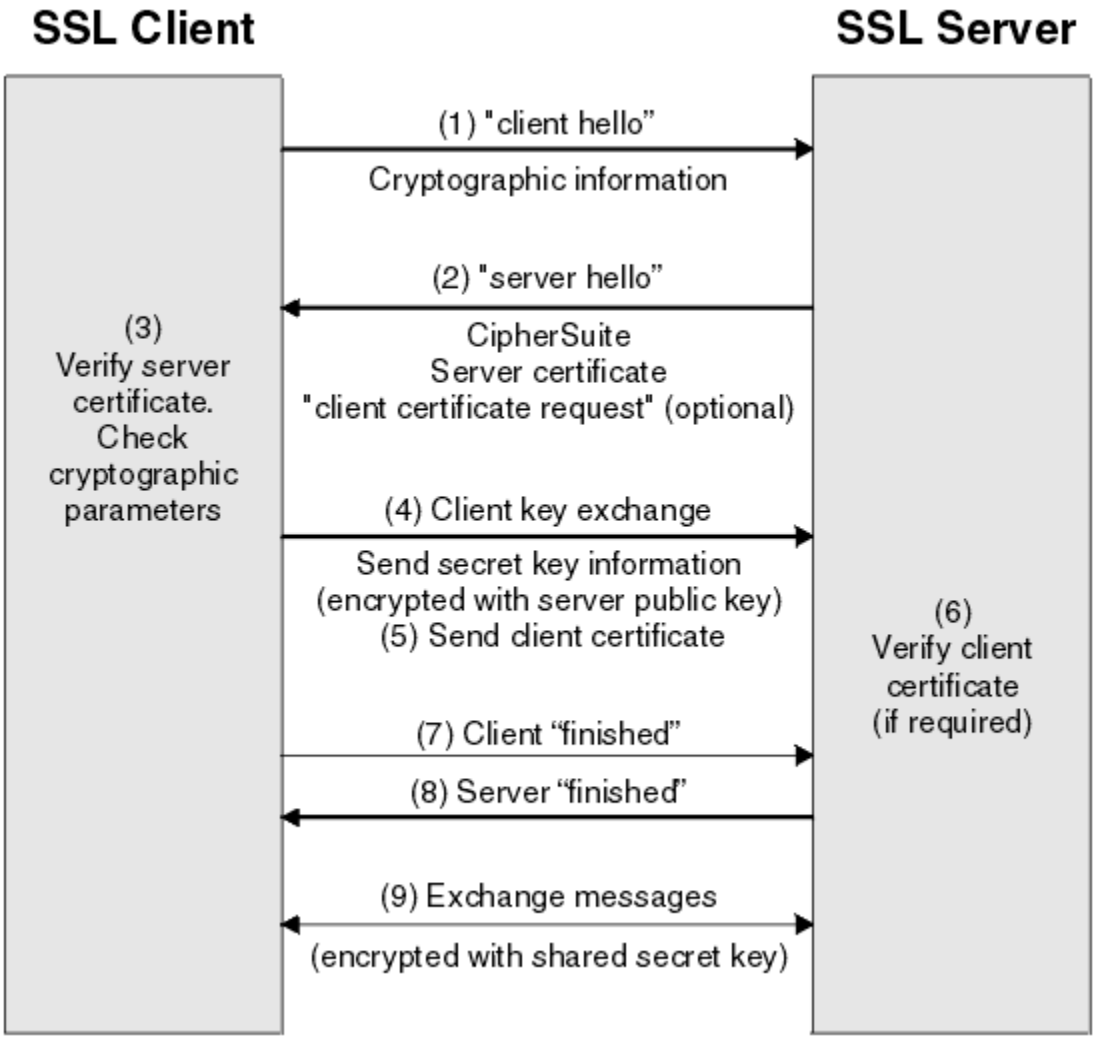



그림 *5. SSL* 또는 *TLS* 데이터 교환의 개요

### *SSL* 및 *TLS*가 *ID*, 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 핸드쉐이크에 관련된 단계에 대한 개요는 14 페이지의 『SSL 또는 TLS 데이터 교환의 개요』의 내용을 참조하십시오.

### SSL 및 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.

클라이언트 인증에서는 서버가 데이터 교환의 15 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 15 페이지의 『7』 및 15 페이지의 『8』)으로 인증이 완료되었음을 확인합니다.

임의 인증 단계가 실패하면, 데이터 교환이 실패하고 세션이 종료됩니다.

SSL 또는 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 SSL 또는 TLS 클라이언트에게 인증서를 발행하고, CA Y는 SSL 또는 TLS 서버에게 인증서를 발행합니다.

서버 인증의 경우에 한해, SSL 또는 TLS 서버는 다음을 필요로 합니다.

- CA Y에 의해 서버에게 발행된 개인 인증서

• 서버 개인 키

또한 SSL 또는 TLS 클라이언트는 다음을 필요로 합니다.

• CA Y에 대한 CA 인증서

SSL 또는 TLS 서버가 클라이언트 인증을 요구할 경우, 서버는 클라이언트에게 개인 인증서를 발행한 CA(이 경우, CA X)의 공개 키로 클라이언트의 디지털 인증서를 확인하고, 이를 통해 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

SSL 또는 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 15 페이지의 『3』 및 15 페이지의 『6』 단계에 기술된 대로 SSL 또는 TLS 클라이언트는 서버의 인증서를 확인하고 SSL 또는 TLS 서버는 클라이언트의 인증서를 확인합니다. 이 확인에 대한 네 가지 관점이 있습니다.

1. 디지털 서명을 점검합니다(18 페이지의 『SSL 및 TLS의 디지털 서명』 참조).
2. 인증서 체인을 검사합니다. 중간 CA 인증서가 있어야 합니다(12 페이지의 『인증서 체인 작동 방법』 참조).
3. 만기 및 활성화 날짜와 검증 기간을 검사합니다.
4. 인증서의 폐기 상태를 검사합니다(137 페이지의 『폐기된 인증서에 대한 작업』 참조).

## 보안 키 재설정

SSL 또는 TLS 데이터 교환 중에 SSL 또는 TLS 클라이언트와 서버 간의 데이터를 암호화하기 위해 비밀 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 9 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.

## SSL 및 TLS이 기밀성을 제공하는 방법

SL 및 TLS는 메시지 비공개를 보장하기 위해 대칭과 비대칭 암호화의 결합을 사용합니다. SSL 또는 TLS 데이터 교환 중에 SSL 또는 TLS 클라이언트와 서버는 하나의 세션에만 사용될 암호화 알고리즘과 공유 비밀 키에 동의합니다. SSL 또는 TLS 클라이언트와 서버 사이에 전송되는 모든 메시지는 해당 알고리즘과 키를 사용하여 암호화되며, 메시지가 인터셉트되어도 반드시 개인용으로 남아있게 됩니다. SSL은 광범위한 암호화 알고리즘을 지원합니다. SSL 및 TLS가 공유 비밀 키를 전송할 때는 비대칭 암호화를 사용하기 때문에, 키 분배 문제점이 없습니다. 암호화 기술에 대한 자세한 정보는 7 페이지의 『암호화』를 참조하십시오.

## SSL 및 TLS가 무결성을 제공하는 방법

SSL 및 TLS는 메시지 요약을 계산하여 데이터 무결성을 제공합니다. 자세한 정보는 207 페이지의 『메시지의 데이터 무결성』의 내용을 참조하십시오.

채널 정의의 CipherSpec이 199 페이지의 『CipherSpec 지정』의 표에 설명된 해시 알고리즘을 사용하는 경우 SSL 또는 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 또는 SSL 연결의 양 측은 동일한 CipherSpec이 통신 가능하도록 동의해야 합니다.

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

CipherSpec에 대한 자세한 정보는 199 페이지의 『CipherSpec 지정』의 내용을 참조하십시오.

CipherSuite는 SSL 또는 TLS 연결이 사용하는 암호화 알고리즘을 모아 놓은 것입니다. 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 또는 SSL 연결에 대해 지정되는 경우 특정 결합만 유효합니다. 유효한 CipherSuite의 이름은 사용되는 알고리즘의 결합을 정의합니다. 예를 들면, CipherSuite SSL_RSA_WITH_RC4_128_MD5는 다음을 지정합니다.

- RSA 키 교환과 인증 알고리즘
- 128비트 키를 사용하는 RC4 암호화 알고리즘
- MD5 MAC 알고리즘

키 교환과 인증을 위해 여러 알고리즘을 사용할 수 있으나, RSA 알고리즘이 현재 가장 많이 사용되고 있습니다. 암호화 알고리즘 및 MAC 알고리즘은 보다 다양하게 사용되고 있습니다.

### *SSL* 및 *TLS*의 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.

19 페이지의 그림 6에서 이 프로세스를 설명합니다.

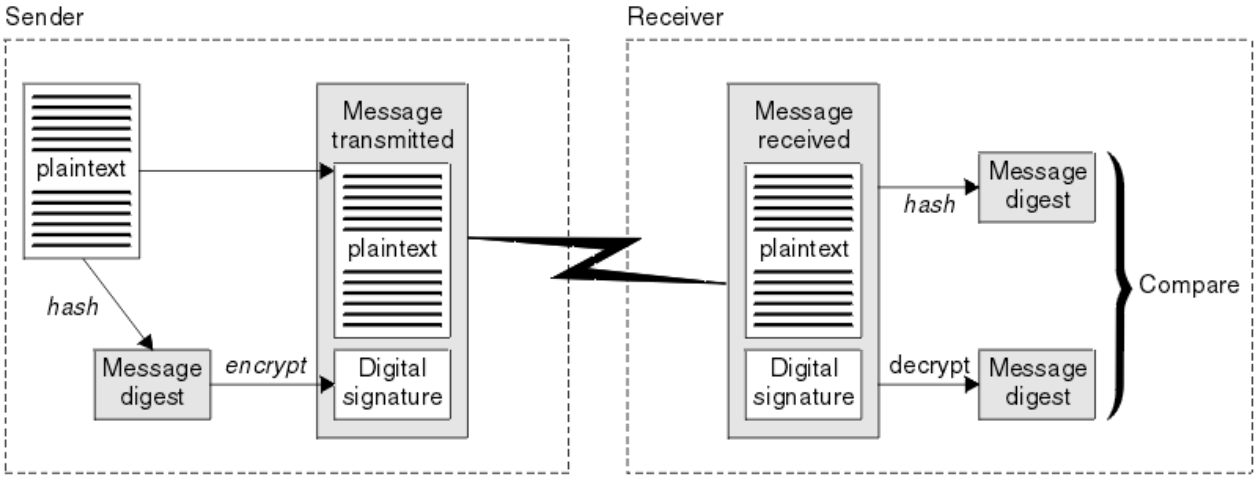



그림 *6.* 디지털 서명 프로세스

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.

IBM WebSphere 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.

### *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 버전 1.2에 사용하려는 스위트 B 준수 프로파일. 스위트 B 준수 조작에 대해 구성할 때 위에 나열된 제한된 세트의 암호화 알고리즘만이 사용됩니다.

2. TLS 버전 1.0 또는 TLS 버전 1.1에 사용하려는 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.

윈도우, 유닉스 및 Linux 시스템에서 WebSphere MQ은 Suite B 호환 TLS 1.2 프로파일을 준수하도록 구성할 수 있지만 Suite B 전환 프로파일을 지원하지 않습니다. 추가 정보에 대해서는 29 페이지의 『IBM WebSphere MQ에서의 NSA 스위트 B Cryptography』의 내용을 참조하십시오.

**관련 정보**

19 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 WebSphere MQ 보안 메커니즘

이 주제 모음에서는 IBM WebSphere MQ에서 다양한 보안 개념을 구현할 수 있는 방법에 대해 설명합니다.

IBM WebSphere MQ 는 5 페이지의 『보안 개념 및 메커니즘』에 소개된 모든 보안 개념을 구현하는 메커니즘을 제공합니다. 해당 내용은 다음 절에서 더 자세하게 설명됩니다.

## IBM WebSphere MQ의 식별 및 인증

IBM WebSphere MQ에서 메시지 컨텍스트 정보 및 상호 인증을 사용하여 식별 및 인증을 구현할 수 있습니다.

다음은 IBM WebSphere MQ 환경에서의 식별 및 인증에 대한 몇 개의 예제입니다.

- 모든 메시지에는 메시지 컨텍스트 정보가 있을 수 있습니다. 이 정보는 메시지 디스크립터에 보유됩니다. 이는 애플리케이션이 메시지를 큐에 넣을 때 큐 관리자에서 생성 가능합니다. 또는, 애플리케이션에 연관된 사용자 ID가 권한이 부여되어 있으면, 애플리케이션이 정보를 제공할 수 있습니다.

  메시지의 컨텍스트 정보를 사용하여 수신하는 애플리케이션이 메시지의 진원지에 대해 알 수 있습니다. 예를 들어, 메시지를 넣는 애플리케이션의 이름과 애플리케이션에 연관된 사용자 ID가 포함됩니다.
- 메시지 채널이 시작되면, 채널 각 측의 메시지 채널 에이전트(MCA)가 그 파트너를 인증하는 것이 가능합니다. 이 기술을 상호 인증이라고 합니다. MCA 전송의 경우, 메시지를 전송하는 파트너가 진짜 파트너이도록 보증합니다. 수신 MCA의 경우 이와 유사하게 진짜 파트너에서 메시지를 수신하도록 보증됩니다.

**관련 개념**

5 페이지의 『식별 및 인증』
식별은 시스템 사용자 또는 시스템에서 실행 중인 애플리케이션을 고유하게 식별하는 기능입니다. 인증은 사용자 또는 애플리케이션이 청구되는 진짜 사용자나 애플리케이션임을 증명해주는 기능입니다.

## IBM WebSphere MQ 의 권한

권한 부여를 사용하여 IBM WebSphere MQ 환경에서 수행할 수 있는 특정 개인 또는 응용프로그램을 제한할 수 있습니다.

다음은 IBM WebSphere MQ 환경에서의 권한의 예입니다.

- 권한 부여된 관리자만 IBM WebSphere MQ 자원을 관리하는 명령을 실행하도록 허용합니다.
- 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.

권한 설정 방법에 대한 자세한 정보는 45 페이지의 『권한 부여 계획』 및 관련 하위 주제를 참조하십시오.

**관련 개념**

6 페이지의 『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 WebSphere MQ에서 감사

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

다음은 IBM WebSphere 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 채널 열기를 시도하지만 SSL에서 연결을 허용하지 않기 때문에 실패합니다. 도구 이벤트 메시지가 발행됩니다. 이벤트 메시지를 검사하여 이 시도가 발행하는지 검색하고 필요한 조치를 결정할 수 있습니다.

**관련 개념**

6 페이지의 『감사』

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

## IBM WebSphere MQ 의 기밀성

메시지를 암호화하여 IBM WebSphere MQ에서 기밀성을 구현할 수 있습니다.

다음은 IBM WebSphere MQ 환경에서 기밀성이 구현되는 몇 가지 예입니다.

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

**관련 개념**

6 페이지의 『기밀성』

기밀성 서비스는 기밀 정보가 권한 없이 노출되는 것을 막습니다.

## IBM WebSphere MQ 의 데이터 무결성

메시지 수정 여부를 감지하기 위해 데이터 무결성 서비스를 사용할 수 있습니다.

다음은 IBM WebSphere MQ 환경에서 데이터 무결성을 보장하는 방법에 대한 몇 가지 예입니다.

- SSL 또는 TLS를 사용하여 메시지 컨텐츠가 네트워크를 통해 전송 중에 의도적으로 수정되었는지 감지할 수 있습니다. SSL 및 TLS에서 메시지 요약 알고리즘은 전송 중인 수정된 메시지 감지를 제공합니다. 모든 IBM WebSphere MQ CipherSpecs는 메시지 데이터 무결성을 제공하지 않는 TLS_RSA_WITH_NULL_NULL만 제외하고 메시지 요약 알고리즘을 제공합니다.
- 메시지가 로컬 큐에 저장되는 경우 IBM WebSphere MQ에서 제공되는 메시지 제어 메커니즘은 메시지 컨텐츠의 고의적인 수정을 보호하기에 충분합니다. 그렇지만 최고 레벨을 보안을 위해 IBM WebSphere MQ Advanced Message Security를 사용하여 큐에 메시지를 넣을 때부터 큐에서 검색되는 사이의 시간 동안 메시지 컨텐츠가 고의적으로 수정되었는지를 감지할 수 있습니다.

**관련 개념**

7 페이지의 『데이터 무결성』

데이터 무결성 서비스는 데이터가 권한 없이 수정되었는지를 감지합니다.

## IBM WebSphere MQ 의 암호화

IBM WebSphere MQ는 SSL(Secure Sockets Layer) 및 TLS(Transport Security Layer) 프로토콜을 사용하여 암호를 제공합니다.

자세한 정보는 22 페이지의 『IBM WebSphere MQ 의 보안 프로토콜』의 내용을 참조하십시오.

**관련 개념**

7 페이지의 『암호화 개념』
이 주제 콜렉션에서는 WebSphere MQ에 적용 가능한 암호화 개념을 설명합니다.

## IBM WebSphere MQ 의 보안 프로토콜

IBM WebSphere MQ는 메시지 채널 및 MQI 채널에 대한 링크 레벨의 보안을 제공하기 위해 TLS(Transport Layer Security) 및 SSL(Secure Sockets Layer) 프로토콜 모두를 지원합니다.

메시지 채널 및 MQI 채널은 SSL 또는 TLS 프로토콜을 사용하여 링크 레벨의 보안을 제공합니다.호출자 MCA는 SSL 또는 TLS 클라이언트이며 응답자 MCA는 SSL 또는 TLS 서버입니다. WebSphere MQ는 SSL 프로토콜 버전 3.0과 TLS(Transport Layer Security) 프로토콜 버전 1.0 및 버전 1.2를 지원합니다. 채널 정의의 일부로 CipherSpec을 제공하여 SSL 또는 TLS 프로토콜에서 사용하는 암호화 알고리즘을 지정하십시오.

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

큐 관리자 및 WebSphere MQ 클라이언트 사용자는 SSLCAUTH (REQUIRED) 가 채널의 서버 측에 지정되어 있지 않으면 SSL 또는 TLS 클라이언트로 작동할 때 이와 연관된 개인 디지털 인증서를 가질 필요가 없습니다.

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

**관련 개념**

14 페이지의 『암호화 보안 프로토콜: SSL 및 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 WebSphere MQ는 SSL과 TLS를 모두 지원합니다.

### *SSL* 및 *TLS*용 *IBM WebSphere MQ* 지원

IBM WebSphere MQ는 SSL(Secure Sockets Layer) 프로토콜 및 TLS(Transport Layer Security) 프로토콜을 모두 지원합니다.

SSL 및 TLS 프로토콜에 대한 자세한 정보는 관련 정보를 참조하십시오.

IBM WebSphere MQ는 SSL 버전 3.0 및 TLS 1.0 및 TLS 1.2에 다음과 같은 지원을 제공합니다.

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

**UNIX, Linux, and Windows 및 HP Integrity NonStop Server 시스템**
UNIX, Linux, and Windows 및 HP Integrity NonStop Server 시스템의 경우, SSL 및 TLS 지원은 IBM WebSphere MQ와 함께 설치됩니다.

IBM WebSphere MQ SSL및 TLS 지원의 전제조건에 대한 정보는 IBM WebSphere MQ에 대한 시스템 요구사항의 내용을 참조하십시오.

*SSL* 또는 *TLS* 키 저장소
서로 인증된 SSL 또는 TLS 연결의 경우 연결의 각 끝에 키 저장소(여러 플랫폼에서 다양한 이름으로 확인할 수 있음)가 있어야 합니다. 키 저장소는 디지털 인증서와 개인 키를 포함합니다.

이 정보는 일반 용어 키 저장소를 사용하여 디지털 인증서 및 해당 관련 개인 키에 대한 저장소를 설명합니다. SSL 및 TLS를 지원하는 플랫폼과 환경에서 사용되는 특정 저장소 이름은 다음과 같습니다.

| | |
|---|---|
| Java및 JMS | 키 저장소 및 신뢰 저장소 |
| Windows UNIX<br>Linux Windows<br>UNIX Linux<br>Windows, UNIX and Linux 시스템 | 키 데이터베이스 파일 |

자세한 정보는 9 페이지의 『디지털 인증서』 및 14 페이지의 『SSL(Secure Socket Layer) 및 TLS(Transport Layer Security) 개념』을 참조하십시오.

상호 인증된 SSL 또는 TLS 연결에는 각 연결 끝에 키 저장소가 필요합니다. 키 저장소에는 다음이 포함될 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. 개별 개인 인증서를 각 큐 관리자나 WebSphere MQ MQI 클라이언트와 연관시킵니다. 개인 인증서는 상호 인증이 필요한 경우 SSL 또는 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 관한 자세한 정보는 23 페이지의 『IBM WebSphere MQ 키 저장소 보호』의 내용을 참조하십시오.

키 저장소의 위치는 사용하고 있는 플랫폼에 따라 다릅니다.

Windows UNIX Linux **Windows, UNIX and Linux 시스템**

UNIX and Linux 시스템은 키 저장소가 키 데이터베이스 파일입니다. 키 데이터베이스 파일의 이름은 `.kdb` 파일 확장자를 사용해야 합니다. 예를 들어, UNIX and Linux에서 큐 관리자 `QM1` 의 기본 키 데이터베이스 파일은 `/var/mqm/qmgrs/QM1/ssl/key.kdb` 입니다. IBM WebSphere MQ 가 기본 위치에 설치된 경우 윈도우 에서 해당 경로는 `C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\key.kdb`입니다.

Windows, UNIX and Linux 시스템에서 각 키 데이터베이스 파일에는 연관된 비밀번호 숨김(stash) 파일이 있습니다. 이 파일은 프로그램이 키 데이터베이스에 액세스하도록 허용하는 인코딩된 비밀번호를 보유합니다. 비밀번호 숨김(stash) 파일은 키 데이터베이스와 동일한 디렉토리에 있어야 하고 동일한 파일 스템을 가져야 하며 접미부 `.sth`로 끝나야 합니다(예: `/var/mqm/qmgrs/QM1/ssl/key.sth`).

**참고:** UNIX and Linux 시스템에서 PKCS #11 암호화 하드웨어 카드에는 키 데이터베이스 파일에 있는 인증서 및 키가 포함될 수 있습니다. 인증서와 키가 PKCS #11 카드에 있으면, WebSphere MQ가 여전히 키 데이터베이스 파일과 비밀번호 숨김(stash) 파일에 모두 액세스해야 합니다.

Windows 및 UNIX 시스템에서는 키 데이터베이스에 큐 관리자 또는 WebSphere MQ MQI 클라이언트와 연관된 개인 인증서의 개인 키도 포함될 수 있습니다.

*IBM WebSphere MQ* 키 저장소 보호

IBM WebSphere 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 WebSphere 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.

### 큐 관리자의 키 저장소 새로 고치기

키 저장소 컨텐츠를 변경하는 경우 큐 관리자는 즉시 새 컨텐츠를 선택하지 않습니다. 큐 관리자가 새 키 저장소 컨텐츠를 사용하도록 하려면 REFRESH SECURITY TYPE(SSL) 명령을 발행해야 합니다.

이 프로세스는 의도적이며 여러 개의 실행 채널이 다른 버전의 키 저장소를 사용하는 상황을 방지합니다. 보안 제어로 언제든지 큐 관리자는 한 개 버전의 키 저장소만 로드할 수 있습니다.

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

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

**관련 개념**

24 페이지의 『SSL 키 저장소 컨텐츠 및 SSL 설정의 클라이언트 보기 새로 고치기』
키 저장소에 대해 새로 고친 컨텐츠가 포함된 클라이언트 애플리케이션을 업데이트하려면 클라이언트 애플리케이션을 중지한 후에 다시 시작해야 합니다.

### *SSL* 키 저장소 컨텐츠 및 *SSL* 설정의 클라이언트 보기 새로 고치기

키 저장소에 대해 새로 고친 컨텐츠가 포함된 클라이언트 애플리케이션을 업데이트하려면 클라이언트 애플리케이션을 중지한 후에 다시 시작해야 합니다.

WebSphere MQ 클라이언트에서 보안을 새로 고칠 수 없습니다. 클라이언트에 대해 REFRESH SECURITY TYPE(SSL) 명령과 동등한 명령이 없습니다. 자세한 정보는 REFRESH SECURITY를 참조하십시오.

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**

24 페이지의 『큐 관리자의 키 저장소 새로 고치기』
키 저장소 컨텐츠를 변경하는 경우 큐 관리자는 즉시 새 컨텐츠를 선택하지 않습니다. 큐 관리자가 새 키 저장소 컨텐츠를 사용하도록 하려면 REFRESH SECURITY TYPE(SSL) 명령을 발행해야 합니다.

### *FIPS(Federal Information Processing Standard)*

이 토픽에서는 미국 국립 표준 기술원(US National Institute of Standards and Technology)의 FIPS(Federal Information Processing Standards) Cryptomodule 유효성 검증 프로그램 및 Windows, UNIX and Linux 및 z/OS 시스템의 경우 SSL 또는 TLS 채널에 사용할 수 있는 암호화 기능에 대해 소개합니다.

UNIX, Linux및 Windows 시스템에서 IBM WebSphere MQ SSL 또는 TLS 연결의 FIPS 140-2준수가 여기 25 페이지의 『 UNIX, Linux및 Windows 용 FIPS (Federal Information Processing Standards)』에 있습니다.

암호화 하드웨어가 있는 경우 IBM WebSphere 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 WebSphere MQ도 최신 표준 구현으로 업데이트됩니다. 그 결과 유지보수를 적용한 후에는 동작이 변경됨을 알 수 있습니다.

**관련 개념**

99 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.

105 페이지의 『iKeyman, iKeycmd, runmqakm 및 runmqckm 사용』

UNIX, Linux 및 Windows 시스템에서는 iKeyman GUI를 사용하거나 명령행에서 iKeycmd 또는 runmqakm을 사용하여 키 및 디지털 인증서를 관리합니다.

**관련 태스크**

Java용 WebSphere MQ 클래스에서 SSL 사용

JMS용 WebSphere MQ 클래스와 함께 SSL(Secure Sockets Layer) 사용

**관련 참조**

JMS 오브젝트의 SSL 특성

**관련 정보**

19 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.

*UNIX, Linux*및 *Windows* 용 *FIPS (Federal Information Processing Standards)*

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

Windows, UNIX and Linux 시스템에서 WebSphere MQ SSL 또는 TLS 연결의 FIPS 140-2준수는 다음과 같습니다.

- 다음과 같은 조건이 충족되면 모든 IBM WebSphere MQ 메시지 채널(CLNTCONN 채널 유형 제외)에 대해 연결은 FIPS를 준수합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2 준수가 인증되었습니다.
  - 큐 관리자의 SSLFIPS 속성이 YES로 설정되었습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
- 모든 IBM WebSphere MQ MQI 클라이언트 애플리케이션의 경우, 연결은 GSKit을 사용하며 다음 조건이 충족되는 경우 FIPS를 준수합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2 준수가 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.
- 클라이언트 모드를 사용하는 Java 애플리케이션용 IBM WebSphere MQ 클래스의 경우, 연결은 JRE의 SSL및 TLS 구현을 사용하며 다음 조건이 충족되면 FIPS를 준수합니다.
  - 애플리케이션을 실행하는 데 사용되는 JRE (Java Runtime Environment) 는 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS를 준수합니다.
  - Java 클라이언트의 관련 주제에 설명된 대로 FIPS 인증 암호화만 사용하도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
- 클라이언트 모드를 사용하는 JMS 애플리케이션용 IBM WebSphere MQ 클래스의 경우, 연결은 JRE의 SSL 및 TLS 구현을 사용하며 다음 조건이 충족되는 경우 FIPS를 준수합니다.
  - 애플리케이션을 실행하는 데 사용되는 JRE (Java Runtime Environment) 는 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS를 준수합니다.
  - JMS 클라이언트에 대한 관련 항목에 설명된 대로 FIPS 인증 암호화만이 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.

- 관리되지 않는 .NET 클라이언트 애플리케이션에 대해 다음 조건이 충족되는 경우 연결은 FIPS를 준수하는 GSKit을 사용합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2 준수가 인증되었습니다.
  - .NET 클라이언트에 대한 관련 항목에 설명된 대로 FIPS 인증 암호화만이 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.
- 관리되지 않는 XMS 및 .NET 클라이언트 애플리케이션에 대해 다음 조건이 충족되는 경우 연결은 FIPS를 준수하는 GSKit을 사용합니다.
  - 설치된 GSKit ICC 버전은 설치된 운영 체제 버전 및 하드웨어 아키텍처에서 FIPS 140-2 준수가 인증되었습니다.
  - XMS .NET 문서에 설명된 대로 FIPS 인증 암호화만이 사용되도록 지정했습니다.
  - 모든 키 저장소가 FIPS 준수 소프트웨어(예: `-fips` 옵션이 지정된 **runmqakm**)만을 사용하여 작성되고 조작됩니다.

지원되는 모든 AIX®, Linux, HP-UX, Solaris, Windows및 z/OS 플랫폼은 각 수정팩 또는 Refresh Pack에 포함된 readme 파일에 명시된 경우를 제외하고 FIPS 140-2인증 플랫폼입니다.

GSKit을 사용하는 SSL 및 TLS 연결에 대해 FIPS 140-2 인증된 컴포넌트 이름은 *ICC*입니다. 이는 지정된 모든 플랫폼에서 GSKit FIPS 준수를 판별하는 이 컴포넌트 버전입니다. 현재 설치된 ICC 버전을 판별하려면 **dspmqver -p 64 -v** 명령을 실행하십시오.

다음은 ICC에 관련된 **dspmqver -p 64 -v** 출력에 대한 추출 예입니다.

```
  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 WebSphere MQ에서 사용되는 암호화 모듈은 하드웨어 생산자가 제공하는 모듈로 구성 가능합니다. 모듈 구성을 완료하면 구성이 해당 암호화 모듈이 FIPS에서 인증된 경우에만 FIPS를 준수합니다.

**참고:** FIPS 140-2 준수 조작을 위해 구성된 32비트 Solaris x86 SSL 및 TLS 클라이언트는 Intel 시스템에서 실행하는 경우 실패합니다. FIPS 140-2 준수 GSKit-Crypto Solaris x86 32비트 라이브러리 파일은 Intel 칩 세트에 로드되지 않기 때문에 이와 같은 실패가 발생합니다. 영향받는 시스템에서 오류 AMQ9655가 클라이언트 오류 로그에 보고됩니다. 이 문제를 해결하려면 FIPS 140-2 준수를 사용 안함으로 설정하거나, 64비트 코드는 영향을 받지 않으므로 클라이언트 애플리케이션 64비트를 재컴파일하십시오.

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

WebSphere MQ가 FIPS 140-2를 준수하여 작동하도록 구성되어 있을 때에는 3중 DES(3DES) CipherSpecs에 관련하여 추가 제한이 시행됩니다. 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 WebSphere MQ는 시크릿 세션 키를 재설정하지 않으므로 이 재설정을 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 205 페이지의 『SSL 및 TLS 비밀 키 재설정』의 내용을 참조하십시오.

WebSphere MQ 는 이미 규칙 1및 2를 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**

99 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.

105 페이지의 『iKeyman, iKeycmd, runmqakm 및 runmqckm 사용』
UNIX, Linux 및 Windows 시스템에서는 iKeyman GUI를 사용하거나 명령행에서 iKeycmd 또는 runmqakm을 사용하여 키 및 디지털 인증서를 관리합니다.

**관련 태스크**

Java용 WebSphere MQ 클래스에서 SSL 사용

JMS용 WebSphere MQ 클래스와 함께 SSL(Secure Sockets Layer) 사용

**관련 참조**

JMS 오브젝트의 SSL 특성

**관련 정보**

19 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 WebSphere MQ MQI* 클라이언트의 *SSL*및 *TLS*
IBM WebSphere MQ는 클라이언트에서 SSL 및 TLS를 지원합니다. 다양한 방식으로 SSL 또는 TLS 사용을 조정할 수 있습니다.

IBM WebSphere MQ 는 UNIX and Linux 시스템에서 IBM WebSphere MQ MQI 클라이언트에 대한 SSL및 TLS 지원을 제공합니다. Java에 대해 IBM WebSphere MQ 클래스를 사용하는 경우 WebSphere MQ 을 참조하십시오. JMS에 대해 IBM WebSphere MQ 클래스를 사용하는 경우에는 JMS용 WebSphere MQ 클래스 사용을 참조하십시오. 이 섹션의 나머지 부분은 Java 또는 JMS 환경에 적용되지 않는다.

애플리케이션에서 MQCONNX를 호출할 때나 IBM WebSphere MQ 클라이언트 구성 파일에 있는 MQSSLKEYR 값을 사용하여 IBM WebSphere MQ MQI 클라이언트에 대한 키 저장소를 지정할 수 있습니다. 세 가지 옵션을 사용하여 채널이 SSL을 사용하도록 지정할 수 있습니다.

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

MQSERVER 환경 변수를 사용해서는 채널이 SSL을 사용하도록 지정할 수 없습니다.

SSL이 채널의 반대 측에 지정되지 않은 경우에는 SSL 없이 기존 IBM WebSphere MQ MQI 클라이언트 애플리케이션을 계속 실행할 수 있습니다.

클라이언트 시스템에서 SSL 키 저장소의 컨텐츠, SSL 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 SSL 연결을 종료해야 합니다. 모든 연결이 종료되면 SSL 채널을 재시작하십시오. 모든 새 SSL 설정이 사용됩니다. 이 설정은 큐 관리자 시스템에서 REFRESH SECURITY TYPE(SSL) 명령으로 새로 고치기되는 설정과 유사합니다.

IBM WebSphere MQ MQI 클라이언트가 암호화 하드웨어로 윈도우 UNIX and Linux 시스템에서 실행되는 경우, MQSSLCRYP 환경 변수로 해당 하드웨어를 구성합니다. 이 변수는 ALTER QMGR MQSC 명령에서의 SSLCRYP 매개변수와 동일합니다. ALTER QMGR MQSC 명령의 SSLCRYP 매개변수에 대한 설명은 ALTER QMGR을 참조하

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

SSL 비밀 키 재설정 및 FIPS는 IBM WebSphere MQ MQI 클라이언트에서 지원됩니다. 자세한 정보는 205 페이지의 『SSL 및 TLS 비밀 키 재설정』 및 25 페이지의 『 UNIX, Linux및 Windows 용 FIPS (Federal Information Processing Standards)』의 내용을 참조하십시오.

IBM WebSphere MQ MQI 클라이언트에 대한 SSL 지원에 대한 자세한 정보는 99 페이지의 『IBM WebSphere MQ MQI 클라이언트 보안 설정』 를 참조하십시오.

**관련 태스크**

구성 파일을 사용하여 클라이언트 구성

*MQI* 채널이 *SSL*을 사용하도록 지정

MQI 채널이 SSL을 사용하도록 클라이언트 연결 채널의 *SSLCipherSpec* 속성 값은 클라이언트 플랫폼의 IBM WebSphere 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. WebSphere MQ MQI 클라이언트 애플리케이션이 MQCONNX 호출을 발행하는 경우입니다.

   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 활성 디렉토리에서 클라이언트 연결 채널 정의를 발행할 수 있습니다. 이 정의 중 하나 이상은 CipherSpec 이름을 지정할 수 있습니다.

예를 들어, 클라이언트 애플리케이션이 MQCONNX 호출의 MQCD 구조에 클라이언트 연결 채널 정의를 제공하는 경우 WebSphere MQ 클라이언트가 액세스할 수 있는 클라이언트 채널 정의 테이블에 있는 모든 항목에 우선하여 이 정의가 사용됩니다.

SSL을 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**

204 페이지의 『IBM WebSphere MQ MQI 클라이언트에 대해 CipherSpec 지정』
IBM WebSphere MQ MQI 클라이언트에 대해 CipherSpec 을 지정하는 세 가지 옵션이 있습니다.

*IBM WebSphere MQ* 의 *CipherSpecs* 및 *CipherSuites*

IBM WebSphere MQ는 SSL과 TLS CipherSpecs 및 RSA와 Diffie-Hellman 알고리즘을 모두 지원합니다.

WebSphere MQ는 SSL V3, TLS V1.0 및 V1.2 CipherSpec을 지원합니다.

WebSphere MQ는 RSA와 Diffie-Hellman 키 교환 및 인증 알고리즘을 지원합니다. SSL 데이터 교환 중 사용되는 키의 크기는 사용자가 사용하는 디지털 인증서에 따라 결정될 수 있지만 일부 CiperSpec은 데이터 교환 키 크기의 스펙을 포함합니다. 데이터 교환 키 크기가 클수록 보다 강력한 인증이 제공됩니다. 키 크기가 작아지면 데이터 교환이 보다 빨라집니다.

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

*IBM WebSphere MQ*에서의 *NSA* 스위트 *B Cryptography*
이 주제에서는 Windows, Linux및 UNIX 시스템에서 Suite B 호환 TLS 1.2 프로파일을 준수하도록 IBM WebSphere MQ 를 구성하는 방법에 대한 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 WebSphere MQ도 최신 표준 구현으로 업데이트됩니다. 그 결과 유지보수를 적용한 후에는 동작이 변경됨을 알 수 있습니다. IBM WebSphere MQ Version 7.5 Readme 파일은 각 제품 유지보수 레벨에서 적용된 스위트 B의 버전을 나열합니다. 스위트 B 준수를 적용하도록 IBM WebSphere MQ를 구성하는 경우, 유지보수를 적용할 계획인 경우 항상 readme 파일을 참조하십시오(IBM MQ, WebSphere MQ 및 MQSeries 제품 READMEs 참조).

윈도우, 유닉스및 Linux 시스템에서 IBM WebSphere MQ 는 테이블 1에 표시된 보안 레벨에서 Suite B 호환 TLS 1.2 프로파일을 준수하도록 구성할 수 있습니다.

| 표 *1.* 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 WebSphere MQ 의 기본 동작은 Suite B 표준을 준수하지 않지만 윈도우, 유닉스 및 Linux 시스템의 보안 레벨 중 하나 또는 둘 다에 맞도록 IBM WebSphere MQ 를 구성할 수 있습니다. 스위트 B를 사용하도록 IBM WebSphere 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.

WebSphere MQ CipherSpec에 대한 자세한 정보는 199 페이지의 『CipherSpec 지정』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. 그러므로 원격 파트너의 구성된 스위트 B 보안 레벨에서 허용되는 디지털 서명 알고리즘 및 공개 키 유형이 있는 인증서를 사용하도록 WebSphere MQ를 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 runmqckm 및 runmqakm 옵션을 참조하십시오.

**참고: iKeycmd** 및 **iKeyman** 도구는 스위트 B 준수 조작에 대한 디지털 인증서 작성을 지원하지 않습니다.

## 디지털 인증서 작성 및 요청

스위트 B 테스트를 위해 자체 서명 디지털 인증서를 작성하려는 경우에는 111 페이지의 『UNIX, Linux, and Windows 시스템에서 자체 서명한 개인 인증서 작성』의 내용을 참조하십시오.

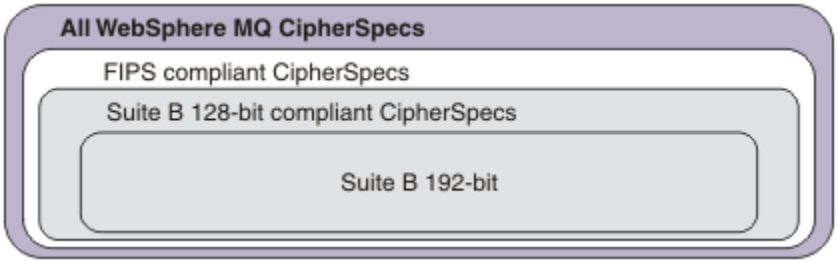
스위트 B 프로덕션 사용을 위해 CA 서명 디지털 인증서를 요청하려는 경우에는 114 페이지의 『UNIX, Linux, and Windows 시스템에서 개인 인증서 요청』의 내용을 참조하십시오.

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

## FIPS 140-2 및 스위트 B

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 WebSphere MQ가 FIPS 140-2 준수 조작용으로 구성된 경우에 사용할 수 있습니다. 따라서 두 가지 제한사항 세트가 적용되는 경우 FIPS 및 스위트 B를 동시에 준수하도록 WebSphere MQ를 구성할 수 있습니다.

다음 다이어그램은 이러한 서브세트 간의 관계를 보여줍니다.




## 스위트 B 준수 조작을 위해 WebSphere MQ 구성

Suite B 준수 조작에 대해 Windows, UNIX및 Linux 에서 IBM WebSphere MQ 를 구성하는 방법에 대한 정보는 30 페이지의 『스위트 B용 IBM WebSphere MQ 구성』의 내용을 참조하십시오.

IBM WebSphere MQ는 IBM i 및 z/OS 플랫폼에서 스위트 B 준수 조작을 지원하지 않습니다. WebSphere MQ Java 및 JMS 클라이언트도 스위트 B 준수 조작을 지원하지 않습니다.

**관련 개념**

99 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.

스위트 *B*용 *IBM WebSphere MQ* 구성
IBM WebSphere MQ는 UNIX, Linux, and Windows 시스템에 대한 NSA 스위트 B 표준에 맞게 작동하도록 구성할 수 있습니다.

스위트 B는 보증된 보안 레벨을 제공하기 위해서 사용 가능한 암호화 알고리즘 세트를 제한합니다.IBM WebSphere MQ 는 향상된 레벨의 보안을 제공하기 위해 스위트 B에 순응하여 동작하도록 구성될 수 있다. 스위

트 B에 대한 추가 정보는 19 페이지의 『NSA(National Security Agency) 스위트 B 암호화』 의 내용을 참조하십시오. 스위트 B 구성 및 SSL 및 TLS 채널에 대한 해당 영향의 자세한 정보는 29 페이지의 『IBM WebSphere 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. 추가 정보는 ALTER QMGR을 참조하십시오.

또한 **MQIA_SUITE_B_STRENGTH** 매개변수와 함께 PCF **MQCMD_CHANGE_Q_MGR** 명령을 사용하여 스위트 B 준수 조작을 위한 큐 관리자를 구성할 수도 있습니다.

## MQI 클라이언트

기본적으로 MQI 클라이언트는 스위트 B 준수를 강제 적용하지 않습니다. 다음 옵션 중 하나를 실행하여 MQI 클라이언트에서 스위트 B 준수를 사용하도록 설정할 수 있습니다.

1. MQCONNX 호출의 MQSCO 구조에 **EncryptionPolicySuiteB** 필드를 설정하여 아래의 값 중 하나 이상을 호출하십시오.
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

   다음 값이 포함된 MQ_SUITE_B_NONE 사용은 올바르지 않습니다.
2. MQSUITEB 환경 변수를 다음 값 중 하나 이상으로 설정:
   - NONE
   - 128_BIT
   - 192_BIT

   쉼표로 구분된 목록을 사용하여 여러 개의 값을 지정할 수 있습니다. 다른 값이 포함된 NONE 값 사용은 올바르지 않습니다.
3. MQI 클라이언트 구성 파일의 SSL 스탠자에 **EncryptionPolicySuiteB** 속성을 다음 값 중 하나 이상으로 설정하십시오.
   - NONE
   - 128_BIT
   - 192_BIT

   쉼표로 구분된 목록을 사용하여 여러 개의 값을 지정할 수 있습니다. 어떠한 다른 값이 포함된 NONE 사용도 올바르지 않습니다.

**참고:** MQI 클라이언트 설정은 우선순위대로 나열됩니다. MQCONNX 호출의 MSCO 구조는 MQSUITED 환경 변수의 설정을 대체하며 이는 SSL 스탠자 속성을 대체합니다.

MQSCO 구조의 전체 설명은 MQSCO - SSL 구성 옵션을 참조하십시오.

클라이언트 구성 파일에서 스위트 B의 사용에 대한 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자 를 참조하십시오.

MQSUITED 환경 변수 사용에 관한 추가 정보는 환경 변수를 참조하십시오.

## .NET

.NET 관리되지 않는 클라이언트의 경우 **MQC.ENCRYPTION_POLICY_SUITE_B** 특성은 필요한 스위트 B 보안 유형을 표시합니다.

.NET용 IBM WebSphere MQ 클래스에서 스위트 B 사용에 대한 정보는 MQEnvironment .NET 클래스를 참조하십시오.

*IBM WebSphere 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 Java 런타임 환경의 JSSE 컴포넌트에 따라 인증서 유효성 검증 정책이 다릅니다. 인증서 유효성 검증 정책에 대한 자세한 정보는 JRE에 대한 문서를 참조하십시오.
- UNIX, Linux, and Windows 시스템의 경우 인증서 유효성 검증 정책은 GSKit에서 제공되며 이는 구성 가능합니다. 두 개의 서로 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.

UNIX, Linux, and Windows 시스템에서 인증서 유효성 검증 정책 구성 방법에 대한 정보는 32 페이지의 『IBM WebSphere MQ에서 인증서 유효성 검증 정책 구성』 의 내용을 참조하십시오. 기본 및 표준 인증 유효성 검증 정책 간의 차이점에 대한 자세한 정보는 UNIX, Linux 및 시스템 인증서 유효성 검증 및 신뢰 정책 설계 를 참조하십시오.

*IBM WebSphere MQ*에서 인증서 유효성 검증 정책 구성
네 가지 방법으로 원격 파트너 시스템에서 수신된 디지털 인증서를 유효성 검증하는 데 사용되는 SSL/TLS 인증서 유효성 검증 정책을 지정할 수 있습니다.

큐 관리자에서 인증서 유효성 검증 정책은 다음 방법으로 설정할 수 있습니다.

- 큐 관리자 속성 *CERTVPOL*을 사용. 이 속성 설정에 대한 자세한 정보는 ALTER QMGR 을 참조하십시오.

클라이언트에서는 인증서 유효성 검증 정책을 설정하는 데 사용할 수 있는 몇 가지 방법이 있습니다. 정책을 설정하는 데 둘 이상의 방법이 사용되는 경우 클라이언트는 다음 우선순위에 따라 설정을 사용합니다.

1. 클라이언트 MQSCO 구조의 *CertificateValPolicy* 필드 사용. 이 필드를 사용하는 데 대한 자세한 정보는 MQSCO - SSL 구성 옵션을 참조하십시오.
2. 클라이언트 환경 변수 *MQCERTVPOL* 사용. 이 변수 사용에 대한 자세한 정보는 MQCERTVPOL 을 참조하십시오.
3. 클라이언트 SSL 스탠자 성능 조정 매개변수 설정 *CertificateValPolicy* 사용. 이 설정 사용에 대한 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자 를 참조하십시오.

인증서 유효성 검증 정책에 대한 자세한 정보는 32 페이지의 『IBM WebSphere MQ 의 인증서 유효성 검증 정책』 의 내용을 참조하십시오.

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

이전 릴리스의 IBM WebSphere MQ에서는 지원되는 모든 SSL 및 TLS CipherSpec이 디지털 서명 및 키 계약에 대해 RSA 알고리즘을 사용했습니다. 디지털 인증서의 지원되는 모든 유형은 지원되는 모든 CipherSpec과 호환되었기 때문에 디지털 인증서를 변경하지 않고도 모든 채널에 대해 CipherSpec을 변경할 수 있었습니다.

IBM WebSphere MQ v7.5에서는 지원되는 모든 유형의 디지털 인증서에 지원되는 CipherSpec의 서브세트만을 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 신규 CipherSpec을 사용하려는 경우, 디지털 인증서가 해당 디지털 서명에 MD5 알고리즘을 사용하지 않도록 해야 합니다. SSL 3.0 및 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 보려면 **runmqakm** 명령을 사용하면 됩니다.

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

여기서 cert_label은 표시해야 하는 디지털 서명 알고리즘의 인증서 레이블입니다.

**참고: iKeycmd (runmqckm)** 도구와 **iKeyman (strmqikm)** GUI를 사용하여 디지털 서명 알고리즘의 선택사항을 볼 수 있지만 **runmqakm** 도구는 더 광범위한 범위를 제공합니다.

**runmqakm** 명령을 실행하면 지정된 서명 알고리즘의 사용을 표시하는 출력이 생성됩니다.

```
Label : ibmwebspheremqexample
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. CipherSpec 이름 접두부로 표시되는 세 가지 유형의 CipherSpec이 있습니다. 각 유형의 CipherSpec은 사용 가능한 디지털 인증서의 유형별로 다른 제한을 시행합니다. 이 제한사항은 모든 WebSphere MQ SSL 및 TLS 연결에 적용되지만 특히 Elliptic Curve 암호화의 사용자와 관련이 있습니다.

CipherSpec 및 디지털 인증서 간의 관계는 다음 테이블에 요약되어 있습니다.

표 *2. 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 및 Elliptic Curve 비밀 키 및 Elliptic Curve 디지털 서명 알고리즘을 사용하는 CipherSpec | RSA | RSA | ECDHE |
| 3 | (나머지 모두) | RSA 공개 키 및 RSA 디지털 서명 알고리즘을 사용하는 CipherSpec. | RSA | RSA | RSA |

**참고:** 유형 1 및 2 CipherSpec은 UNIX, Linux, and Windows 플랫폼의 WebSphere MQ 큐 관리자 및 MQI 클라이언트에서만 지원됩니다.

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.

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 WebSphere MQ가 FIPS 140-2 또는 NSA 스위트 B 표준을 준수하도록 구성된 경우 허용 가능한 구성의 범위로 제한됩니다. 자세한 정보는 다음 절을 참조하십시오.

WebSphere MQ 큐 관리자는 단일 개인 인증서만을 사용하여 해당 큐 관리자를 식별할 수 있습니다. 이는 큐 관리자의 모든 채널이 동일한 디지털 인증서를 사용하므로 각 큐 관리자는 한 번에 한 유형의 CipherSpec만을 사용할 수 있음을 의미합니다. 이와 비슷하게 WebSphere MQ 클라이언트 애플리케이션은 단일 개인 인증서만을 사용하여 해당 큐 관리자를 식별할 수 있습니다. 이는 단일 애플리케이션 프로세스에 있는 모든 SSL 및 TLS 연결이 동일한 디지털 인증서를 사용하므로 각 클라이언트 애플리케이션 프로세스는 한 번에 한 유형의 CipherSpec만을 사용할 수 있음을 의미합니다.

세 가지 유형의 CipherSpec은 직접 상호 운용되지 않으며 이는 현재 SSL 및 TLS 표준의 제한사항입니다. 예를 들어, QM1 큐 관리자에서 TO.QM1 수신자 채널에 대해 ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec을 사용하도록 선택했다고 가정해 보십시오. ECDHE_ECDSA_AES_128_CBC_SHA256은 유형 1 CipherSpec이므

로 QM1에는 Elliptic Curve 키 및 ECDSA 기반 디지털 서명이 있는 개인 인증서가 있어야 합니다. 따라서 QM1과 직접 통신하는 모든 클라이언트 및 기타 큐 관리자에는 유형 1 CipherSpec의 요구사항을 충족하는 디지털 서명이 있어야 합니다. 큐 관리자 QM1과 연결되는 다른 채널은 다른 CipherSpec(예: ECDHE_ECDSA_3DES_EDE_CBC_SHA256)을 사용할 수 있지만 QM1과 통신하려면 유형 1 CipherSpec만을 사용할 수 있습니다.

WebSphere MQ 네트워크를 계획할 때 어떤 채널에서 SSL 또는 TLS가 필요한지 주의깊게 고려하고 상호 운용해야 할 모든 클라이언트 및 큐 관리자가 동일한 유형의 CipherSpec 및 적절한 디지털 인증서를 사용하는지 확인하십시오. IETF 표준 RFC 4492, RFC 5246 및 RFC 6460은 TLS 1.2에서 Elliptic Curve CipherSpec의 자세한 사용법에 대해 설명합니다.

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.

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

```
Label : ibmwebspheremqexample
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.

동일한 매개변수를 사용하여 **iKeycmd (runmqckm)** 도구를 사용할 수도 있습니다. 또한, **iKeyman (strmqikm)** GUI는 키 저장소를 열고 인증서 레이블을 두 번 클릭하는 경우 디지털 서명 알고리즘 확인에 사용할 수도 있습니다. 그렇지만 **runmqakm** 도구를 사용하여 디지털 인증서를 보도록 권장되는데 이는 광범위한 범위의 알고리즘을 지원하기 때문입니다.

## Elliptic Curve CipherSpec 및 NSA 스위트 B

스위트 B 준수 TLS 1.2 프로파일을 준수하도록 WebSphere MQ가 구성된 경우 29 페이지의 『IBM WebSphere MQ에서의 NSA 스위트 B Cryptography』에서 설명된 대로 허용되는 CipherSpec 및 디지털 서명 알고리즘이 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 *3.* 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**

199 페이지의 『CipherSpec 지정』
**DEFINE CHANNEL** MQSC 명령 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec 을 지정하십시오.

99 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.

29 페이지의 『IBM WebSphere MQ에서의 NSA 스위트 B Cryptography』
이 주제에서는 Windows, Linux및 UNIX 시스템에서 Suite B 호환 TLS 1.2 프로파일을 준수하도록 IBM WebSphere MQ 를 구성하는 방법에 대한 정보를 제공합니다.

19 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.

## *IBM WebSphere MQ* 에서 지원되는 *CipherSpec* 값

기본 CipherSpec 세트는 다음과 같은 값만 허용합니다.

**TLS 1.0**

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

**TLS 1.2**

- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384
- 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

## 더 이상 사용되지 않는 CipherSpec 사용

기본적으로 채널 정의에 더 이상 사용되지 않는 CipherSpec을 지정하도록 허용되지 않습니다. 더 이상 사용되지 않는 CipherSpec을 지정하려고 시도하면 큐 관리자의 오류 로그에서 AMQ9788 메시지를 수신합니다.

`qm.ini` 파일을 편집하여 더 이상 사용되지 않는 CipherSpecs를 다시 사용할 수 있습니다. `qm.ini` 파일의 SSL 스탠자 내에 다음 행을 추가하십시오.

```
SSL:
AllowWeakCipherSpec=Yes
```

또한 환경 변수 *AMQ_SSL_WEAK_CIPHER_ENABLE*을 임의의 값으로 설정하여 런타임 시 더 이상 사용되지 않는 CipherSpecs 중 하나 이상을 다시 사용할 수 있습니다. 이 환경 변수를 사용하면 `qm.ini` 파일에 지정된 값에 관계없이 CipherSpec을 사용할 수 있습니다.

## 채널 인증 레코드

채널 레벨에서 연결 시스템에 허용된 액세스에 대해 보다 정교한 제어를 실행하려면 채널 인증 레코드를 사용할 수 있습니다.

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 WebSphere 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 *4.* 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. 그러나 WebSphere MQ 시스템에 액세스하지 않아야 하는 IP 주소에서 연결 시도가 발생하는 경우가 있을 수 있으며, 방화벽을 업데이트하려면 먼저 임시로 주소를 차단해야 합니다. 이 연결은 WebSphere MQ 채널에서 시도되는 것이 아니라 WebSphere MQ 리스너를 대상으로 잘못 구성한 기타 소켓 애플리케이션에서 시도되는 중일 수 있습니다. 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.

166 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.

167 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.

170 페이지의 『클라이언트 확인 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.

169 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.

170 페이지의 『SSL 식별 이름의 액세스 블로킹』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.

171 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.

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

### 클라이언트가 사용한 사용자 ID를 사용할 사용자 ID에 맵핑

특정 사용자 ID가 WebSphere MQ MQI 클라이언트로부터의 연결에서 사용되는 경우 서로 다른 지정된 MCAUSER를 사용하도록 지정하려면 유형 USERMAP의 채널 인증 레코드를 설정하십시오. 사용자 ID 맵핑에는 와일드카드를 사용하지 않습니다.

예는 168 페이지의 『MCAUSER 사용자 ID에 클라이언트 확인 사용자 ID 맵핑』의 내용을 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.

169 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.

예는 168 페이지의 『MCAUSER 사용자 ID에 리모트 큐 관리자 맵핑』의 내용을 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 채널 인증 레코드가 채널 이름, IP 주소, 큐 관리자 이름 또는 SSL이나 TLS DN과일치하는 경우 가장 구체적인 일치가 사용됩니다. 가장 구체적인 항목으로 간주되는 일치는 다음과 같이 판별됩니다.

- 채널 이름의 경우:
  - 가장 구체적인 일치는 와일드카드가 없는 이름(예: A, B, C)입니다.
  - 가장 일반적인 일치는 모든 채널 이름과 일치하는 하나의 별표(*)입니다.
  - 맨 왼쪽에 별표가 있는 패턴은 맨 왼쪽에 정의된 값이 있는 패턴보다 더 일반적입니다. 따라서 *.B.C는 A.*보다 더 일반적입니다.
  - 두 번째 위치에 별표가 있는 패턴은 두 번째 위치에 정의된 값이 있는 패턴보다 더 일반적이며 각 후속 위치도 마찬가지입니다. 따라서 A.*.C는 A.B.*보다 더 일반적입니다.
  - 두 개 이상의 패턴에서 동일한 위치에 별표가 있는 경우 별표 뒤에 몇 개의 노드가 있는 항목이 더 일반적입니다. 따라서, A. 는 A.*.C보다 일반적입니다.
- IP 주소의 경우:
  - 가장 구체적인 일치는 와일드카드가 없는 이름(예: 192.0.2.6)입니다.
  - 가장 일반적인 일치는 모든 채널 이름과 일치하는 하나의 별표(*)입니다.
  - 맨 왼쪽에 별표가 있는 패턴은 맨 왼쪽에 정의된 값이 있는 패턴보다 더 일반적입니다. 따라서 *.0.2.6은 192.*보다 더 일반적입니다.
  - 두 번째 위치에 별표가 있는 패턴은 두 번째 위치에 정의된 값이 있는 패턴보다 더 일반적이며 각 후속 위치도 마찬가지입니다. 따라서 192.*.2.6은 192.0.*보다 더 일반적입니다.

- 두 개 이상의 패턴에서 동일한 위치에 별표가 있는 경우 별표 뒤에 몇 개의 노드가 있는 항목이 더 일반적입니다. 따라서, 192 .* 는 192.* .2.*보다 일반적입니다.
- 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 문자 별표에 인접한 별표가 없다면 패턴에 별표를 추가할 수 있습니다. 예를 들어, 192.* .2.* 는 유효하지만 192.0.*.* 유효하지 않습니다.
- 결과 주소가 분명하지 않을 수 있으므로 IPv6 주소 패턴에는 이중 콜론 및 후미 별표가 포함될 수 없습니다. 예를 들어, 2001::*는 2001:0000:*, 2001:0000:0000:* 등으로 확장될 수 있습니다.

- 큐 관리자 이름의 경우:
  - 가장 구체적인 일치는 와일드카드가 없는 이름(예: 192.0.2.6)입니다.
  - 가장 일반적인 일치는 모든 채널 이름과 일치하는 하나의 별표(*)입니다.
  - 맨 왼쪽에 별표가 있는 패턴은 맨 왼쪽에 정의된 값이 있는 패턴보다 더 일반적입니다. 따라서 *QUEUEMANAGER는 QUEUEMANAGER*보다 더 일반적입니다.
  - 두 번째 위치에 별표가 있는 패턴은 두 번째 위치에 정의된 값이 있는 패턴보다 더 일반적이며 각 후속 위치도 마찬가지입니다. 따라서 Q*MANAGER는 QUEUE*보다 더 일반적입니다.
  - 두 개 이상의 패턴에서 동일한 위치에 별표가 있는 경우 별표 뒤에 몇 개의 문자가 있는 항목이 더 일반적입니다. 따라서 Q*는 Q*MGR보다 더 일반적입니다.
- SSL이나 TLS 식별 이름(DN)의 경우 하위 문자열의 우선 순위는 다음과 같습니다.

표 *5.* 하위 문자열의 우선 순위

| **순서** | **DN 하위 문자열** | **이름** |
|---|---|---|
| 1 | SERIALNUMBER= | 인증서 일련 번호 |
| 2 | MAIL= | 이메일 주소 |
| 3 | 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 |

| 표 5. 하위 문자열의 우선 순위 (계속) | | |
|---|---|---|
| **순서** | **DN 하위 문자열** | **이름** |
| 15 | UNSTRUCTUREDNAME= | 호스트 이름 |
| 16 | UNSTRUCTUREDADDRESS= | IP 주소 |
| 17 | DNQ= | 식별 이름 규정자 |

따라서 SSL 또는 TSL 인증서가 하위 문자열 O=IBM 및 C=UK를 포함하는 DN과 함께 표시되는 경우 WebSphere MQ는 둘 다 존재할 때 C=UK에 대한 항목 대신, O=IBM에 대한 채널 인증 레코드를 사용합니다.

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 명령 **DISPLAY CHLAUTH** 또는 PCF 명령 **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**

51 페이지의 『원격 메시징의 보안』

이 절에서는 원격 메시징 보안 측면을 다룹니다.

## IBM WebSphere MQ 의 메시지 보안

IBM WebSphere MQ 인프라의 메시지 보안은 별도의 라이센스가 있는 컴포넌트 IBM WebSphere MQ Advanced Message Security에서 제공됩니다.

IBM WebSphere MQ Advanced Message Security(AMS)에서는 IBM WebSphere 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**

246 페이지의 『IBM WebSphere MQ Advanced Message Security』

IBM WebSphere MQ Advanced Message Security(AMS)은 종료 애플리케이션에 영향을 주지 않으면서 IBM WebSphere MQ Advanced Message Security 네트워크를 통한 민감한 데이터에 대한 상위 레벨 보호를 제공하는 별도로 라이센스가 있는 IBM WebSphere MQ Advanced Message Security의 컴포넌트입니다.

# 보안 요구사항 계획

이 주제 모음에서는 IBM WebSphere MQ 환경에서 보안을 계획할 때 고려해야 하는 사항에 대해 설명합니다.

플랫폼 범위에서 다양한 애플리케이션에 대해 IBM WebSphere MQ를 사용할 수 있습니다. 보안 요구사항은 각 애플리케이션별로 다를 수 있습니다. 어떤 경우에는 보안이 중요한 고려사항입니다.

WebSphere MQ는 SSL(Secure Socket Layer) 및 TLS(Transport Layer Security)의 지원을 포함하여 다양한 링크 레벨 보안 서비스를 제공합니다.

WebSphere를 구현할 때는 특정 보안 측면을 고려해야 합니다. UNIX, Linux 및 Windows 시스템에서는 이러한 고려사항을 무시하고 아무 작업도 하지 않으면 WebSphere MQ를 사용할 수 없습니다.

보안 고려사항에 대해 아래에서 설명합니다.

## WebSphere MQ 관리 권한

WebSphere MQ 관리자는 다음을 수행하기 위해 권한이 필요합니다.

- WebSphere MQ를 관리하기 위한 명령 발행
- IBM WebSphere MQ 탐색기 사용

자세한 정보는 다음을 참조하십시오.

- 180 페이지의 『UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 관리 권한』

## WebSphere MQ 오브젝트에 대해 작업하기 위한 권한

애플리케이션은 MQI 호출을 발행하여 다음 WebSphere MQ 오브젝트에 액세스할 수 있습니다.

- 큐 관리자
- 큐
- Processes
- 이름 목록
- 토픽

애플리케이션은 PCF(Programmable Command Format) 명령을 사용하여 이러한 WebSphere MQ 오브젝트에 액세스하고, 채널 및 인증 정보 오브젝트에도 액세스할 수 있습니다. 이러한 오브젝트는 WebSphere MQ에 의해 보호되고 애플리케이션과 연관된 사용자 ID는 이에 액세스할 수 있는 권한이 있어야 합니다.

자세한 정보는 46 페이지의 『IBM WebSphere MQ 를 사용하기 위한 애플리케이션의 권한』의 내용을 참조하십시오.

## 채널 보안

메시지 채널 에이전트(MCA)와 연관된 사용자 ID는 다양한 WebSphere MQ 자원에 액세스하려면 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 63 페이지의 『채널 권한 부여』의 내용을 참조하십시오.

## 추가적인 고려사항

특정 WebSphere MQ 기능 또는 기본 제품 확장을 사용 중인 경우에만 다음 보안 측면을 고려해야 합니다.

- 71 페이지의 『큐 관리자 클러스터의 보안』

- 72 페이지의 『IBM WebSphere MQ 발행/구독에 대한 보안』
- 73 페이지의 『IBM WebSphere MQ Internet Pass-Thru에 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 WebSphere 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. SSL 또는 TLS를 사용하는 IBM WebSphere 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 WebSphere MQ 인증에 사용될 수 있는 디지털 인증서를 판별합니다. 자세한 정보는 다음을 참조하십시오.

- 168 페이지의 『MCAUSER 사용자 ID에 리모트 큐 관리자 맵핑』
- 168 페이지의 『MCAUSER 사용자 ID에 클라이언트 확인 사용자 ID 맵핑』
- 169 페이지의 『MCAUSER 사용자 ID에 SSL 또는 TLS 식별 이름 맵핑』
- 171 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 WebSphere MQ MQI 클라이언트를 표시합니다. 보안은 다음 텍스트에 설명되어 있는 것처럼 네 개의 레벨에서 적용됩니다. MCA는 메시지 채널 에이전트입니다.

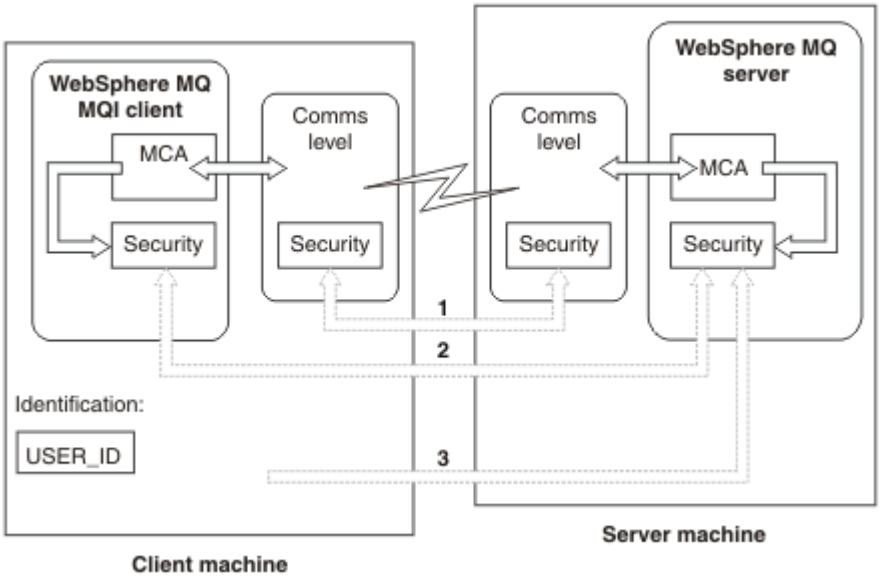



그림 7. 클라이언트/서버 연결에서의 보안

1. 통신 레벨

   화살표 1을 보십시오. 통신 레벨에서 보안을 구현하려면 SSL 또는 TLS를 사용한다. 추가 정보는 14 페이지의 『암호화 보안 프로토콜: SSL 및 TLS』의 내용을 참조하십시오.

2. 채널 인증 레코드

   화살표 2 및 3을 참조하십시오. 인증은 보안 레벨에서 IP 주소 또는 SSL/TLS 식별 이름을 사용하여 제어할 수 있다. 사용자 ID는 차단될 수도 있으며 확인된 사용자 ID는 올바른 사용자 ID에 맵핑될 수 있습니다. 전체 설명은 37 페이지의 『채널 인증 레코드』에서 제공됩니다.

3. 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.

4. 채널 보안 엑시트에 전달된 ID

   화살표 3을 참조하십시오. 클라이언트 대 서버 통신에서 채널 보안 엑시트는 쌍으로 작동할 필요가 없습니다. IBM WebSphere 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.

   Windows 클라이언트는 또한 식별을 지원하기 위해 추가 정보를 전송합니다.

   - 서버로 전달되는 사용자 ID는 클라이언트에 현재 로그온된 사용자 ID입니다.
   - 현재 로그온된 사용자의 보안 ID입니다.

   HP Integrity NonStop Server의 IBM WebSphere MQ 클라이언트에 대한 식별을 지원하기 위해 클라이언트는 클라이언트 애플리케이션이 실행 중인 OSS Safe가드 별명을 전달합니다. 이 ID의 양식은 일반적으로 <PRIMARYGROUP>.<ALIAS>입니다. 필요한 경우에는 채널 인증 레코드 또는 보안 엑시트를 사용하여 이 사용자 ID를 큐 관리자에 있는 대체 사용자 ID에 맵핑할 수 있습니다. 메시지 엑시트에 대한 자세한 정보는 136 페이지의 『메시지 엑시트에서 ID 맵핑』의 내용을 참조하십시오. 채널 인증 레코드 정의에 대한 자세한 정보는 168 페이지의 『MCAUSER 사용자 ID에 클라이언트 확인 사용자 ID 맵핑』의 내용을 참조하십시오.

   사용자 ID 및 보안 ID(사용 가능한 경우)의 값은 서버 보안 엑시트가 IBM WebSphere MQ MQI 클라이언트의 ID를 설정하는 데 사용될 수 있습니다.

### 사용자 *ID*

IBM WebSphere MQ MQI 클라이언트가 에 고 IBM WebSphere MQ 서버도 있고 클라이언트 사용자 ID가 정의된 도메인에 액세스할 수 있는 경우, IBM WebSphere MQ 는 최대 20자의 사용자 ID를 지원합니다. UNIX and Linux 플랫폼 및 구성에서는 최대 12자가 지원됩니다.

A WebSphere MQ for 윈도우 server does not support the connection of a 윈도우 client if the client is running under a user ID that contains the @ character, for example, abc@d. 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.

UNKNOWN은 예약된 사용자 ID이며 NOBODY 사용자 ID 또한 WebSphere MQ에 대해 특별한 의미가 있습니다. 운영 체제에서 UNKNOWN 또는 NOBODY로 사용자 ID를 작성하면 예기치 못한 결과를 초래합니다.

사용자 ID가 인증에 사용됨에도 불구하고 그룹이 권한 부여에 사용됩니다. 단, Windows는 예외입니다.

그룹에 주의하지 않고 서비스 계정을 작성하고 모든 사용자 ID를 다르게 권한 부여하면 모든 사용자가 다른 모든 사용자의 정보에 액세스할 수 있습니다.

## 권한 부여 계획

관리 권한이 있는 사용자를 계획하고 애플리케이션의 사용자에게 권한을 부여하는 방법을 계획하여 IBM WebSphere MQ MQI 클라이언트에서의 연결을 포함하여 IBM WebSphere MQ 오브젝트를 적절하게 사용하십시오.

IBM WebSphere MQ를 사용하려면 개인 또는 애플리케이션에 액세스가 부여되어야 합니다. 필요한 액세스는 수행하는 역할 및 수행해야 하는 태스크에 따라 다릅니다. IBM WebSphere MQ에서의 권한 부여는 다음과 같은 두 개의 기본 범주로 구분됩니다.

- 관리 조작을 수행하기 위한 권한 부여
- 애플리케이션이 IBM WebSphere MQ를 사용하는 권한

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 WebSphere MQ 관리 권한

IBM WebSphere MQ 관리자는 다양한 기능 수행 권한이 필요합니다. 이 권한은 다른 플랫폼에서 다양한 방식으로 확보합니다.

IBM WebSphere MQ 관리자는 다음 권한이 필요합니다.

- IBM WebSphere MQ 를 관리하기 위한 명령 실행
- 사용IBM WebSphere MQ Explorer

자세한 정보는 운영 체제에 해당하는 주제를 참조하십시오.

### 시스템 *IBM WebSphere MQ* 를 관리하는 권한

IBM WebSphere MQ 관리자는 mqm 그룹의 구성원입니다. 이 그룹은 모든 IBM WebSphere MQ 자원에 대한 액세스를 가지며 IBM WebSphere MQ 제어 명령을 발행할 수 있습니다. 관리자는 다른 사용자에게 특정 권한을 부여할 수 있습니다.

시스템 IBM WebSphere MQ 관리자가 되려면 사용자가 *mqm* 그룹의 구성원이어야 합니다. 이 그룹은 WebSphere MQ를 설치할 때 자동으로 작성됩니다. 사용자가 제어 명령을 발행하도록 허용하려면 사용자를 mqm 그룹에 추가해야 합니다. 여기에는 UNIX 시스템의 루트 사용자도 포함됩니다.

mqm 그룹의 구성원이 아닌 사용자에게 관리 권한을 부여할 수 있지만 IBM WebSphere MQ 제어 명령을 발행할 수 없고 액세스가 부여된 명령만 실행할 권리가 있습니다.

또는 Windows 시스템에서 SYSTEM 및 Administrator 계정은 IBM WebSphere MQ 자원에 대해 전체 액세스 권한을 가집니다.

mqm 그룹의 모든 구성원들은 시스템에서 실행 중인 모든 큐 관리자에 액세스할 수 있는 것을 포함하여 시스템에 있는 모든 WebSphere MQ 자원에 액세스할 수 있습니다. 이 액세스는 사용자를 mqm 그룹에서 제거해야만 취소할 수 있습니다. Windows 시스템에서는 Administrators 그룹의 구성원들도 모든 WebSphere MQ 자원에 액세스할 수 있습니다.

관리자는 제어 명령 **runmqsc**를 사용하여 WebSphere MQ 스크립트(MQSC) 명령을 실행할 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.

WebSphere MQ Explorer는 PCF 명령을 사용하여 관리 태스크를 수행합니다. 관리자는 로컬 시스템에서 큐 관리자를 관리하기 위해 WebSphere MQ Explorer를 사용할 수 있는 추가 권한이 필요하지 않습니다. WebSphere MQ 탐색기가 다른 시스템에 있는 큐 관리자를 관리하는 데 사용되면, 관리자가 리모트 큐 관리자에 의해 처리되는 PCF 명령에 대한 필수 권한을 가져야 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 46 페이지의 『IBM WebSphere MQ 를 사용하기 위한 애플리케이션의 권한』의 내용을 참조하십시오.
- 채널, 채널 이니시에이터, 리스너 및 클러스터에 대해 실행되는 명령에 대해서는 채널 보안을 참조하십시오.

UNIX 및 Windows 시스템에서 WebSphere MQ를 관리하기 위해 필요한 권한에 대한 자세한 정보는 관련 정보를 참조하십시오.

## IBM WebSphere MQ 를 사용하기 위한 애플리케이션의 권한

애플리케이션이 오브젝트에 액세스하는 경우 애플리케이션에 연관되는 사용자 ID에는 적절한 권한이 있어야 합니다.

애플리케이션은 MQI 호출을 발행하여 다음 IBM WebSphere MQ 오브젝트에 액세스할 수 있습니다.

- 큐 관리자
- 큐
- Processes
- 이름 목록
- 토픽

애플리케이션은 PCF 명령을 사용하여 IBM WebSphere MQ 오브젝트를 관리할 수도 있습니다. PCF 명령이 처리되면 PCF 메시지를 넣는 사용자 ID의 권한 컨텍스트를 사용합니다.

이 컨텍스트에서 애플리케이션은 사용자 및 벤더가 작성한 애플리케이션을 포함합니다.

Java의 IBM WebSphere MQ 클래스, JMS의 IBM WebSphere MQ 클래스, .NET의 IBM WebSphere MQ 클래스 또는 Message Service Clients for C/C++ and .NET 를 사용하는 애플리케이션은 MQI를 간접적으로 사용합니다.

MCA도 MQI 호출을 발행하며 MCA에 연관된 사용자 ID는 이러한 WebSphere MQ 오브젝트에 액세스할 수 있는 권한이 있어야 합니다. 이 사용자 ID와 그들이 필요로 하는 권한에 대한 자세한 정보는 63 페이지의 『채널 권한 부여』 을 참조하십시오.

## 권한 검사가 수행되는 시기

권한 검사는 애플리케이션이 큐 관리자, 큐, 프로세스 또는 이름 목록에 액세스를 시도할 때 수행됩니다.

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 WebSphere MQ에 사인온할 필요는 없습니다. IBM WebSphere MQ에서는 사용자가 근본적인 운영 체제에 사인온하고 이미 인증되었다고 간주합니다.

**애플리케이션이 MQOPEN 또는 MQPUT1 호출을 사용하여 IBM WebSphere MQ 오브젝트를 열 때**

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. 권한 검사는 리모트 큐 관리자와 동일한 이름의 전송 큐에 대해 수행됩니다. UNIX, Linux, and Windows에서, 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.QUEUE에 대해 수행됩니다.

동적 큐에 대한 권한은 이 동적 큐가 도출된 모델 큐의 권한을 기본으로 하지만 반드시 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 두 개 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.

**WebSphere MQ 오브젝트에서 작용하는 PCF 명령을 명령 서버에서 처리할 때**

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 필드 및 필드 설정 방법에 관한 자세한 정보는 48 페이지의 『메시지 컨텍스트』의 내용을 참조하십시오.

## 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. 이것이 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 넣은 애플리케이션의 유형. CICS® 트랜잭션이 그 예입니다.

**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.

## *UNIX, Linux* 및 *Windows* 시스템에서 *IBM WebSphere MQ* 오브젝트에 대한 작업 권한

IBM WebSphere MQ에서 제공되는 권한 서비스 컴포넌트는 오브젝트 권한 관리자*(OAM)*라고 합니다. 이는 인증 및 권한 검사를 통해 액세스 제어를 제공합니다.

1. 인증.

   IBM WebSphere MQ에서 제공되는 OAM이 수행하는 인증 검사는 기본이며 특정 상황에서만 수행됩니다. 매우 안전한 환경에서 예상되는 엄격한 요구사항을 충족시키기 위한 것은 아닙니다.

   OAM은 애플리케이션이 큐 관리자에 연결할 때 해당 인증 검사를 수행하고 다음 조건이 적용됩니다.

   MQCSP 구조가 애플리케이션 연결로 제공되고 MQCSP 구조의 *AuthenticationType* 속성이 MQCSP_AUTH_USER_ID_AND_PWD 값을 제공하는 경우 검사는 해당 MQZID_AUTHENTICATE_USER 함

수에서 OAM으로 수행됩니다. 이는 검사입니다. MQCSP 구조의 사용자 ID는 *IdentityContext*(MQZIC)의 사용자 ID와 비교되어 일치 여부가 판별됩니다. 일치하지 않으면 검사가 실패합니다.

이 기본 검사는 사용자의 전체 인증을 위한 것이 아닙니다. 예를 들어, MQCSP 구조에서 제공되는 비밀번호를 검사하여 사용자의 권한을 검사하지 않습니다. 또한, 애플리케이션이 MQCSP 구조를 생략하면 검사가 수행되지 않습니다.

전체 인증 서비스가 권한 서비스 컴포넌트를 통해 큐 관리자에서 필요한 경우 IBM WebSphere MQ에서 제공하는 OAM은 이를 제공하지 않습니다. 새 권한 서비스 컴포넌트를 작성하거나 벤더에서 이를 확보해야 합니다.

2. 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.

UNIX, Linux 및 Windows 시스템에서, 권한 서비스는 애플리케이션이 MQI 호출을 실행하여 큐 관리자, 큐, 프로세스, 토픽 또는 이름 목록인 IBM WebSphere MQ 오브젝트에 액세스할 때의 액세스 제어를 제공합니다. 여기에는 대체 사용자 권한 및 컨텍스트 정보 설정 또는 전달 권한에 대한 검사가 포함됩니다.

Windows에서 OAM은 관리 그룹 구성원에서 모든 IBM WebSphere MQ 오브젝트 액세스 권한을 부여하며 UAC가 사용되는 경우도 포함됩니다.

또한 Windows 시스템에서 SYSTEM 계정에는 IBM WebSphere MQ 자원에 대한 전체 액세스 권한이 있습니다.

권한 서비스는 PCF 명령이 다음 IBM WebSphere 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.

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 WebSphere MQ 제품에서 제공되는 컴포넌트를 기능 보강하거나 대체하기 위해 컴포넌트를 제공할 수 있습니다.

IBM WebSphere 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 WebSphere MQ 오브젝트에 대해 액세스 제어 목록(ACL)을 유지보수합니다. UNIX and Linux 시스템에서 그룹 ID만 ACL에 표시 가능합니다. 이는 그룹의 모든 구성원에 같은 권한이 있다는 것을 의미합니다. Windows 시스템에서 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 Linux 가 이 한계를 설정했지만 IBM WebSphere MQ 가 모든 UNIX 플랫폼에서 12개의 문자 제한을 계속 관찰합니다. 12자 이상의 사용자 ID를 사용하면 IBM WebSphere MQ가 이를 값 "UNKNOWN"으로 대체합니다. 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 WebSphere 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.

UNIX, Linux 및 Windows 시스템에서 제어 명령 **setmqaut**는 권한을 부여하고 취소하며, ACL을 유지보수하는 데 사용됩니다. 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
```

그룹 VOYAGER의 구성원이 MOON. 문자로 시작하는 이름의 큐에 메시지를 넣을 수 있도록 합니다. MOON.* 은 일반 프로파일의 이름이다.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.

권한 검사를 하지 않는 경우(예를 들어, 테스트 환경), OAM을 사용하지 않을 수 있습니다.

*PCF*를 사용하여 *OAM* 명령에 액세스
UNIX, Linux 및 Windows 시스템에서는 PCF 명령을 사용하여 OAM 관리 명령에 액세스할 수 있습니다.

PCF 명령과 대응되는 OAM 명령은 다음과 같습니다.

표 *6. 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.

이러한 명령 사용에 대한 자세한 정보는 프로그램 가능 명령 형식 소개 를 참조하십시오.

## 원격 메시징의 보안

이 절에서는 원격 메시징 보안 측면을 다룹니다.

IBM WebSphere 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 리모트 IP 주소, 리모트 사용자 ID, 제공된 SSL 또는 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.

### *UNIX and Linux* 시스템에서 오브젝트의 보안

이 ID가 IBM WebSphere MQ 관리 명령을 사용하는 경우, 관리 사용자는 시스템에서 mqm 그룹의 파트여야 합니다(루트 포함).

항상 "mqm" 사용자 ID로 amqcrsta를 실행해야 합니다.

## UNIX and Linux 시스템의 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* 시스템에서 오브젝트의 보안

이 ID가 IBM WebSphere MQ 관리 명령을 사용하는 경우, 관리 사용자는 시스템에서 mqm 그룹 및 관리자 그룹의 일부여야 합니다

## Windows 시스템의 사용자 ID

Windows 시스템에서, 설치된 메시지 엑시트가 없으면 큐 관리자는 모두 대문자로 되었거나 대소문자가 섞인 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*

Windows, UNIX and Linux 시스템 이외의 플랫폼은 메시지에서 사용자 ID에 대문자를 사용합니다.

Windows, UNIX and Linux 시스템에서 메시지에 소문자 사용자 ID를 허용하기 위해 이러한 플랫폼에서 메시지 채널 에이전트(MCA)가 다음 변환을 수행합니다.

**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 이유로 UNIX, Linux 및 Windows 시스템에서 메시지 엑시트를 제공하면 자동 변환이 수행되지 않습니다.

## 사용자 정의 권한 서비스 사용

IBM WebSphere MQ에서는 설치 가능한 권한 서비스를 제공합니다. 대체 서비스 설치를 선택할 수 있습니다.

IBM WebSphere 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 WebSphere 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 7. 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 53 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 도구에 의해 설정된 사용 ID가 없으면 사용됩니다. |
| 서버 연결 채널을 시작한 사용자 ID | 다른 방법으로 설정된 사용자 ID가 없고 클라이언트 사용자 ID가 플로우되지 않는 경우 사용됨. 자세한 정보는 54 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 WebSphere MQ MQI 클라이언트의 경우, MQI 호출을 실행하는 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.

Windows 플랫폼에서 MCAUSER 필드는 12자로 제한되며 권한 부여 실패를 일으킬 수 있는 추가 문자는 잘려집니다.

클라이언트측 보안 엑시트가 사용 중일 때 IBM WebSphere 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/OS의 경우 z/OS 시작 프로시저 테이블에서 채널 시작기 시작 태스크에 지정된 사용자 ID.
- TCP/IP(비z/OS)의 경우, `inetd.conf` 항목의 사용자 ID 또는 리스너를 시작한 사용자 ID입니다.
- SNA(비z/OS)의 경우, SNA 서버 항목의 사용자 ID 또는 (없는 경우) 수신되는 첨부 요청 또는 리스너를 시작한 사용자 ID입니다.
- 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 WebSphere MQ 기본 서버 연결 채널(SYSTEM.DEF.SVRCONN)에는 MCAUSER 속성이 공백으로 설정되어 있습니다. 무단 액세스를 방지하려면 기본 정의의 MCAUSER 속성을 IBM WebSphere 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.

UNIX, Linux, Windows 시스템의 서버의 경우, 클라이언트에 수신한 `MCAUserIdentifier` 필드 컨텐츠는 소문자로 변경됩니다.

IBM i의 서버의 경우, 클라이언트에서 수신한 `LongMCAUserIdentifier` 필드 컨텐츠는 대문자로 변경됩니다.

UNIX and Linux 시스템의 서버의 경우, 클라이언트에서 수신한 `LongMCAUserIdentifier` 필드의 컨텐츠는 소문자로 변경됩니다.

기본적으로 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. SSL 또는 TLS를 사용하도록 선택할 수 있으며 어떤 경우에도 디지털 인증서 사용을 계획해야 합니다. 또한, 표준 기능이 요구사항에 맞지 않는 경우에는 채널 엑시트 프로그램을 사용할 수도 있습니다.

**관련 개념**

55 페이지의 『링크 레벨의 보안과 애플리케이션 레벨의 보안 비교』
이 토픽에는 다양한 측면의 링크 레벨 보안 및 애플리케이션 레벨 보안에 대한 정보가 들어 있고 두 보안 레벨을 비교합니다.

59 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.

65 페이지의 『SSL을 사용하는 채널 보호』
IBM WebSphere MQ의 SSL 지원은 큐 관리자 인증 정보 오브젝트 및 다양한 MQSC 명령어를 사용합니다. 또한, 디지털 인증서 사용도 고려해야 합니다.

## 링크 레벨의 보안과 애플리케이션 레벨의 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 55 페이지의 그림 8에서 설명됩니다.

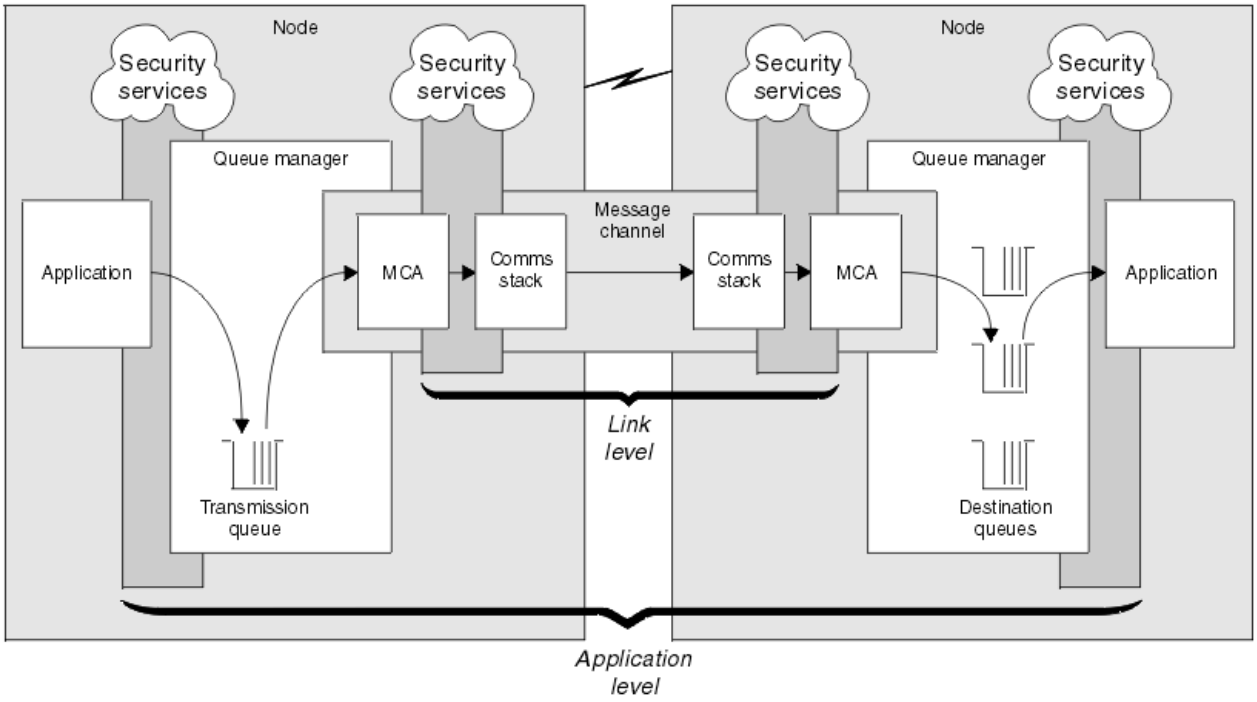



그림 *8.* 링크 레벨 보안 및 애플리케이션 레벨 보안

## 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.

애플리케이션 레벨 보안은 메시지가 큐에 저장되어 있는 동안 이를 보호할 수 있으며 분산 큐잉이 사용되지 않는 경우에도 적용됩니다. 이것이 링크 레벨 보안과 응용프로그램 레벨 보안 사이의 주요한 차이점이고 55 페이지의 그림 8에 나타나 있습니다.

## 큐 관리자가 제어되고 신뢰되는 환경에서 실행 중이 아님

큐 관리자가 제어되고 신뢰되는 환경에서 실행 중인 경우 WebSphere MQ가 제공하는 액세스 제어 메커니즘은 해당 큐에 저장된 메시지를 보호하기에 충분한 것으로 간주될 수도 있습니다. 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. 또한 메시지 데이터 이외의 WebSphere 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 55 페이지의 그림 8에서 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 WebSphere MQ에서 제공되는 링크 레벨 보안

IBM WebSphere MQ에서의 기밀성 및 데이터 무결성의 프로비저닝을 위한 기본 방법은 SSL 또는 TLS 사용입니다. IBM WebSphere MQ에서 SSL 및 TLS을 사용하는 것에 관한 자세한 정보는 22 페이지의 『SSL 및 TLS용 IBM WebSphere MQ 지원』 의 내용을 참조하십시오. 인증을 위해 IBM WebSphere MQ는 채널 인증 레코드 사용 기능을 제공합니다. 채널 인증 레코드는 개별 채널이나 채널 그룹의 레벨에서 연결 시스템에 부여되는 액세스에 대해 세밀한 제어를 제공합니다. 추가 정보는 37 페이지의 『채널 인증 레코드』 의 내용을 참조하십시오.

### 사용자 고유의 링크 레벨 보안 제공

이 토픽 콜렉션은 사용자 고유의 링크 레벨 보안 서비스를 제공하는 방법에 대해 설명합니다. 사용자 고유의 채널 엑시트 프로그램을 작성하는 것이 사용자 고유의 링크 레벨 보안 서비스를 제공하는 기본적인 방법입니다.

채널 엑시트 프로그램은 59 페이지의 『채널 엑시트 프로그램』 에 소개되어 있습니다. 동일한 주제에서 Windows (SSPI 채널 엑시트 프로그램) 에 대해 IBM WebSphere MQ 와 함께 제공되는 채널 엑시트 프로그램에 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. 애플리케이션, 큐 관리자, WebSphere MQ를 지원하는 다른 제품 또는 함께 작동하는 이들 중의 어떠한 결합에 의해 서비스가 직접 또는 간접적으로 호출될 수 있습니다. 애플리케이션 레벨 보안은 55 페이지의 그림 8에서 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.

계획의 대상 *Advanced Message Security*
IBM WebSphere MQ Advanced Message Security(AMS)은 종료 애플리케이션에 영향을 주지 않으면서 IBM WebSphere MQ 네트워크를 통한 민감한 데이터에 대한 상위 레벨 보호를 제공하는 별도로 라이센스가 있는 IBM WebSphere 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 WebSphere MQ에 대해 자체적으로 보안 확장을 개발할 수도 있습니다. 그렇지만 이런 솔루션의 경우 전문가의 기술이 필요하며 유지보수가 복잡하고 비용이 많이 들 수 있습니다. IBM WebSphere MQ Advanced Message Security는 기업 내에서 정보를 이동할 때 거의 모든 상용 IT 시스템 유형 사이에서 발생하는 이런 문제 해결에 사용할 수 있습니다.

IBM WebSphere MQ Advanced Message Security는 다음과 같은 방식으로 IBM WebSphere 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 WebSphere MQ 서버 및 클라이언트 모두를 지원합니다.
- 엔드-투-엔드 안전 메시징 솔루션을 제공하기 위해 IBM WebSphere MQ Managed File Transfer와 통합됩니다.

추가 정보는 246 페이지의 『IBM WebSphere MQ Advanced Message Security』의 내용을 참조하십시오.

사용자 고유의 애플리케이션 레벨 보안 제공
이 토픽 콜렉션은 자체적으로 애플리케이션 레벨 보안 서비스를 제공하는 방법에 대해 설명합니다.

애플리케이션 레벨 보안을 구현할 수 있도록 IBM WebSphere MQ는 두 개의 엑시트인 API 엑시트와 API 교차 엑시트를 제공합니다.

이 엑시트는 식별 및 인증, 액세스 제어, 기밀성, 데이터 무결성, 부인 방지 서비스 및 보안에 연관되지 않은 다른 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 WebSphere 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 WebSphere 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 60 페이지의 그림 9에 설명되어 있으며 다음 주제에서도 설명됩니다.

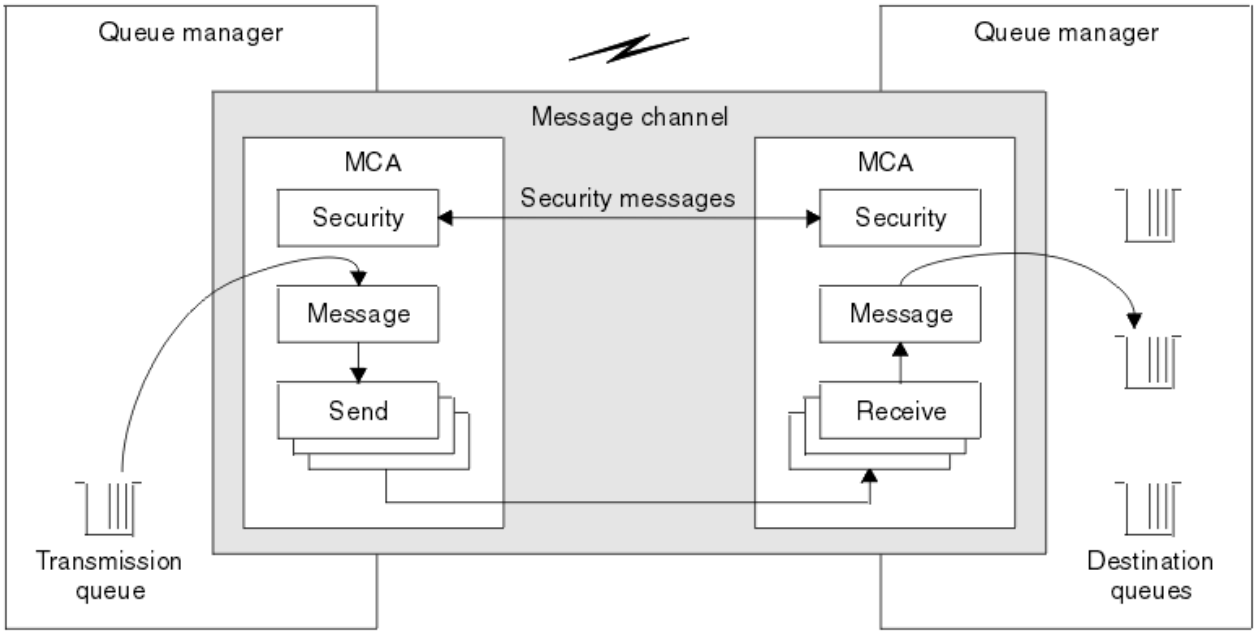



그림 9. 메시지 채널의 보안, 메시지, 송신, 수신

**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 57 페이지의 『보안 엑시트를 사용하는 링크 레벨 보안』를 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. MQI 채널의 목적이 MQI 호출의 입출력 매개변수가 IBM WebSphere MQ MQI 클라이언트 애플리케이션과 큐 관리자 사이에서 플로우하도록 하는 것이기 때문입니다.

메시지 엑시트의 이름은 양측 채널에 있는 채널 정의에서 매개변수로 지정됩니다. 연속적으로 실행되도록 메시지 엑시트 목록을 지정할 수도 있습니다.

메시지 엑시트에 대한 자세한 정보는 57 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 WebSphere 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 WebSphere 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 57 페이지의 『송신 및 수신 엑시트를 사용하는 링크 레벨 보안』를 참조하십시오.

## 데이터 무결성 계획

데이터 무결성 보존 방법을 계획합니다.

애플리케이션 레벨 또는 링크 레벨에서 데이터 무결성을 구현할 수 있습니다.

애플리케이션 레벨에서 인증되지 않은 수정이 이루어지지 않도록 보호하기 위해 IBM WebSphere MQ Advanced Message Security를 사용하여 메시지를 디지털로 서명하도록 선택할 수 있습니다. 표준 기능이 요구사항을 만족하지 않는 경우 API 엑시트 프로그램도 사용할 수 있습니다.

링크 레벨에서 SSL 또는 TLS를 사용하도록 선택할 수 있으며 어떤 경우에도 디지털 인증서 사용을 계획해야 합니다. 또한, 표준 기능이 요구사항에 맞지 않는 경우에는 채널 엑시트 프로그램을 사용할 수도 있습니다.

**관련 개념**

65 페이지의 『SSL을 사용하는 채널 보호』
IBM WebSphere MQ의 SSL 지원은 큐 관리자 인증 정보 오브젝트 및 다양한 MQSC 명령어를 사용합니다. 또한, 디지털 인증서 사용도 고려해야 합니다.

21 페이지의 『IBM WebSphere MQ 의 데이터 무결성』
메시지 수정 여부를 감지하기 위해 데이터 무결성 서비스를 사용할 수 있습니다.

58 페이지의 『계획의 대상 Advanced Message Security』
IBM WebSphere MQ Advanced Message Security(AMS)은 종료 애플리케이션에 영향을 주지 않으면서 IBM WebSphere MQ 네트워크를 통한 민감한 데이터에 대한 상위 레벨 보호를 제공하는 별도로 라이센스가 있는 IBM WebSphere 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 WebSphere 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 다룹니다.

## 감사하려는 데이터

다음 절에서 설명하는 것처럼 감사해야 하는 데이터 또는 활동 유형을 고려하십시오.

**IBM WebSphere MQ를 사용하여 작성한 IBM WebSphere MQ의 변경사항**
도구 이벤트, 특히 명령 이벤트 및 구성 이벤트를 발행하도록 IBM WebSphere MQ를 구성하십시오.

**해당 제어를 벗어난 IBM WebSphere MQ에 대한 변경**
일부 변경은 IBM WebSphere MQ 작동 방식에 영향을 줄 수 있지만 IBM WebSphere 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 WebSphere MQ 의 운영 제어**
큐 관리자 시작 및 중지와 같은 활동을 감사하기 위해 운영 체제를 사용해야 할 수도 있습니다. IBM WebSphere MQ가 도구 이벤트를 발행하도록 구성 가능한 경우도 있습니다.

**IBM WebSphere MQ에서의 애플리케이션 활동**
큐 열기 및 메시지 넣기 및 가져오기와 같은 애플리케이션 조치를 감사하려면 적절한 이벤트를 발행하도록 IBM WebSphere 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 WebSphere 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

IBM WebSphere MQ Explorer에서는 테스트 세트가 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 보내거나 받을 때 다양한 IBM WebSphere MQ 자원에 대한 액세스 권한이 있는 사용자 ID가 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.

WebSphere MQ에서는 SSL 또는 TLS 지원을 통해 채널을 보호할 수 있습니다.

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 메시지를 큐에 넣으려는 대상 큐를 열 수 있습니다.

  MQI(Message queuing Interface)를 포함하므로, WebSphere MQ OAM(Object Authority Manager)을 사용하는 경우 추가 액세스 제어 검사를 수행해야 할 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 기본값으로 큐 관리자 서비스 계정을 사용하며 이는 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.
- WebSphere 관리 권한이 없는 사용자 ID를 사용하는 경우, 채널에 대한 dsp 및 ctrlx 권한을 채널에 대한 사용자 ID에 부여하여 작동하도록 허용해야 합니다. 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. 예를 들어, 다음 명령을 실행하십시오.

```
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**
37 페이지의 『채널 인증 레코드』
채널 레벨에서 연결 시스템에 허용된 액세스에 대해 보다 정교한 제어를 실행하려면 채널 인증 레코드를 사용할 수 있습니다.

채널 인증 레코드 특성

## 채널 시작기 정의 보호
mqm 그룹의 구성원만 채널 시작기를 조작할 수 있습니다.

IBM WebSphere MQ 채널 시작기는 IBM WebSphere MQ 오브젝트가 아닙니다. 이에 대한 액세스는 OAM으로 제어되지 않습니다. IBM WebSphere MQ는 해당 사용자 ID가 mqm 그룹의 구성원인 경우를 제외하고는 사용자나 애플리케이션이 해당 오브젝트를 조작하는 것을 허용하지 않습니다. PCF 명령 `StartChannelInitiator`를 실행하는 애플리케이션을 사용하는 경우, PCF 메시지의 메시지 디스크립터에 지정된 사용자 ID는 대상 큐 관리자의 mqm 그룹 멤버여야 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. 82 페이지의 표 10의 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 59 페이지의 『채널 엑시트 프로그램』의 내용을 참조하십시오.

### *SSL*을 사용하는 채널 보호

IBM WebSphere MQ의 SSL 지원은 큐 관리자 인증 정보 오브젝트 및 다양한 MQSC 명령어를 사용합니다. 또한, 디지털 인증서 사용도 고려해야 합니다.

### SSL을 지원하는 명령 및 속성

SSL(Secure Sockets Layer) 프로토콜은 도청, 도용 및 위장에 대한 보호와 함께 채널 보안을 제공합니다. SSL에 대한 IBM WebSphere MQ 지원을 사용하면 채널 정의에서 특정 채널이 SSL 보안을 사용하도록 지정할 수 있습니다. 또한, 사용하려는 암호화 알고리즘과 같은 필요한 보안 유형의 세부사항을 지정할 수도 있습니다.

다음 MQSC 명령이 SSL을 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 SSL을 지원합니다.

**SSLCRLNL**
SSLCRLNL 속성은 향상된 TLS/SSL 인증서 검사를 허용하기 위해 인증서 폐기 위치를 제공하는 데 사용되는 인증 정보 오브젝트 이름 목록을 지정합니다.

**SSLCRYP**
윈도우시스템에서 UNIX and Linux 시스템은 SSLCryptoHardware 큐 관리자 속성을 설정합니다. 이 속성은 시스템에 있는 암호화 하드웨어를 구성하는 데 사용할 수 있는 매개변수 문자열의 이름입니다.

**SSLEV**
SSL을 사용하는 채널이 SSL 연결 설정에 실패할 경우 SSL 이벤트 메시지를 보고할 것인지를 판별합니다.

**SSLFIPS**
암호화가 암호화 하드웨어가 아니라 IBM WebSphere 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**
의,UNIX and Linux 시스템은 키 저장소를 큐 관리자와 연관시킵니다. 키 데이터베이스는 *GSKit* 키 데이터베이스에 보유됩니다. (IBM GSKit(Global Security Kit)은 Windows,UNIX and Linux 시스템에서 SSL 보안을 사용할 수 있도록 합니다.)

**SSLRKEYC**
비밀 키가 재결정되기 전에 SSL 대화 안에서 송신 및 수신된 바이트 수. 바이트 수에는 MCA에서 전송된 제어 정보가 포함됩니다.

다음 채널 매개변수가 SSL을 지원합니다.

**SSLCAUTH**
IBM WebSphere MQ가 SSL 클라이언트의 인증서를 요구하며 유효성 검증하는지 여부를 정의합니다.

**SSLCIPH**
암호화 강도 및 기능(CipherSpec)(예: NULL_MD5 또는 RC4_MD5_US)을 지정합니다. CipherSpec은 채널의 양 측과 일치해야 합니다.

**SSLPEER**
허용된 파트너의 식별 이름(고유 ID)을 지정합니다.

이 절에서는 인증 정보 오브젝트를 지원하기 위한 `setmqaut`, `dspmqaut`, `dmpmqaut`, `rcrmqobj`, `rcdmqimg` 및 `dspmqfls` 명령에 대해 설명합니다. 또한 UNIX and Linux 시스템에서 인증서를 관리하기 위한 iKeycmd 명령과 UNIX, Linux 및 시스템 인증서를 관리하기 위한 runmqakm 도구에 대해서도 설명합니다. 다음 절을 참조하십시오.

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- 키 및 인증서 관리

SSL을 사용한 채널 보안 개요는 다음을 참조하십시오.

- 22 페이지의 『SSL 및 TLS용 IBM WebSphere MQ 지원』

SSL과 연관된 MQSC 명령에 대한 세부사항은 다음을 참조하십시오.

- ALTER AUTHINFO
- DEFINE AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

SSL에 연관된 PCF 명령에 대한 자세한 내용은 다음을 참조하십시오.

- 인증 정보 오브젝트 변경, 복사 및 작성
- 인증 정보 오브젝트 삭제
- 인증 정보 오브젝트 조회

## 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.

Windows, UNIX및 Linux 시스템에서 자체 서명 인증서를 사용할 수 있습니다. 지시사항은 111 페이지의 『UNIX, Linux, and Windows 시스템에서 자체 서명한 개인 인증서 작성』을 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 CipherSpec를 선택해야 합니다. 이와 비슷하게 조직의 보안 정책에서 특정 CipherSpec을 사용할 것을 요구하는 경우 적당한 디지털 인증서를 가지고 있어야 합니다.

CipherSpec과 디지털 인증서 사이의 관계에 대한 자세한 정보는 32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 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. WebSphere MQ의 인증서 유효성 검증 정책에 대한 자세한 정보는 32 페이지의 『IBM WebSphere MQ 의 인증서 유효성 검증 정책』을 참조하십시오.

## *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 네트워크에서는 WebSphere MQ MCA가 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/OS용 Triple DES 24바이트 암호화와 같이 사용자 플랫폼에서 사용 가능한 더 강력한 형식의 암호화에 대한 정보도 동일한 문서를 참조하십시오.

세션 레벨 암호화에 대한 일반적인 정보는 *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808을 참조하십시오.

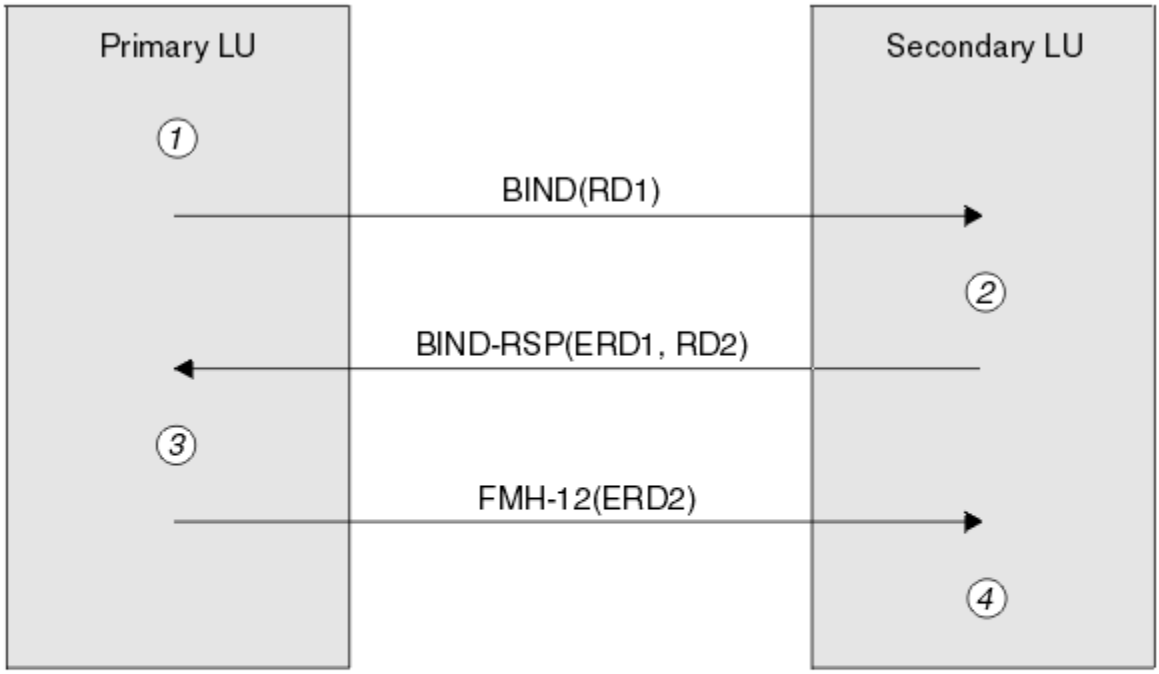
### 세션 레벨 인증

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.

69 페이지의 그림 10는 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 *10.* 세션 레벨 인증 플로우

세션 레벨 인증의 프로토콜은 다음과 같습니다. 프로시저에 있는 번호는 69 페이지의 그림 10의 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 WebSphere 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 특정 정보는 *z/OS MVS™ Planning: APPC/MVS Management*, SA22-7599를 참조하십시오.

CPI-C에 대한 자세한 정보는 *Common Programming Interface Communications CPI-C Specification*, SC31-6180을 참조하십시오. APPC/MVS TP Conversation Callable Services에 대한 자세한 정보는 *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*, SA22-7621을 참조하십시오.

시스템 시스템 *IBM WebSphere MQ* 의 대화 레벨 인증 지원
이 토픽을 사용하여 UNIX, Linux, and Windows에서 대화 레벨 인증이 작동하는 방법에 대한 개요를 확인하십시오.

The support for conversation level authentication in IBM WebSphere MQ for WebSphere MQ on 유닉스 systems, and WebSphere MQ for 윈도우 is illustrated in 70 페이지의 그림 11. 다이어그램에 있는 번호는 그 뒤에 오는 설명에 있는 번호에 해당합니다.

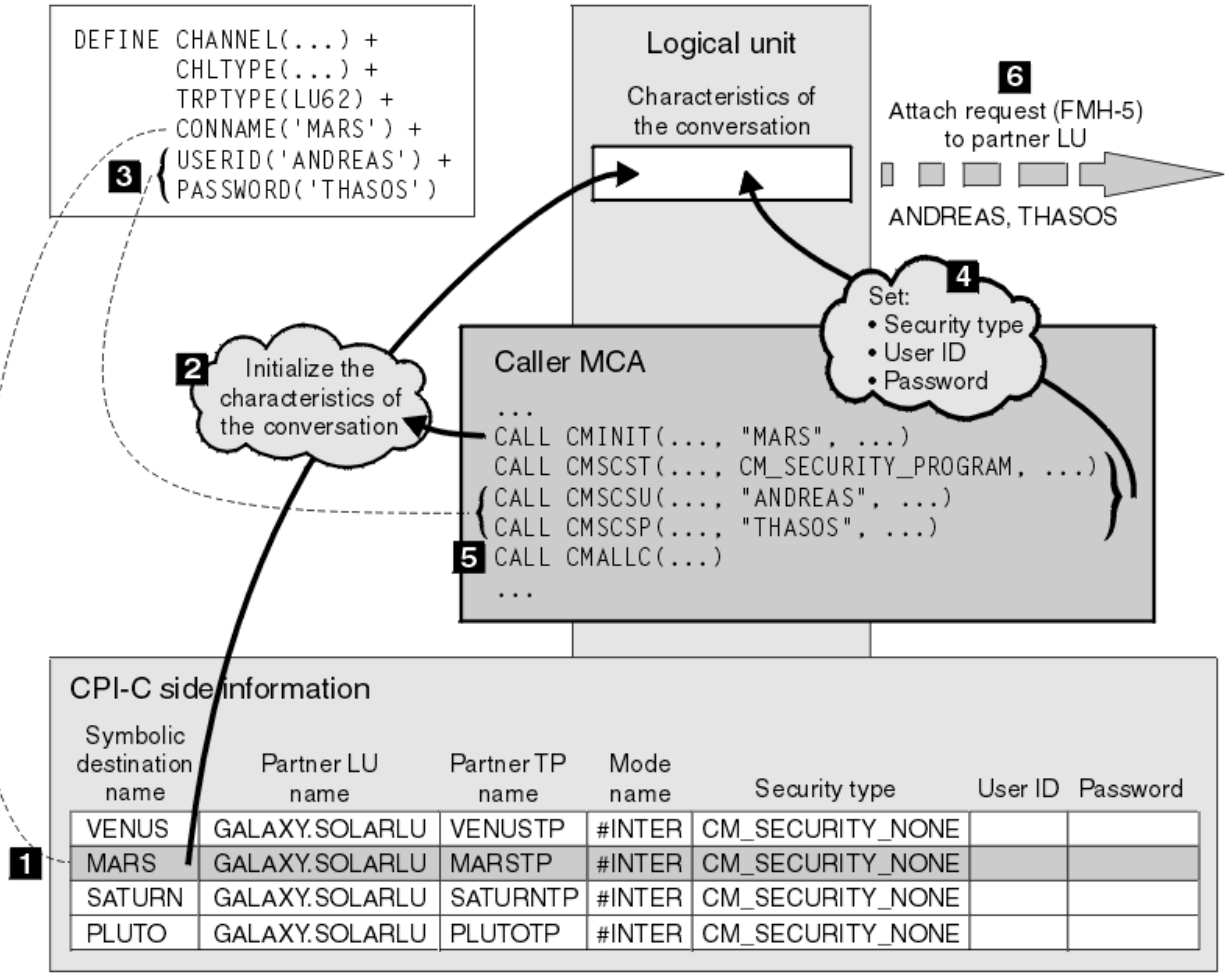



그림 *11.* 대화 레벨 인증에 대한 *WebSphere MQ* 지원

IBM i, UNIX 시스템 및 Windows 시스템에서 MCA는 CPI-C(Common Programming Interface Communications) 호출을 사용하여 SNA 네트워크를 통해 상대방 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.

## 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 WebSphere MQ는 이를 송신할 큐 관리자를 결정합니다. 여러 애플리케이션이 클러스터 큐에 메시지를 송신하면, WebSphere MQ가 큐의 인스턴스를 가지는 큐 관리자 각각에서 워크로드의 균형을 맞춥니다. 복제된 큐 관리자를 호스트하는 시스템 중 하나가 실패하면, 실패한 시스템이 재시작할 때까지 WebSphere MQ가 나머지 큐 관리자에서 워크로드의 균형을 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.

**관련 태스크**

227 페이지의 『큐 관리자가 메시지 수신하는 것을 방지』
엑시트 프로그램을 사용하여 클러스터 큐 관리자가 수신할 권한이 없는 메시지를 수신하는 것을 막을 수 있습니다.

## IBM WebSphere MQ 발행/구독에 대한 보안

IBM WebSphere MQ 발행/구독을 사용 중인 경우에는 추가 보안 고려사항이 있습니다.

발행/구독 시스템에는 발행자와 구독자의 두 애플리케이션 유형이 있습니다. 발행자는 IBM WebSphere 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 WebSphere 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 WebSphere MQ 멀티캐스트에 필요한 이유를 이해하십시오.

IBM WebSphere MQ Multicast는 기본 제공 보안이 필요하지 않습니다. 보안 검사는 MQOPEN 시에 큐 관리자에서 처리되고 MQMD 필드 설정은 클라이언트에서 처리됩니다. 네트워크의 일부 애플리케이션은 IBM WebSphere MQ 애플리케이션이 아닐 수 있으므로(예를 들어, LLM 애플리케이션의 경우 자세한 정보는 WebSphere MQ 낮은 대기 시간 메시징을 사용하는 멀티캐스트 상호 운용성 참조), 수신하는 애플리케이션이 컨텍스트 필드의 유효성을 확신할 수 없으므로 자체 보안 프로시저를 구현해야 합니다.

고려해야 하는 보안 프로세스는 세 가지입니다.

**액세스 제어**

IBM WebSphere MQ의 액세스 제어는 사용자 ID를 기반으로 합니다. 이 주제에 대한 자세한 정보는 52 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 7 페이지의 『암호화 개념』의 내용을 참조하십시오.

# 방화벽 및 인터넷 Passthru

일반적으로 방화벽을 사용하여 적대적인 IP 주소(예: 서비스 거부 공격)에서 액세스를 방지합니다. 그렇지만 보안 관리자가 방화벽 규칙을 업데이트하기를 대기하는 동안처럼 IBM WebSphere MQ 내에서 임시로 IP 주소를 차단해야 할 수도 있습니다.

하나 이상의 IP 주소를 차단하려면 유형 BLOCKADDR 또는 ADDRESSMAP의 채널 인증 레코드를 작성하십시오. 추가 정보는 166 페이지의 『특정 IP 주소 차단』의 내용을 참조하십시오.

## IBM WebSphere MQ Internet Pass-Thru에 대한 보안

인터넷 Passthru는 방화벽을 통해 통신을 단순화할 수 있지만 이는 보안에 영향을 줍니다.

IBM WebSphere MQ internet pass-thru is a IBM WebSphere MQ base product extension that is supplied in SupportPac MS81.

WebSphere MQ internet passthru를 사용하면 직접적인 TCP/IP 연결 없이 인터넷에서 두 개의 큐 관리자가 메시지를 교환하거나, WebSphere MQ 클라이언트 애플리케이션이 큐 관리자로 연결될 수 있습니다. 방화벽이 두 시스템 사이에 직접적인 TCP/IP 연결을 막는 경우에 이 기능이 유용합니다. HTTP 내부에서 플로우를 터널링하

거나 프록시로 작동하여 WebSphere MQ 채널 프로토콜이 방화벽 안팎으로 흐르는 것을 더 간단하고 관리하기 쉽도록 해줍니다. SSL(Secure Sockets Layer)을 사용하여 인터넷에서 송신되는 메시지를 암호화하고 복호화하는 데 사용할 수도 있습니다.

WebSphere MQ 시스템이 IPT와 통신할 때, IPT의 SSLProxyMode를 사용하지 않는 한 WebSphere MQ가 사용하는 CipherSpec이 IPT가 사용하는 CipherSuite와 일치하는지 확인하십시오.

- IPT가 SSL 또는 TLS 서버로 작동하고 있고 WebSphere MQ가 SSL 또는 TLS 클라이언트로 연결되어 있으면, WebSphere MQ가 사용하는 CipherSpec이 관련 IPT 키 링에서 사용할 수 있게 되어 있는 CipherSuite에 해당되어야 합니다.
- IPT가 SSL 또는 TLS 클라이언트로 작동하고 있고 WebSphere MQ SSL 또는 TLS 서버에 연결되어 있으면, IPT CipherSuite가 WebSphere MQ 채널을 수신할 때 정의되는 CipherSpec과 일치해야 합니다.

IPT에서 통합 WebSphere MQ SSL 및 TLS 지원으로 마이그레이션하는 경우 iKeyman을 사용하는 IPT에서 디지털 인증서를 전송하십시오.

자세한 정보는 WebSphere MQ Internet Pass-Thru(SupportPac MS81)를 참조하십시오.

# 보안 설정

이 토픽 콜렉션에는 다른 운영 체제 및 클라이언트 사용에 특정되는 정보가 포함됩니다.

## UNIX, Linux, and Windows 시스템에서 보안 설정

UNIX, Linux, and Windows 시스템 특정 보안 고려사항입니다.

IBM WebSphere 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 WebSphere MQ 를 관리할 수 있는 사용자**
IBM WebSphere MQ 관리를 위해 명령을 발행할 수 있는 사용자 세트를 정의할 수 있습니다.

**IBM WebSphere 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 WebSphere MQ에 액세스할 수 있는 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 WebSphere MQ에 대해 개인용입니다. MQI 자원에 권한을 부여하거나 취소하는 경우에는 표준 운영 체제 명령을 사용하지 마십시오.

### 터미널 서비스를 사용하여 IBM WebSphere MQ 연결

터미널 서비스를 사용하는 경우 **Create global objects** 사용자 권한이 문제를 일으킬 수 있습니다.

터미널 서비스를 사용하여 윈도우 시스템에 연결하고 큐 관리자를 작성하거나 시작하는 데 문제점이 있는 경우, 이는 윈도우의 최신 버전에서 사용자 권한 **Create global objects**때문일 수 있습니다.

**Create global objects** 사용자 권한은 글로벌 네임스페이스에서 오브젝트를 작성하도록 권한이 부여된 사용자를 제한합니다. 애플리케이션이 글로벌 오브젝트를 작성하려면 글로벌 네임스페이스에서 실행 중이거나 애플리케이션이 실행 중인 사용자에게 **Create global objects** 사용자 권한이 적용되어 있어야 합니다.

관리자는 기본적으로 **Create global objects** 사용자 권한이 적용되므로, 관리자는 사용자 권한을 변경하지 않고 터미널 서비스를 사용하여 연결할 때 큐 관리자를 작성하고 시작할 수 있습니다.

터미널 서비스를 사용할 때 WebSphere MQ 관리 방법이 작동하지 않는 경우 **Create global objects** 사용자 권한 설정을 시도하십시오.

1. 관리 도구 패널을 여십시오.

   **Windows 2003 및 Windows XP**
   **제어판** > **관리 도구**를 사용하여 이 패널에 액세스하십시오.

   **Windows Vista 및 Windows Server 2008**
   **제어판** > **시스템 및 유지보수** > **관리 도구**를 사용하여 이 패널에 액세스하십시오.
2. **로컬 보안 정책**을 두 번 클릭하십시오.
3. `Local Policies`을 펼치십시오.
4. `User Rights Assignment`을(를) 클릭하십시오.
5. **Create global objects** 정책에 새 사용자 또는 그룹을 추가하십시오.

## Windows에서 그룹 작성 및 관리

이 지시사항은 워크스테이션 또는 구성원 서버 시스템에서 그룹을 관리하는 프로세스를 통해 안내합니다.

도메인 제어기에 대해 사용자 및 그룹은 Active Directory를 통해 관리됩니다. Active Directory 사용에 대한 자세한 정보는 해당하는 운영 체제 지시사항을 참조하십시오.

프린시펄 그룹 멤버십에 대한 모든 변경사항은 큐 관리자가 재시작되거나 MQSC 명령 REFRESH SECURITY(또는 PCF 대응 명령)을 발행할 때까지 인식되지 않습니다.

컴퓨터 관리 패널을 사용하여 사용자 및 그룹에 대해 작업하십시오. 현재 로그온한 사용자에 대한 모든 변경사항은 사용자가 다시 로그온할 때까지 적용되지 않을 수도 있습니다.

**Windows 2003 및 Windows XP**
**제어판** > **관리 도구** > **컴퓨터 관리**를 사용하여 이 패널에 액세스하십시오.

**Windows Vista 및 Windows Server 2008**
**제어판** > **시스템 및 유지보수** > **관리 도구** > **컴퓨터 관리**를 사용하여 이 패널에 액세스하십시오.

**Windows 7**
**관리 도구** > **컴퓨터 관리**를 사용하여 이 패널에 액세스하십시오.

### *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*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 현재 선택한 위치에서 사용 가능한 그룹을 나열하려면 **지금 찾기**를 선택하십시오. 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*에서 그룹의 멤버 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*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.

## HP-UX에서 그룹 작성 및 관리
HP-UX에서 NIS 또는 NIS+를 사용하지 않는 경우, 그룹에 대해 작업하려면 시스템 관리 관리자(SAM)를 사용하십시오.

### *HP-UX*에서 그룹 작성
System Administration Manager를 사용하여 그룹에 사용자를 추가합니다.

### 프로시저
1. SAM(System Administration Manager)에서 사용자 및 그룹 계정을 두 번 클릭하십시오.
2. 그룹을 두 번 클릭하십시오.
3. 조치에서 추가 풀다운을 선택하여 새 그룹 추가 패널을 표시하십시오.
4. 그룹 이름을 입력하고 그룹에 추가하려는 사용자를 선택하십시오.
5. 적용을 클릭하여 그룹을 작성하십시오.

### 결과
이제 그룹을 작성했습니다.

### *HP-UX*에서 그룹에 사용자 추가
System Administration Manager를 사용하여 그룹에 사용자를 추가합니다.

### 프로시저
1. SAM(System Administration Manager)에서 사용자 및 그룹 계정을 두 번 클릭하십시오.
2. 그룹을 두 번 클릭하십시오.
3. 그룹 이름을 강조표시하고 조치에서 수정 풀다운을 선택하여 기존 그룹 수정 패널을 표시하십시오.
4. 그룹에 추가하려는 사용자를 선택하여 추가를 클릭하십시오.
5. 다른 그룹에 사용자 추가하려면 각 사용자에 대해 4단계를 반복하십시오.
6. 목록에 이름 추가를 완료하면 확인을 클릭하십시오.

### 결과
이제 그룹에 사용자를 추가했습니다.

### *HP-UX*에서 그룹의 멤버 표시
System Administration Manager를 사용하여 그룹의 구성원을 표시합니다.

### 프로시저

1. SAM(System Administration Manager)에서 사용자 및 그룹 계정을 두 번 클릭하십시오.
2. 그룹을 두 번 클릭하십시오.
3. 그룹 이름을 강조표시하고 조치에서 수정 풀다운을 선택하여 그룹의 사용자 목록을 보여주는 기존 그룹 수정 패널을 표시하십시오.

### 결과

그룹 구성원이 표시됩니다.

### *HP-UX*에서 그룹에서 사용자 제거

System Administration Manager를 사용하여 그룹에서 사용자를 제거합니다.

### 프로시저

1. SAM(System Administration Manager)에서 사용자 및 그룹 계정을 두 번 클릭하십시오.
2. 그룹을 두 번 클릭하십시오.
3. 그룹 이름을 강조표시하고 조치에서 수정 풀다운을 선택하여 기존 그룹 수정 패널을 표시하십시오.
4. 그룹에서 제거하려는 사용자를 선택하여 제거를 클릭하십시오.
5. 그룹에서 다른 사용자를 제거하려면 각 사용자에 대해 4단계를 반복하십시오.
6. 목록에서 이름 제거를 완료하면 확인을 클릭하십시오.

### 결과

이제 그룹에서 사용자를 제거했습니다.

## AIX에서 그룹 작성 및 관리

AIX에서 NIS 또는 NIS+를 사용하지 않는 경우, 그룹에 대해 작업하려면 SMITTY를 사용하십시오.

### 그룹 작성

SMITTY를 사용하여 그룹을 작성합니다.

### 프로시저

1. SMITTY에서 보안 및 사용자를 선택하고 Enter를 누르십시오.
2. 그룹을 선택하고 Enter를 누르십시오.
3. 그룹 추가를 선택하고 Enter를 누르십시오.
4. 그룹 이름 및 그룹에 추가하려는 모든 사용자 이름을 쉼표로 구분하여 입력하십시오.
5. Enter를 눌러 그룹을 작성하십시오.

### 결과

이제 그룹을 작성했습니다.

### 그룹에 사용자 추가

SMITTY를 사용하여 그룹에 사용자 추가합니다.

### 프로시저

1. SMITTY에서 보안 및 사용자를 선택하고 Enter를 누르십시오.
2. 그룹을 선택하고 Enter를 누르십시오.
3. 그룹 특성 변경 / 표시를 선택하고 Enter를 누르십시오.
4. 그룹 구성원 목록을 표시하려는 그룹 이름을 입력하십시오.
5. 그룹에 추가하려는 사용자 이름을 쉼표로 구분하여 추가하십시오.

6. 이름을 그룹에 추가하려면 Enter를 누르십시오.

### 그룹에 있는 사용자 표시
SMITTY를 사용하여 그룹에 있는 구성원을 표시합니다.

### 프로시저
1. SMITTY에서 보안 및 사용자를 선택하고 Enter를 누르십시오.
2. 그룹을 선택하고 Enter를 누르십시오.
3. 그룹 특성 변경 / 표시를 선택하고 Enter를 누르십시오.
4. 그룹 구성원 목록을 표시하려는 그룹 이름을 입력하십시오.

### 결과
그룹 구성원이 표시됩니다.

### 그룹에서 사용자 제거
SMITTY를 사용하여 그룹에서 사용자를 제거합니다.

### 프로시저
1. SMITTY에서 보안 및 사용자를 선택하고 Enter를 누르십시오.
2. 그룹을 선택하고 Enter를 누르십시오.
3. 그룹 특성 변경 / 표시를 선택하고 Enter를 누르십시오.
4. 그룹 구성원 목록을 표시하려는 그룹 이름을 입력하십시오.
5. 그룹에서 제거하려는 사용자 이름을 삭제하십시오.
6. 그룹에서 이름을 제거하려면 Enter를 누르십시오.

### 결과
이제 그룹에서 사용자를 제거했습니다.

## Solaris에서 그룹 작성 및 관리
Solaris에서 NIS 또는 NIS+를 사용하지 않는 경우 그룹에 대해 작업하려면 `/etc/group` 파일을 사용하십시오.

### *Solaris*에서 그룹 작성
**groupadd** 명령을 사용하여 그룹을 작성합니다.

### 프로시저
다음 명령을 입력하십시오. `groupadd` *`group-name`*
여기서, *group-name*은 그룹 이름입니다.

### 결과
파일 `/etc/group`은 그룹 정보를 보유합니다.

### *Solaris*에서 그룹에 사용자 추가
**usermod** 명령을 사용하여 그룹에 사용자를 추가합니다.

### 프로시저
보충 그룹에 구성원을 추가하려면 `usermod` 명령을 실행하여 사용자가 현재 구성원인 보충 그룹 및 사용자가 구성원이 되는 보충 그룹을 나열하십시오.
예를 들어, 사용자가 그룹 `groupa`의 멤버이고 또한 `groupb`의 멤버가 될 경우, `usermod -G groupa,groupb` *`user-name`* 명령을 사용하십시오. 여기서 *user-name*은 사용자 이름입니다.

### *Solaris*에서 그룹에 있는 멤버 표시
그룹의 구성원을 알려면 /etc/group 파일에서 해당 그룹의 항목을 검사하십시오.

### *Solaris*의 그룹에서 사용자 제거
**usermod** 명령을 사용하여 그룹에서 사용자를 제거합니다.

#### 프로시저
보충 그룹에서 구성원을 제거하려면 **usermod** 명령을 실행하여 사용자를 구성원으로 남겨두려는 보충 그룹을 표시하십시오.
예를 들어, 사용자의 기본 그룹이 users이고 사용자가 mqm, groupa, groupb의 구성원이기도 한 경우 사용자를 mqm 그룹에서 제거하려면 `usermod -G groupa,groupb user-name` 명령을 사용하십시오. 여기서, *user-name*은 사용자 이름입니다.

## Linux에서 그룹 작성 및 관리
Linux에서 NIS 또는 NIS+를 사용하지 않는 경우 /etc/group 파일을 사용하여 그룹에 대해 작업하십시오.

### *Linux* 에서 그룹 작성
**groupadd** 명령을 사용하여 그룹을 작성합니다.

#### 프로시저
새 그룹을 작성하려면 다음 명령을 입력하십시오. `groupadd -g group-ID group-name`
여기서, *group-ID*는 그룹의 숫자 ID이고 *group-name*은 그룹 이름입니다.

#### 결과
파일 /etc/group은 그룹 정보를 보유합니다.

### *Linux*에서 그룹에 사용자 추가
**usermod** 명령을 사용하여 그룹에 사용자를 추가합니다.

#### 프로시저
보충 그룹에 구성원을 추가하려면 usermod 명령을 실행하여 사용자가 현재 구성원인 보충 그룹 및 사용자가 구성원이 되는 보충 그룹을 나열하십시오.
예를 들어, 사용자가 groupa 그룹의 구성원이고 groupb 그룹의 구성원도 되는 경우 `usermod -G groupa,groupb user-name` 명령을 사용하십시오.
여기서, *user-name*은 사용자 이름입니다.

### *Linux*의 그룹에 있는 구성원 표시
**getent** 명령을 사용하여 그룹의 구성원을 표시합니다.

#### 프로시저
그룹의 구성원인 사용자를 표시하려면 다음 명령을 입력하십시오. `getent group group-name`
여기서, *group-name*은 그룹 이름입니다.

### 그룹에서 사용자 제거
**usermod** 명령을 사용하여 그룹에서 사용자를 제거합니다.

#### 프로시저
보충 그룹에서 구성원을 제거하려면 **usermod** 명령을 실행하여 사용자를 구성원으로 남겨두려는 보충 그룹을 표시하십시오.

예를 들어, 사용자의 기본 그룹이 users이고 사용자가 mqm, groupa 및 groupb의 멤버이기도 한 경우 mqm 그룹에서 사용자를 제거하려면 usermod -G groupa,groupb *user-name* 명령을 사용합니다.

여기서 *user-name*은 사용자 이름입니다.

## 권한 부여 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.

테이블에서 접두부가 MQZAO_인 상수는 특정 엔티티를 위한 setmqaut 명령에 대한 권한 목록 키워드에 대응합니다. 예를 들어, MQZAO_BROWSE는 키워드 +browse에 대응하고 MQZAO_SET_ALL_CONTEXT는 키워드 +setall 등에 대응합니다. 이런 상수는 제품에서 제공하는 헤더 파일 cmqzc.h에 정의됩니다.

### *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 MQPUT1의 경우, 권한 검사는 이름이 해석된 후에 발생되는 이름이 아니라 열리고 있는 오브젝트의 이름에서 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.

82 페이지의 표 8, 82 페이지의 표 9, 82 페이지의 표 10, 83 페이지의 표 11에는 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 *8. MQCONN* 호출에 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(83 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| **MQCONN** | 적용할 수 없음 | 적용할 수 없음 | MQZAO_CONNECT |

| 표 *9. MQOPEN* 호출에 대해 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(83 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 **(83 페이지의 『2』)** | MQZAO_INPUT | 적용할 수 없음 | 적용할 수 없음 |
| MQOO_OUTPUT(정상 큐) **(83 페이지의 『3』)** | MQZAO_OUTPUT | 적용할 수 없음 | 적용할 수 없음 |
| MQOO_PASS_IDENTITY_CONTEXT **(83 페이지의 『4』)** | MQZAO_PASS_IDENTITY_CONTEXT | 적용할 수 없음 | 검사 안함 |
| MQOO_PASS_ALL_CONTEXT**(83 페이지의 『4』, 83 페이지의 『5』)** | MQZAO_PASS_ALL_CONTEXT | 적용할 수 없음 | 검사 안함 |
| MQOO_SET_IDENTITY_CONTEXT**(83 페이지의 『4』, 83 페이지의 『5』)** | MQZAO_SET_IDENTITY_CONTEXT | 적용할 수 없음 | MQZAO_SET_IDENTITY_CONTEXT **(83 페이지의 『6』)** |
| MQOO_SET_ALL_CONTEXT**(83 페이지의 『4』, 83 페이지의 『7』)** | MQZAO_SET_ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT **(83 페이지의 『6』)** |
| MQOO_OUTPUT(전송 큐) **(83 페이지의 『8』)** | MQZAO_SET_ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ALL_CONTEXT **(83 페이지의 『6』)** |
| MQOO_SET | MQZAO_SET | 적용할 수 없음 | 검사 안함 |
| MQOO_ALTERNATE_USER_AUTHORITY | **(83 페이지의 『9』)** | **(83 페이지의 『9』)** | MQZAO_ALTERNATE_USER_AUTHORITY**(83 페이지의 『9』, 84 페이지의 『10』)** |

| 표 *10. MQPUT1* 호출에 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(83 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQPMO_PASS_IDENTITY_CONTEXT | MQZAO_PASS_IDENTITY_CONTEXT **(84 페이지의 『11』)** | 적용할 수 없음 | 검사 안함 |

| 표 *10. MQPUT1* 호출에 필요한 보안 권한 *(계속)* | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(83 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQPMO_PASS_ALL _CONTEXT | MQZAO_PASS_ ALL_CONTEXT **(84 페이지의 『11』)** | 적용할 수 없음 | 검사 안함 |
| MQPMO_SET_ IDENTITY_CONTEXT | MQZAO_SET_ IDENTITY_CONTEXT **(84 페이지의 『11』)** | 적용할 수 없음 | MQZAO_SET_ IDENTITY_CONTEXT **(83 페이지의 『6』)** |
| MQPMO_SET_ ALL_CONTEXT | MQZAO_SET_ ALL_CONTEXT **(84 페이지의 『11』)** | 적용할 수 없음 | MQZAO_SET_ ALL_CONTEXT **(83 페이지의 『6』)** |
| (전송 큐) **(83 페이지의 『8』)** | MQZAO_SET_ ALL_CONTEXT | 적용할 수 없음 | MQZAO_SET_ ALL_CONTEXT **(83 페이지의 『6』)** |
| MQPMO_ALTERNATE_ USER_AUTHORITY | **(84 페이지의 『12』)** | 적용할 수 없음 | MQZAO_ALTERNATE_ USER_AUTHORITY **(84 페이지의 『10』)** |

| 표 *11. MQCLOSE* 호출에 대해 필요한 보안 권한 | | | |
|---|---|---|---|
| **다음에 대한 권한 필요** | **큐 오브젝트(83 페이지의 『1』)** | **프로세스 오브젝트** | **큐 관리자 오브젝트** |
| MQCO_DELETE | MQZAO_DELETE **(84 페이지의 『13』)** | 적용할 수 없음 | 적용할 수 없음 |
| MQCO_DELETE _PURGE | MQZAO_DELETE **(84 페이지의 『13』)** | 적용할 수 없음 | 적용할 수 없음 |

**표에 대한 참고 사항:**

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(83 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 모두에 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.

## 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 |
| 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 |

**CLEAR *object***

| **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 |
| 채널 | 적용할 수 없음 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 클라이언트 연결 채널 | 적용할 수 없음 |
| 리스너 | 적용할 수 없음 |
| 서비스 | 적용할 수 없음 |
| 통신 정보 | 적용할 수 없음 |

**DEFINE *object* NOREPLACE (88 페이지의 『1』)**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 주제 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 프로세스 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 큐 관리자 | 적용할 수 없음 |
| 이름 목록 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 인증 정보 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 채널 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 클라이언트 연결 채널 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 리스너 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 서비스 | MQZAO_CREATE **(88 페이지의 『2』)** |
| 통신 정보 | MQZAO_CREATE **(88 페이지의 『2』)** |

**DEFINE *object* REPLACE (88 페이지의 『1』, 88 페이지의 『3』)**

| 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 |

**DELETE *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DELETE |
| 주제 | MQZAO_DELETE |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 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 |
| 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 |

**START *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 주제 | 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 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 통신 정보 | 적용할 수 없음 |

**STOP *object***

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 주제 | 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 | 주제 | MQZAO_CONTROL |
| DEFINE SUB | 주제 | MQZAO_CONTROL |
| DELETE SUB | 주제 | MQZAO_CONTROL |
| DISPLAY SUB | 주제 | 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 |

| 명령 | 오브젝트 | 필수 권한 |
|---|---|---|
| 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

## *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 |
| 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 |

***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 |
| 통신 정보 | 적용할 수 없음 |

***object* 복사 (바꾸지 않음) ( 1)**

| **오브젝트** | **필수 권한** |
|---|---|
| 큐 | MQZAO_CREATE **( 2)** |
| 토픽 | 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 **(94 페이지의 『2』 )** |

***object* 복사 (대체 포함) ( 1, 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 |
| 통신 정보 | MQZAO_CHANGE |

***object* 를 작성하십시오 (교체하지 않음) ( 3)**

| **오브젝트** | **필수 권한** |
|---|---|
| 큐 | MQZAO_CREATE **( 2)** |
| 토픽 | 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)** |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 클라이언트 연결 채널 | MQZAO_CREATE **( 2)** |
| 리스너 | MQZAO_CREATE **( 2)** |
| 서비스 | MQZAO_CREATE **( 2)** |
| 통신 정보 | MQZAO_CREATE **( 2)** |

***object* 작성 (대체 포함) ( 3, 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 |
| 통신 정보 | MQZAO_CHANGE |

***object* 삭제**

| 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 |

***object* 조회**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | MQZAO_DISPLAY |
| 토픽 | MQZAO_DISPLAY |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 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 |
| 주제 | 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 |
| 주제 | 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 |

| 오브젝트 | 필수 권한 |
| --- | --- |
| 통신 정보 | 적용할 수 없음 |

***object* 중지**

| 오브젝트 | 필수 권한 |
| --- | --- |
| 큐 | 적용할 수 없음 |
| 주제 | 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 | 주제 | MQZAO_CONTROL |
| 구독 작성 | 주제 | MQZAO_CONTROL |
| 구독 삭제 | 주제 | MQZAO_CONTROL |
| 구독 조회 | 주제 | 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 |

| 명령 | 오브젝트 | 필수 권한 |
|---|---|---|
| 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 | |
| 클러스터 재설정 | 'mqm' 그룹 멤버십 필수 | |
| 큐 관리자 클러스터 일시중단 | 'mqm' 그룹 멤버십 필수 | |
| 큐 관리자 클러스터 재개 | '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 |
| 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. 작성 명령의 경우, 해당 SYSTEM.DEFAULT.* 객체 (object)

4. 이는 바꿀 오브젝트가 이미 있는 경우에도 적용됩니다. 없는 경우, 검사는 복사 또는 작성(바꾸기 없음)에 대해 수행됩니다.

## Windows에서 보안에 대한 특별 고려사항

일부 보안 기능은 여러 가지 Windows 버전에서 각각 다르게 작동합니다.

IBM WebSphere MQ 보안은 사용자 권한 및 그룹 멤버십에 대한 정보의 운영 체제 API에 대한 호출에 의존합니다. 일부 기능은 Windows 시스템에서 동등하게 작동하지 않습니다. 이 주제 콜렉션에는 환경 IBM WebSphere MQ 를 실행할 때 이러한 차이점이 IBM WebSphere MQ 보안에 미치는 영향에 대한 설명이 포함되어 있습니다.

### *SSPI* 채널 엑시트 프로그램

WebSphere MQ for 윈도우 supplies a security exit program, which can be used on both message and MQI channels. 엑시트는 소스 및 오브젝트 코드로 제공되며 단방향 및 양방향 인증을 제공합니다.

보안 엑시트는 Windows 플랫폼의 통합된 보안 기능을 제공하는 SSPI(Security Support Provider Interface)를 사용합니다.

보안 엑시트는 다음의 식별 및 인증 서비스를 제공합니다.

**단방향 인증**

이것은 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 WebSphere MQ 버전 7.0에서 지원하는 모든 플랫폼 지원됩니다.

이 서비스는 일반적으로 MQI 채널에서 서버 큐 관리자가 WebSphere MQ MQI 클라이언트 애플리케이션을 인증할 수 있게 하게 하기 위해 사용됩니다. 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 WebSphere MQ 버전 7.0에서 지원하는 모든 플랫폼 지원됩니다.

이 서비스는 메시지와 MQI 채널 둘 다에서 사용될 수 있습니다. 메시지 채널에서는 두 개의 큐 관리자의 상호 인증을 제공합니다. MQI 채널에서는 서버 큐 관리자와 WebSphere MQ MQI 클라이언트 애플리케이션이 서로를 인증하게 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 방식에 대한 자세한 정보와 이 프로그램을 구현하는 방법에 대한 지시사항은 Windows 시스템에서 SSPI 보안 엑시트 사용을 참조하십시오.

### *Windows*에서 '그룹을 찾을 수 없음' 오류가 발생할 경우

This problem can arise because WebSphere MQ loses access to the local mqm group when 윈도우 servers are promoted to, or demoted from, domain controllers. 이러한 문제점을 해결하려면 로컬 mqm 그룹을 재작성하십시오.

증상은 로컬 mqm 그룹 부족을 표시하는 오류로 예를 들어 다음과 같습니다.

```
>crtmqm qm0
AMQ8066:Local mqm group not found.
```

WebSphere MQ가 로컬로 정의된 mqm 그룹을 사용하기 때문에 서버와 도메인 제어기 사이의 시스템 상태를 대체하면 WebSphere MQ의 조작에 영향을 줄 수 있습니다. 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.

이러한 문제점을 해결하려면 표준 Windows 관리 도구를 사용하여 로컬 mqm 그룹을 재작성하십시오. 모든 그룹 멤버십 정보가 삭제되므로 권한이 있는 WebSphere MQ 사용자를 새로 작성한 로컬 mqm 그룹에서 다시 인스턴스화해야 합니다. 시스템이 도메인 멤버일 경우, 또한 도메인 mqm 그룹을 로컬 mqm 그룹에 추가하여 권한이 있는 도메인 WebSphere MQ 사용자 ID에게 필요한 레벨의 권한을 부여해야 합니다.

### 윈도우 에서 *IBM WebSphere MQ* 및 도메인 컨트롤러에 문제가 있는 경우

Windows 서버를 도메인 제이기로 승격할 때 보안 설정값에 특정한 문제점이 발생할 수 있습니다.

Windows 2000, Windows 2003 또는 Windows Server 2008 서버를 도메인 제어기로 승격하는 동안 사용자 및 그룹 권한에 관련된 기본 또는 기본이 아닌 보안 설정값을 선택하는 옵션이 제공됩니다. 이 옵션은 임의 사용자가 Active Directory에서 그룹 멤버십을 검색할 수 있도록 하는지 여부를 제어합니다. WebSphere MQ는 그룹 멤버십 정보에 따라 보안 정책을 구현하므로 WebSphere MQ 조작을 수행하고 있는 사용자 ID는 다른 사용자의 그룹 멤버십을 판별할 수 있어야 합니다.

Windows 2000에서 기본 보안 옵션을 사용하여 도메인을 작성하면, 설치 프로세스 동안 WebSphere MQ에서 작성된 기본 사용자 ID는 필요한 다른 사용자의 그룹 멤버십을 확보할 수 있습니다. 그러면 제품이 정상적으로 설치되어 기본 오브젝트를 작성하고 큐 관리자는 필요에 따라 로컬 및 도메인 사용자의 액세스 권한을 판별할 수 있습니다.

Windows 2000에서 기본이 아닌 보안 옵션을 사용하여 도메인을 작성하거나 Windows 2003 및 Windows Server 2008에서 기본 보안 옵션을 사용하여 도메인을 작성할 경우, 설치 프로세스 동안 WebSphere MQ에서 작성된 사용자 ID가 필요한 그룹 멤버십을 항상 판별할 수는 없습니다. 이런 경우 다음을 알아야 합니다.

- 기본이 아닌 보안 옵션을 사용하는 Windows2000 또는 기본 보안 옵션을 사용하는 Windows 2003및 Windows Server 2008에서 보안 권한의 작동 방법
- 도메인 mqm 그룹 멤버가 그룹 멤버십을 읽는 방법
- 도메인 사용자에서 실행되도록 IBM WebSphere MQ Windows 서비스를 구성하는 방법

기본이 아닌 보안 권한이 있는 *Windows 2000* 도메인 또는 기본 보안 권한이 있는 *Windows 2003* 및 *Windows Server 2008* 도메인
WebSphere MQ 설치는 로컬 사용자가 설치를 수행하는지 도메인 사용자가 수행하는지에 따라 이러한 운영 체제에서 다르게 작동합니다.

**로컬** 사용자가 WebSphere MQ를 설치하는 경우 WebSphere MQ 준비 마법사는 IBM WebSphere MQ Windows 서비스를 위해 작성된 로컬 사용자가 설치 중인 사용자의 그룹 멤버십 정보를 검색할 수 있는지 감지합니다. WebSphere MQ 준비 마법사는 사용자에게 Windows 2000 이상에서 실행 중인 도메인 제어기에 정의된 기타 사용자 계정이 있는지 여부를 판별하기 위해 네트워크 구성에 관한 질문을 묻습니다. 그런 경우, IBM WebSphere MQ Windows 서비스는 특정 설정 및 권한의 도메인 사용자 계정으로 실행되어야 합니다. WebSphere MQ 준비 마법사는 사용자에게 이 사용자의 계정 세부사항에 대해 프롬프트합니다. 해당 온라인 도움말은 도메인 관리자에게 송신 가능한 필요한 도메인 사용자 계정 세부사항을 제공합니다.

**도메인** 사용자가 WebSphere MQ를 설치하는 경우 WebSphere MQ 준비 마법사는 IBM WebSphere MQ Windows 서비스를 위해 작성된 로컬 사용자가 설치 중인 사용자의 그룹 멤버십 정보를 검색할 수 없는지 감지합니다. 이 경우, WebSphere MQ 준비 마법사는 사용자에게 사용할 IBM WebSphere MQ Windows 서비스에 대한 도메인 사용자 계정의 계정 세부사항에 대해 항상 프롬프트합니다.

IBM WebSphere MQ Windows 서비스가 도메인 사용자 계정을 사용해야 하는 경우, WebSphere MQ는 WebSphere MQ 준비 마법사를 사용하여 구성될 때까지 올바로 작동할 수 없습니다. WebSphere MQ 준비 마법사는 Windows 서비스가 적당한 계정으로 구성될 때까지 사용자가 다른 태스크를 계속 수행하지 못하게 합니다.

Windows 2000 도메인이 기본이 아닌 보안 권한으로 구성된 경우, WebSphere MQ가 올바로 작업할 수 있게 하는 일반적인 방법은 위에 설명된 대로 적당한 도메인 사용자 계정으로 구성하는 것입니다.

자세한 정보는 WebSphere MQ에 대한 도메인 계정 작성 및 설정을 참조하십시오.

*Windows* 에서 도메인 사용자가 실행하도록 *IBM WebSphere MQ* 서비스 구성
IBM WebSphere MQ 준비 마법사를 사용하여 도메인 사용자 계정 세부사항을 입력하십시오. 또는 컴퓨터 관리 패널을 사용하여 설치 특정 IBM WebSphere MQ 서비스에 대한 **로그온** 세부사항을 변경할 수 있습니다.

자세한 정보는 IBM WebSphere MQ Windows 서비스 사용자 계정의 비밀번호 변경 을 참조하십시오.

## *Windows*에 보안 템플리트 파일 적용

템플리트를 적용하면 WebSphere MQ 파일 및 디렉토리에 적용된 보안 설정에 영향을 줄 수 있습니다. 고급 보안 템플리트를 사용할 경우, WebSphere MQ를 설치하기 전에 이를 적용하십시오.

Windows는 보안 구성 및 분석 MMC 스냅인으로 하나 이상의 컴퓨터에 동일한 보안 설정을 적용하는 데 사용할 수 있는 텍스트 기반 보안 템플리트 파일을 지원합니다. 특히 Windows는 특정 레벨의 보안을 제공하기 위해 다양한 보안 설정이 포함된 다중 템플리트를 제공합니다. 이런 템플리트에는 Compatible, Secure, Highly Secure가 포함됩니다.

이러한 템플리트 중 하나를 적용하면 WebSphere MQ 파일 및 디렉토리에 적용된 보안 설정에 영향을 줄 수 있습니다. 고급 보안 템플리트를 사용하려는 경우 WebSphere MQ를 설치하기 전에 시스템을 구성하십시오.

WebSphere MQ가 이미 설치되어 있는 시스템에 고급 보안 템플리트를 적용할 경우, WebSphere MQ 파일 및 디렉토리에 설정한 모든 권한이 제거됩니다. 이 권한이 제거되기 때문에 오류 디렉토리에서 관리자, *mqm*, 적용되는 경우 모든 사용자 그룹 액세스가 유실됩니다.

## 중첩 그룹

중첩 그룹 사용에는 제한이 있습니다. 이 제한사항의 결과는 도메인 기능 레벨에서 일부 나타나고 WebSphere MQ 제한사항에서 일부 나타납니다.

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 2003 , Windows XP, Windows Vista 및 Windows Server 2008은 모두 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 요구사항 외에도 WebSphere MQ에서 사용되는 ID에는 추가 제한사항이 적용됩니다. WebSphere MQ에서 사용되는 네트워크 API는 도메인 기능 레벨이 지원하는 모든 구성을 지원하지 않습니다. 따라서 WebSphere MQ는 로컬 그룹에 중첩된 도메인 로컬 그룹에 있는 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.

## *IBM WebSphere MQ* 에 연결하는 애플리케이션 대한 추가 권한 구성

IBM WebSphere MQ 프로세스가 실행되는 계정은 애플리케이션 프로세스에 대해 SYNCHRONIZE 액세스가 부여되기 전에 추가 인증이 필요할 수도 있습니다.

애플리케이션 예: ASP 페이지) 이 일반적인 보안 레벨에서 실행되도록 구성된 IBM WebSphere MQ 에 연결되어 있는 경우 문제점이 발생할 수 있습니다.

IBM WebSphere MQ의 경우 특정 조치를 통합하려면 애플리케이션 프로세스에 대해 SYNCHRONIZE 액세스가 필요합니다. APAR IC35116은 적절한 권한이 지정되도록 IBM WebSphere MQ를 변경했습니다. 그렇지만 IBM WebSphere MQ 프로세스가 실행되는 계정은 요청된 액세스가 부여되기 전에 추가 권한이 필요할 수도 있습니다.

서버 애플리케이션이 처음으로 큐 관리자 연결을 시도하는 경우, IBM WebSphere MQ는 IBM WebSphere MQ 관리자에 대해 SYNCHRONIZE 권한을 부여하기 위해 프로세스를 수정합니다. IBM WebSphere MQ 프로세스가 실행되는 사용자 ID에 대해 추가 권한을 구성하려면 다음 단계를 완료하십시오.

1. 로컬 보안 정책 도구를 시작하고, 보안 설정 - > 로컬 정책 - > 사용자 권한 지정을 클릭하고, "프로그램 디버그" 를 클릭하십시오.
2. "디버그 프로그램"을 두 번 클릭하고 IBM WebSphere MQ 사용자 ID를 목록에 추가하십시오.

시스템이 Windows 도메인에 있으며 유효한 정책 설정이 여전히 설정되지 않은 경우, 로컬 정책 설정이 설정되어 있더라도 사용자 ID는 도메인 보안 정책 도구를 사용하여 도메인 레벨과 같은 방법으로 권한 부여되어야 합니다.

# HP Integrity NonStop Server에서 보안 설정

HP Integrity NonStop Server 시스템 특정 보안 고려사항입니다.

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 큐 관리자에 연결할 때 링크 레벨 보안을 제공하기 위해 TLS (Transport Layer Security) 및 SSL (Secure Sockets Layer) 프로토콜을 모두 지원합니다. 이러한 프로토콜은 OpenSSL의 구현을 사용하여 지원됩니다. OpenSSL은 강력한 암호화 연산을 제공하기 위해 무작위 데이터의 소스를 필요로 합니다.

## OpenSSL

HP Integrity NonStop Server의 IBM WebSphere MQ 클라이언트에 대한 OpenSSL 보안 개요

OpenSSL 툴킷은 네트워크를 통한 안전한 통신을 목적으로 하는 SSL(Secure Sockets Layer) 및 TLS(Transport Layer Security) 프로토콜의 오픈 소스 구현입니다.

이 툴킷은 OpenSSL Project에 의해 개발되었습니다. OpenSSL Project에 대한 자세한 정보는 https://www.openssl.org의 내용을 참조하십시오. HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트에는 OpenSSL 라이브러리의 수정된 버전 및 **openssl** 명령이 있습니다. 이 라이브러리와 **openssl** 명령은 OpenSSL 툴킷 1.0.1c로부터 포트되었으며 오브젝트 코드로만 제공됩니다. 소스 코드는 제공되지 않습니다.

OpenSSL 라이브러리는 필요에 따라 동적으로 IBM WebSphere MQ 클라이언트 애플리케이션 프로그램에 의해 로드됩니다. IBM WebSphere MQ에서 제공하는 OpenSSL 라이브러리만 IBM WebSphere MQ 클라이언트 애플리케이션용으로 지원됩니다.

인증서 관리 목적을 위해 사용할 수 있는 **openssl** 명령은 OSS 디렉토리 `opt_installation_path/opt/mqm/bin`에 설치되어 있습니다.

**openssl** 명령을 사용하면 다양한 공통 데이터 형식으로 키 및 디지털 인증서를 작성하고 관리할 수 있으며 간단한 인증 기관(CA) 태스크를 수행할 수 있습니다.

OpenSSL에 의해 처리되는 키 및 인증서 데이터의 기본 형식은 PEM(Privacy Enhanced Mail) 형식입니다. PEM 형식의 데이터는 base64로 인코딩된 ASCII 데이터입니다. 따라서 이 데이터는 이메일과 같은 텍스트 기반 시스템을 사용하여 전송될 수 있으며, 텍스트 편집기 및 웹 브라우저를 사용하여 잘라내고 붙여넣을 수 있습니다. PEM은 텍스트 기반 암호화된 교환에 대한 인터넷 표준이며 인터넷 RFC 1421, 1422, 1423 및 1424에 지정되어 있습니다. IBM WebSphere MQ 는 확장자가 `.pem` 인 파일이 PEM 형식의 데이터를 포함한다고 가정합니다. PEM 형식의 파일에는 여러 인증 및 기타 인코드된 오브젝트가 포함될 수 있으며 주석도 포함될 수 있습니다.

다른 운영 체제에서의 IBM WebSphere MQ SSL 지원에서는 파일 내의 키 및 인증서 데이터를 DER(Distinguished Encoding Rules)을 사용하여 인코딩하도록 요구할 수 있습니다. DER은 보안 통신에서 ASN.1 표시를 사용하는 인코딩 규칙 세트입니다. DER을 사용하여 인코딩된 데이터는 2진 데이터이며 DER를 사용하여 인코딩된 키 및 인증서 데이터의 형식은 PKCS#12 또는 PFX라고도 합니다. 이 데이터를 포함하는 파일은 일반적으로 `.p12` 또는 `.pfx` 확장자를 갖고 있습니다. **openssl** 명령은 PEM과 PKCS#12 형식을 서로 변환할 수 있습니다.

## 엔트로피 디먼

OpenSSL은 강력한 암호화 연산을 제공하기 위해 무작위 데이터의 소스를 필요로 합니다. 난수 생성은 운영 체제 또는 시스템 전체 디먼 프로세스에 의해 제공되는 기능입니다. HP Integrity NonStop Server 운영 체제는 운영 체제 내에서 이 기능을 제공하지 않습니다.

HP Integrity NonStop Server에 대해 IBM WebSphere MQ 클라이언트와 함께 제공되는 SSL및 TLS 지원을 사용하는 경우 무작위 데이터의 소스를 제공하려면 엔트로피 디먼이라고 하는 프로세스가 필요합니다. SSL 또는 TLS를 필요로 하는 클라이언트 채널을 시작할 때 OpenSSL은 엔트로피 디먼이 실행 중이며 `/etc/egd-pool`에 있는 OSS 파일 시스템의 소켓에서 서비스를 제공하고 있다고 가정합니다.

엔트로피 디먼은 HP Integrity NonStop Server의 IBM WebSphere MQ 클라이언트에서 제공되지 않습니다. HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 다음 엔트로피 디먼으로 테스트됩니다.

- amqjkdm0(IBM WebSphere MQ 5.3 서버에서 제공)
- `/usr/local/bin/prngd`(버전 0.9.27, HP Integrity NonStop Server Open Source Technical Library에서 제공)

# IBM WebSphere MQ MQI 클라이언트 보안 설정

클라이언트 애플리케이션에 서버의 자원에 대한 무제한 액세스 권한이 없도록 IBM WebSphere MQ MQI 클라이언트 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.

  SSL 또는 TLS에서 상호 인증을 사용해야 합니다. 자세한 정보는 101 페이지의 『SSL 또는 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 검증하기 위해 사용 중인 사용자 ID를 변경하는 기능이 있습니다.

  액세스 제어에 대한 자세한 정보는 145 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 168 페이지의 『MCAUSER 사용자 ID에 클라이언트 확인 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 37 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.

런타임 시에 FIPS 준수를 위해서는 -fips 옵션과 함께 runmqakm 등과 같은 FIPS 준수 소프트웨어만을 사용하여 작성 및 관리되었어야 합니다.

SSL 또는 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 MQSSLFIPS를 YES로 설정하십시오.
3. 클라이언트 구성 파일의 SSLFipsRequired 속성을 YES로 설정하십시오.

기본적으로 FIPS-인증 CipherSpec은 필요하지 않습니다.

이러한 값은 ALTER QMGR SSLFIPS(ALTER QMGR 참조)에서 동등한 매개변수 값과 동일한 의미가 있습니다. 현재 클라이언트 프로세스에 활성 SSL 또는 TLS 연결이 없고, FipsRequired 값이 SSL MQCONNX에 올바르게 지정된 경우에는 이 프로세스와 연관된 모든 후속 SSL 연결은 이 값과 연관된 CipherSpec만을 사용해야 합니다. 이는 이 SSL 및 다른 모든 SSL 또는 TLS 연결이 중지될 때까지 적용되고, 이 단계에서는 후속 MQCONNX는 FipsRequired의 새 값을 제공할 수 있습니다.

암호화 하드웨어가 존재하는 경우 WebSphere MQ가 사용하는 암호화 모듈은 하드웨어 제품에서 제공한 모듈로 구성될 수 있으며, 이러한 모듈은 특정 레벨로 FIPS 인증될 수 있습니다. 구성 가능한 모듈 및 이들이 FIPS-인증인지 여부는 사용 중인 하드웨어 제품에 따라 다릅니다.

가능한 경우 FIPS-전용 CipherSpec이 구성되면 MQI 클라이언트는 MQRC_SSL_INITIALIZATION_ERROR와 함께 비FIPS CipherSpec을 지정하는 연결을 거부합니다. WebSphere MQ가 이러한 연결 모두를 거부하도록 보장하지는 않습니다. WebSphere MQ 구성이 FIPS와 호환되는지 여부를 판별하는 것은 사용자의 책임입니다.

**관련 개념**

25 페이지의 『 UNIX, Linux및 Windows 용 FIPS (Federal Information Processing Standards)』

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

클라이언트 구성 파일의 SSL 스탠자

**관련 참조**

FipsRequired(MQLONG)

MQSSLFIPS

## AIX

AIX의 SSL 또는 TLS 클라이언트 애플리케이션은 GSKit V8.0이 여러 개 설치된 AIX 시스템에서 실행될 때 MQRC_CHANNEL_CONFIG_ERROR 및 AMQ6175 오류가 발생할 수 있습니다.

다중 GSKit V8.0 설치와 함께 AIX 시스템에서 클라이언트 애플리케이션을 실행하면 클라이언트 연결 호출은 SSL 또는 TLS를 사용할 때 MQRC_CHANNEL_CONFIG_ERROR를 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 설정 때문에 IBM WebSphere MQ 클라이언트가 다른 두 GSKit V8.0 설치에서 혼합된 라이브러리 세트를 로드하기 때문입니다. DB2® 환경에서IBM WebSphere MQ 클라이언트 애플리케이션을 실행하면 이 오류가 발생할 수 있습니다.

이 오류를 피하려면 IBM WebSphere MQ 라이브러리가 우선할 수 있도록 라이브러리 경로 앞에 IBM WebSphere MQ 라이브러리 디렉토리를 포함하십시오. 이는 **-k** 매개변수와 함께 **setmqenv** 명령을 사용하여 달성할 수 있습니다. 예를 들면, 다음과 같습니다.

```
. /usr/mqm/bin/setmqenv -s -k
```

**setmqenv** 명령 사용에 대한 자세한 정보는 setmqenv(WebSphere MQ 환경 설정)를 참조하십시오.

## UNIX, Linux, and Windows 시스템에서 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. UNIX, Linux 및 Windows 시스템에서 자체 서명 인증서로 테스트를 수행할 수 있습니다.

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 104 페이지의 『UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 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 WebSphere MQ 구현을 사용하는 경우 SSL 또는 TLS 서버는 항상 클라이언트로부터 인증서를 요청합니다.

UNIX, Linux 및 Windows 시스템의 경우 SSL 또는 TLS 클라이언트는 올바른 IBM WebSphere MQ 형식으로 레이블 지정된 인증서가 있는 경우에만 인증서를 전송합니다.

- 큐 관리자의 경우 형식은 소문자로 변경된 큐 관리자 이름이 뒤따라오는 ibmwebspheremq입니다. 예를 들어, QM1의 경우 ibmwebspheremqqm1입니다.
- IBM WebSphere MQ 클라이언트의 경우 소문자로 변경된 로그온 사용자 ID가 뒤따라오는 ibmwebspheremq입니다(예: ibmwebspheremqmyuserid).

IBM WebSphere 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 188 페이지의 『SSL 또는 TLS를 사용하여 두 개의 큐 관리자 연결』의 내용을 참조하십시오.

## SSL 또는 TLS에 대한 작업

이러한 주제는 IBM WebSphere MQ을 사용하여 SSL 또는 TLS 사용과 관련된 단일 태스크를 수행하기 위한 지시사항을 제공합니다.

이들 대부분은 다음 절에 설명된 고급 수준 태스크에서 단계로서 사용됩니다.

- 132 페이지의 『사용자 식별 및 인증』
- 145 페이지의 『오브젝트에 액세스 권한 부여』
- 187 페이지의 『메시지의 기밀성』
- 207 페이지의 『메시지의 데이터 무결성』
- 224 페이지의 『클러스터 보안 유지』

## HP Integrity NonStop Server

보안 서비스, 구성요소, 지원되는 프로토콜 버전, 지원되는 CipherSpecs및 지원되지 않는 보안 기능을 포함하여 HP Integrity NonStop Server OpenSSL 보안 구현을 위한 IBM WebSphere MQ 클라이언트를 설명합니다.

IBM WebSphere MQ SSL (SSL) /TLS 지원은 클라이언트 채널에 다음과 같은 보안 서비스를 제공합니다.

- 서버 인증과 선택적인 클라이언트 인증
- 채널을 통해 플로우되는 데이터의 암호화 및 복호화
- 채널을 통해 플로우되는 데이터에 대한 무결성 검사

HP Integrity NonStop Server 에 대해 IBM WebSphere MQ 클라이언트와 함께 제공되는 SSL및 TLS 지원은 다음 컴포넌트를 포함합니다.

- OpenSSL 라이브러리 및 **openssl** 명령
- IBM WebSphere MQ 비밀번호 보관 명령 **amqrsslc**

SSL 또는 TLS 클라이언트 채널 조작에 필요한 다음 구성요소는 HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트와 함께 제공되지 않습니다.

- OpenSSL 암호화를 위한 무작위 데이터의 소스를 제공하는 엔트로피 디먼

### 지원되는 프로토콜 버전

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 다음 프로토콜 버전을 지원합니다.

- SSL 3.0
- TLS 1.0
- TLS 1.2

### 지원되는 CipherSpec

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 다음 CipherSpecs 버전을 지원합니다.

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- RC4_SHA_US
- RC4_MD5_US
- TRIPLE_DES_SHA_US
- TLS_RSA_WITH_3DES_EDE_CBC_SHA(더 이상 사용되지 않음)
- DES_SHA_EXPORT1024
- RC4_56_SHA_EXPORT1024
- RC4_MD5_EXPORT
- RC2_MD5_EXPORT
- DES_SHA_EXPORT
- TLS_RSA_WITH_DES_CBC_SHA
- NULL_SHA
- NULL_MD5

- FIPS_WITH_DES_CBC_SHA
- FIPS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_NULL_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

## 지원되지 않는 보안 기능

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 현재 다음을 지원하지 않습니다.

- PKCS#11 암호화 하드웨어 지원
- LDAP 인증서 폐기 목록 검사
- OCSP 온라인 인증서 상태 프로토콜 검사
- FIPS 140-2, NSA 스위트 B cipher suite 제어

## 인증서 관리

파일 세트를 사용하여 디지털 인증서 및 인증서 폐기 정보를 저장하십시오.

IBM WebSphere MQ SSL 및 TLS 지원은 디지털 인증서 및 인증서 폐기 정보를 저장하기 위해 파일 세트를 사용합니다. 이러한 파일은 MQCONNX 호출에 전달되는 MQSCO 구조의 KeyRepository 필드, *MQSSLKEYR* 환경 변수, 또는 SSLKeyRepository 속성을 사용한 `mqclient.ini`의 SSL 스탠자를 통해 프로그래밍 방식으로 지정된 디렉토리에 있습니다.

MQSCO 구조는 ini 파일 스탠자 값에 대해 우선순위를 갖는 MQSSLKEYR 환경 변수에 대해 우선순위를 갖습니다.

**중요사항:** 키 저장소 위치는 디렉토리 위치를 지정하며 HP Integrity NonStop Server 플랫폼에 있는 파일 이름을 지정하지 않습니다.

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 키 저장소 위치에서 다음과 같은 대소문자를 구분하여 이름이 지정된 파일을 사용합니다.


- 103 페이지의 『개인 인증서 저장소』
- 104 페이지의 『인증서 신뢰 저장소』
- 104 페이지의 『암호 문구 보관 파일』
- 104 페이지의 『인증서 폐기 목록 파일』


개인 인증서 저장소
개인 인증서 저장소 파일은 `cert.pem`입니다.

이 파일은 사용할 클라이언트를 위한 개인 인증서와 암호화된 개인 키를 PEM 형식으로 포함하고 있습니다. 클라이언트 인증을 필요로 하지 않는 SSL 또는 TLS 채널을 사용하고 있는 경우 이 파일의 존재는 선택적입니다. 채널에서 클라이언트 인증을 필요로 하며 채널 정의에 SSLCAUTH(REQUIRED)가 지정된 위치에서는 이 파일이 반드시 있어야 하며 이 파일은 인증서와 암호화된 개인 키를 포함하고 있어야 합니다.

인증서 저장소의 소유자에게 읽기 액세스를 허용하기 위해 이 파일에는 파일 권한이 설정되어야 합니다.

올바르게 형식화된 `cert.pem` 파일은 다음과 같은 헤더와 푸터가 있는 두 섹션을 정확히 포함해야 합니다.

```
-----BEGIN PRIVATE KEY-----
Base 64 ASCII encoded private key data here
-----END PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
Base 64 ASCII encoded certificate data here
-----END CERTIFICATE-----
```

암호화된 개인 키에 대한 암호 문구는 암호 문구 보관 파일인 `Stash.sth`에 저장되어 있습니다.

인증서 신뢰 저장소
인증서 신뢰 저장소 파일은 `trust.pem`입니다.

이 파일은 클라이언트가 연결하는 큐 관리자에서 사용하는 개인 인증서를 유효성 검증하는 데 필요한 인증서를 PEM 형식으로 포함하고 있습니다. 인증서 신뢰 저장소는 모든 SSL 또는 TLS 클라이언트 채널에서 필수입니다.

이 파일에 대한 쓰기 액세스를 제한하기 위해 파일 권한을 설정해야 합니다.

올바르게 형식화된 `trust.pem` 파일은 다음과 같은 헤더와 푸터가 있는 하나 이상의 섹션을 포함해야 합니다.

```
-----BEGIN CERTIFICATE-----
Base 64 ASCII encoded certificate data here
-----END CERTIFICATE-----
```

암호 문구 보관 파일
암호 문구 보관 파일은 `Stash.sth`입니다.

이 파일은 IBM WebSphere MQ 전용 2진 형식이며 `cert.pem` 파일에 보유하고 있는 개인 키에 액세스할 때 사용할 암호화된 암호 문구를 포함하고 있습니다. 개인 키 자체는 `cert.pem` 인증서 저장소에 저장됩니다.

이 파일은 IBM WebSphere MQ **amqrsslc** 명령행 도구를 **-s** 매개변수와 함께 사용하여 작성되거나 변경됩니다. 예를 들어, 디렉토리 `/home/alice`가 `cert.pem` 파일을 포함하고 있는 경우 이 도구는 다음과 같이 사용됩니다.

```
amqrsslc -s /home/alice/cert

  Enter password for Keystore /home/alice/cert.pem :
    password

  Stashed the password in file /home/alice/Stash.sth
```

연관된 인증서 저장소의 소유자에게 읽기 액세스를 허용하기 위해 이 파일에는 파일 권한이 설정되어야 합니다.

인증서 폐기 목록 파일
인증서 폐기 목록 파일은 `crl.pem`입니다.

이 파일은 클라이언트가 디지털 인증서를 유효성 검증하는 데 사용하는 인증서 폐기 목록(CRL)을 PEM 형식으로 포함하고 있습니다. 이 파일의 존재는 선택적입니다. 이 파일이 존재하지 않는 경우에는 인증서를 유효성 검증할 때 인증서 폐기 검사가 수행되지 않습니다.

이 파일에 대한 쓰기 액세스를 제한하기 위해 파일 권한을 설정해야 합니다.

올바르게 형식화된 `crl.pem` 파일은 다음과 같은 헤더와 푸터가 있는 하나 이상의 섹션을 포함해야 합니다.

```
-----BEGIN X509 CRL-----
Base 64 ASCII encoded CRL data here
-----END X509 CRL-----
```

## UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 작업

UNIX, Linux 및 Windows 시스템에서 SSL (Secure Sockets Layer) 지원은 IBM WebSphere MQ와 함께 설치됩니다.

인증서 유효성 검증 정책에 대한 보다 자세한 정보는 인증서 유효성 검증 및 신뢰 정책 설계의 내용을 참조하십시오.

### *iKeyman, iKeycmd, runmqakm* 및 *runmqckm* 사용

UNIX, Linux 및 Windows 시스템에서는 iKeyman GUI를 사용하거나 명령행에서 iKeycmd 또는 runmqakm을 사용하여 키 및 디지털 인증서를 관리합니다.

- **UNIX and Linux** 시스템의 경우:
  - **strmqikm** 명령을 사용하여 iKeyman GUI를 시작하십시오.
  - **runmqckm** 명령을 사용하여 iKeycmd 명령행 인터페이스를 사용하여 태스크를 수행하십시오.
  - runmqakm 명령행 인터페이스로 태스크를 수행하려면 **runmqakm** 명령을 사용하십시오. **runmqakm**의 명령 구문은 **runmqckm**의 구문과 같습니다.

    FIPS를 준수하는 방법으로 SSL 인증서를 관리해야 하는 경우에는 **runmqckm** 또는 **strmqikm** 명령 대신에 **runmqakm** 명령을 사용하십시오.

**runmqckm** 및 **runmqakm** 명령에 대한 명령행 인터페이스의 전체 설명은 키 및 인터페이스 관리를 참조하십시오.

PKCS #11 암호화 하드웨어에 저장된 인증서 또는 키를 사용 중인 경우 iKeycmd 및 iKeyman은 64비트 프로그램입니다. PKCS #11 지원에 필요한 외부 모듈이 64비트 프로세스에 로드되므로 암호화 하드웨어 관리를 위해서는 64비트 PKCS #11 라이브러리를 설치해야 합니다. Windows및 Linux x86 32비트플랫폼은 해당 플랫폼에서 iKeyman 및 iKeycmd 프로그램이 32비트이므로 유일한 예외입니다.

제품의 이전 버전에서 JRE가 32비트이지만 IBM WebSphere MQ Version 7.5에서만 64비트인 다음 플랫폼에서는 **iKeyman** 및 **iKeycmd** JRE의 주소 지정 모드에 적합한 추가 PKCS#11 드라이버를 설치해야 합니다. 이는 PKCS#11 드라이버가 동일한 주소 지정 모드를 JRE로 사용해야 하기 때문입니다. 다음 테이블에서는 IBM WebSphere MQ Version 7.5 JRE 주소 지정 모드를 표시합니다.

표 *12. IBM WebSphere MQ Version 7.5 JRE* 주소 지정 모드

| **플랫폼** | **JRE 주소 지정 모드** |
|---|---|
| Windows(32비트 또는 64비트) | 32 |
| Linux for System x(32비트) | 32 |
| System x용 Linux 64비트 | 64 |
| Linux (System p의 경우) | 64 |
| System z의 경우 Linux | 64 |
| HP-UX | 64 |
| Solaris Sparc | 64 |
| Solaris x86-64 | 64 |
| AIX | 64 |

**strmqikm** 명령을 실행하여 iKeyman GUI를 시작하기 전에 X Window 시스템을 실행할 수 있는 시스템에서 작업 중인지와 다음을 수행하는지를 확인하십시오.

- DISPLAY 환경 변수를 다음과 같이 설정하십시오.

```
export DISPLAY=mypc:0
```

- PATH 환경 변수에 **/usr/bin** 및 **/bin**이 포함되어 있는지 확인하십시오. 이는 **runmqckm** 및 **runmqakm** 명령에도 필요합니다. 예를 들면, 다음과 같습니다.

```
export PATH=$PATH:/usr/bin:/bin
```

- **Windows** 시스템의 경우:
  - **strmqikm** 명령을 사용하여 iKeyman GUI를 시작하십시오.
  - **runmqckm** 명령을 사용하여 iKeycmd 명령행 인터페이스를 사용하여 태스크를 수행하십시오.

    FIPS를 준수하는 방법으로 SSL 인증서를 관리해야 하는 경우에는 **runmqckm** 또는 **strmqikm** 명령 대신에 **runmqakm** 명령을 사용하십시오.

UNIX, Linux 또는 시스템 SSL 추적을 요청하려면 strmqtrc를 참조하십시오.

**관련 참조**

runmqckm 및 runmqakm 명령

## *UNIX, Linux, and Windows* 시스템에서 키 저장소 설정

iKeyman 사용자 인터페이스 또는 **iKeycmd** 또는 **runmqakm** 명령을 사용하여 키 저장소를 설정할 수 있습니다.

### 이 태스크 정보

SSL 또는 TLS 연결에는 연결 양끝에 키 저장소가 있어야 합니다. 각 IBM WebSphere MQ 큐 관리자 및 IBM WebSphere MQ MQI client에는 키 저장소에 대한 액세스 권한이 있어야 합니다. 자세한 정보는 22 페이지의 『SSL 또는 TLS 키 저장소』의 내용을 참조하십시오.

UNIX, Linux, and Windows 시스템에서 디지털 인증서는 **iKeyman** 사용자 인터페이스를 사용하거나 **iKeycmd** 또는 **runmqakm** 명령을 사용하여 관리되는 키 데이터베이스 파일에 저장됩니다. 이러한 디지털 인증서에는 레이블이 있습니다. 특정 레이블은 개인 인증서를 큐 관리자 또는 IBM WebSphere MQ MQI client와 연관시킵니다. SSL 및 TLS는 인증 목적을 위해 해당 인증서를 사용합니다. UNIX, Linux, and Windows 시스템에서, IBM WebSphere MQ는 `ibmwebspheremq`를 레이블 접두부로 사용하여 다른 제품의 인증서와 혼동되지 않도록 합니다. 접두부 뒤에는 소문자로 변경된 큐 관리자의 이름 또는 IBM WebSphere MQ MQI client 사용자 로그온 ID가 옵니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

키 데이터베이스 파일 이름은 경로와 스템 이름으로 구성됩니다.

- UNIX and Linux 시스템에서 큐 관리자의 기본 경로(큐 관리자를 작성할 때 설정됨)는 `/var/mqm/qmgrs/<queue_manager_name>/ssl`입니다.

  Windows 시스템에서 기본 경로는 *MQ_INSTALLATION_PATH*`\Qmgrs\`*queue_manager_name*`\ssl`입니다. 여기서 *MQ_INSTALLATION_PATH* 는 IBM WebSphere MQ 가 설치된 디렉토리입니다. (예: `C:\program files\IBM\WebSphere MQ\Qmgrs\QM1\ssl`).

  기본 스템 이름은 key입니다. 또는 사용자 고유의 경로 및 스템 이름을 선택할 수도 있지만 확장자는 `.kdb`이어야 합니다.

  사용자 고유의 경로나 파일 이름을 선택하는 경우에는 이에 대한 액세스를 엄격하게 제어하기 위해 파일에 권한을 설정하십시오.
- WebSphere MQ 클라이언트에서는 기본 경로나 스템 이름이 없습니다. 이 파일에 대한 액세스를 엄격하게 제어하십시오. 확장자는 `.kdb`이어야 합니다.

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 110 페이지의 『UNIX, Linux 또는 Windows 시스템에서 큐 관리자에 대한 키 저장소 위치 변경』의 내용을 참조하십시오. 키 데이터베이스 파일을 작성하기 전후에 키 데이터베이스 파일 이름을 지정할 수 있습니다.

**iKeyman** 또는 **iKeycmd** 명령을 실행하는 사용자 ID에는 키 데이터베이스 파일을 작성하거나 업데이트한 디렉토리에 대한 쓰기 권한이 있어야 합니다. 기본 `ssl` 디렉토리를 사용하는 큐 관리자의 경우 **iKeyman** 또는 **iKeycmd**를 실행하는 사용자 ID는 mqm 그룹의 구성원이어야 합니다. IBM WebSphere MQ MQI client의 경우, 클라이언트를 실행하는 데 사용한 것과 다른 사용자 ID로 **iKeyman** 또는 **iKeycmd**를 실행하는 경우, IBM WebSphere MQ MQI client가 런타임 시 키 데이터베이스 파일에 액세스할 수 있도록 파일 권한을 변경해야 합니다. 자세한 정보는 108 페이지의 『Windows에서 키 데이터베이스 파일에 액세스 및 보안 설정』 또는 108 페이지의 『UNIX and Linux 시스템에서 키 데이터베이스 파일 액세스 및 보안』의 내용을 참조하십시오.

**iKeyman** 또는 **iKeycmd** 버전 7.0에서, 새로운 키 데이터베이스가 자동으로 미리 정의된 인증 기관(CA) 인증서 세트로 채워집니다. **iKeyman** 또는 **iKeycmd** 버전 8.0에서, 키 데이터베이스가 자동으로 채워지지 않으며 키 데이터베이스 파일에 원하는 CA 인증서만을 포함시키므로 초기 설정이 보다 안전합니다.

**참고:** GSKit 버전 8.0에 대한 작동 변경으로 인해 CA 인증서가 더 이상 자동으로 저장소에 추가되지 않습니다. 사용자가 수동으로 선호하는 CA 인증서를 추가해야 합니다. 이 동작의 변화는 사용된 CA 인증서에 대한 보다 세부 단위의 제어를 제공합니다. 109 페이지의 『GSKit 버전 8.0을 사용하는 UNIX, Linux, and Windows 시스템의 비어 있는 키 저장소에 기본 CA 인증서 추가』의 내용을 참조하십시오.

명령행을 사용하거나 **strmqikm** (iKeyman) 사용자 인터페이스를 사용하여 키 데이터베이스를 작성합니다.

**참고:** FIPS를 준수하는 방법으로 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오. **strmqikm** 사용자 인터페이스는 FIPS 준수 옵션을 제공하지 않습니다.

## 프로시저

명령행을 사용하여 키 데이터베이스를 작성한다.

1. 다음 명령 중 하나를 실행하십시오.

   - UNIX, Linux, and Windows 시스템의 경우:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - runmqakm 사용:

     ```
     runmqakm -keydb -create -db filename -pw password -type cms
         -stash -fips -strong
     ```

   설명:

   **-db *filename***
   CMS 키 데이터베이스의 완전한 파일 이름을 지정하고 `.kdb`의 파일 확장자가 있어야 합니다.

   **-pw *password***
   CMS 키 데이터베이스의 비밀번호를 지정합니다.

   **-type *cms***
   데이터베이스의 유형을 지정합니다. (IBM WebSphere MQ의 경우, `cms`여야 합니다.)

   **-stash**
   키 데이터베이스 비밀번호를 파일에 저장합니다.

   **-fips**
   BSafe 암호화 라이브러리의 사용을 불가능하게 만듭니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서는 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **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 내에 있는 표준 ASCII 인쇄 가능 문자 세트로 되어 있습니다.

   또는 **strmqikm** (iKeyman) 사용자 인터페이스를 사용하여 키 데이터베이스를 작성하십시오.

2. UNIX and Linux 시스템에서 루트 사용자로서 로그인하십시오. Windows 시스템에서 관리자 또는 MQM 그룹의 구성원으로서 로그인하십시오.

3. **strmqikm** 명령을 실행하여 iKeyman 사용자 인터페이스를 시작하십시오.
4. **키 데이터베이스 파일** 메뉴에서 **새로 작성**을 클릭하십시오.
   새 창이 열립니다.
5. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
6. **파일 이름** 필드에 파일 이름을 입력하십시오.
   이 필드에는 이미 key.kdb 텍스트가 있습니다. 스템 이름이 key이면 이 필드를 변경하지 말고 두십시오. 다른 스템 이름을 지정한 경우에는 key를 사용자의 스템 이름으로 바꾸십시오. 그러나 .kdb 확장자는 변경하지 않아야 합니다.
7. **위치** 필드에 경로를 입력하십시오.
   예를 들면, 다음과 같습니다.
   - 큐 관리자의 경우: /var/mqm/qmgrs/QM1/ssl ( UNIX and Linux 시스템의 경우) 또는 C:\Program Files\IBM\WebSphere MQ\qmgrs\QM1\ssl ( Windows 시스템의 경우).

     경로 이름은 큐 관리자의 **SSLKeyRepository** 속성의 값과 일치해야 합니다.
   - IBM WebSphere MQ 클라이언트의 경우: /var/mqm/ssl(UNIX and Linux 시스템에서) 또는 C:\mqm\ssl(Windows 시스템에서).
8. **열기**를 클릭하십시오.
   비밀번호 프롬프트 창이 열립니다.
9. **비밀번호** 필드에 비밀번호를 입력하고 **비밀번호 확인** 필드에 다시 입력하십시오.
10. **파일에 비밀번호 숨김** 선택란을 선택하십시오.

    **참고:** 비밀번호를 숨기지 않은 경우에는 키 데이터베이스 파일에 액세스하는 데 필요한 비밀번호를 구할 수 없으므로 SSL 또는 TLS 채널을 시작하려는 시도가 실패합니다.
11. **확인**을 클릭하십시오.
    개인 인증서 창이 열립니다.
12. 108 페이지의 『Windows에서 키 데이터베이스 파일에 액세스 및 보안 설정』 또는 108 페이지의 『UNIX and Linux 시스템에서 키 데이터베이스 파일 액세스 및 보안』에 설명된 대로 액세스 권한을 설정하십시오.

*Windows*에서 키 데이터베이스 파일에 액세스 및 보안 설정
키 데이터베이스 파일에는 적합한 액세스 권한이 없을 수도 있습니다. 이러한 파일에 대한 적합한 액세스를 설정해야 합니다.

파일 *key*.kdb, *key*.sth, *key*.crl 및 *key*.rdb에 대한 액세스 제어를 설정하십시오. 여기서 *key*는 제한된 사용자 세트에 권한을 부여하기 위한 키 데이터베이스의 스템 이름입니다.

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.

*UNIX 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 사용자로서 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 경

우에는 권한이 충분할 것입니다. 클라이언트 프로세스 사용자가 아니라 해당 그룹의 다른 사용자라면 그룹의 다른 사용자에게 읽기 권한을 부여해야 할 수도 있습니다.

*key*.kdb, *key*.sth, *key*.crl 및 *key*.rdb 파일에 권한을 설정하십시오. 여기서 *key*는 파일 소유자가 읽기와 쓰기를 위하고, mqm 또는 클라이언트 사용자 그룹(-rw-r-----)이 읽기를 위한 키 데이터베이스의 스템 이름입니다.

*GSKit* 버전 *8.0*을 사용하는 *UNIX, Linux, and Windows* 시스템의 비어 있는 키 저장소에 기본 *CA* 인증서 추가
GSKit 버전 8이 있는 빈 키 저장소에 기본 CA 인증서를 하나 이상 추가하려면 이 프로시저를 따르십시오.

GSKit 버전 7.0에서는 새로운 키 저장소를 작성할 때 수행되는 동작이 일반적으로 사용되는 인증 기관의 기본 CA 인증서 세트에 자동으로 추가되었습니다. GSKit 버전 8에서는 이 동작이 CA 인증서가 더 이상 저장소에 자동으로 추가되지 않도록 변경되었습니다. 사용자는 이제 키 저장소에 CA 인증서를 수동으로 추가해야 합니다.

### iKeyman 사용

CA 인증서를 추가하고 싶은 시스템에서 다음 단계를 수행하십시오.

1. **strmqikm** 명령 ( 유닉스, Linux 및 윈도우 시스템) 을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 인증서를 추가하려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **키 데이터베이스 컨텐츠** 필드에서 **서명자 인증서**를 선택하십시오.
9. **채우기**를 클릭하십시오. CA의 인증서 추가 창이 열립니다.
10. 저장소에 추가할 수 있는 CA 인증서가 계층 구조 트리 구조에 표시됩니다. 유효한 CA 인증서의 전체 목록을 보려면 신뢰하려는 CA 인증서가 포함된 조직의 맨 위 레벨 항목을 선택하십시오.
11. 목록에서 신뢰하려는 CA 인증서를 선택하고 **확인**을 클릭하십시오. 인증서가 키 저장소에 추가됩니다.

### 명령행 사용

다음 명령을 사용하여 나열한 다음 iKeycmd를 사용하여 CA 인증서를 추가하십시오.

- 다음 명령을 실행하여 기본 CA 인증서를 이를 발행한 조직과 함께 나열하십시오.

```
runmqckm -cert -listsigners
```

- 레이블 필드에 지정된 조직의 모든 CA 인증서를 추가하려면 다음 명령을 실행하십시오.

```
runmqckm -cert -populate -db filename -pw password -label label
```

설명:

| | |
|---|---|
| -db *filename* | 키 데이터베이스의 완전한 경로 이름입니다. |
| -pw *password* | 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 인증서에 첨부된 레이블입니다. |

**참고:** CA 인증서를 키 저장소에 추가하면 WebSphere MQ가 해당 CA 인증서에 서명한 모든 개인 인증서를 신뢰할 수 있게 됩니다. 어떤 인증 기관을 신뢰할지를 신중히 고려하고 클라이언트와 관리자를 인증하는 데 필요한 CA 인증서 세트만을 추가하십시오. 보안 정책의 결정적인 요구사항이 아닌 한 기본 CA 인증서의 전체 세트를 추가하는 것은 바람직하지 않습니다.

## *UNIX, Linux, and Windows* 시스템에서 큐 관리자에 대한 키 저장소 위치 지정

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 WebSphere MQ 탐색기 또는 PCF 명령을 사용하여 큐 관리자의 속성을 표시할 수도 있습니다.
2. 키 데이터베이스 파일의 경로 및 스템 이름의 명령 출력을 검사하십시오.
   예를 들면 다음과 같습니다.

   a. UNIX and Linux 시스템에서: `/var/mqm/qmgrs/QM1/ssl/key`. 여기서 `/var/mqm/qmgrs/QM1/ssl`은 경로이고 `key`는 스템 이름입니다.

   b. 의: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`, 여기서 *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` 는 경로이고 `key` 은 스템 이름입니다. *MQ_INSTALLATION_PATH* WebSphere MQ 가 설치된 상위 레벨 디렉토리를 나타냅니다.

## *UNIX, Linux* 또는 *Windows* 시스템에서 큐 관리자에 대한 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, UNIX and Linux 시스템의 경우는 다음과 같습니다.

```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey')
```

키 데이터베이스 파일에는 완전한 파일 이름(`/var/mqm/qmgrs/QM1/ssl/MyKey.kdb`)이 있습니다.

Windows의 경우:

```
ALTER QMGR SSLKEYR('C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\Mykey')
```

키 데이터베이스 파일에는 완전한 파일 이름(`C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\Mykey.kdb`)이 있습니다.

**주의:** SSLKEYR 키워드에서 파일 이름에 `.kdb` 확장자를 포함하지 않도록 하십시오. 큐 관리자가 이 확장자를 자동으로 추가하기 때문입니다.

WebSphere 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 122 페이지의 『UNIX, Linux, and Windows 시스템에서 개인 인증서를 키 저장소에 가져오기』에 설명된 대로 필요한 CA및 개인용 인증서로 채워야 합니다.

## *UNIX, Linux, and Windows* 시스템에서 *IBM WebSphere MQ MQI* 클라이언트에 대한 키 저장소 찾기

키 저장소의 위치는 MQSSLKEYR 변수에 의해 제공되거나 MQCONNX 호출에 지정됩니다.

MQSSLKEYR 환경 변수를 검사하여 IBM WebSphere MQ MQI 클라이언트의 키 데이터베이스 파일 위치를 확인하십시오. 예를 들면, 다음과 같습니다.

```
echo $MQSSLKEYR
```

또한 키 데이터베이스 파일 이름이 111 페이지의 『UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ MQI 클라이언트의 키 저장소 위치 지정』에서 설명하는 대로 MQCONNX 호출에서 설정될 수도 있으므로 애플리케이션을 점검하십시오. MQCONNX 호출에 설정한 값은 MQSSLKEYR의 값을 대체합니다.

## *UNIX, Linux, and Windows* 시스템에서 *IBM WebSphere MQ MQI* 클라이언트의 키 저장소 위치 지정

IBM WebSphere MQ MQI 클라이언트에 대한 기본 키 저장소는 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.

다음 두 가지 방법 중 하나로 IBM WebSphere MQ MQI 클라이언트의 키 데이터베이스 파일 위치를 지정할 수 있습니다.

- MQSSLKEYR 환경 변수 설정. 예를 들어, UNIX and Linux 시스템의 경우는 다음과 같습니다.

  ```
  export MQSSLKEYR=/var/mqm/ssl/key
  ```

  키 데이터베이스 파일에는 완전한 파일 이름이 있습니다.

  ```
  /var/mqm/ssl/key.kdb
  ```

  Windows의 경우:

  ```
  set MQSSLKEYR=C:\Program Files\IBM\WebSphere MQ\ssl\key
  ```

  키 데이터베이스 파일에는 완전한 파일 이름이 있습니다.

  ```
  C:\Program Files\IBM\WebSphere MQ\ssl\key.kdb
  ```

  **참고:** .kdb 확장자는 파일 이름의 필수 부분이지만 환경 변수의 값의 일부로는 포함되지 않습니다.
- 애플리케이션이 MQCONNX 호출을 할 때 MQSCO 구조의 *KeyRepository* 필드에 키 데이터베이스 파일의 경로 및 스템 이름 제공. MQCONNX에서 MQSCO 구조 사용에 대한 자세한 정보는 MQSCO 개요를 참조하십시오.

## 인증서 또는 인증서 저장소의 변경사항이 *UNIX, Linux* 또는 *Windows* 시스템에서 적용되는 시점

인증서 저장소에서 인증서 또는 인증서 저장소의 위치를 변경할 때 변경사항은 채널 유형 및 채널이 실행 중인 방법에 따라 적용됩니다.

키 데이터베이스 파일의 인증서 및 키 저장소 속성의 변경사항은 다음 경우에 적용됩니다.

- 새 아웃바운드 단일 프로세스가 처음으로 SSL 채널을 실행할 때.
- 새 인바운드 TCP/IP 단일 채널 프로세스가 처음으로 SSL 채널을 시작하라는 요청을 수신할 때.
- MQSC 명령 REFRESH SECURITY TYPE(SSL)이 Websphere MQ SSL 환경을 새로 고치기 위해 발행될 때.
- 클라이언트 애플리케이션 프로세스의 경우 프로세스의 마지막 SSL 연결이 처리완료될 때. 다음 SSL 연결에는 인증서 변경사항이 적용됩니다.
- 프로세스 풀링 프로세스(amqrmppa)의 스레드로서 실행되는 채널의 경우 프로세스 풀링 프로세스가 시작되거나 재시작되고 SSL 채널을 처음 시작할 때. 프로세스 풀링 프로세스가 이미 SSL 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 SSL 채널을 처음 실행할 때. 채널 시작기 프로세스가 이미 SSL 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 SSL 채널을 시작하라는 요청을 처음으로 받을 때. 리스너가 이미 SSL 채널을 실행했고 변경사항을 즉시 적용하고 싶으면 MQSC 명령 REFRESH SECURITY TYPE(SSL)을 실행하십시오.

IBM WebSphere MQ Explorer 또는 PCF 명령을 사용하여 WebSphere MQ SSL 환경을 새로 고칠 수도 있습니다.

## *UNIX, Linux, and Windows* 시스템에서 자체 서명한 개인 인증서 작성

iKeyman, iKeycmd 또는 runmqakm을 사용하여 자체 서명된 인증서를 작성할 수 있습니다.

**참고:** IBM WebSphere 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.

디지털 서명 알고리즘 이름 SHA3WithRSA 및 SHA5WithRSA는 각각 SHA384WithRSA 및 SHA512WithRSA의 축약된 양식이므로 더 이상 사용되지 않습니다.

자체 서명된 인증서 사용 이유에 대한 자세한 정보는 188 페이지의 『두 개의 큐 관리자의 상호 인증을 위해 자체 서명 인증서 사용』을 참조하십시오.

모든 디지털 인증서를 모든 CipherSpec과 함께 사용할 수 있는 것은 아닙니다. 사용해야 하는 CipherSpec과 호환되는 인증서를 작성하는지 확인하십시오. WebSphere MQ는 세 가지 유형의 서로 다른 CipherSpec을 지원합니다. 자세한 내용은 32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 호환성』 주제에서 33 페이지의 『Elliptic Curve 및 RSA CipherSpec의 상호 운용성』의 내용을 참조하십시오. 유형 1 CipherSpec(ECDHE_ECDSA_로 시작하는 이름 사용)을 사용하려면 **runmqakm** 명령을 사용하여 인증서를 작성하고 Elliptic Curve ECDSA 서명 알고리즘 매개변수(예: **-sig_alg** EC_ecdsa_with_SHA384)를 지정해야 합니다.

## iKeyman 사용

iKeyman은 FIPS 준수 옵션을 제공하지 않습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

큐 관리자나 WebSphere MQ MQI 클라이언트를 위해 자체 서명된 인증서를 확보하려면 다음 프로시저를 사용하십시오.

1. **strmqikm** 명령을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 표시됩니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 예를 들어, key.kdb와 같이, 인증서를 저장하려는 키 데이터베이스 파일을 선택하십시오.
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 표시됩니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **작성** 메뉴에서 **새 자체 서명 인증서**를 클릭하십시오. 새 자체 서명 인증서 작성 창이 표시됩니다.
9. **키 레이블** 필드에서 다음을 입력하십시오.
   - 큐 관리자에서는 ibmwebspheremq 뒤에 큐 관리자의 이름이 소문자로 변경됩니다. 예를 들면, QM1의 경우, ibmwebspheremqqm1입니다.
   - WebSphere MQ 클라이언트에서는 ibmwebspheremq 뒤의 로그온 사용자 ID가 소문자로 되어 있습니다(예: ibmwebspheremqmyuserid).
10. **Distinguished name**또는 **Subject alternative name** 필드에 있는 필드의 값을 입력하거나 선택하십시오.
11. 나머지 필드의 경우 기본값을 승인하거나 새 값을 입력하거나 선택하십시오. 식별 이름에 대한 자세한 정보는 10 페이지의 『식별 이름』의 내용을 참조하십시오.
12. **확인**을 클릭하십시오. **개인 인증서** 목록은 사용자가 작성한 자체 서명 개인 인증서의 레이블을 보여줍니다.

## 명령행 사용

iKeycmd 또는 runmqakm을 사용하여 자체 서명된 개인 인증서를 작성하려면 다음 명령을 사용하십시오.

- Linux 및 시스템 iKeycmd 를 사용하는 경우:

```
runmqckm -cert -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
```

```
 -x509version version -expire days
 -sig_alg algorithm
```

-dn *distinguished_name* 대신 -san_dsname *DNS_names* , -san_emailaddr *email_addresses* 또는 -san_ipaddr *IP_addresses* 을(를) 사용할 수 있습니다.

- runmqakm 사용:

```
runmqakm -cert -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -x509version version -expire days

        -fips -sig_alg algorithm
```

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 파일 이름입니다. |
| -pw *password* | CMS 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 인증서에 첨부된 키 레이블입니다. |
| -dn *distinguished_name* | X.500 식별 이름은 큰따옴표 안에 있습니다. 하나 이상의 속성이 필요합니다. 여러 OU 또는 DC 속성을 제공할 수 있습니다. |
| -size *key_size* | 키 크기입니다. iKeycmd의 경우 값은 512 또는 1024가 될 수 있습니다. runmqakm의 경우 값은 512, 1024, 2048 또는 4096일 수 있습니다. |
| -x509version *version* | 작성할 X.509 인증서의 버전입니다. 값은 1, 2 또는 3일 수 있습니다. 기본은 3입니다. |
| -expire *days* | 인증서의 만기 시간(일)입니다. 기본값은 인증서의 경우 365일입니다. |
| -fips | 명령이 FIPS 모드에서 실행되도록 지정합니다. 이 모드는 BSafe 암호화 라이브러리의 사용을 불가능하게 만듭니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **runmqakm** 명령이 실패합니다. |
| -sig_alg | runmqakm의 경우 자체 서명 인증서 작성 중에 해시 알고리즘이 사용됩니다. 이 해시 알고리즘은 새로 작성된 자체 서명 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_SHA512가 될 수 있습니다. 기본값은 SHA1WithRSA입니다. |
| -sig_alg | iKeycmd의 경우 항목의 키 쌍 작성에 사용된 비대칭 서명 알고리즘. 값은 MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithRSA, SHA256_WITH_RSA, SHA256WithRSA, SHA2WithRSA, SHA384_WITH_RSA, SHA384WithRSA, SHA512_WITH_RSA, SHA512WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA 또는 SHAWithRSA가 될 수 있습니다. 기본값은 SHA1WithRSA입니다. |
| -san_dnsname *DNS_names* | 작성 중인 항목의 DNS 이름의 쉼표 또는 공백으로 구분된 목록. |

| | |
|---|---|
| -san_emailaddr *email_addresses* | 작성 중인 항목의 이메일 주소의 쉼표 또는 공백으로 구분된 목록. |
| -san_ipaddr *IP_addresses* | 작성 중인 항목의 IP 주소의 쉼표 또는 공백으로 구분된 목록. |

## distributed *UNIX, Linux, and Windows* 시스템에서 개인 인증서 요청

**strmqikm** (iKeyman)을 사용하거나 **runmqckm** 또는 **runmqakm** 명령을 사용하여 개인 인증서를 요청할 수 있습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

### 이 태스크 정보

iKeyman GUI를 사용하거나 명령행에서 개인 인증서를 요청할 수 있으며 다음 고려사항이 적용됩니다.

- WebSphere 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.
- 디지털 서명 알고리즘 이름 SHA3WithRSA 및 SHA5WithRSA는 각각 SHA384WithRSA 및 SHA512WithRSA의 축약된 양식이므로 더 이상 사용되지 않습니다.
- 모든 디지털 인증서를 모든 CipherSpec과 함께 사용할 수 있는 것은 아닙니다. 사용해야 하는 CipherSpec과 호환 가능한 인증서를 요청하는지 확인하십시오. WebSphere MQ는 세 가지 유형의 CipherSpec을 지원합니다. 자세한 내용은 32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 호환성』 주제에서 33 페이지의 『Elliptic Curve 및 RSA CipherSpec의 상호 운용성』의 내용을 참조하십시오.
- 유형 1 CipherSpecs ( ECDHE_ECDSA_로 시작하는 이름 포함) 를 사용하려면 **runmqakm** 명령을 사용하여 인증서를 요청하고, Elliptic Curve ECDSA 서명 알고리즘 매개변수 (예: **-sig_alg** EC_ecdsa_with_SHA384) 를 지정해야 합니다.
- runmqakm 명령만 FIPS 준수 옵션을 제공합니다.
- 암호화 하드웨어를 사용 중인 경우에는 129 페이지의 『PKCS #11 하드웨어의 개인 인증서 요청』의 내용을 참조하십시오.

*iKeyman* 사용자 인터페이스 사용

### 이 태스크 정보

iKeyman은 FIPS 준수 옵션을 제공하지 않습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

### 프로시저

iKeyman 사용자 인터페이스를 사용하여 개인 인증서를 신청하려면 다음 단계를 완료하십시오.

1. **strmqikm** 명령을 사용하여 iKeyman 사용자 인터페이스를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오.
   **열기** 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 요청을 생성하려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭하십시오.
   **비밀번호 프롬프트** 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오.
   키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **작성** 메뉴에서 **새 인증서 요청**을 클릭하십시오. **새 키 및 인증서 요청 작성** 창이 열립니다.
9. **키 레이블** 필드에서 다음 레이블을 입력하십시오.

- 큐 관리자에서는 ibmwebspheremq 뒤에 소문자로 변경한 큐 관리자의 이름을 입력하십시오. 예를 들어, QM1라는 큐 관리자의 경우 ibmwebspheremqqm1를 입력하십시오.
- IBM WebSphere MQ MQI client의 경우, ibmwebspheremq 를 입력한 다음 로그온 사용자 ID를 소문자로 입력하십시오 (예: ibmwebspheremqmyuserid ).

10. **식별 이름** 필드 또는 **제목 대체 이름** 필드에서 임의의 필드의 값을 입력하거나 선택하십시오. 나머지 필드의 경우 기본값을 승인하거나 새 값을 입력하거나 선택하십시오.

    식별 이름에 대한 자세한 정보는 10 페이지의 『식별 이름』의 내용을 참조하십시오.
11. **인증서 요청을 저장할 파일의 이름 입력** 필드에 기본 certreq.arm을 승인하거나 전체 경로가 있는 새 값을 입력하십시오.
12. **확인**을 클릭하십시오.

    확인 창이 표시됩니다.
13. **확인**을 클릭하십시오.

    **개인 인증서 요청** 목록에 사용자가 작성한 새 개인 인증서 요청의 레이블이 표시됩니다. 인증서 요청이 115 페이지의 『11』 단계에서 선택한 파일에 저장됩니다.
14. 파일을 인증 기관(CA)으로 보내거나 파일을 CA의 웹 사이트에서 요청 양식으로 복사하여 새 개인 인증서를 요청하십시오.

명령행 사용

## 프로시저

**runmqckm** 또는 **runmqakm** 명령을 사용하여 개인 인증서를 요청하려면 다음 명령을 사용하십시오.

- **runmqckm** 사용:

```
runmqckm -certreq -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -file filename -sig_alg algorithm
```

  -dn *distinguished_name* 대신 -san_dsname *DNS_names* , -san_emailaddr *email_addresses* 또는 -san_ipaddr *IP_addresses* 을(를) 사용할 수 있습니다.

- runmqakm 사용:

```
runmqakm -certreq -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -file filename -fips
        -sig_alg algorithm
```

설명:

**-db *filename***
CMS 키 데이터베이스의 완전한 파일 이름을 지정합니다.

**-pw *password***
CMS 키 데이터베이스의 비밀번호를 지정합니다.

**-label *label***
인증서에 첨부된 키 레이블을 지정합니다.

**-dn *distinguished_name***
큰따옴표로 묶인 X.500 식별 이름을 지정합니다. 하나 이상의 속성이 필요합니다. 여러 OU 및 DC 속성을 제공할 수 있습니다.

**-size *key_size***
키 크기를 지정합니다. **runmqckm**을 사용하는 경우, 값은 512 또는 1024일 수 있습니다. **runmqakm**을 사용하는 경우, 값은 512, 1024 또는 2048일 수 있습니다.

**-file *filename***
인증서 요청의 파일 이름을 지정합니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. 이 모드는 BSafe 암호화 라이브러리의 사용을 불가능하게 만듭니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서는 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **runmqakm** 명령이 실패합니다.

**-sig_alg**
**runmqckm**의 경우, 항목의 키 쌍 작성에 사용된 비대칭 서명 알고리즘을 지정합니다. 값은 MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithRSA, SHA256_WITH_RSA, SHA256WithRSA, SHA2WithRSA, SHA384_WITH_RSA, SHA384WithRSA, SHA512_WITH_RSA, SHA512WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA 또는 SHAWithRSA가 될 수 있습니다. 기본값은 SHA1WithRSA입니다.

**-sig_alg**
**runmqakm**의 경우 인증서 요청의 작성 동안에 사용된 해싱 알고리즘을 지정합니다. 이 해싱 알고리즘은 새로 작성된 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_SHA512가 될 수 있습니다. 기본값은 SHA1WithRSA입니다.

**-san_dnsname *DNS_names***
작성 중인 항목의 쉼표로 구분되거나 공백으로 구분된 DNS 이름 목록을 지정합니다.

**-san_emailaddr *email_addresses***
작성 중인 항목의 쉼표로 구분되거나 공백으로 구분된 이메일 주소 목록을 지정합니다.

**-san_ipaddr *IP_addresses***
작성 중인 항목의 쉼표로 구분되거나 공백으로 구분된 IP 주소 목록을 지정합니다.

### *UNIX, Linux, and Windows* 시스템에서 기존 개인 인증서 갱신
iKeyman 사용자 인터페이스를 사용하거나 **iKeycmd** 또는 **runmqakm** 명령을 사용하여 개인 인증을 갱신할 수 있습니다.

### 시작하기 전에

개인 인증서에 대한 키 크기를 더 크게 사용해야 하는 경우, 다시 작성된 인증서 요청이 기존 키에서 생성되기 때문에 아래에 설명된 갱신 단계가 작동하지 않습니다.

114 페이지의 『UNIX, Linux, and Windows 시스템에서 개인 인증서 요청』 에 설명된 단계에 따라 필요한 키 크기를 사용하여 새 인증서 요청을 작성하십시오. 이 프로세스는 기존 키를 바꿉니다.

### 이 태스크 정보

개인 인증서에는 만기 날짜가 있고, 그 후에는 인증서를 더 이상 사용할 수 없습니다. 이 태스크는 인증서가 만기되기 전에 기존 개인 인증서를 갱신하는 방법을 설명합니다.

*iKeyman* 사용자 인터페이스 사용

### 이 태스크 정보
iKeyman은 FIPS 준수 옵션을 제공하지 않습니다. FIPS를 준수하는 방법으로 SSL 또는 TLS 인증서를 관리해야 하는 경우, **runmqakm** 명령을 사용하십시오.

## 프로시저

iKeyman 사용자 인터페이스를 사용하여 개인 인증서를 신청하려면 다음 단계를 완료하십시오.

1. UNIX, Linux, and Windows 시스템에서 **strmqikm** 명령을 사용하여 iKeyman 사용자 인터페이스를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오.
   **열기** 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 요청을 생성하려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭하십시오.
   **비밀번호 프롬프트** 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오.
   키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. 드롭 다운 선택 메뉴에서 **개인 인증서**를 선택하고 갱신하려는 목록에서 인증서를 선택하십시오.
9. **요청 재작성 ...** 을 클릭하십시오. 표시됩니다.
   파일 이름과 파일 위치 정보를 입력하기 위한 창이 열립니다.
10. **파일 이름** 필드에서 기본값 certreq.arm을 승인하거나 전체 파일 경로를 포함한 새 값을 입력하십시오.
11. **확인**을 클릭하십시오. 인증서 요청이 117 페이지의 『9』 단계에서 선택한 파일에 저장됩니다.
12. 파일을 인증 기관(CA)으로 보내거나 파일을 CA의 웹 사이트에서 요청 양식으로 복사하여 새 개인 인증서를 요청하십시오.

명령행 사용

## 프로시저

**iKeycmd** 또는 **runmqakm** 명령을 사용하여 개인 인증서를 요청하려면 다음 명령을 사용하십시오.

- **iKeycmd**를 UNIX, Linux, and Windows 시스템에서 사용:

```
runmqckm -certreq -recreate -db filename -pw
password -label label
 -target filename
```

- runmqakm 사용:

```
runmqakm -certreq -recreate -db filename -pw
password -label label
  -target filename
```

설명:

**-db *filename***
CMS 키 데이터베이스의 완전한 파일 이름을 지정합니다.

**-pw *password***
CMS 키 데이터베이스의 비밀번호를 지정합니다.

**-target *filename***
인증서 요청의 파일 이름을 지정합니다.

## 다음에 수행할 작업

인증 기관으로부터 서명된 개인 인증서를 받은 후에는 이를 118 페이지의 『UNIX, Linux 및 Windows 시스템에서 키 저장소에 개인 인증서 수신』에 설명된 단계를 사용하여 키 데이터베이스에 추가할 수 있습니다.

### *UNIX, Linux* 및 *Windows* 시스템에서 키 저장소에 개인 인증서 수신

개인 인증서를 키 데이터베이스 파일에 수신하려면 이 프로시저를 사용하십시오. 키 저장소는 인증서 요청을 작성한 곳과 같은 저장소여야 합니다.

CA가 새 개인 인증서를 송신한 후에 새 인증서 요청을 생성한 위치인 키 데이터베이스 파일에 이를 추가합니다. CA가 이메일 메시지의 일부로서 인증서를 송신하면 인증서를 별도의 파일로 복사하십시오.

#### iKeyman 사용

FIPS를 준수하는 방식으로 SSL 인증서를 관리해야 하는 경우 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

들여올 인증서 파일이 현재 사용자에 대해 쓰기 권한이 있는지 확인한 후 다음 프로시저를 사용하여 큐 관리자나 WebSphere MQ MQI 클라이언트에서 개인 인증서를 키 데이터베이스 파일에 수신하십시오.

1. **strmqikm** 명령 (Windows UNIX and Linux에서)을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 인증서를 추가하려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭한 다음 **확인**을 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다. **개인 인증서** 보기를 선택하십시오.
8. **수신**을 클릭하십시오. 파일에서 인증서 수신 창이 열립니다.
9. 새 개인 인증서의 인증서 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름 및 위치를 선택하십시오.
10. **확인**을 클릭하십시오. 키 데이터베이스에 이미 개인 인증서가 있는 경우, 추가 중인 키를 데이터베이스에서 기본 키로 설정할지를 묻는 창이 열립니다.
11. **예** 또는 **아니오**를 누르십시오. 레이블 입력 창이 열립니다.
12. **확인**을 클릭하십시오. **개인 인증서** 필드가 사용자가 추가한 새 개인 인증서의 레이블을 보여줍니다.

#### 명령행 사용

iKeycmd 를 사용하여 키 데이터베이스 파일에 개인 인증서를 추가하려면 다음 명령을 사용하십시오.

- Linux 및 는 다음 명령을 실행하십시오.

```
runmqckm -cert -receive -file filename -db filename -pw
password
        -format ascii
```

설명:

| | |
|---|---|
| `-file filename` | 개인 인증서를 포함하는 파일의 완전한 파일 이름입니다. |
| `-db filename` | CMS 키 데이터베이스의 완전한 파일 이름입니다. |
| `-pw password` | CMS 키 데이터베이스의 비밀번호입니다. |
| `-format ascii` | 인증서의 형식입니다. 값은 Base64-encoded ASCII의 경우 `ascii` 또는 2진 DER 데이터의 경우 `binary`일 수 있습니다. 기본값은 `ascii`입니다. |

암호화 하드웨어를 사용하는 경우 131 페이지의 『PKCS #11 하드웨어로 개인 인증서 들여오기』 를 참조하십시오.

### 키 저장소에서 *CA* 인증서 추출

CA 인증서를 추출하려면 이 프로시저를 따르십시오.

### iKeyman 사용

FIPS를 준수하는 방식으로 SSL 인증서를 관리해야 하는 경우 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

CA 인증서를 추출하려는 시스템에서 다음 단계를 수행하십시오.

1. **strmqikm** 명령을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 추출하려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **키 데이터베이스 컨텐츠** 필드에서 **서명자 인증서**를 선택하고 추출하려는 인증서를 선택하십시오.
9. **추출**을 클릭하십시오. 파일에 인증서 추출 창이 열립니다.
10. 인증서의 **데이터 유형**을 선택하십시오. 예를 들어, .arm 확장자가 있는 파일의 경우 **Base64 인코딩된 ASCII 데이터**입니다.
11. 인증서를 저장하려는 인증서 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름과 위치를 선택하십시오.
12. **확인**을 클릭하십시오. 인증서가 사용자가 지정한 파일에 쓰여집니다.

### 명령행 사용

다음 명령을 사용하여iKeycmd을 사용하여 CA 인증서를 추출하십시오.

- Linux 및 의:

```
runmqckm -cert -extract -db filename -pw password -label label -target filename
         -format ascii
```

설명:

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 경로 이름입니다. |
| -pw *password* | CMS 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 인증서에 첨부된 레이블입니다. |
| -target *filename* | 대상 파일의 이름입니다. |
| -format *ascii* | 인증서의 형식입니다. 값은 Base64 인코딩된 ASCII의 경우 ascii이거나 2진 DER 데이터의 경우 binary입니다. 기본값은 ascii입니다. |

## *UNIX, Linux* 및 *Windows* 시스템의 키 저장소에서 자체 서명된 인증서의 공용 부분 추출

자체 서명 인증서의 공용 부분을 추출하려면 이 프로시저를 따르십시오.

### iKeyman 사용

FIPS를 준수하는 방식으로 SSL 인증서를 관리해야 하는 경우 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

자체 서명 인증서의 공용 부분을 추출하려는 시스템에서 다음 단계를 수행하십시오.

1. **strmqikm** 명령 ( Linux 및 )사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.

4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 예를 들어, key.kdb와 같이 인증서를 추출하려는 키 데이터베이스 파일을 선택하십시오.
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **키 데이터베이스 컨텐츠** 필드에서 **개인 인증서**를 선택한 후 인증서를 선택하십시오.
9. **인증서 추출**을 클릭하십시오. 파일에 인증서 추출 창이 열립니다.
10. 인증서의 **데이터 유형**을 선택하십시오. 예를 들어, .arm 확장자가 있는 파일의 경우 **Base64 인코딩된 ASCII 데이터**입니다.
11. 인증서를 저장하려는 인증서 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름과 위치를 선택하십시오.
12. **확인**을 클릭하십시오. 인증서가 사용자가 지정한 파일에 쓰여집니다. 인증서를 추출(내보내기가 아니라)할 때는 인증서의 공용 부분만 포함되므로 비밀번호가 필요하지 않습니다.

### 명령행 사용

iKeycmd 또는 runmqakm을 사용하여 자체 서명 인증서의 공용 부분을 추출하려면 다음 명령을 사용하십시오.

- Linux 및 의:

```
runmqckm -cert -extract -db filename -pw password -label label -target filename
         -format ascii
```

- runmqakm 사용:

```
runmqakm -cert -extract -db filename -pw password -label label
            -target filename -format ascii -fips
```

설명:

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 경로 이름입니다. |
| -pw *password* | CMS 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 인증서에 첨부된 레이블입니다. |
| -target *filename* | 대상 파일의 이름입니다. |
| -format *ascii* | 인증서의 형식입니다. 값은 Base64 인코딩된 ASCII의 경우 ascii이거나 2진 DER 데이터의 경우 binary입니다. 기본값은 ascii입니다. |

## *UNIX, Linux, and Windows* 시스템에서 *CA* 인증서(또는 자체 서명된 인증서의 공용 부분)를 키 저장소에 추가

CA 인증서 또는 자체 서명 인증서의 공용 부분을 키 저장소에 추가하려면 이 프로시저를 따르십시오.

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.

여기서 다음 지시사항은 CA 인증서를 가리키고, 이들은 또한 자체 서명 인증서의 공용 부분에도 적용됩니다.

**참고:** 추가할 인증서가 인증서 체인에 있으면 체인에서 해당 인증서 위에 있는 인증서도 모두 추가해야 합니다. 인증서가 ASCII(UTF-8) 또는 2진(DER) 인코딩인지 확인해야 합니다. IBM GSKit(Global Secure Toolkit)은 다른 인코딩 유형의 인증서를 지원하지 않습니다. 루트에서 시작하여 체인에서 해당 인증서 바로 아래에 있는 CA 인증서가 뒤에 오는 방식으로 엄격한 내림차순으로 인증서를 추가해야 합니다.

### iKeyman 사용

FIPS를 준수하는 방식으로 SSL 인증서를 관리해야 하는 경우 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

CA 인증서를 추가하고 싶은 시스템에서 다음 단계를 수행하십시오.

1. **strmqikm** 명령 ( 유닉스, Linux 및 윈도우 시스템) 을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 인증서를 추가하려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **키 데이터베이스 컨텐츠** 필드에서 **서명자 인증서**를 선택하십시오.
9. **추가**를 클릭하십시오. 파일에서 CA 인증 추가 창이 열립니다.
10. 인증서가 저장된 인증서 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름 및 위치를 선택하십시오.
11. **확인**을 클릭하십시오. 레이블 입력 창이 열립니다.
12. 레이블 입력 창에서 인증서 이름을 입력하십시오.
13. **확인**을 클릭하십시오. 인증서가 키 데이터베이스에 추가됩니다.

### 명령행 사용

iKeycmd 를 사용하여 CA 인증서를 추가하려면 다음 명령을 사용하십시오.

- Linux 및 는 다음 명령을 실행하십시오.

```
runmqckm -cert -add -db filename -pw password -label label -file filename
        -format ascii
```

설명:

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 경로 이름입니다. |
| -pw *password* | CMS 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 인증서에 첨부된 레이블입니다. |
| -file *filename* | 인증서를 포함하는 파일의 이름입니다. |
| -format *ascii* | 인증서의 형식입니다. 값은 Base64 인코딩된 ASCII의 경우 ascii이거나 2진 DER 데이터의 경우 binary입니다. 기본값은 ascii입니다. |

## 키 저장소에서 개인 인증서 내보내기

개인 인증서를 내보내려면 이 프로시저를 따르십시오.

### iKeyman 사용

FIPS를 준수하는 방식으로 SSL 인증서를 관리해야 하는 경우 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

개인 인증서를 내보내려는 시스템에서 다음 단계를 수행하십시오.

1. **strmqikm** 명령 (Windows UNIX and Linux에서)을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.

3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 인증서를 내보내려는 키 데이터베이스 파일을 선택하십시오(예: key.kdb).
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **키 데이터베이스 컨텐츠** 필드에서 **개인 인증서**를 선택하고 내보내려는 인증서를 선택하십시오.
9. **내보내기/가져오기**를 클릭하십시오. 키 내보내기/가져오기 창이 열립니다.
10. **키 내보내기**를 선택하십시오.
11. 내보내려는 인증서의 **키 파일 유형**을 선택하십시오(예: **PKCS12**).
12. 인증서를 내보내려는 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름과 위치를 선택하십시오.
13. **확인**을 클릭하십시오. 비밀번호 프롬프트 창이 열립니다. 인증서를 내보낼 때(추출이 아니라) 인증서의 공용 및 개인용 부분 모두가 포함된다는 점을 유의하십시오. 내보낸 파일이 비밀번호로 보호되기 때문입니다. 인증서를 추출할 때 인증서의 공용 부분만이 포함되므로 비밀번호는 필요하지 않습니다.
14. **비밀번호** 필드에 비밀번호를 입력하고 **비밀번호 확인** 필드에 다시 입력하십시오.
15. **확인**을 클릭하십시오. 인증서는 지정한 파일로 내보내기됩니다.

### 명령행 사용

iKeycmd를 사용하여 개인 인증서를 내보내려면 다음 명령을 사용하십시오.

- Linux 및 의:

```
runmqckm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
```

설명:

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 경로 이름입니다. |
| -pw *password* | CMS 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 인증서에 첨부된 레이블입니다. |
| -type *cms* | 데이터베이스의 유형입니다. |
| -target *filename* | 대상 파일의 완전한 경로 이름입니다. |
| -target_pw *password* | 인증서를 암호화하기 위한 비밀번호입니다. |
| -target_type *pkcs12* | 인증서의 유형입니다. |

## *UNIX, Linux, and Windows* 시스템에서 개인 인증서를 키 저장소에 가져오기

개인 인증서를 가져오려면 이 프로시저를 따르십시오.

PKCS #12 형식의 개인 인증서를 키 데이터베이스 파일로 들여오기 전에 먼저 발행 CA 인증서의 유효한 전체 체인을 키 데이터베이스 파일에 추가해야 합니다(120 페이지의 『UNIX, Linux, and Windows 시스템에서 CA 인증서(또는 자체 서명된 인증서의 공용 부분)를 키 저장소에 추가』 참조).

PKCS #12 파일은 임시로 간주되어야 하고 사용 후에 삭제되어야 합니다.

### iKeyman 사용

FIPS를 준수하는 방법으로 SSL 인증서를 관리해야 하는 경우에는 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

개인 인증서를 가져오려는 시스템에서 다음 단계를 수행하십시오.

1. **strmqikm** 명령을 사용하여 iKeyman GUI를 시작하십시오.

2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 표시됩니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 인증서를 추가하려는 키 데이터베이스 파일을 선택하십시오(예: `key.kdb`).
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 표시됩니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. **키 데이터베이스 컨텐츠** 필드에서 **개인 인증서**를 선택하십시오.
9. 개인 인증서 보기에 인증서가 있는 경우 다음 단계를 따르십시오.
   a. **내보내기/가져오기**를 클릭하십시오. 키 내보내기/가져오기 창이 표시됩니다.
   b. **키 가져오기**를 선택하십시오.
10. 개인 인증서 보기에 인증서가 없는 경우 **가져오기**를 클릭하십시오.
11. 가져오려는 인증서의 **키 파일 유형**을 선택하십시오(예: PKCS12).
12. 인증서가 저장된 인증서 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름 및 위치를 선택하십시오.
13. **확인**을 클릭하십시오. 비밀번호 프롬프트 창이 표시됩니다.
14. **비밀번호** 필드에 인증서가 내보내기되었을 때 사용된 비밀번호를 입력하십시오.
15. **확인**을 클릭하십시오. 레이블 변경 창이 표시됩니다. 이 창에서, 예를 들어 대상 키 데이터베이스에 동일한 레이블을 가지는 인증서가 이미 존재하는 경우 들여오는 중인 인증서의 레이블을 변경할 수 있습니다. 인증서 레이블 변경은 인증서 체인 유효성 검증에 영향을 미치지 않습니다. 이는 인증서를 특정 큐 관리자 또는 클라이언트와 연관시키기 위해 WebSphere MQ 에 필요한 개인 인증 레이블을 변경하는 데 사용할 수 있습니다 (예:`ibmwebspheremqqm1` ).
16. 레이블을 변경하려면 **변경할 레이블 선택** 목록에서 필수 레이블을 선택하십시오. 레이블이 **새 레이블 입력** 입력 항목 필드로 복사됩니다. 레이블 텍스트를 새 레이블의 레이블 텍스트로 바꾸고 **적용**을 클릭하십시오.
17. **새 레이블 입력** 입력 항목 필드의 텍스트가 다시 **변경할 레이블 선택** 필드로 복사되어 원래 선택된 레이블을 대체하고 해당하는 인증서를 다시 레이블 지정합니다.
18. 변경해야 하는 모든 레이블을 변경한 후에는 **확인**을 클릭하십시오. 레이블 변경 창이 닫히고 원래 IBM 키 관리 창이 올바르게 레이블된 인증서로 업데이트된 **개인 인증서** 및 **서명자 인증서** 필드와 함께 다시 표시됩니다.
19. 인증서가 대상 키 데이터베이스로 가져오기됩니다.

## 명령행 사용

iKeycmd를 사용하여 개인 인증서를 가져오려면 다음 명령을 사용하십시오.

- Linux 및 의:

```
runmqckm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label
```

설명:

| | |
|---|---|
| `-file filename` | PKCS #12 인증서를 포함하는 파일의 완전한 파일 이름입니다. |
| `-pw password` | PKCS #12 인증서의 비밀번호입니다. |
| `-type pkcs12` | 파일의 유형입니다. |
| `-target filename` | 대상 CMS 키 데이터베이스의 이름입니다. |
| `-target_pw password` | CMS 키 데이터베이스의 비밀번호입니다. |
| `-target_type cms` | `-target`에 의해 지정된 데이터베이스의 유형입니다. |

| | |
|---|---|
| -label *label* | 소스 키 데이터베이스로부터 가져오려는 인증서의 레이블입니다. |
| -new_label *label* | 대상 데이터베이스에서 인증서에 지정될 레이블입니다. -new_label 옵션을 생략하면 기본값은 -label 옵션과 같은 옵션을 사용합니다. |

iKeycmd는 인증서 레이블을 직접 변경하는 명령을 제공하지 않습니다. 인증서 레이블을 변경하려면 다음 단계를 사용하십시오.

1. **-cert -export** 명령을 사용하여 인증서를 PKCS #12 파일로 내보내십시오. -label 옵션의 기존 인증서 레이블을 지정하십시오.
2. **-cert -delete** 명령을 사용하여 원래 키 데이터베이스에서 인증서의 기존 사본을 제거하십시오.
3. **-cert -import** 명령을 사용하여 PKCS #12 파일에서 인증서를 가져오십시오. -label 옵션의 이전 레이블을 지정하고 -new_label 옵션에 필수 새 레이블을 지정하십시오. 인증서가 필수 레이블과 함께 키 데이터베이스로 다시 가져오기됩니다.

### *Microsoft .pfx* 파일에서 들여오기

iKeyman을 사용하여 Microsoft .pfx 파일에서 가져오려면 이 프로시저를 따르십시오. .pfx 파일을 들여오는 데 runmqakm를 사용할 수 없습니다.

.pfx 파일은 동일 키와 관련된 두 개의 인증서를 포함할 수 있습니다. 하나는 개인 또는 사이트 인증서(공용 및 개인용 키 모두 포함)입니다. 다른 하나는 CA(서명자) 인증서(공용 키만을 포함)입니다. 이러한 인증서는 동일한 CMS 키 데이터베이스 파일에 공존할 수 없으므로 이들 중 하나만을 가져올 수 있습니다. 또한 "친근한 이름" 또는 레이블은 서명자 인증서에만 첨부됩니다.

개인 인증서는 시스템 생성 UUID(Unique User Identifier)에 의해서 식별됩니다. 이 섹션은 이전에 CA(서명자) 인증서에 지정한 친근한 이름으로 레이블을 지정하는 동안 pfx 파일에서 개인 인증서를 가져오기를 보여줍니다. 발행 CA(서명자) 인증서는 이미 대상 키 데이터베이스에 추가되어야 합니다. PKCS#12 파일은 임시적인 것으로 간주되고 사용 후 삭제되어야 합니다.

소스 pfx 키 데이터베이스에서 개인 인증서를 가져오려면 다음 단계를 수행하십시오.

1. **strmqikm** 명령 ( Linux, )사용하여 iKeyman GUI를 시작하십시오. IBM 키 관리 창이 표시됩니다.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 표시됩니다.
3. 키 데이터베이스 유형 **PKCS12**를 선택하십시오.
4. **이 단계를 수행하기 전에 pfx 데이터베이스를 백업하는 것이 좋습니다.** 가져오려는 pfx 키 데이터베이스를 선택하십시오. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 표시됩니다.
5. 키 데이터베이스 비밀번호를 입력하고 **확인**을 클릭하십시오. IBM 키 관리 창이 표시됩니다. 제목 표시줄은 선택된 pfx 키 데이터베이스 파일의 이름을 보여주고 파일이 열려 있고 준비되어 있음을 나타냅니다.
6. 목록에서 **서명자 인증서**를 선택하십시오. 필수 인증서의 "친근한 이름"이 서명자 인증서 패널에 레이블로서 표시됩니다.
7. 레이블 항목을 선택하고 **삭제**를 클릭하여 서명자 인증서를 제거하십시오. 확인 창이 표시됩니다.
8. **예**를 클릭하십시오. 선택한 레이블이 더 이상 서명자 인증서 패널에 표시되지 않습니다.
9. 모든 서명자 인증서에 대해 6, 7 및 8단계를 반복하십시오.
10. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 표시됩니다.
11. pfx 파일이 가져오기되고 있는 대상 키 CMS 데이터베이스를 선택하십시오. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 표시됩니다.
12. 키 데이터베이스 비밀번호를 입력하고 **확인**을 클릭하십시오. IBM 키 관리 창이 표시됩니다. 제목 표시줄은 선택한 키 데이터베이스 파일의 이름을 표시하여 파일이 열려 있고 준비가 되었음을 나타냅니다.
13. 목록에서 **개인 인증서**를 선택하십시오.
14. 개인 인증서 보기에 인증서가 있는 경우 다음 단계를 따르십시오.
    a. **키 가져오기/내보내기**를 클릭하십시오. 키 내보내기/가져오기 창이 표시됩니다.
    b. 조치 유형 선택에서 **가져오기**를 선택하십시오.
15. 개인 인증서 보기에 인증서가 없는 경우 **가져오기**를 클릭하십시오.

16. PKCS12 파일을 선택하십시오.
17. 4단계에서 사용된 pfx 파일의 이름을 입력하십시오. **확인**을 클릭하십시오. 비밀번호 프롬프트 창이 표시됩니다.
18. 서명자 인증서를 삭제할 때 지정한 것과 같은 비밀번호를 지정하십시오. **확인**을 클릭하십시오.
19. 레이블 변경 창이 표시됩니다(가져오기에는 단 하나의 인증서만이 사용 가능해야 하므로). 인증서의 레이블은 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 형식인 UUID여야 합니다.
20. 레이블을 변경하려면 **변경할 레이블 선택** 패널에서 UUID를 선택하십시오. 레이블이 **새 레이블 입력:** 필드로 복제됩니다. 레이블 텍스트를 7단계에서 삭제된 친숙한 이름의 레이블 텍스트로 바꾸고 **적용**을 클릭하십시오. 친숙한 이름은 ibmwebspheremq양식이어야 하며, 큐 관리자 이름 또는 WebSphere MQ MQI 클라이언트 사용자 로그온 ID가 소문자로 되어 있어야 합니다.
21. **확인**을 클릭하십시오. 이제 레이블 변경 창이 제거되고 원래 IBM 키 관리 창이 올바르게 레이블된 개인 인증서로 업데이트된 개인 인증서 및 서명자 인증서 패널과 함께 다시 표시됩니다.
22. pfx 개인 인증서가 이제 (대상)데이터베이스로 가져오기됩니다.

iKeycmd 를 사용하여 인증서 레이블을 변경할 수 없습니다.

### *PKCS #7* 파일에서 가져오기

iKeyman 및 iKeycmd 도구는 PKCS #7(.p7b) 파일을 지원하지 않습니다. 인증서를 PKCS #7 파일로부터 가져오려면 runmqckm 도구를 사용하십시오.

CA 인증서를 PKCS #7 파일로부터 추가하려면 다음 명령을 사용하십시오.

```
runmqckm -cert -add -db filename -pw password -type cms -file filename
-label label
```

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 파일 이름입니다. |
| -pw *password* | 키 데이터베이스의 비밀번호입니다. |
| -type *cms* | 키 데이터베이스의 유형입니다. |
| -file *filename* | PKCS #7 파일의 이름입니다. |
| -label *label* | 대상 데이터베이스에서 인증서에 지정된 레이블입니다. 첫 번째 인증서는 제공된 레이블을 받습니다. 있는 경우, 다른 모든 인증서에는 제목 이름이 레이블 지정됩니다. |

개인 인증서를 PKCS #7 파일에서 가져오려면 다음 명령을 사용하십시오.

```
runmqckm -cert -import -db filename -pw password -type pkcs7 -target filename
-target_pw password -target_type cms -label label -new_label label
```

| | |
|---|---|
| -db *filename* | PKCS #7 인증서를 포함하는 파일의 완전한 파일 이름입니다. |
| -pw *password* | PKCS #7 인증서의 비밀번호입니다. |
| -type *pkcs7* | 파일의 유형입니다. |
| -target *filename* | 목적지 키 데이터베이스의 이름입니다. |
| -target_pw *password* | 목적지 키 데이터베이스의 비밀번호입니다. |
| -target_type *cms* | -target에 의해 지정된 데이터베이스의 유형입니다. |
| -label *label* | 가져오려는 인증서의 레이블입니다. |
| -new_label *label* | 대상 데이터베이스에서 인증서에 지정될 레이블입니다. -new_label 옵션을 생략하면 사용할 기본값은 -label 옵션과 같습니다. |

### *UNIX, Linux, and Windows* 시스템의 키 저장소에서 인증서 삭제

개인 또는 CA 인증서를 제거하려면 이 프로시저를 사용하십시오.

### iKeyman 사용

FIPS를 준수하는 방식으로 SSL 인증서를 관리해야 하는 경우 runmqakm 명령을 사용하십시오. iKeyman은 FIPS 준수 옵션을 제공하지 않습니다.

1. **strmqikm** 명령 ( 유닉스, Linux 및 윈도우 시스템) 을 사용하여 iKeyman GUI를 시작하십시오.
2. **키 데이터베이스 파일** 메뉴에서 **열기**를 클릭하십시오. 열기 창이 열립니다.
3. **키 데이터베이스 유형**을 클릭하고 **CMS**(인증서 관리 시스템)를 선택하십시오.
4. 키 데이터베이스 파일이 있는 디렉토리로 이동하려면 **찾아보기**를 클릭하십시오.
5. 예를 들어, key.kdb와 같이 인증서를 삭제하려는 키 데이터베이스 파일을 선택하십시오.
6. **열기**를 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
7. 키 데이터베이스를 작성할 때 설정한 비밀번호를 입력하고 **확인**을 클릭하십시오. 키 데이터베이스 파일의 이름이 **파일 이름** 필드에 표시됩니다.
8. 드롭 다운 목록에서 **개인 인증서** 또는 **서명자 인증서**를 선택하십시오.
9. 삭제하려는 인증서를 선택하십시오.
10. 인증서 사본을 이미 가지고 있지 않고 인증서를 저장하려는 경우 **내보내기/들여오기**를 누르고 인증서를 내보내십시오(121 페이지의 『키 저장소에서 개인 인증서 내보내기』 참조).
11. 인증서를 선택한 상태에서 **삭제**를 클릭하십시오. 확인 창이 열립니다.
12. **예**를 클릭하십시오. **개인 인증서** 필드가 더 이상 사용자가 삭제한 인증서의 레이블을 보여주지 않습니다.

### 명령행 사용

iKeycmd 또는 runmqakm을 사용하여 인증서를 삭제하려면 다음 명령을 사용하십시오.

- Linux 및 의:

```
runmqckm -cert -delete -db filename -pw password -label label
```

설명:

| | |
|---|---|
| -db *filename* | CMS 키 데이터베이스의 완전한 파일 이름입니다. |
| -pw *password* | CMS 키 데이터베이스의 비밀번호입니다. |
| -label *label* | 개인 인증서에 첨부된 레이블입니다. |
| -fips | 명령이 FIPS 모드에서 실행되도록 지정합니다. 이 모드는 BSafe 암호화 라이브러리의 사용을 불가능하게 만듭니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **runmqakm** 명령이 실패합니다. |

### 키 저장소 보호를 위한 강력한 비밀번호 생성

**runmqakm** 명령을 사용하여 키 저장소 보호를 위한 강력한 비밀번호를 생성할 수 있습니다.

강력한 비밀번호를 생성하기 위해 다음 매개변수와 함께 **runmqakm** 명령을 사용할 수 있습니다.

```
runmqakm -random -create -length 14 -strong -fips
```

후속 인증서 관리 명령의 **-pw** 매개변수에서 생성된 비밀번호를 사용할 때 항상 비밀번호 주변에 큰따옴표를 두십시오. UNIX and Linux 시스템에서 다음 문자가 비밀번호 문자열에 나타나는 경우 백슬래시 문자를 사용하여 이스케이프 처리해야 합니다.

```
!   \   "    '
```

**runmqckm**, **runmqakm** 또는 iKeyman GUI로부터 프롬프트에 대한 응답에 비밀번호를 입력할 때는 비밀번호를 따옴표로 묶거나 이스케이프 처리할 필요가 없습니다. 운영 체제 쉘이 이러한 경우 데이터 입력 항목에 영향을 주지 않기 때문에 이는 필요하지 않습니다.

## *UNIX, Linux, and Windows* 시스템에서 암호화 하드웨어 구성

큐 관리자 또는 클라이언트의 암호화 하드웨어를 여러 방법으로 구성할 수 있습니다.

다음 방법 중 하나를 사용하여 UNIX, Linux 또는 Windows 시스템에 큐 관리자에 대한 암호화 하드웨어를 구성할 수 있습니다.

- ALTER QMGR에 설명된 대로 SSLCRYP 매개변수와 함께 ALTER QMGR MQSC 명령을 사용하십시오.
- IBM WebSphere MQ 탐색기를 사용하여 UNIX, Linux 또는 Windows 시스템에서 암호화 하드웨어를 구성하십시오. 자세한 정보는 온라인 도움말을 참조하십시오.

다음 방법 중 하나를 사용하여 UNIX, Linux 또는 윈도우 시스템에서 WebSphere MQ 클라이언트에 대한 암호화 하드웨어를 구성할 수 있습니다.

- MQSSLCRYP 환경 변수를 설정하십시오. MQSSLCRYP에 허용되는 값은 ALTER QMGR에 설명된 대로 SSLCRYP 매개변수에서와 같습니다. SSLCRYP 매개변수의 GSK_PCS11 버전을 사용하는 경우에는 PKCS #11 토큰 레이블은 전적으로 소문자로 지정되어야 합니다.
- MQCONNX 호출에서 SSL 구성 옵션 구조의 **CryptoHardware** 필드, MQSCO를 설정하십시오. 자세한 정보는 MQSCO의 개요를 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 127 페이지의 『PKCS #11 하드웨어에서 인증서 관리』에서 설명합니다.

*PKCS #11* 하드웨어에서 인증서 관리
PKCS #11 인터페이스를 지원하는 암호화 하드웨어에서 디지털 인증서를 관리할 수 있습니다.

### 이 태스크 정보

인증 기관(CA) 인증서를 저장할 의도가 없더라도 IBM WebSphere MQ 환경을 준비하기 위해 키 데이터베이스를 작성해야 하고 모든 인증서를 암호화 하드웨어에 저장합니다. 키 데이터베이스는 SSLKEYR 필드에서 참조하기 위해 큐 관리자에 필요하거나 클라이언트 애플리케이션이 MQSSLKEYR 환경 변수에서 참조하기 위해 필요합니다. 이 키 데이터베이스는 또한 인증서 요청을 작성 중인 경우에 필수입니다.

명령행을 사용하거나 **strmqikm** (iKeyman) 사용자 인터페이스를 사용하여 키 데이터베이스를 작성합니다.

### 프로시저

명령행을 사용하여 키 데이터베이스를 작성한다.

1. 다음 명령 중 하나를 실행하십시오.
   - UNIX, Linux, and Windows 시스템의 경우:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - runmqakm 사용:

     ```
     runmqakm -keydb -create -db filename -pw password -type cms
         -stash -fips -strong
     ```

   설명:

   **-db *filename***
   CMS 키 데이터베이스의 완전한 파일 이름을 지정하고 .kdb의 파일 확장자가 있어야 합니다.

   **-pw *password***
   CMS 키 데이터베이스의 비밀번호를 지정합니다.

   **-type *cms***
   데이터베이스의 유형을 지정합니다. (IBM WebSphere MQ의 경우, cms여야 합니다.)

**-stash**
키 데이터베이스 비밀번호를 파일에 저장합니다.

**-fips**
BSafe 암호화 라이브러리의 사용을 불가능하게 만듭니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서는 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **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 내에 있는 표준 ASCII 인쇄 가능 문자 세트로 되어 있습니다.

또는 **strmqikm** (iKeyman) 사용자 인터페이스를 사용하여 키 데이터베이스를 작성하십시오.

2. UNIX and Linux 시스템에서 루트 사용자로서 로그인하십시오. Windows 시스템에서 관리자 또는 MQM 그룹의 구성원으로서 로그인하십시오.
3. **strmqikm** 명령을 실행하여 iKeyman 사용자 인터페이스를 시작하십시오.
4. **키 데이터베이스 파일** > **열기**를 클릭하십시오.
5. **키 데이터베이스 유형**을 클릭하고 **PKCS11Direct**를 선택하십시오.
6. **파일 이름** 필드에 암호화 하드웨어를 관리하기 위한 모듈 이름을 입력하십시오(예: PKCS11_API.so).

   PKCS #11 암호화 하드웨어에 저장된 인증서 또는 키를 사용 중인 경우 iKeycmd 및 iKeyman은 64비트 프로그램입니다. PKCS #11 지원에 필요한 외부 모듈이 64비트 프로세스에 로드되므로 암호화 하드웨어 관리를 위해서는 64비트 PKCS #11 라이브러리를 설치해야 합니다. Windows및 Linux x86 32비트플랫폼은 해당 플랫폼에서 iKeyman 및 iKeycmd 프로그램이 32비트이므로 유일한 예외입니다.
7. **위치** 필드에 경로를 입력하십시오.

   - UNIX and Linux 시스템에서 이는 예를 들어, /usr/lib/pksc11일 수 있습니다.
   - Windows 시스템에서 라이브러리 이름을 입력할 수 있습니다(예: cryptoki).

   **확인**을 클릭하십시오. 암호화 토큰 열기 창이 열립니다.
8. **암호화 토큰 비밀번호** 필드에 암호화 하드웨어를 구성할 때 설정한 비밀번호를 입력하십시오.
9. 암호화 하드웨어에 개인 인증서를 받거나 가져오는 데 필요한 서명자 인증서를 보유하기 위한 용량이 있는 경우에는 보조 키 데이터베이스 확인란을 둘 모두 지우고 129 페이지의 『13』 단계부터 계속하십시오.

   서명자 인증서를 보유하기 위해 보조 CMS 키 데이터베이스가 필요한 경우에는 **기존 보조 키 데이터베이스 파일 열기** 또는 **새 보조 키 데이터베이스 파일 작성**을 선택하십시오.
10. **파일 이름** 필드에 파일 이름을 입력하십시오. 이 필드에는 이미 key.kdb 텍스트가 있습니다. 스템 이름이 key이면 이 필드를 변경하지 말고 두십시오. 다른 스템 이름을 지정한 경우에는 key를 사용자의 스템 이름으로 바꾸십시오. .kdb 접미부를 변경하지 않아야 합니다.
11. **위치** 필드에 경로를 입력하십시오. 예:

    - 큐 관리자의 경우: /var/mqm/qmgrs/QM1/ssl
    - IBM WebSphere MQ MQI 클라이언트의 경우: /var/mqm/ssl

    **확인**을 클릭하십시오. 비밀번호 프롬프트 창이 열립니다.
12. 비밀번호를 입력하십시오.

    128 페이지의 『9』 단계에서 **기존 보조 키 데이터베이스 파일 열기**를 선택한 경우에는 **비밀번호** 필드에 비밀번호를 입력하십시오.

128 페이지의 『9』 단계에서 **새 보조 키 데이터베이스 파일 작성** 을 선택한 경우 다음 하위 단계를 완료하십시오.

a) **비밀번호** 필드에 비밀번호를 입력하고 **비밀번호 확인** 필드에 다시 입력하십시오.

b) **파일에 비밀번호 숨김**을 선택하십시오. 비밀번호를 숨기지 않은 경우에는 키 데이터베이스 파일에 액세스하는 데 필요한 비밀번호를 구할 수 없으므로 SSL 채널 시작 시도에 실패합니다.

c) **확인**을 클릭하십시오. 비밀번호가 `key.sth` 파일에 있음을 확인하는 창이 열립니다(다른 스템 이름을 지정하지 않은 경우).

13. **확인**을 클릭하십시오. 키 데이터베이스 컨텐츠 프레임이 표시됩니다.

*PKCS #11* 하드웨어의 개인 인증서 요청
다음 프로시저를 큐 관리자나 IBM WebSphere MQ MQI 클라이언트에 사용하여 사용자의 암호화 하드웨어에 대한 개인 인증서를 요청합니다.

*iKeyman* 사용자 인터페이스 사용

## 이 태스크 정보

**참고:** WebSphere 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.

디지털 서명 알고리즘 이름 SHA3WithRSA 및 SHA5WithRSA는 각각 SHA384WithRSA 및 SHA512WithRSA의 축약된 양식이므로 더 이상 사용되지 않습니다.

## 프로시저

iKeyman 사용자 인터페이스로부터 개인 인증서를 요청하려면 다음 단계를 완료하십시오.

1. 암호화 하드웨어에 대해 작업하기 위한 단계를 완료하십시오. 127 페이지의 『PKCS #11 하드웨어에서 인증서 관리』 의 내용을 참조하십시오.
2. **작성** 메뉴에서 **새 인증서 요청**을 클릭하십시오.
   새 키 및 인증서 요청 작성 창이 열립니다.
3. **키 레이블** 필드에서 다음 레이블을 입력하십시오.
   - 큐 관리자에서는 `ibmwebspheremq` 뒤에 소문자로 변경한 큐 관리자의 이름을 입력하십시오. 예를 들어, `QM1`라는 큐 관리자의 경우 `ibmwebspheremqqm1`를 입력하십시오.
   - IBM WebSphere MQ MQI client의 경우, `ibmwebspheremq` 를 입력한 다음 로그온 사용자 ID를 소문자로 입력하십시오 (예: `ibmwebspheremqmyuserid` ).
4. **공용 이름** 및 **조직**에 값을 입력하고 **국가**를 선택하십시오. 나머지 선택적 필드에는 기본값을 승인하거나 새 값을 입력하거나 선택하십시오.
   **조직 단위** 필드에 단 하나의 이름만을 제공할 수 있음을 유의하십시오. 이러한 필드에 대한 자세한 정보는 10 페이지의 『식별 이름』 의 내용을 참조하십시오.
5. **인증서 요청을 저장할 파일의 이름 입력** 필드에 기본 `certreq.arm`을 승인하거나 전체 경로가 있는 새 값을 입력하십시오.
6. **확인**을 클릭하십시오.
   확인 창이 열립니다.
7. **확인**을 클릭하십시오.
   **개인 인증서 요청** 목록에 사용자가 작성한 새 개인 인증서 요청의 레이블이 표시됩니다. 인증서 요청이 129 페이지의 『5』 단계에서 선택한 파일에 저장됩니다.
8. 파일을 인증 기관(CA)으로 보내거나 파일을 CA의 웹 사이트에서 요청 양식으로 복사하여 새 개인 인증서를 요청하십시오.

## 프로시저

**runmqckm** 또는 **runmqakm** 명령을 사용하여 개인 인증서를 요청하려면 다음 명령을 사용하십시오.

- **runmqckm** 사용:

```
runmqckm -certreq -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -file filename -sig_alg algorithm
```

  -dn *distinguished_name* 대신 -san_dsname *DNS_names* , -san_emailaddr *email_addresses* 또는 -san_ipaddr *IP_addresses* 을(를) 사용할 수 있습니다.

- runmqakm 사용:

```
runmqakm -certreq -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -file filename -fips
        -sig_alg algorithm
```

설명:

**-db *filename***
CMS 키 데이터베이스의 완전한 파일 이름을 지정합니다.

**-pw *password***
CMS 키 데이터베이스의 비밀번호를 지정합니다.

**-label *label***
인증서에 첨부된 키 레이블을 지정합니다.

**-dn *distinguished_name***
큰따옴표로 묶인 X.500 식별 이름을 지정합니다. 하나 이상의 속성이 필요합니다. 여러 OU 및 DC 속성을 제공할 수 있습니다.

**-size *key_size***
키 크기를 지정합니다. **runmqckm**을 사용하는 경우, 값은 512 또는 1024일 수 있습니다. **runmqakm**을 사용하는 경우, 값은 512, 1024 또는 2048일 수 있습니다.

**-file *filename***
인증서 요청의 파일 이름을 지정합니다.

**-fips**
명령이 FIPS 모드에서 실행되도록 지정합니다. 이 모드는 BSafe 암호화 라이브러리의 사용을 불가능하게 만듭니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서는 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **runmqakm** 명령이 실패합니다.

**-sig_alg**
**runmqckm**의 경우, 항목의 키 쌍 작성에 사용된 비대칭 서명 알고리즘을 지정합니다. 값은 MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithRSA, SHA256_WITH_RSA, SHA256WithRSA, SHA2WithRSA, SHA384_WITH_RSA, SHA384WithRSA, SHA512_WITH_RSA, SHA512WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA 또는 SHAWithRSA가 될 수 있습니다. 기본값은 SHA1WithRSA입니다.

**-sig_alg**
**runmqakm**의 경우 인증서 요청의 작성 동안에 사용된 해싱 알고리즘을 지정합니다. 이 해싱 알고리즘은 새로 작성된 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_SHA512가 될 수 있습니다. 기본값은 SHA1WithRSA입니다.

**-san_dnsname *DNS_names***
작성 중인 항목의 쉼표로 구분되거나 공백으로 구분된 DNS 이름 목록을 지정합니다.

**-san_emailaddr *email_addresses***
작성 중인 항목의 쉼표로 구분되거나 공백으로 구분된 이메일 주소 목록을 지정합니다.

**-san_ipaddr *IP_addresses***
작성 중인 항목의 쉼표로 구분되거나 공백으로 구분된 IP 주소 목록을 지정합니다.

*PKCS #11* 하드웨어로 개인 인증서 들여오기
다음 프로시저를 큐 관리자나 IBM WebSphere MQ MQI 클라이언트에 사용하여 사용자의 암호화 하드웨어로 개인 인증서를 가져옵니다.

*iKeyman* 사용

## 프로시저

iKeyman 사용자 인터페이스로부터 개인 인증서를 요청하려면 다음 단계를 완료하십시오.

1. 암호화 하드웨어에 대해 작업하기 위한 단계를 완료하십시오. 127 페이지의 『PKCS #11 하드웨어에서 인증서 관리』의 내용을 참조하십시오.
2. **수신**을 클릭하십시오. 파일에서 인증서 수신 창이 열립니다.
3. 새 개인 인증서의 **데이터 유형**을 선택하십시오. 예를 들어, `.arm` 확장자가 있는 파일의 Base64-encoded ASCII data입니다.
4. 새 개인 인증서의 인증서 파일 이름 및 위치를 입력하거나 **찾아보기**를 클릭하여 이름 및 위치를 선택하십시오.
5. **확인**을 클릭하십시오. 키 데이터베이스에 개인 인증서가 이미 있는 경우, 사용자가 추가 중인 키를 데이터베이스에서 기본 키로 설정할지 여부를 묻는 창이 열립니다.
6. **예** 또는 **아니오**를 클릭하십시오. 레이블 입력 창이 열립니다.
7. 레이블을 입력하십시오.
   예를 들어, 개인 인증서를 요청할 때 사용한 레이블과 동일한 레이블을 사용할 수 있습니다. 레이블은 정확한 IBM WebSphere MQ 형식이어야 합니다.
   - 큐 관리자에서는, ibmwebspheremq 뒤에 큐 관리자의 이름이 소문자입니다. 예를 들어, QM1이라는 큐 관리자의 경우 레이블은 ibmwebspheremqqm1입니다.
   - IBM WebSphere MQ MQI 클라이언트의 경우, ibmwebspheremq 뒤에 소문자로 사용자의 로그인 사용자 ID가 위치합니다. 예를 들어, 사용자 ID MyUserID인 경우 레이블은 ibmwebspheremqmyuserid입니다.
8. **확인**을 클릭하십시오. **개인 인증서** 목록이 추가한 새 개인 인증서의 레이블을 표시합니다. 이 레이블은 제공한 레이블 앞에 암호화 토큰 레이블을 추가하여 구성합니다.

명령행 사용

## 프로시저

명령행으로부터 개인 인증서를 요청하려면 다음 단계를 완료하십시오.

1. 사용자 환경에 대해 구성된 명령 창을 여십시오.
2. 사용자의 운영 체제와 구성에 적합한 명령을 입력하십시오.
   - Windows, UNIX and Linux 시스템에서는 다음 명령 중 하나를 사용하십시오.

```
runmqckm -cert -receive -file filename -crypto path
```

```
-tokenlabel hardware_token -pw hardware_password -format cert_format
```

```
runmqakm -cert -receive -file filename -crypto path
-tokenlabel hardware_token -pw hardware_password -format cert_format -fips
```

설명:

**-file *filename***
개인 인증서를 포함하고 있는 파일의 완전한 파일 이름을 지정합니다.

**-crypto *path***
하드웨어와 함께 제공된 PKCS #11 라이브러리에 완전한 경로를 지정합니다.

**-tokenlabel *hardware_token***
설치 중에 암호화 하드웨어의 스토리지 부분에 제공된 레이블을 지정합니다.

**-pw *hardware_password***
하드웨어의 액세스 비밀번호를 지정합니다.

**-format *cert_format***
인증서의 형식을 지정합니다. 값은 Base64 인코딩된 ASCII의 경우 `ascii`이거나 2진 DER 데이터의 경우 `binary`입니다. 기본값은 ASCII입니다.

**-fips**
이 모드는 FIPS 모드에서 실행됨을 지정합니다. 이 모드는 BSafe 암호화 라이브러리의 사용을 불가능하게 합니다. ICC 컴포넌트만 사용되며 이 컴포넌트는 FIPS 모드에서 성공적으로 초기화되어야 합니다. FIPS 모드에서는 ICC 컴포넌트는 FIPS 140-2 유효성 검증된 알고리즘을 사용합니다. ICC 컴포넌트가 FIPS 모드에서 초기화되지 않는 경우에는 **runmqakm** 명령이 실패합니다.

# 사용자 식별 및 인증

MQCSP 구조를 사용하거나 몇몇 사용자 엑시트 프로그램 유형에서 사용자를 식별하고 인증할 수 있습니다.

## MQCSP 구조 사용

MQCONNX 호출에서 MQCSP 연결 보안 매개변수 구조를 지정할 수 있습니다. 이 구조에는 사용자 ID 및 비밀번호가 포함됩니다. 필요한 경우 보안 엑시트에서 MQCSP를 변경할 수 있습니다.

**참고:** 오브젝트 권한 관리자(OAM)는 비밀번호를 사용하지 않습니다. 그러나 OAM은 사용자 ID를 사용하여 인증의 사소한 양식으로 간주될 수 있는 몇몇 제한된 작업을 수행합니다. 이러한 검사는 사용자가 애플리케이션에서 이러한 매개변수를 사용하는 경우 사용자가 또 다른 사용자 ID를 채택하는 것을 중지합니다.

## 보안 엑시트에서 식별 및 인증 구현

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 쪽에서 MCA는 일반적으로 WebSphere MQ 클라이언트 애플리케이션의 사용자 대신 작동합니다. 이 상황에서는, 상호 인증이 두 큐 관리자 사이나 큐 관리자와 WebSphere MQ MQI 클라이언트 애플리케이션의 사용자 사이에서 실제로 일어납니다.

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 95 페이지의 『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 유

효성을 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.

SSL(Secure Sockets Layer) 및 TLS(Transport Layer Security)는 방금 설명한 것과 같은 PKI 기술을 사용합니다. SSL 및 TLS가 인증을 수행하는 방법에 대한 자세한 정보는 14 페이지의 『SSL(Secure Socket Layer) 및 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 206 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 68 페이지의 『세션 레벨 인증』에 설명되어 있습니다.

상호 인증을 위한 모든 앞선 기술은 단방향 인증을 제공하기 위해 채택될 수 있습니다.

## 메시지 엑시트에서 식별 및 인증 구현

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 136 페이지의 『API 엑시트와 API 교차 엑시트에서 ID 맵핑』의 내용을 참조하십시오.

## API 엑시트 및 API 교차 엑시트에서 식별 및 인증 구현

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.

## 권한이 있는 사용자

권한있는 사용자는 WebSphere MQ에 대한 전체 관리 권한이 있는 사용자입니다.

다음 표에 나열된 사용자 외에도 큐에 대한 `+crt` 권한이 있는 모든 그룹의 사용자가 간접적 관리자입니다. 이와 유사하게 큐 관리자에 대한 `+set` 권한이 있으며 명령 큐에 대한 `+put` 권한이 있는 모든 사용자가 관리자입니다.

이러한 권한을 일반 사용자 및 애플리케이션에 부여해서는 안됩니다.

표 *13.* 플랫폼별 권한이 있는 사용자.

권한이 있는 사용자의 표. Windows에서 SYSTEM, mqm 그룹의 모든 구성원 및 관리자 그룹의 모든 구성원은 권한있는 사용자입니다. UNIX and Linux 시스템에서 mqm 그룹의 모든 구성원은 권한이 있는 사용자입니다. IBM i에서 프로파일(사용자) qmqm 및 qmqmadm, qmqmadm 그룹의 모든 구성원 및 *ALLOBJ 설정이 정의된 모든 사용자는 권한이 있는 사용자입니다.

| 플랫폼 | 권한이 있는 사용자 |
| --- | --- |
| Windows 시스템 | • SYSTEM<br>• mqm 그룹의 구성원<br>• 관리자 그룹의 구성원 |

표 *13.* 플랫폼별 권한이 있는 사용자.

권한이 있는 사용자의 표. Windows에서 SYSTEM, mqm 그룹의 모든 구성원 및 관리자 그룹의 모든 구성원은 권한이 있는 사용자입니다. UNIX and Linux 시스템에서 mqm 그룹의 모든 구성원은 권한이 있는 사용자입니다. IBM i에서 프로파일(사용자) qmqm 및 qmqmadm, qmqmadm 그룹의 모든 구성원 및 *ALLOBJ 설정이 정의된 모든 사용자는 권한이 있는 사용자입니다.

*(계속)*

| 플랫폼 | 권한이 있는 사용자 |
| --- | --- |
| UNIX and Linux 시스템 | • mqm 그룹의 구성원 |

## MQCSP 구조를 사용하여 사용자 식별 및 인증

MQCONNX 호출에서 MQCSP 연결 보안 매개변수 구조를 지정할 수 있습니다.

MQCSP 연결 보안 매개변수 구조에는 권한 서비스가 사용자를 식별하고 인증하는 데 사용할 수 있는 사용자 ID 및 비밀번호가 포함되어 있습니다.

IBM WebSphere MQ에서 제공되는 권한 서비스 컴포넌트는 오브젝트 권한 관리자(OAM)라고 합니다. OAM은 MQCSP에 포함된 ID를 기반으로 사용자에게 권한을 부여하지만 비밀번호의 유효성을 검증하지 않습니다. OAM과 함께 체인 연결된 엑시트를 사용하거나 OAM을 대체 권한 서비스로 바꿔서 권한 서비스에서 비밀번호 유효성 검증을 구현할 수 있습니다.

보안 엑시트에서 MQCSP를 변경할 수 있습니다.

## 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 쪽에서 MCA는 일반적으로 WebSphere MQ MQI 클라이언트 애플리케이션의 사용자 대신 작동합니다. 이 상황에서는, 상호 인증이 두 큐 관리자 사이나 큐 관리자와 WebSphere MQ MQI 클라이언트 애플리케이션의 사용자 사이에서 실제로 일어납니다.

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 95 페이지의 『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.

SSL(Secure Sockets Layer) 및 TLS(Transport Layer Security)는 방금 설명한 것과 같은 PKI 기술을 사용합니다. SSL(Secure Sockets Layer)의 인증 수행에 대한 자세한 정보는 14 페이지의 『SSL(Secure Socket Layer) 및 TLS(Transport Layer Security) 개념』을 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 206 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 68 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 136 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.

## 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.

SSL 데이터 교환 중 통신하고 있는 파트너가 디지털 인증서를 이용해 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.

UNIX, Linux 및 윈도우 시스템에서 WebSphere MQ SSL 지원은 OCSP (Online Certificate Status Protocol) 를 사용하거나 LDAP (Lightweight Directory Access Protocol) 서버에서 CRL및 ARL을 사용하여 해지된 인증서를 확인합니다. OCSP가 기본 메소드입니다. IBM WebSphere MQ classes for Java 및 IBM WebSphere MQ classes for JMS 는 클라이언트 채널 정의 테이블 파일에서 OCSP 정보를 사용할 수 없습니다. 그러나 온라인 인증서 프로토콜 사용 절에서 설명하는 대로 OCSP를 구성할 수 있습니다.

z/Os및 IBM i WebSphere MQ SSL 지원은 LDAP 서버에서만 CRL및 ARL을 사용하여 해지된 인증서를 확인합니다.

인증서의 자세한 정보

권한은 9 페이지의 『디지털 인증서』의 내용을 참조하십시오.

### 폐기된 인증서 및 OCSP

IBM WebSphere MQ는 사용할 OCSP(Online Certificate Status Protocol) 응답자를 판별하고, 수신한 응답을 핸들링합니다. OCSP 응답자에 액세스할 수 있도록 하는 단계를 수행해야 할 수도 있습니다.

**참고:** This information applies only to WebSphere MQ on 윈도우, UNIX and Linux systems.

OCSP를 사용하여 디지털 인증서의 폐기 상태를 점검하려면 WebSphere MQ가 두 가지 방법을 사용하여 접속할 OCSP 응답자를 판별하면 됩니다.

- 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 또는 SSL 인증서의 해지 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.

**보통**
인증서가 올바릅니다.

**해지됨**
인증서가 폐기되었습니다.

**알 수 없음**
이 결과는 세 가지 중 하나의 이유로 발생할 수 있습니다.

- IBM WebSphere MQ가 OCSP 응답자에 액세스할 수 없습니다.
- OCSP 응답자가 응답을 송신했으나 WebSphere MQ가 응답의 디지털 서명을 확인할 수 없습니다.
- OCSP 응답자가 인증서에 대한 폐기 데이터를 가지고 있지 않음을 표시하는 응답을 송신했습니다.

IBM WebSphere MQ에서 알 수 없음의 OCSP 결과를 수신하는 경우 이에 따른 작동은 OCSPAuthentication 속성의 설정에 따라 다릅니다. 큐 관리자의 경우 이 속성은 `qm.ini` 파일의 SSL 스탠자(UNIX and Linux 시스템의 경우) 또는 Windows 레지스트리에 보유됩니다. IBM WebSphere MQ 탐색기를 사용하여 설정될 수 있습니다. 클라이언트의 경우, 이 속성은 클라이언트 구성 파일의 SSL 스탠자에 보유됩니다.

알 수 없음 결과가 수신되고 OCSPAuthentication이 REQUIRED(기본값)로 설정되면 WebSphere MQ는 연결을 거부하고 오류 메시지 유형 AMQ9716을 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 결과가 수신되고 OCSPAuthentication이 OPTIONAL로 설정된 경우 WebSphere MQ는 SSL 채널이 시작되도록 허용하며 경고나 SSL 이벤트 메시지는 생성되지 않습니다.

알 수 없음 결과를 수신하고 OCSPAuthentication이 WARN으로 설정된 경우, SSL 채널은 시작되지만 IBM WebSphere 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. SSL 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 SSL 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. 120 페이지의 『UNIX, Linux, and Windows 시스템에서 CA 인증서(또는 자체 서명된 인증서의 공용 부분)를 키 저장소에 추가』의 설명을 참조하십시오. CA 인증서가 추가되면, 기본적으로 신뢰 루트로서 추가되며 이는 이 컨텍스트에서 필수 설정입니다. 이 인증서가 추가되지 않으면 WebSphere MQ는 OCSP 응답에 대한 디지털 서명을 확인할 수 없고 OCSP 검사 결과를 알 수 없는 결과로 확인할 수 없는 경우, OCSPAuthentication의 값에 따라 IBM WebSphere MQ가 채널을 닫게 할 수 있습니다.

## Java 및 JMS 클라이언트 애플리케이션에서의 OCSP(Online Certificate Status Protocol)

Java API의 제한사항으로 인해, WebSphere MQ는 전체 JVM(Java virtual machine) 프로세스에 대해 OCSP(Online Certificate Status Protocol)가 사용되는 경우에만 SSL 및 TLS 보안 소켓에 대해 OCSP 인증서 해지 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.
- Java 보안 관리자 정책에 적용되는 조건으로 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. |

| 특성 이름 | 설명 |
| --- | --- |
| 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. 일부의 경우 JVM이 SSL 또는 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 파일이 덮어쓰여지지 않도록 Java 임시 수정사항 및 제품 유지보수를 적용할 때 주의하십시오. 유지보수를 적용한 후 java.security 변경사항을 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.
- 고유 인터셉터에서 OCSP 검사 사용 설정에서 설명한 AMS Java 인터셉터를 사용 중인 경우, 키 저장소 구성 파일에서 AMS OCSP 구성과 충돌하는 java.security OCSP 구성은 사용하지 마십시오.

## 인증서 폐기 목록(CRL) 및 권한 취소 목록(ARL) 관련 작업

CRL 및 ARL에 대한 WebSphere MQ의 지원은 플랫폼별로 다양합니다.

각 플랫폼의 CRL 및 ARL 지원은 다음과 같습니다.

- z/OS에서 시스템 SSL은 Tivoli® Public Key Infrastructure 제품에 의해 LDAP 서버에 저장된 CRL 및 ARL을 지원합니다.
- 다른 플랫폼에서 CRL 및 ARL 지원은 PKIX X.509 V2 CRL 프로파일 권장사항을 준수합니다.

WebSphere MQ가 지난 12시간 동안 액세스되었던 CRL 및 ARL의 캐시를 유지보수합니다.

큐 관리자나 WebSphere MQ MQI 클라이언트가 인증서를 수신하면, 인증서가 아직 유효한지 확인하기 위해 CRL을 점검합니다. 캐시가 있으면, WebSphere MQ가 먼저 캐시를 점검합니다. CRL이 캐시에 없으면 WebSphere MQ가 사용 가능한 CRL을 찾을 때까지 *SSLCRLNamelist* 속성에 지정된 인증 정보 오브젝트의 이름 목록에 발생하는 순서대로 WebSphere MQ가 LDAP CRL 서버 위치를 조사합니다. 이름 목록이 지정되어 있지 않거나, 빈 값으로 지정되어 있으면, CRL이 검사되지 않습니다.

LDAP에 대한 자세한 정보는 Using lightweight directory access protocol services with WebSphere MQ for 윈도우의 내용을 참조하십시오.

### *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.

WebSphere MQ SSL과 함께 사용하려면 이러한 속성에 대한 2진 데이터가 DER(Definite Encoding Rules) 형식을 준수해야 합니다. LDIF 파일에 대한 자세한 정보는 LDAP 서버와 함께 제공된 문서를 참조하십시오.

141 페이지의 그림 12 는 CA1이 발행한 CRL및 ARL을 로드하기 위해 LDAP 서버에 입력으로 작성할 수 있는 샘플 LDIF 파일을 표시합니다. 이는 IBM의 테스트 조직에서 식별 이름이 "CN=CA1, OU=Test, O=IBM"인 가상 인증 기관입니다.

```
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 *12.* 인증 기관의 샘플 *LDIF* 파일 이는 구현에 따라 달라질 수 있습니다.

141 페이지의 그림 13에서는 141 페이지의 그림 12에 표시된 샘플 LDIF 파일을 로드할 때 LDAP 서버가 작성하는 DIT 구조를, 마찬가지로 IBM 내부의 PKI 조직에서 설정한 가상의 인증 기관인 CA2에 대한 유사한 파일과 함께 표시합니다.

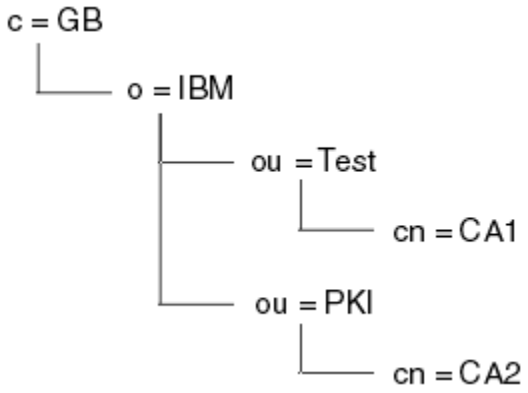



그림 *13. LDAP* 디렉토리 정보 트리*(DIT)* 구조의 예

WebSphere MQ 는 CRL및 ARL을 모두 검사합니다.

**참고:** LDAP 서버의 액세스 제어 목록을 사용하여 권한이 부여된 사용자가 CRL과 ARL을 보유하는 항목을 읽고 검색하고 비교할 수 있는지 확인하십시오. WebSphere MQ 는 AUTHINFO 오브젝트의 LDAPUSER및 LDAPPWD 등록 정보를 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. 141 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.

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 이름 목록에 보유되어 있으며 이 이름 목록은 *SSLCRLNamelist* 큐 관리자 속성에 지정되어 있습니다.

다음 예에서는 매개변수를 지정하는 데 MQSC가 사용됩니다.

1. AUTHTYPE 매개변수를 CRLLDAP으로 설정한 DEFINE AUTHINFO MQSC 명령을 사용하여 인증 정보 오브젝트를 정의하십시오.

   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, 이들이 가리키는 LDAP 서버는 동일한 정보를 가져야 합니다. 이렇게 하면 하나 이상의 LDAP 서버가 실패하는 경우에 서비스의 지속성을 제공합니다.

   모든 플랫폼에서 사용자 ID와 비밀번호는 암호화되지 않은 LDAP 서버에 전송됩니다.
2. DEFINE NAMELIST MQSC 명령을 사용하여, 인증 정보 오브젝트의 이름에 이름 목록을 정의하십시오.
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 *SSLCRLNamelist*라는 큐 관리자 속성을 설정합니다. 이 속성에 대한 큐 관리자의 초기값은 공백입니다.

최대 10개의 대체 LDAP 서버에 대한 연결을 이름 목록에 추가하여 하나 이상의 LDAP 서버가 실패하는 경우에 서비스의 지속성을 보장할 수 있습니다. 여러 LDAP 서버가 반드시 동일한 정보를 가지고 있어야 한다는 점에 유의하십시오.

### *IBM WebSphere MQ Explorer* 를 사용하여 *CRL* 및 *ARL* 액세스

IBM WebSphere 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 이름(142 페이지의 『2.a』 단계로부터)을 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(142 페이지의 『3.a』 단계로부터)을 입력하십시오.

### *IBM WebSphere MQ MQI* 클라이언트를 사용하여 *CRL*및 *ARL* 액세스

IBM WebSphere MQ MQI 클라이언트를 사용하여 확인하기 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 Directory 사용(Active Directory가 지원되는 Windows 시스템에서)

자세한 내용은 관련 정보를 참조하십시오.

10개까지의 연결을 대체 LDAP 서버에 포함시켜 하나 이상의 LDAP 서버가 실패할 경우 서비스의 지속성을 보장할 수 있습니다. 여러 LDAP 서버가 반드시 동일한 정보를 가지고 있어야 한다는 점에 유의하십시오.

Linux ( zSeries 플랫폼) 에서 실행 중인 WebSphere MQ MQI 클라이언트 채널에서 LDAP CRL에 액세스할 수 없습니다.

*OCSP* 응답자 및 *CRL*을 보유하는 *LDAP* 서버의 위치

IBM WebSphere MQ MQI 클라이언트 시스템에서 인증 취소 목록 (CRL) 을 보유하는 LDAP (Lightweight Directory Access Protocol) 서버 및 OCSP 응답자의 위치를 지정할 수 있습니다.

여기에 나와 있는 세 가지 방식으로(아래로 갈수록 우선순위가 낮아짐) 이 위치를 지정할 수 있습니다.

### WebSphere MQ MQI 클라이언트 애플리케이션이 MQCONNX 호출을 발행하는 경우

**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 WebSphere MQ 클라이언트가 CRL을 보유하는 LDAP 서버 또는 OCSP 응답자에 액세스하는 데 필요한 모든 정보가 들어 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

WebSphere MQ MQI 클라이언트가 CRL을 보유하고 있는 LDAP 서버 또는 OCSP 응답자에 액세스할 수 있도록 하기 위해 하나 이상의 인증 정보 오브젝트 속성을 클라이언트 채널 정의 테이블에 포함시킬 수 있습니다.

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.

AUTHTYPE(OCSP)의 인증 정보 오브젝트는 IBM i 또는 z/OS 큐 관리자에 사용하도록 적용되지 않지만, 클라이언트가 사용하도록 클라이언트 채널 정의 테이블(CCDT)에 복사될 플랫폼에는 지정될 수 있습니다.

WebSphere MQ MQI 클라이언트가 OCSP 응답자 또는 CRL을 보유하는 LDAP 서버에 액세스할 수 있도록 하기 위해, 하나 이상의 인증 정보 오브젝트의 속성이 클라이언트 채널 정의 테이블에 포함될 수 있습니다. 다음 방법 중 하나로 이러한 속성을 포함할 수 있습니다.

**서버 플랫폼에서 AIX, HP-UX, Linux, Solaris및 윈도우**

하나 이상의 인증 정보 오브젝트의 이름을 포함하는 이름 목록을 정의할 수 있습니다. 그런 다음 **SSLCRLNameList** 큐 관리자 속성을 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. 어떤 LDAP 서버가 필요할 때 사용 불가능하면 WebSphere MQ MQI 클라이언트는 다른 LDAP 서버에 액세스를 시도할 수 있습니다.

이름 목록으로 식별되는 인증 정보 오브젝트의 속성이 여기에서는 집합적으로 인증서 폐기 위치로 참조됩니다. 큐 관리자 속성, **SSLCRLNameList**를 이름 목록의 이름으로 설정하면 인증서 폐기 위치가 큐 관리자와 연관된 클라이언트 채널 정의 테이블로 복사됩니다. 클라이언트 시스템에서 공유 파일로서 CCDT에 액세스할 수 있거나 그런 다음 CCDT가 클라이언트 시스템으로 복사되는 경우, 해당 시스템의 WebSphere MQ MQI 클라이언트는 CCDT에 있는 인증서 취소 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, **SSLCRLNameList**가 공백으로 설정된 경우에는 인증서 폐기 위치가 CCDT에서 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.

## Windows의 Active Directory 사용

Windows 시스템에서는 **setmqcrl** 제어 명령을 사용하여 Active Directory에 클라이언트 연결 채널 정의를 발행할 수 있습니다.

**setmqcrl** 명령은 OCSP 정보를 공개하지 않습니다.

이 명령 및 해당 구문에 대한 자세한 정보는 setmqcrl의 내용을 참조하십시오.

## *Java*용 *IBM WebSphere MQ* 클래스 및 *JMS*의 *IBM WebSphere MQ* 클래스를 사용하여 *CRL* 및 *ARL* 액세스

Java및 IBM WebSphere MQ 클래스의 IBM WebSphere MQ 클래스는 다른 플랫폼과 다르게 CRL로 액세스합니다.

Java용 IBM WebSphere MQ 클래스를 사용하여 CRL및 ARL 작업에 대한 정보는 인증서 폐기 목록 사용을 참조하십시오.

JMS에 대한 IBM WebSphere MQ 클래스를 사용하여 CRL및 ARL 작업에 대한 정보는 SSLCERTSTORES 오브젝트 특성을 참조하십시오.

## 인증 정보 오브젝트 조작

MQSC 또는 PCF 명령 또는 IBM WebSphere 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 전체 설명은 Script (MQSC) 명령 을 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 명령에 대한 자세한 설명은 프로그램 가능 명령 형식 정의 를 참조하십시오.

사용 가능한 플랫폼에서 WebSphere MQ Explorer를 사용할 수도 있습니다.

# 오브젝트에 액세스 권한 부여

이 절에는 오브젝트에 대한 액세스를 제어하기 위해 오브젝트 권한 관리자 및 채널 엑시트 프로그램 사용에 대한 정보가 포함되어 있습니다.

UNIX, 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. 또한 시스템에 보안을 적용하기 위해 수행하는 태스크를 판별하기 위해 사용할 수 있는 체크리스트 및 사용자가 IBM WebSphere MQ를 관리하고 IBM WebSphere MQ 오브젝트를 관리하기 위한 권한을 부여하기 위한 고려사항이 포함됩니다. 제공된 보안 메커니즘이 사용자의 필요를 충족시키지 않은 경우 사용자 고유의 채널 엑시트 프로그램을 개발할 수 있습니다.

## UNIX, Linux 및 Windows 시스템에서 OAM을 사용하여 오브젝트에 대한 액세스 제어

오브젝트 권한 관리자(OAM)는 WebSphere MQ 오브젝트에 권한을 부여하고 취소하기 위한 명령행 인터페이스를 제공합니다.

180 페이지의 『UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 관리 권한』 에서 설명한 바와 같이 이러한 명령을 사용할 적합한 권한을 가지고 있어야 합니다. WebSphere MQ를 관리하도록 권한 부여된 사용자 ID는 큐 관리자에 대해 수퍼 사용자 권한을 가집니다. 이는 MQI 요청이나 명령을 실행하기 위해 추가 권한이 부여될 필요가 없음을 의미합니다.

### UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 오브젝트에 대한 액세스 부여

**setmqaut** 제어 명령 또는 **MQCMD_SET_AUTH_REC** PCF 명령을 사용하여 사용자 및 사용자 그룹을 IBM WebSphere MQ 오브젝트에 액세스할 수 있습니다.

**setmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 setmqaut를 참조하고 **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 프로파일로 지정합니다. 이는 오브젝트의 명시적 이름이거나 와일드카드 문자가 포함된 일반 이름입니다. 일반 프로파일에 대한 자세한 설명과 일반 프로파일에서 와일드카드 문자 사용에 대해서는 146 페이지의 『UNIX, Linux, and Windows 시스템에서 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 시스템에서는 도메인 이름으로 사용자 ID를 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. UNIX 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
```

## 다른 권한 서비스로 명령 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.

# UNIX, Linux, and Windows 시스템에서 OAM 일반 프로파일 사용

OAM 일반 프로파일을 사용하면 작성 시 각 개별 오브젝트에 대해 별도의 **setmqaut** 명령을 실행하지 않고 사용자가 한 번에 여러 오브젝트에 대해 갖는 권한을 설정할 수 있습니다.

**setmqaut** 명령에 일반 프로파일을 사용하면 그 프로파일에 해당되는 모든 오브젝트에 대한 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 지정할 경우 해당 프로파일에 부여하는 권한은 이름이 ABC.DEF, ABC.CEF, ABC.BEF 등인 오브젝트에 적용됩니다.

사용 가능한 와일드카드 문자는 다음과 같습니다.

**?**

단일 문자 대신 물음표(?) 사용. 예를 들어, AB.?D는 AB.CD, AB.ED 및 AB.FD 등의 오브젝트에 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.*.JKL 는 ABC.DEF.JKL및 ABC.GHI.JKL오브젝트에 적용됩니다. (ABC.JKL에는 적용되지 **않는** 점을 주지하십시오. 이 컨텍스트에서 사용된 *는 항상 하나의 규정자를 표시합니다).
- 오브젝트 이름에 있는 규정자 내에서 0개 또는 그 이상의 문자와 일치하는 프로파일 이름의 규정자 문자.

  예를 들어, ABC.DE*.JKL 는 ABC.DE.JKL, ABC.DEF.JKL및 ABC.DEGH.JKL 오브젝트에 적용됩니다.

******

다음과 같이 프로파일 이름에서는 이중 별표(**)를 **한 번만** 사용하십시오.

- 모든 오브젝트 이름과 일치하는 전체 프로파일 이름. 예를 들어 -t prcs를 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, **.ABC 는 최종 규정자 ABC를 사용하여 모든 오브젝트를 식별합니다.

**참고:** UNIX 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 프로파일과 일치하는 이름을 가진 프린시펄 fred에 대한 모든 큐에 대한 권한을 부여한다. 두 번째는 프로파일 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 위의 예에서 큐 AB.CD는 **Get** 권한을 가집니다 (AB.C*가 AB.*보다 더 특정함).

일반 문자를 비교할 때 특이성의 순서는 다음과 같습니다.

1. ?
2. *
3. **

## 프로파일 설정 버리기

**dmpmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 dmpmqaut를 참조하고 **MQCMD_INQUIRE_AUTH_RECS** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 조회를 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:** UNIX and Linux 사용자는 **dmpmqaut** 명령에 -p 옵션을 사용할 수 있지만, 권한을 정의할 때는 -g groupname을 대신 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 프로파일 a.b에 대한 모든 권한 레코드를 덤프합니다. *, 큐에 넣을 수 있다.

```
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:** For WebSphere MQ for 윈도우 only, all principals displayed include domain information, for example:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

### *OAM* 프로파일에서 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. (ABC.JKL에는 적용되지 **않는** 점을 주지하십시오. 이 컨텍스트에서 사용된 *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 prcs를 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.

**참고:** UNIX and Linux 시스템에서 와일드카드 문자를 사용할 때 프로파일 이름을 작은따옴표에 **묶어야 합니다**.

## 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 프로파일과 일치하는 이름을 가진 프린시펄 fred에 대한 모든 큐에 대한 권한을 부여한다. 두 번째는 프로파일 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 위의 예에서 큐 AB.CD는 **Get** 권한을 가집니다 (AB.C*가 AB.*보다 더 특정함).

일반 문자를 비교할 때 특이성의 순서는 다음과 같습니다.

1. ?
2. *
3. **

## 프로파일 설정 버리기

지정된 프로파일과 연관된 현재 권한을 덤프하려면 **dmpmqaut** 제어 명령 또는 **MQCMD_INQUIRE_AUTH_RECS** PCF 명령을 사용하십시오.

**dmpmqaut** 제어 명령 및 해당 구문의 전체 정의에 대해서는 dmpmqaut를 참조하고 **MQCMD_INQUIRE_AUTH_RECS** PCF 명령 및 해당 구문의 전체 정의에 대해서는 권한 레코드 조회를 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:** UNIX and Linux 사용자는 `-p` 옵션을 사용할 수 없습니다. 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 프로파일 a.b에 대한 모든 권한 레코드를 덤프합니다. *, 큐에 넣을 수 있다.

```
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:** Windows용 WebSphere MQ에서만 표시된 모든 프린시펄에 도메인 정보가 포함됩니다. 예를 들면, 다음과 같습니다.

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

## 액세스 설정 표시

특정 프린시펄 또는 그룹이 특정 오브젝트에 대해 갖는 권한을 보려면 **dspmqaut** 제어 명령 또는 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 명령을 사용하십시오.

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 dmpmqaut를 참조하고 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 명령 및 해당 구문의 전체 정의에 대해서는 엔티티 권한 조회를 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
```

## IBM WebSphere MQ 오브젝트에 대한 액세스 변경 및 취소

사용자 또는 그룹이 오브젝트에 대해 갖는 액세스 레벨을 변경하려면 **setmqaut** 명령을 사용하십시오. 권한을 가진 그룹의 멤버인 특정 사용자의 액세스를 취소하려면 그룹에서 사용자를 제거하십시오.

사용자를 그룹에서 제거하는 프로세스는 다음에서 설명됩니다.

- 75 페이지의 『Windows에서 그룹 작성 및 관리』
- 77 페이지의 『HP-UX에서 그룹 작성 및 관리』
- 78 페이지의 『AIX에서 그룹 작성 및 관리』
- 79 페이지의 『Solaris에서 그룹 작성 및 관리』
- 80 페이지의 『Linux에서 그룹 작성 및 관리』

.

IBM WebSphere MQ 오브젝트를 작성하는 사용자 ID에는 해당 오브젝트에 대한 전체 제어 권한이 부여됩니다. 로컬 mqm 그룹(또는 Windows 시스템의 관리자 그룹)에서 이 사용자 ID를 제거할 경우 이들 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 구문의 전체 정의는 setmqaut를 참조하고 **MQCMD_INQUIRE_ENTITY_AUTH** PCF 명령 및 구문의 전체 정의에 대해서는 엔티티 권한 조회를 참조하십시오.

Windows에서 사용자 프로파일을 삭제하기 전에 특정 Windows 사용자 계정에 해당하는 OAM 입력 항목을 삭제하십시오. 사용자 계정을 제거한 후에는 OAM 입력 항목을 제거할 수 없습니다.

## UNIX, Linux, and Windows 시스템에서 보안 액세스 검사 방지

모든 보안 검사를 끄려면 OAM을 사용 안함으로 설정하면 됩니다. 이는 테스트 환경에 적합할 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 MQSNOAUT를 설정하십시오(이를 수행하는 경우 나중에 OAM을 추가할 수 없습니다).

  MQSNOAUT 변수 설정의 포함 에 대한 자세한 정보는 환경 변수 를 참조하십시오.
- 서비스를 제거하려면 큐 관리자 구성 파일을 편집하십시오. (이와 같이 하면, 나중에 OAM을 추가할 수 없습니다.)

OAM이 사용 안함으로 설정된 상태에서 setmqaut 또는 dspmqaut를 사용하는 경우에는 다음 사항을 유의하십시오.

- OAM은 지정된 프린시펄 또는 그룹의 유효성을 검증하지 않습니다. 즉, 명령이 올바르지 않은 값을 승인 할 수 있습니다.

- OAM은 보안 검사를 수행하지 않고 모든 프린시펄 및 그룹이 모든 적용 가능한 오브젝트 조작을 수행할 권한이 있음을 나타냅니다.

**경고:** OAM이 제거되면 이를 기존 큐 관리자에 다시 넣을 수 없습니다. 이 는 오브젝트 작성 시 OAM이 제 위치에 있어야 하기 때문입니다. WebSphere MQ OAM이 제거된 후 다시 사용하려면 큐 관리자를 다시 빌드해야 합니다.

**관련 개념**

설치 가능 서비스

# 자원에 대한 필수 액세스 부여

이 주제에서 WebSphere MQ 시스템에 보안을 적용하기 위해 수행할 태스크를 판별할 수 있습니다.

## 이 태스크 정보

이 태스크 중 WebSphere MQ 설치 요소에 적절한 레벨의 보안을 적용하기 위해 필요한 조치를 결정합니다. 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: 153 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: 158 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: 162 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: 163 페이지의 『큐 관리자에서 모든 자원에 전체 관리 액세스 부여』의 내용을 참조하십시오.
6. 사용자 애플리케이션을 큐 관리자에 연결해야 합니까?
   a) 아니오: 164 페이지의 『큐 관리자에 대한 연결성 제거』에 설명된 대로 연결성을 사용 안함으로 설정하십시오.
   b) 예: 164 페이지의 『사용자 애플리케이션이 큐 관리자에 연결할 수 있도록 허용』의 내용을 참조하십시오.

# 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 *14.* 큐 관리자 자원의 서브세트에 부분 관리 액세스 부여

| **사용자가 관리해야 하는 오브젝트 유형** | **수행 조치** |
|---|---|
| 큐 | 154 페이지의 『일부 큐에 제한된 관리 액세스 부여』에 설명된 대로 필수 큐에 부분 관리 액세스를 부여하십시오. |

| 표 *14.* 큐 관리자 자원의 서브세트에 부분 관리 액세스 부여 *(계속)* | |
|---|---|
| **사용자가 관리해야 하는 오브젝트 유형** | **수행 조치** |
| 토픽 | 155 페이지의 『일부 토픽에 제한된 관리 액세스 부여』에 설명된 대로 필수 토픽에 부분 관리 액세스를 부여하십시오. |
| 채널 | 155 페이지의 『일부 채널에 제한된 관리 액세스 부여』에 설명된 대로 필수 채널에 부분 관리 액세스를 부여하십시오. |
| 큐 관리자 | 155 페이지의 『큐 관리자에 제한된 관리 액세스 부여』에 설명된 대로 큐 관리자에 부분 관리 액세스를 부여하십시오. |
| Processes | 156 페이지의 『일부 프로세스에 제한된 관리 액세스 부여』에 설명된 대로 필수 프로세스에 부분 관리 액세스를 부여하십시오. |
| 이름 목록 | 157 페이지의 『일부 이름 목록에 제한된 관리 액세스 부여』에 설명된 대로 필수 이름 목록에 부분 관리 액세스를 부여하십시오. |
| 서비스 | 157 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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
```

- 변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

  **ReqdAction**
  그룹이 수행할 수 있는 조치:

  – UNIX, Linux 및 Windows 시스템에서는 +chg, +clr, +dlt, +dsp 권한을 함께 사용할 수 있습니다. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.

  **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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- 변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

  **ReqdAction**
  그룹이 수행할 수 있는 조치:

  - UNIX, Linux 및 시스템 다음 권한의 조합:+chg,+clr,+crt,+dlt,+dsp. +만 표시됩니다. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.

## 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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- 변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **ObjectProfile**
  권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

  **GroupName**
  액세스가 부여될 그룹의 이름입니다.

  **ReqdAction**
  그룹이 수행할 수 있는 조치:

  - UNIX, Linux 및 시스템 다음 권한의 조합:+chg,+clr,+crt,+dlt,+dsp. +α+ctrlx,+ctrlx 이다. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.

## 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.

### 프로시저
- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction)
MQMNAME('QMgrName')
```

### 결과
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**
대기열 관리자의 이름입니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- UNIX, Linux 및 Windows 시스템에서는 +chg, +clr, +crt, +dlt, +dsp 권한을 함께 사용할 수 있습니다. 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.

## 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.

### 프로시저
- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
```

- 변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- UNIX, Linux 및 Windows 시스템에서는 +chg, +clr, +crt, +dlt, +dsp 권한을 함께 사용할 수 있습니다. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.

## 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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- 변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- UNIX, Linux 및 Windows 시스템에서는 +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp 권한을 함께 사용할 수 있습니다. 권한 +alladm은 +chg +clr +dlt +dsp와 동등합니다.

## 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.

**참고:** 서비스 오브젝트는 z/OS에 존재하지 않습니다.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction)
MQMNAME('QMgrName')
```

## 결과

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**
대기열 관리자의 이름입니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

**ReqdAction**
그룹이 수행할 수 있는 조치:

- UNIX, Linux 및 Windows 시스템에서는 +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp 권한을 함께 사용할 수 있습니다. 권한 +alladm은 +chg +clr +dlt +dsp와 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 *15.* 큐 관리자 자원의 서브세트에 전체 관리 액세스 부여

| **사용자가 관리해야 하는 오브젝트 유형** | **수행 조치** |
|---|---|
| 큐 | 159 페이지의 『일부 큐에 전체 관리 액세스 부여』 에 설명된 대로 필수 큐에 전체 관리 액세스를 부여하십시오. |
| 토픽 | 159 페이지의 『일부 토픽에 전체 관리 액세스 부여』 에 설명된 대로 필수 토픽에 전체 관리 액세스를 부여하십시오. |
| 채널 | 160 페이지의 『일부 채널에 전체 관리 액세스 부여』 에 설명된 대로 필수 채널에 전체 관리 액세스를 부여하십시오. |
| 큐 관리자 | 160 페이지의 『큐 관리자에 전체 관리 액세스 부여』 에 설명된 대로 큐 관리자에 전체 관리 액세스를 부여하십시오. |
| Processes | 161 페이지의 『일부 프로세스에 전체 관리 액세스 부여』 에 설명된 대로 필수 프로세스에 전체 관리 액세스를 부여하십시오. |
| 이름 목록 | 161 페이지의 『일부 이름 목록에 전체 관리 액세스 부여』 에 설명된 대로 필수 이름 목록에 전체 관리 액세스를 부여하십시오. |
| 서비스 | 162 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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.QUEUE.ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM)
MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.TOPIC.ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.CHANNEL.ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.CHANNEL.ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM)
MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.NAMELIST.ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

### 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQADMIN QMgrName.SERVICE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SERVICE.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

### 프로시저

- 마법사 사용:

  a) WebSphere MQ Explorer 네비게이터 분할창에서 큐 관리자를 마우스의 오른쪽 단추로 클릭한 후 **오브젝트 권한 > 역할 기반 권한 추가**를 클릭하십시오.

  역할 기반 권한 추가 마법사가 열립니다.

- UNIX 및 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
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp +inq
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
```

MQ Explorer를 사용하려는 경우에만 SYSTEM.ADMIN.COMMAND.QUEUE 및 SYSTEM.MQEXPLORER.REPLY.MODEL에 대한 특정 권한이 필요합니다.

- 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/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQQUEUE) ID(GroupName) ACCESS(READ)
RDEFINE MQTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQTOPIC) 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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 사용하십시오.

### 프로시저

- 마법사 사용:

  a) WebSphere MQ Explorer 네비게이터 분할창에서 큐 관리자를 마우스의 오른쪽 단추로 클릭한 후 **오브젝트 권한 > 역할 기반 권한 추가**를 클릭하십시오.

  역할 기반 권한 추가 마법사가 열립니다.

- UNIX 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.QUEUE -t queue -g GroupName +dsp +inq +get
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 +conn
```

- Windows 시스템의 경우, UNIX and Linux 시스템과 동일한 명령을 실행하지만 @class대신 프로파일 이름 @CLASS 를 사용하십시오.
- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER('GroupName') AUT(*ALLADM) MQMNAME('QMgrName')
```

- 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

### 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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 171 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 165 페이지의 『큐 관리자에 대한 원격 연결성 보안』에 설명된 대로 원격 연결성을 보안 설정하십시오.

두 경우 모두 171 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.

| 설명 | 수행 조치 |
|---|---|
| 큐를 사용하는 애플리케이션이 있습니다. | 172 페이지의 『큐에 대한 사용자 액세스 제어』 참조 |
| 토픽을 사용하는 애플리케이션이 있습니다. | 177 페이지의 『토픽에 대한 사용자 액세스 제어』 참조 |
| 큐 관리자 오브젝트에서 조회한 애플리케이션이 있습니다. | 178 페이지의 『큐 관리자에서 조사하기 위한 권한 부여』 참조 |
| 프로젝트 오브젝트를 사용하는 애플리케이션이 있습니다. | 178 페이지의 『프로세스에 액세스하기 위한 권한 부여』 참조 |
| 이름 목록을 사용하는 애플리케이션이 있습니다. | 179 페이지의 『이름 목록에 액세스하기 위한 권한 부여』 참조 |

## 큐 관리자에 대한 원격 연결성 보안

SSL 또는 TLS, 보안 엑시트, 채널 인증 레코드 또는 세 메소드의 결합을 사용하여 큐 관리자에 대한 원격 연결성을 보안화할 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 SSL 또는 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 SSL 또는 TLS 사용:
   a) 서버 연결 채널의 MCAUSER를 권한이 없는 사용자 ID로 설정하십시오.
   b) MQCD 구조에서 엑시트로 전달된 SSLPeerNamePtr 및 SSLPeerNameLength 필드에서 수신하는 SSL DN의 값에 따라 MCAUSER 값을 지정하도록 보안 엑시트를 작성하십시오.
3. 고정된 채널 정의 값과 함께 SSL 및 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. SSL 또는 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 사용하여 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 제공하고 싶은 경우에는 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 `Addr`입니다.
   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**

166 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**를 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 134 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. 아래 표시된 명령에서 서버 연결 채널의 이름을 지정하는 경우 아무 영향을 주지 않습니다.

## 프로시저

- MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**를 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 주소 를 참조하십시오.

**관련 참조**

SET CHLAUTH

*MCAUSER* 사용자 *ID*에 클라이언트 확인 사용자 *ID* 맵핑

클라이언트에서 수신한 원래 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** 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(MAP)
MCAUSER(
user)
```

*generic-channel-name*은 액세스를 제어하려는 채널의 이름이거나 채널 이름과 일치하는 와일드카드로서 별표(*) 기호를 포함하는 패턴입니다.
*client-user-name*은 클라이언트가 확인한 사용자 ID입니다.
*user*는 클라이언트 사용자 이름 대신 사용될 사용자 ID입니다.

**관련 참조**
SET CHLAUTH

*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**를 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
) 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 WebSphere MQ 규칙을 따르는 문자열입니다. SSLPEER 값의 WebSphere MQ 규칙을 참조하십시오.
*user*는 지정된 DN을 사용하는 모든 연결에 사용할 사용자 ID입니다.

**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. 아래 표시된 명령에서 서버 연결 채널의 이름을 지정하는 경우 아무 영향을 주지 않습니다.

### 프로시저
MQSC 명령 **SET CHLAUTH**또는 PCF 명령 **Set Channel Authentication Record**를 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 확인 사용자 *ID*에 대한 액세스 차단
채널 인증 레코드를 사용하여 클라이언트 확인 사용자 ID가 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 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 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 클라이언트가 확인한 사용자 ID입니다.

**관련 참조**

SET CHLAUTH

## *SSL* 식별 이름의 액세스 블로킹
채널 인증 레코드를 사용하여 SSL 식별 이름이 채널을 시작하지 못하도록 할 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**를 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')
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 WebSphere MQ 규칙을 따르는 문자열입니다. SSLPEER 값의 WebSphere MQ 규칙을 참조하십시오.

**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**를 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**를 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.

### 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM)  USER('GroupName') AUT(*CONNECT)
```

- z/OS의 경우 다음 명령을 실행하십시오.

```
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)
```

이들 명령은 배치, CICS, IMS 및 채널 시작기(CHIN)에 연결 권한을 부여합니다. 특정 연결 유형을 사용하지 않으면 관련 명령을 생략하십시오.

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

## 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.

| 설명 | Action |
|---|---|
| 애플리케이션은 큐로부터 메시지를 받습니다. | 172 페이지의 『큐로부터 메시지를 가져오는 권한 부여』 참조 |
| 애플리케이션이 컨텍스트를 설정합니다. | 173 페이지의 『컨텍스트를 설정하기 위한 권한 부여』 참조 |
| 애플리케이션이 컨텍스트를 전달합니다. | 174 페이지의 『컨텍스트를 전달하기 위한 권한 부여』 참조 |
| 애플리케이션이 메시지를 클러스터된 큐에 넣습니다. | 225 페이지의 『리모트 클러스터 큐에 메시지를 넣는 권한 부여』 참조 |
| 애플리케이션이 메시지를 로컬 큐에 넣습니다. | 175 페이지의 『메시지를 로컬 큐에 넣기 위한 권한 부여』 참조 |
| 애플리케이션이 메시지를 모델 큐에 넣습니다. | 175 페이지의 『메시지를 모델 큐에 넣기 위한 권한 부여』 참조 |
| 애플리케이션이 메시지를 리모트 큐에 넣습니다. | 176 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.

## 프로시저
- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME('QMgrName')
```

- 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**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **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.

## 프로시저
- UNIX, Linux 및 시스템 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
```

- IBM i의 경우 다음 명령 중 하나를 발행하십시오.
  - 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/OS의 경우 다음 명령 세트 중 하나를 발행하십시오.
  - 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 시스템 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 i의 경우 다음 명령 중 하나를 발행하십시오.
  - 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/OS에서는 다음 명령을 실행하여 ID 컨텍스트 또는 모든 컨텍스트를 전달하십시오.

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

변수 이름에는 다음과 같은 의미가 있습니다.

**QMgrName**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME('QMgrName')
```

- 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**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **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.

## 프로시저

- UNIX, Linux 및 시스템 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 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/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**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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 사용할 경우 로컬 오브젝트에 넣기 위한 권한이 필요합니다. 175 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.

UNIX, Linux및 Windows 시스템에서는 SET AUTHREC 명령을 사용할 수도 있습니다.

## 프로시저

- UNIX, Linux 및 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 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/OS의 경우 다음 명령을 실행하십시오.

```
RDEFINE MQQUEUE  QMgrNameObjectProfile UACC(NONE)
PERMIT QMgrNameObjectProfile CLASS(MQADMIN)
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 큐 공유 그룹의 이름이기도 합니다.

  **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 *16.* 토픽에 대한 사용자 액세스 제어

| **설명** | **Action** |
| --- | --- |
| 애플리케이션이 토픽에 메시지를 발행합니다. | 177 페이지의 『토픽에 메시지를 발행하기 위한 권한 부여』 참조 |
| 애플리케이션이 토픽을 구독합니다. | 177 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.

### 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
  ```

- IBM i의 경우 다음 명령을 실행하십시오.

  ```
  GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('QMgrName')
  ```

- 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**
  대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

  **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.

### 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
  ```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('QMgrName')
```

- 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME('QMgrName')
```

- 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME('QMgrName')
```

- 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**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.

## 프로시저

- UNIX, Linux 및 Windows 시스템의 경우 다음 명령을 실행하십시오.

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
 +all
```

- IBM i의 경우 다음 명령을 실행하십시오.

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- 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**
대기열 관리자의 이름입니다. z/OS에서 이 값은 큐 공유 그룹의 이름이기도 합니다.

**ObjectProfile**
권한을 변경한 오브젝트 또는 일반 프로파일의 이름입니다.

**GroupName**
액세스가 부여될 그룹의 이름입니다.

# UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 관리 권한

IBM WebSphere MQ 관리자는 모든 IBM WebSphere MQ 명령을 사용하고 다른 사용자에게 권한을 부여할 수 있습니다. 관리자가 리모트 큐 관리자에게 명령을 발행할 때 관리자는 리모트 큐 관리자에 대해 필요한 권한이 있어야 합니다. 추가적인 고려사항은 Windows 시스템에 적용됩니다.

IBM WebSphere MQ 관리자는 모든 WebSphere MQ 명령(다른 사용자에게 WebSphere MQ 권한을 부여하는 명령 포함)을 사용할 권한이 있습니다.

IBM WebSphere MQ 관리자가 되려면, *mqm* 그룹(또는 Windows 시스템의 관리자 그룹의 구성원)이라는 특별한 그룹의 구성원이어야 합니다. mqm 그룹은 WebSphere MQ가 설치될 때 자동으로 작성되며 추가 사용자를 그룹에 추가하여 관리를 수행할 수 있도록 합니다. 이 그룹의 모든 구성원은 모든 자원에 대한 액세스 권한을 가집니다. 이 액세스는 사용자를 mqm 그룹에서 제거하고 REFRESH SECURITY 명령을 실행하는 방법으로만 취소할 수 있습니다. 관리자는 제어 명령을 사용하여 WebSphere MQ를 관리할 수 있습니다. 이러한 제어 명령 중 하나가 **setmqaut**이며 이 명령은 권한을 다른 사용자에게 부여하는 데 사용되어 사용자가 WebSphere MQ 자원을 액세스 또는 제어할 수 있게 합니다. 권한 레코드 관리를 위한 PCF 명령은 큐 관리자에서 dsp 및 chg 권한을 부여받은 비관리자가 사용할 수 있습니다. PCF 명령을 사용하여 권한 관리에 대한 자세한 정보는 프로그램 가능한 명령 형식을 참조하십시오.

관리자는 제어 명령 **runmqsc**를 사용하여 MQSC(IBM WebSphere 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. WebSphere MQ Explorer는 PCF 명령을 사용하여 관리 태스크를 수행합니다. 관리자는 로컬 시스템에서 큐 관리자를 관리하기 위해 WebSphere MQ Explorer를 사용할 수 있는 추가 권한이 필요하지 않습니다. IBM WebSphere MQ 탐색기가 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 때의 권한 확인에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 큐 관리자, 큐, 프로세스, 이름 목록 및 인증 정보 오브젝트에 대해 실행되는 PCF 명령에 대해서는 WebSphere MQ 오브젝트에 대한 작업 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.

또한 시스템 SYSTEM 계정에는 WebSphere MQ 자원에 대한 전체 액세스 권한이 있습니다.

UNIX and Linux 플랫폼에서 제품 전용 사용을 위한 mqm의 특수 사용자 ID도 작성됩니다. 권한을 부여받지 못한 사용자에게 사용 가능해서는 안됩니다. 모든 WebSphere MQ 오브젝트는 사용자 ID mqm이 소유합니다.

시스템 관리자 그룹의 구성원은 시스템 계정으로도 큐 관리자를 관리할 수 있습니다. 도메인 내에서 활성인 모든 권한을 가진 사용자 ID를 포함하는 도메인 제어기에 도메인 mqm 그룹을 작성할 수도 있고 이를 로컬 mqm 그룹에 추가할 수 있습니다. **crtmqm**과 같은 일부 명령은 IBM WebSphere MQ 오브젝트에서 권한을 조작하므로 이러한 오브젝트에 대해 작업하기 위한 권한이 필요합니다(다음 절에 설명된 대로). mqm 그룹의 구성원은 모든 오브젝트에 대한 작업 권한을 가지지만, 동일한 이름의 로컬 사용자와 도메인 인증 사용자가 있어서 Windows 시스템에서 권한 부여가 거부되는 상황이 있을 수 있습니다. 이 프로그램은 183 페이지의 『프린시펄 및 그룹』에서 설명합니다.

Windows 버전에서는 사용자가 관리자 그룹의 구성원이 경우에도 특정 운영 체제 기능에 대해 수행할 수 있는 조치를 제한하는 UAC(User Account Control) 기능을 소개합니다. 사용자 ID가 mqm 그룹이 아닌 Administrators 그룹에 있는 경우, 권한 승격된 명령 프롬프트를 사용하여 WebSphere MQ 관리 명령(예: **crtmqm**)을 실행하십시오. 그렇지 않으면, "AMQ7077: 요청된 조작을 수행할 권한이 없습니다" 오류가 생성됩니다. 권한 승격된 명령 프롬프트를 열려면 명령 프롬프트에 대한 시작 메뉴 항목 또는 아이콘을 마우스 오른쪽 단추로 클릭한 후 "관리자로 실행"을 선택하십시오.

다음 작업은 mqm 그룹의 구성원이 아니어도 수행할 수 있습니다.

- PCF 나가기 명령 내에 PCF 명령 또는 MQSC 명령을 발행하는 애플리케이션 프로그램에서 명령을 발행하십시오(채널 이니시에이터를 조작하지 않는 경우). (이 명령은 64 페이지의 『채널 시작기 정의 보호』에 설명되어 있습니다.)
- 애플리케이션 프로그램에서 MQI 호출을 발행하십시오 (MQCONNX 호출에 빠른 경로 바인딩을 사용하지 않는 경우).

- 데이터 유형 구조에 대한 데이터 변환을 수행하는 코드 조각을 작성하려면 `crtmqcvx` 명령을 사용하십시오.
- `dspmq` 명령을 사용하여 큐 관리자를 표시하십시오.
- `dspmqtrc` 명령을 사용하여 WebSphere MQ 형식화된 추적 출력을 표시하십시오.

그룹과 사용자 ID 둘 모두에 12자 제한이 적용됩니다.

일반적으로, UNIX and Linux 플랫폼은 사용자 ID의 길이를 12자로 제한합니다. AIX 버전 5.3에서는 이러한 제한이 약화되었지만, WebSphere MQ는 계속해서 모든 UNIX and Linux 플랫폼에 대해 12자 제한을 적용합니다. 12자를 초과하는 사용자 ID를 사용하면 WebSphere MQ는 이 ID를 `UNKNOWN` 값으로 바꿉니다. 사용자 ID를 `UNKNOWN` 값으로 정의하지 마십시오.

## mqm 그룹 관리

mqm 그룹의 사용자에게는 WebSphere MQ에 대한 전체 관리 권한이 부여됩니다. 이러한 이유로 애플리케이션과 일반 사용자를 mqm 그룹에 등록해서는 안됩니다. mqm 그룹에는 WebSphere MQ 관리자의 계정만 포함되어야 합니다.

이러한 태스크가 다음에 설명되어 있습니다.

- Windows에서 그룹 작성 및 관리
- HP-UX에서 그룹 작성 및 관리
- AIX에서 그룹 작성 및 관리
- Solaris에서 그룹 작성 및 관리
- Linux

도메인 제어기가 Windows 2000 또는 Windows 2003에서 실행하는 경우, 도메인 관리자는 사용할 WebSphere MQ의 특수 계정을 설정할 필요가 있을 수 있습니다. 이 내용은 WebSphere MQ 계정 구성에 설명되어 있습니다.

# UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 오브젝트에 대한 작업 권한

모든 오브젝트는 IBM WebSphere 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 WebSphere MQ에 의해 보호되며 애플리케이션에는 이에 액세스할 수 있는 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 큐에 대해 한 그룹에는 Put과 Get 조작을 둘 다 수행할 수 있도록 허용하고 다른 그룹에는 큐를 찾아보기하는 권한만 (찾아보기 옵션이 포함된 `MQGET`) 허용할 수 있습니다. 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 ID 및 mqm 그룹의 모든 멤버(그리고 Windows 시스템의 로컬 관리자 그룹의 멤버)에게 자동으로 부여됩니다

**관련 개념**

180 페이지의 『UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 관리 권한』
IBM WebSphere MQ 관리자는 모든 IBM WebSphere MQ 명령을 사용하고 다른 사용자에게 권한을 부여할 수 있습니다. 관리자가 리모트 큐 관리자에게 명령을 발행할 때 관리자는 리모트 큐 관리자에 대해 필요한 권한이 있어야 합니다. 추가적인 고려사항은 Windows 시스템에 적용됩니다.

## UNIX, Linux, and Windows 시스템에서 보안 검사가 수행되는 경우

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. 그러면 WebSphere MQ가 사용자 ID에 큐 관리자에 연결할 수 있는 권한이 부여되어 있는지 확인하고 나중에 검사하기 위해 사용자 ID를 보유합니다.

사용자는 WebSphere MQ에 사인온할 필요가 없습니다. WebSphere MQ는 사용자가 기본 운영 체제에 사인온했으며 인증되었다고 가정합니다.

**오브젝트 열기(MQOPEN 또는 MQPUT1 호출)**
WebSphere MQ 오브젝트는 해당 오브젝트를 열고 그에 대한 명령을 실행함으로써 액세스할 수 있습니다. 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.

WebSphere MQ 는 **MQOPEN** 요청에서 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 83 페이지의 『1』에서 설명됩니다.

액세스 검사를 위해 큐 관리자가 사용하는 사용자 ID는 큐 관리자에 연결된 응용프로그램의 운영 환경에서 확보한 사용자 ID입니다. 적합하게 권한이 부여된 애플리케이션은 대체 사용자 ID를 지정하는 **MQOPEN** 호출을 발행할 수 있습니다. 그런 다음 대체 사용자 ID에 대한 액세스 제어 확인이 수행됩니다. 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.

애플리케이션에서 토픽에 구독할 때는 애플리케이션이 구독된 토픽 트리의 지점 및 그 이상에서 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.

## UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ 가 액세스 제어를 구현하는 방법

IBM WebSphere MQ는 오브젝트 권한 관리자를 사용하여 기본 운영 체제가 제공하는 보안 서비스를 사용합니다. IBM WebSphere MQ는 액세스 제어 목록을 작성하고 유지보수하기 위한 명령을 제공합니다.

권한 서비스 인터페이스라고 하는 액세스 제어 인터페이스는 WebSphere MQ의 부분입니다. WebSphere MQ는 오브젝트 권한 관리자*(OAM)*로 알려진 액세스 제어 관리자(권한 서비스 인터페이스에 일치하는)의 구현을 제공

합니다. 이는 사용자가 별도로 지정하지 않는 한 자동으로 설치되고 사용자가 작성하는 각 큐 관리자에서 사용 가능으로 설정됩니다(152 페이지의 『UNIX, 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. 사용자가 올바른 권한을 가진 경우에만 WebSphere MQ 오브젝트에 액세스할 수 있습니다. 145 페이지의 『UNIX, Linux 및 Windows 시스템에서 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 액세스는 mqm 그룹의 사용자로 제한되고 추가적으로 Windows의 경우 관리자 그룹의 사용자 및 SYSTEM ID로 로그인한 사용자로 제한됩니다. 큐에 대한 사용자 액세스는 변경될 수 없습니다.

WebSphere MQ는 액세스 제어 목록을 작성하고 유지보수하기 위한 명령을 제공합니다. 이러한 명령에 대한 자세한 정보는 145 페이지의 『UNIX, Linux 및 Windows 시스템에서 OAM을 사용하여 오브젝트에 대한 액세스 제어』의 내용을 참조하십시오.

WebSphere MQ는 OAM에 프린시펄, 자원 이름 및 액세스 유형이 포함된 요청을 전달합니다. OAM은 유지보수하는 ACL을 기본으로 액세스 권한을 부여하거나 거부합니다. WebSphere MQ는 OAM의 결정을 따릅니다. OAM이 결정하면 WebSphere MQ는 액세스를 허용하지 않습니다.

## UNIX, Linux, and Windows 시스템에서 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 WebSphere 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. (애플리케이션이 큐 관리자에 연결하지 않고 XA 호출을 실행하면 xa_open 호출을 실행한 애플리케이션과 연관된 사용자 ID가 큐 관리자에 의해 권한 검사에 사용됩니다.)

UNIX 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 WebSphere MQ는 시스템으로부터 받은 사용자 ID를 각 메시지의 메시지 헤더(MQMD 구조)에 사용자의 ID로서 전파합니다. 이 ID는 메시지 컨텍스트 정보의 부분이며 185 페이지의 『UNIX, Linux 및 Windows 시스템의 컨텍스트 권한』에 설명되어 있습니다. 애플리케이션은 컨텍스트 정보를 변경할 권한이 부여되지 않는 한 이 정보를 변경할 수 없습니다.

### 프린시펄 및 그룹

프린시펄은 그룹에 속할 수 있습니다. 특정 자원의 액세스 권한을 개인 보다는 그룹에 부여하여 필요한 관리 비용을 줄일 수 있습니다. UNIX and Linux 시스템에서 모든 ACL (Access Control List) 은 그룹을 기반으로 하지만, Windows 시스템에서 ACLS는 사용자 ID및 그룹을 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. 이 프로세스에 대해서는 다음을 참조하십시오.

- Windows에서 그룹 작성 및 관리
- HP-UX에서 그룹 작성 및 관리
- AIX에서 그룹 작성 및 관리
- Solaris에서 그룹 작성 및 관리
- Linux

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 캐시됩니다. 따라서 프린시펄의 그룹 멤버십에 작성하는 모든 변경사항은 MQSC 명령 REFRESH SECURITY(또는 PCF 동등)를 실행하는 경우를 제외하고는 큐 관리자가 재시작될 때까지 인식되지 않습니다.

**UNIX and Linux 시스템**

모든 ACL은 그룹을 기본으로 합니다. 사용자가 특정 자원에 대한 액세스를 부여받은 경우 이 사용자 ID의 1차 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* 그룹에 있는 권한을 변경하여 WebSphere MQ 자원에 대한 액세스를 특정 권한이 없는 사용자에게 부여할 수 있습니다.

"UNKNOWN" 값으로 사용자 ID를 정의하지 마십시오. "UNKNOWN" 값은 사용자 ID가 너무 길 때 사용되므로 임의의 사용자 ID가 UNKNOWN의 액세스 권한을 사용할 수 있습니다.

사용자는 최대 12자를 포함할 수 있고 그룹 이름은 12자를 포함할 수 있습니다.

**Windows 시스템**

ACL은 사용자 ID 및 그룹 둘 모두를 기반으로 합니다. 개별 사용자 ID를 ACL에서도 표시할 수 있는 점을 제외하고는 UNIX 시스템의 경우와 동일하게 검사합니다. 동일한 사용자 ID가 있는 다른 도메인에서는 다른 사용자를 가질 수 있습니다. WebSphere MQ 는 사용자 ID가 도메인 이름으로 규정되도록 허용하므로 이러한 사용자에게 다른 레벨의 액세스 권한을 부여할 수 있습니다.

그룹 이름에는 선택적으로 다음 형식으로 지정된 도메인 이름을 포함할 수 있습니다.

```
GroupName@domain
domain\GroupName
```

글로벌 그룹은 두 가지 경우에만 OAM에 의해 검사됩니다.

1. 큐 관리자 보안 스탠자에는 `GroupModel=GlobalGroups` 설정이 포함됩니다. 보안을 참조하십시오.
2. 큐 관리자가 대체 보안 액세스 그룹을 사용 중입니다. **crtmqm**을 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 WebSphere MQ 오브젝트에 대한 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 가진 도메인 제어기에 의해 인증된 사용자 ID에서 `crtmqm` 명령을 실행할 경우, 로컬 mqm 그룹에 포함되지 않은 동일 이름의 로컬 사용자가 시스템에 있으면 이 명령이 실패합니다.

## *Windows* 보안 *ID(SID)*

WebSphere MQ on 윈도우 uses the SID where it is available. 권한 요청에 Windows SID가 제공되지 않을 경우 WebSphere MQ는 사용자 이름만으로 사용자를 식별하지만 이는 잘못된 권한이 부여되는 결과를 낳을 수 있습니다.

Windows 시스템에서 보안 ID(SID)는 사용자 ID를 보충하는 데 사용됩니다. SID에는 사용자가 정의되어 있는 Windows 보안 계정 관리자(SAM) 데이터베이스의 전체 사용자 계정 세부사항을 설명하는 정보가 포함됩니다. When a message is created on WebSphere MQ for 윈도우, WebSphere MQ stores the SID in the message descriptor. When WebSphere MQ on 윈도우 performs authorization checks, it uses the SID to query the full information from the SAM database. (이 조회가 성공하려면 사용자가 정의된 SAM 데이터베이스에 액세스할 수 있어야 합니다.)

기본적으로 Windows SID가 권한 요청에 제공되지 않을 경우 WebSphere MQ는 사용자 이름만으로 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 않을 경우 올바르지않은 WebSphere MQ 권한이 부여될 수 있습니다. 이러한 문제점을 방지하려면 각 권한 요청에 SID를 포함시키십시오. WebSphere MQ는 SID를 사용하여 사용자 신임 정보를 설정합니다.

모든 권한 요청에 SID가 포함되도록 지정하려면 `regedit`를 사용하십시오. SecurityPolicy를 NTSIDsRequired로 설정하십시오.

## UNIX, Linux 및 Windows 시스템의 대체 사용자 권한

WebSphere MQ 오브젝트에 액세스할 때 사용자 ID가 다른 사용자의 권한을 사용하도록 지정할 수 있습니다. 이를 대체 사용자 권한이라고 하며 WebSphere MQ 오브젝트에 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.

## UNIX, Linux 및 Windows 시스템의 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 제어를 참조하고 컨텍스트와 관련된 메시지 디스크립터 필드의 설명은 MQMD에 대한 개요를 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. MQCD에 있는 필드들의 초기값은 WebSphere MQ 관리자가 작성하는 채널 정의에 의해 판별됩니다. 특히, 필드 중 하나인 *MCAUserIdentifier*의 초기값은 DEFINE CHANNEL 명령에서 MCAUSER 매개변수의 값에 의해 판별되거나, 채널 정의가 다른 방법으로 작성되는 경우에는 MCAUSER에 해당하는 값에 의해 판별됩니다. *MCAUserIdentifier*에는 MCA 사용자 ID의 처음 12바이트가 포함됩니다. MCA 사용자 ID가 공백이 아니면, 이는 MQ 자원에 액세스할 수

있는 권한을 가진 메시지 채널 에이전트가 사용하는 사용자 ID를 지정합니다. MCAUSER가 Windows 플랫폼에서 12자 미만인지 확인하십시오.

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.

IBM i, UNIX, Linux 및 시스템 *MCAUserIdentifier* 의 값이 공백이 아니면 큐 관리자는 *MCAUserIdentifier* 의 값을 MCA가 큐 관리자에 연결한 후에 큐 관리자의 자원에 액세스하려고 시도할 때 권한 검사를 위해 사용자 ID로 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/OS에서는 *MCAUserIdentifier*의 값이 비어 있지 않는 한, 권한 검사를 위해 그것을 사용할 수 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
- 점검에 사용되는 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 일부 예입니다.

- 만일 MQI 채널의 클라이언트 쪽에 보안 엑시트가 없다고 한다면, WebSphere MQ 클라이언트 애플리케이션에 연관된 사용자 ID는 클라이언트 애플리케이션이 MQCONN 호출을 발행할 때 클라이언트 연결 MCA에서 서버 연결 MCA로 플로우됩니다.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. 비슷하게, MQI 채널에서는 채널의 클라이언트 쪽에 보안 엑시트가 WebSphere MQ MQI 클라이언트 애플리케이션에 연관된 사용자 ID를 송신할 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 SSL이 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:** WebSphere MQ v7.1 릴리스 이전에 구성된 보안 엑시트는 업데이트가 필요할 수 있습니다. 자세한 정보는 채널 보안 엑시트 프로그램을 참조하십시오.

### WebSphere MQ OAM(Object Authority Manager) 사용자 인증

WebSphere MQ MQI 클라이언트 연결에서 보안 엑시트를 사용하여 OAM(Object Authority Manager) 사용자 인증에 사용되는 MQCSP 구조를 수정하거나 작성할 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 디스크립터의 *UserIdentifier* 필드에서 사용자 ID를 추출할 수 있고, 만일 대체 사용자 권한을 가지는 경우라면, 큐 관리자에게 클라이언트 대신에 WebSphere MQ 자원에 액세스할 때 권한 검사를 위해 이 사용자 ID를 사용하라고 요청합니다.

채널 정의에서 PUTAUT 매개변수가 CTX(z/OS에서는 ALTMCA)로 설정되어 있으면, 수신되는 각 메시지의 *UserIdentifier* 필드에 있는 사용자 ID가, MCA가 목적지 큐를 열 때 권한 검사를 위해 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 136 페이지의 『메시지 엑시트에서 ID 맵핑』의 내용을 참조하십시오.

## API 엑시트 및 API 교차 엑시트에서 액세스 제어 구현

API 또는 API 교차 엑시트가 WebSphere MQ에 의해 제공되는 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.

# 메시지의 기밀성

기밀성을 유지하려면 메시지를 암호화하십시오. 사용자 요구에 따라 WebSphere MQ에서 메시지를 암호화하는 다양한 메소드가 있습니다.

사용자가 선택한 CipherSpec은 사용자가 가지는 기밀성 레벨을 판별합니다.

포인트-투-포인트 메시징 인프라에 대해 애플리케이션 레벨의 엔드-투-엔드 데이터 보호가 필요한 경우 WebSphere MQ Advanced Message Security를 사용하여 메시지를 암호화하거나 고유 API 엑시트 또는 API 교차 엑시트를 작성할 수 있습니다.

채널을 통해 전송 중인 동안에만 메시지를 암호화해야 하는 경우 큐 관리자에 적절한 보안이 있으므로 SSL 또는 TLS를 사용하거나 고유 보안 엑시트, 메시지 엑시트 또는 송신 및 수신 엑시트 프로그램을 작성할 수 있습니다.

WebSphere MQ Advanced Message Security에 대한 자세한 정보는 58 페이지의 『계획의 대상 Advanced Message Security』을 사용하십시오. WebSphere MQ에서의 SSL 및 TLS 사용은 22 페이지의 『SSL 및 TLS용

IBM WebSphere MQ 지원』에 설명되어 있습니다. 메시지 암호화에서 엑시트 프로그램의 사용은 206 페이지의 『사용자 엑시트 프로그램에서 기밀성 구현』에 설명됩니다.

# SSL 또는 TLS를 사용하여 두 개의 큐 관리자 연결

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 확보하고 관리해야 합니다. 테스트 시스템에서는 자체 서명 인증서 또는 로컬 인증 기관(CA)에서 발행한 인증서를 사용할 수 있습니다. 프로덕션 시스템에서는 자체 서명 인증서를 사용하지 마십시오. 자세한 정보는 ..// zs14140_.dita의 내용을 참조하십시오.

인증서 작성 및 관리에 대한 전체 정보는 104 페이지의 『UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 작업』을 참조하십시오.

이 주제 콜렉션에서는 SSL 통신 설정에 관련된 태스크를 소개하고 이러한 태스크를 완료하는 데 대한 단계별 내용을 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. WebSphere MQ 구현의 경우, SSL 또는 TLS 서버가 항상 클라이언트로부터 인증서를 요청합니다.

**참고사항:**

1. 이 컨텍스트에서는 SSL 클라이언트가 데이터 교환을 시작하는 연결을 의미합니다.
2. 자세한 내용은 용어집을 참조하십시오.

UNIX, Linux 및 Windows 시스템에서 SSL 또는 TLS 클라이언트는 올바른 WebSphere MQ 형식으로 레이블된 인증서가 있는 경우에만 인증서를 송신합니다. MQ형식은 `ibmwebspheremq` 다음에 소문자로 변경된 큐 관리자의 이름이 옵니다. 예를 들어, `QM1`의 경우 `ibmwebspheremqqm1`입니다.

WebSphere MQ는 다른 제품에 대한 인증서와의 혼동을 피하기 위해 레이블에 `ibmwebspheremq` 접두부를 사용합니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). 클라이언트가 인증서를 송신하지 않으면 SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 REQUIRED로 설정된 SSLCAUTH 매개변수 또는 SSLPEER 매개변수 값 세트로 정의된 경우에만 인증에 실패합니다. 익명으로 큐 관리자를 연결하는 데 대한 자세한 정보는(즉, SSL 또는 TLS이 인증서를 송신하지 않는 경우) 192 페이지의 『단방향 인증을 사용하여 두 개의 큐 관리자 연결』의 내용을 참조하십시오.

## 두 개의 큐 관리자의 상호 인증을 위해 자체 서명 인증서 사용

자체 서명 SSL 또는 TLS 인증서를 사용하여 두 큐 관리자 사이의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 안전하게 통신하는 데 필요한 두 개의 큐 관리자 QM1 및 QM2가 있습니다. QM1과 QM2 간에 상호 인증을 수행해야 합니다.
- 자체 서명 인증서를 사용하여 보안 통신을 테스트하기로 결정했습니다.

결과 구성은 다음과 유사합니다.

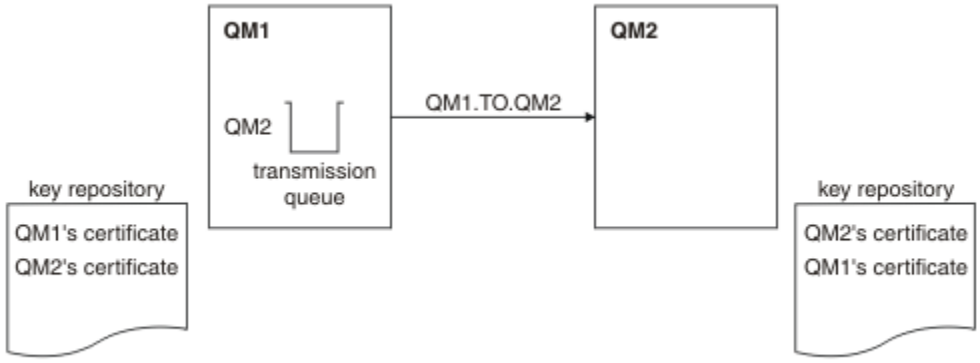



그림 *14.* 이 태스크로부터 나온 구성

189 페이지의 그림 14에서는 QM1용 키 저장소에 QM1에 대한 인증서와 QM2로부터의 공용 인증서가 포함됩니다. QM2용 키 저장소에는 QM2용 인증서와 QM1로부터의 공용 인증서가 포함됩니다.

## 프로시저

1. 운영 체제에 따라 각 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 큐 관리자마다 자체 서명 인증서를 작성하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
3. 각 인증서의 사본을 추출하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. FTP 등의 유틸리티를 사용하여 QM1 인증서의 공용 부분을 QM2 시스템에 전송하거나 반대로 전송하십시오.
5. 큐 관리자마다 파트너 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
6. QM1에서 다음 예와 같은 명령을 발행하여 송신자 채널 및 연관된 전송 큐를 정의하십시오.

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
   SSLCIPH(RC4_MD5_US) DESCR('Sender channel using SSL from QM1 to QM2')

   DEFINE QLOCAL(QM2) USAGE(XMITQ)
   ```

   이 예제에서는 CipherSpec RC4_MD5를 사용합니다. 채널의 각 끝에 있는 CipherSpec은 동일해야 합니다.
7. QM2에서 다음 예와 같은 명령을 발행하여 수신자 채널을 정의하십시오.

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   채널의 이름은 6단계에서 정의한 송신자 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.
8. 채널을 시작합니다.

## 결과

189 페이지의 그림 14에 설명된 대로 키 저장소 및 채널이 작성됩니다.

## 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

큐 관리자 QM2에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

각 경우에, SSLPEER의 값은 2단계에서 작성된 파트너 인증서의 DN의 값과 일치해야 한다. 인증서가 자체 서명되었으므로 발행자 이름이 피어 이름과 일치합니다.

SSLPEER은 선택사항입니다. 지정되면, 파트너 인증서의 DN(2단계에서 지정됨)을 허용할 수 있도록 해당 값을 설정해야 합니다. SSLPEER의 사용에 대한 자세한 정보는 WebSphere MQ SSLPEER값에 대한 규칙 의 내용을 참조하십시오.

## 두 개의 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용

CA 서명 SSL 또는 TLS 인증서를 사용하여 두 큐 관리자 사이의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 안전하게 통신하는 데 필요한 두 개의 큐 관리자 QMA 및 QMB가 있습니다. QMA와 QMB 간에 상호 인증을 수행해야 합니다.
- 추후에 프로덕션 환경에서 이 네트워크를 사용할 계획이어서 시작부터 CA 서명 인증서를 사용하기로 결정했습니다.

결과 구성은 다음과 유사합니다.

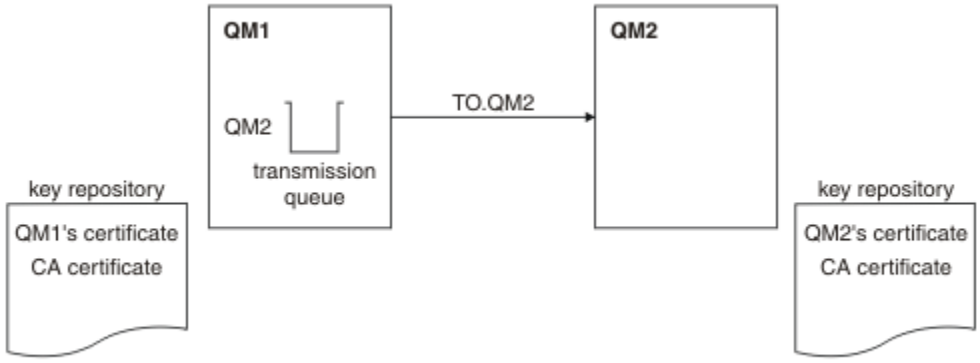



그림 *15.* 이 태스크로부터 나온 구성

191 페이지의 그림 15에서는 QMA용 키 저장소에 QMA의 인증서와 CA 인증서가 포함됩니다. QMB용 키 저장소에는 QMB의 인증서와 CA 인증서가 포함됩니다. 이 예에서는 QMA의 인증서와 QMB의 인증서가 둘 다 동일한 CA에 의해 발행되었습니다. QMA의 인증서와 QMB의 인증서가 다른 CA에 의해 발행된 경우 QMA 및 QMB에 대한 키 저장소에는 두 CA 인증서가 모두 포함되어야 합니다.

## 프로시저

1. 운영 체제에 따라 각 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 큐 관리자마다 CA 서명 인증서를 요청하십시오.
   두 개의 큐 관리자에 대해 서로 다른 CA를 사용할 수도 있습니다.
   - UNIX, Linux 및 Windows 시스템의 경우.
3. 큐 관리자마다 인증 기관 인증서를 키 저장소에 추가하십시오.
   큐 관리자가 다른 인증 기관을 사용 중인 경우, 각 인증 기관에 대한 CA 인증서를 키 저장소 둘 다에 추가해야 합니다.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. 큐 관리자마다 CA 서명 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
5. QMA에서 다음 예와 같은 명령을 발행하여 송신자 채널 및 연관된 전송 큐를 정의하십시오.

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(RC2_MD5_EXPORT) DESCR('Sender channel using SSL from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

   이 예제에서는 CipherSpec RC4_MD5를 사용합니다. 채널의 각 끝에 있는 CipherSpec은 동일해야 합니다.
6. QMB에서 다음 예와 같은 명령을 발행하여 수신자 채널을 정의하십시오.

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
```

   채널의 이름은 6단계에서 정의한 송신자 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.
7. 채널을 시작하십시오.

## 결과

191 페이지의 그림 15에 설명된 대로 키 저장소 및 채널이 작성됩니다.

### 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QMA에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QMB)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QMB)
```

큐 관리자 QMB에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

각 경우에, SSLPEER의 값은 2단계에서 작성된 파트너 인증서의 식별 이름 (DN) 의 값과 일치해야 합니다. 발급자 이름은 4단계에서 추가된 개인 인증서를 서명한 CA 인증서의 주제 DN과 일치합니다.

## 단방향 인증을 사용하여 두 개의 큐 관리자 연결

큐 관리자가 다른 대상에 대해 단방향 인증을 사용하여 연결하도록 하기 위해(즉, SSL 또는 TLS 클라이언트가 인증서를 송신하지 않을 경우) 상호 인증으로 시스템을 수정하려면 이러한 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 두 개의 큐 관리자(QM1 및 QM2)가 190 페이지의 『두 개의 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용』에서와 같이 설정되었습니다.
- QM2에 대해 단방향 인증을 사용하여 연결되도록 QM1을 변경할 수 있습니다.

결과 구성은 다음과 유사합니다.

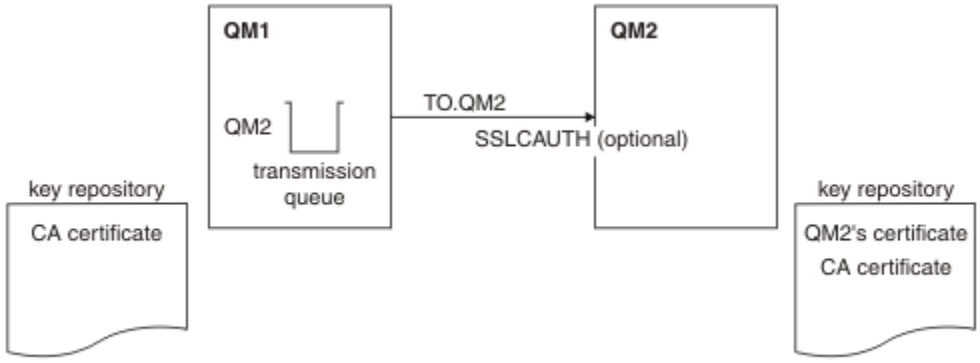



그림 *16.* 단방향 인증을 허용하는 큐 관리자

## 프로시저

1. 운영 체제에 따라 키 저장소에서 QM1의 개인 인증서를 제거하십시오.
   - UNIX, Linux 및 Windows 시스템. 인증서는 다음과 같이 레이블됩니다.
     - `ibmwebspheremq` 다음에 큐 관리자의 이름이 소문자로 접힙니다. 예를 들어, QM1의 경우 `ibmwebspheremqqm1`입니다.
2. 옵션: QM1에서 SSL 또는 TLS 채널이 이전에 실행된 경우 SSL 또는 TLS 환경을 새로 고치십시오.
3. 수신자에서 익명 연결을 허용합니다.

## 결과

193 페이지의 그림 16에 설명된 대로 키 저장소 및 채널이 변경됩니다.

## 다음에 수행할 작업

송신자 채널이 실행되었고 사용자가 REFRESH SECURITY TYPE(SSL) 명령을 발행한 경우(2단계에서), 채널이 자동으로 재시작됩니다. 송신자 채널이 실행되지 않은 경우, 이를 시작하십시오.

채널의 서버 끝에서 채널 상태 표시에 있는 피어 이름 매개변수 값의 존재는 클라이언트 인증서가 플로우되었음을 표시합니다.

일부 DISPLAY 명령을 실행하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

QM1 큐 관리자에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 유사합니다.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

QM2 큐 관리자에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 유사합니다.

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

QM2에서는 SSLPEER 필드가 비어 있습니다. 이는 QM1이 인증서를 송신하지 않았음을 보여줍니다. QM1에서는 SSLPEER의 값이 QM2의 개인 인증서에 있는 DN의 값과 일치합니다.

# 클라이언트를 큐 관리자에 안전하게 연결

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 확보하고 관리해야 합니다. 테스트 시스템에서는 자체 서명 인증서 또는 로컬 인증 기관(CA)에서 발행한 인증서를 사용할 수 있습니다. 프로덕션 시스템에서는 자체 서명 인증서를 사용하지 마십시오. 자세한 정보는 ..// zs14140_.dita의 내용을 참조하십시오.

인증서 작성 및 관리에 대한 전체 정보는 104 페이지의 『UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 작업』을 참조하십시오.

이 주제 콜렉션에서는 SSL 통신 설정에 관련된 태스크를 소개하고 이러한 태스크를 완료하는 데 대한 단계별 내용을 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. WebSphere MQ 구현의 경우, SSL 또는 TLS 서버가 항상 클라이언트로부터 인증서를 요청합니다.

UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 클라이언트는 올바른 WebSphere MQ 형식으로 레이블된 인증서가 있는 경우에만 인증서를 송신합니다. MQ형식은 `ibmwebspheremq` 다음에 소문자로 변경된 로그온 사용자 ID가 옵니다(예: `ibmwebspheremqmyuserid`).

WebSphere MQ는 다른 제품에 대한 인증서와의 혼동을 피하기 위해 레이블에 `ibmwebspheremq` 접두부를 사용합니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). 클라이언트가 인증서를 송신하지 않으면 SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 REQUIRED로 설정된 SSLCAUTH 매개변수 또는 SSLPEER 매개변수 값 세트로 정의된 경우에만 인증에 실패합니다. 익명으로 큐 관리자를 연결하는 데 대한 자세한 정보는 197 페이지의 『클라이언트를 큐 관리자에 익명으로 연결』의 내용을 참조하십시오.

## 클라이언트 및 큐 관리자의 상호 인증을 위해 자체 서명 인증서 사용

자체 서명 SSL 또는 TLS 인증서를 사용하여 클라이언트와 큐 관리자 간의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 현재 안전하게 통신하는 데 필요한 클라이언트 C1과 큐 관리자 QM1이 있습니다. C1과 QM1 간에 상호 인증을 수행해야 합니다.
- 자체 서명 인증서를 사용하여 보안 통신을 테스트하기로 결정했습니다.

IBM i의 DCM은 자체 서명 인증서를 지원하지 않으므로 이 태스크는 IBM i 시스템에서 적용 가능하지 않습니다.

결과 구성은 다음과 유사합니다.

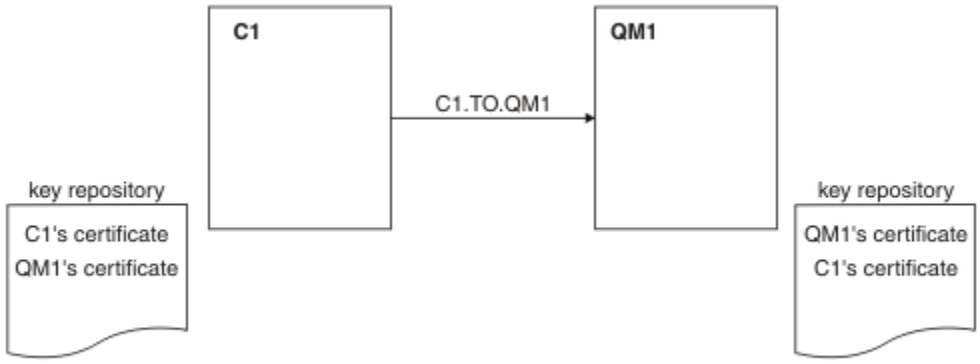



그림 *17.* 이 태스크로부터 나온 구성

195 페이지의 그림 17에서는 QM1용 키 저장소에 QM1에 대한 인증서와 C1로부터의 공용 인증서가 포함됩니다. C1용 키 저장소에는 C1용 인증서와 QM1로부터의 공용 인증서가 포함됩니다.

## 프로시저

1. 운영 체제에 따라 클라이언트 및 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 클라이언트 및 큐 관리자에 대해 자체 서명 인증서를 작성하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
3. 각 인증서의 사본을 추출하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. FTP와 같은 유틸리티를 사용하여 C1 인증서의 공용 부분을 QM1 시스템으로, 그리고 그 반대로 전송하십시오.
5. 클라이언트 및 큐 관리자에 대해 파트너 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
6. 큐 관리자에서 REFRESH SECURITY TYPE(SSL) 명령을 실행하십시오.
7. 다음 방법 중 하나로 클라이언트 연결 채널을 정의하십시오.
   - WebSphere MQ MQI 클라이언트에서 클라이언트 연결 채널 작성에 설명된 대로 C1의 MQSCO 구조와 함께 MQCONNX 호출 사용.
   - 서버 연결 및 클라이언트 연결 정의 작성 에 설명된 대로 클라이언트 채널 정의 테이블을 사용하십시오.
8. QM1에서 다음 예와 같은 명령을 발행하여 서버 연결 채널을 정의하십시오.

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   채널의 이름은 6단계에서 정의한 클라이언트 연결 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.

## 결과

195 페이지의 그림 17에 설명된 대로 키 저장소와 채널이 작성됩니다.

## 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력은 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

채널 정의의 SSLPEER 필터 속성을 설정하는 것은 선택사항입니다. 채널 정의 SSLPEER이 설정된 경우, 이 값은 2단계에서 작성된 파트너 인증서의 주제 DN과 일치해야 합니다. 연결이 성공적으로 완료되면 표시 CHSTATUS 출력의 SSLPEER 필드에 원격 클라이언트 인증서의 주제 DN이 표시된다.

## 클라이언트 및 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용

CA 서명 SSL 또는 TLS 인증서를 사용하여 클라이언트와 큐 관리자 간의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 현재 안전하게 통신하는 데 필요한 클라이언트 C1과 큐 관리자 QM1이 있습니다. C1과 QM1 간에 상호 인증을 수행해야 합니다.
- 추후에 프로덕션 환경에서 이 네트워크를 사용할 계획이어서 시작부터 CA 서명 인증서를 사용하기로 결정했습니다.

결과 구성은 다음과 유사합니다.

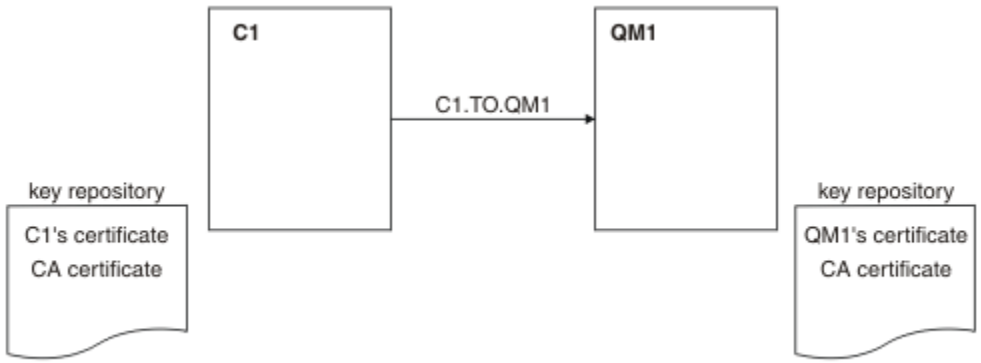



그림 *18.* 이 태스크로부터 나온 구성

196 페이지의 그림 18에서는 C1용 키 저장소에 C1용 인증서와 CA 인증서가 포함됩니다. QM1용 키 저장소에는 QM1용 인증서와 CA 인증서가 포함됩니다. 이 예에서는 C1의 인증서와 QM1의 인증서가 둘 다 동일한 CA에 의해 발행되었습니다. C1의 인증서와 QM1의 인증서가 다른 CA에 의해 발행된 경우 C1 및 QM1에 대한 키 저장소에 두 CA 인증서가 모두 포함되어야 합니다.

### 프로시저

1. 운영 체제에 따라 클라이언트 및 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 클라이언트 및 큐 관리자에 대해 CA 서명 인증서를 요청하십시오.

클라이언트 및 큐 관리자에 대해 서로 다른 CA를 사용할 수도 있습니다.

- UNIX, Linux 및 Windows 시스템의 경우.

3. 클라이언트 및 큐 관리자에 대해 인증 기관 인증서를 키 저장소에 추가하십시오.
   클라이언트 및 큐 관리자가 다른 인증 기관을 사용 중인 경우, 각 인증 기관에 대한 CA 인증서를 키 저장소 둘 다에 추가해야 합니다.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. 클라이언트 및 큐 관리자에 대해 CA 서명 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
5. 다음 방법 중 하나로 클라이언트 연결 채널을 정의하십시오.
   - WebSphere MQ MQI 클라이언트에서 클라이언트 연결 채널 작성에 설명된 대로 C1의 MQSCO 구조와 함께 MQCONNX 호출 사용.
   - 서버 연결 및 클라이언트 연결 정의 작성 에 설명된 대로 클라이언트 채널 정의 테이블을 사용하십시오.
6. QM1에서 다음 예와 같은 명령을 발행하여 서버 연결 채널을 정의하십시오.

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   채널의 이름은 6단계에서 정의한 클라이언트 연결 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.

### 결과

196 페이지의 그림 18에 설명된 대로 키 저장소 및 채널이 작성됩니다.

### 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 출력의 SSLPEER 필드에서는 2단계에서 작성된 원격 클라이언트 인증서의 주체 DN을 보여줍니다. 발급자 이름은 4단계에서 추가된 개인 인증서를 서명한 CA 인증서의 주제 DN과 일치합니다.

## 클라이언트를 큐 관리자에 익명으로 연결

큐 관리자를 익명으로 다른 관리자에 연결하도록 허용하기 위해 상호 인증으로 시스템을 수정하려면 이러한 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 큐 관리자와 클라이언트(QM1 및 C1)가 196 페이지의 『클라이언트 및 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용』 에서와 같이 설정되었습니다.
- C1가 익명으로 QM1에 연결되도록 이를 변경하려고 합니다.

결과 구성은 다음과 유사합니다.

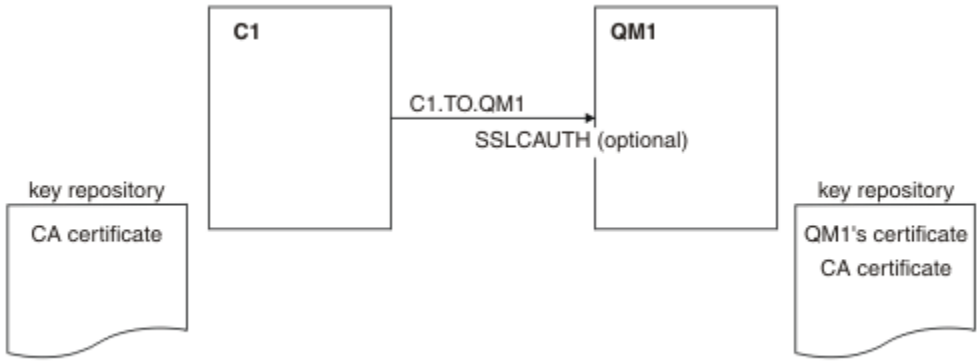



그림 *19.* 익명 연결을 허용하는 클라이언트 및 큐 관리자

## 프로시저

1. 운영 체제에 따라 C1의 키 저장소에서 개인 인증서를 제거하십시오.
   - UNIX, Linux 및 Windows 시스템. 인증서는 다음과 같이 레이블됩니다.
     - `ibmwebspheremq` 다음에 로그온 사용자 ID가 소문자로 접혀 있습니다 (예: `ibmwebspheremqmyuserid`).
2. 클라이언트 애플리케이션을 재시작하거나 클라이언트 애플리케이션을 닫도록 하고 모든 SSL 또는 TLS 연결을 다시 여십시오.
3. 다음 명령을 발행하여 큐 관리자에서 익명 연결을 허용하십시오.

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

## 결과

198 페이지의 그림 19에 설명된 대로 키 저장소 및 채널이 변경됩니다.

## 다음에 수행할 작업

채널의 서버 끝에서 채널 상태 표시에 있는 피어 이름 매개변수 값의 존재는 클라이언트 인증서가 플로우되었음을 표시합니다.

일부 DISPLAY 명령을 실행하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 유사합니다.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

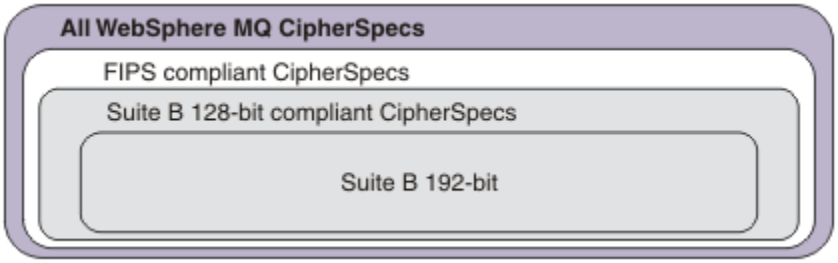
C1이 인증서를 보내지 않았음을 보여주는 SSLCERTI 및 SSLPEER 필드는 비어 있습니다.

## CipherSpec 지정

**DEFINE CHANNEL** MQSC 명령 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec 을 지정하십시오.

IBM WebSphere MQ에 사용할 수 있는 CipherSpec 중 일부는 FIPS를 준수합니다. NULL_MD5와 같이 다른 CipherSpec은 FIPS를 준수하지 않아도 됩니다. 마찬가지로, 일부 FIPS 준수 CipherSpec은 Suite B를 준수하지만 그렇지 않은 경우도 있습니다. 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.




다음 표에 IBM WebSphere MQ SSL 및 TLS 지원과 함께 사용할 수 있는 암호 스펙이 나와 있습니다. 개인 인증서를 요청할 때 공용 및 개인 키 쌍의 키 크기를 지정합니다. SSL 데이터 교환 동안에 사용되는 키 크기는 인증서에 저장된 크기입니다(표에 명시된 것처럼 CipherSpec으로 판별되는 경우는 제외).

| CipherSpec 이름 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 | 암호화 비트 | FIPS[1] | 스위트 B 128 비트 | 스위트 B 192 비트 |
|---|---|---|---|---|---|---|---|
| NULL_MD5 [a] | SSL 3.0 | MD5 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| NULL_SHA [a] | SSL 3.0 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| RC4_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC4 | 40 | 아니오 | 아니오 | 아니오 |
| RC4_MD5_US [a] | SSL 3.0 | MD5 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| RC4_SHA_US [a] | SSL 3.0 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| RC2_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC2 | 40 | 아니오 | 아니오 | 아니오 |
| DES_SHA_EXPORT [2] [a] | SSL 3.0 | SHA-1 | DES | 56 | 아니오 | 아니오 | 아니오 |
| RC4_56_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | RC4 | 56 | 아니오 | 아니오 | 아니오 |
| DES_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | DES | 56 | 아니오 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA [a] | TLS 1.0 | SHA-1 | AES | 128 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA [4] [a] | TLS 1.0 | SHA-1 | AES | 256 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_DES_CBC_SHA [a] | TLS 1.0 | SHA-1 | DES | 56 | No[5] | 아니오 | 아니오 |

| CipherSpec 이름 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 | 암호화 비트 | FIPS[1] | 스위트 B 128 비트 | 스위트 B 192 비트 |
|---|---|---|---|---|---|---|---|
| FIPS_WITH_DES_CBC_SHA [b] | SSL 3.0 | SHA-1 | DES | 56 | No[6] | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 [b] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 [b] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 128 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 256 | 예 | 아니오 | 아니오 |
| ECDHE_ECDSA_RC4_128_SHA256 [b] | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| ECDHE_RSA_RC4_128_SHA256 [b] | TLS 1.2 | SHA_1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 128 | 예 | 아니오 | 아니오 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 [b] | TLS 1.2 | SHA-384 | AES | 256 | 예 | 아니오 | 아니오 |
| ECDHE_RSA_AES_128_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 128 | 예 | 아니오 | 아니오 |
| ECDHE_RSA_AES_256_CBC_SHA384 [b] | TLS 1.2 | SHA-384 | AES | 256 | 예 | 아니오 | 아니오 |
| ECDHE_ECDSA_AES_128_GCM_SHA256 [b] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | 예 | 예 | 아니오 |
| ECDHE_ECDSA_AES_256_GCM_SHA384 [b] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | 예 | 아니오 | 예 |
| ECDHE_RSA_AES_128_GCM_SHA256 [b] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | 예 | 아니오 | 아니오 |
| ECDHE_RSA_AES_256_GCM_SHA384 [b] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_NULL_SHA256 [b] | TLS 1.2 | SHA-256 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| ECDHE_RSA_NULL_SHA256 [b] | TLS 1.2 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| ECDHE_ECDSA_NULL_SHA256 [b] | TLS 1.2 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| TLS_RSA_WITH_NULL_NULL [b] | TLS 1.2 | 없음 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| TLS_RSA_WITH_RC4_128_SHA256 [b] | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |

| CipherSpec 이름 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 | 암호화 비트 | FIPS[1] | 스위트 B 128 비트 | 스위트 B 192 비트 |
|---|---|---|---|---|---|---|---|

**참고:**

1. FIPS 인증 플랫폼에서 CipherSpec이 FIPS 인증 CipherSpec인지 여부를 지정합니다. FIPS에 대한 설명은 FIPS(Federal Information Processing Standards)를 참조하십시오.
2. 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.
3. 데이터 교환 키 크기는 1024비트입니다.
4. 탐색기가 사용하는 JRE에 적절한 제한 없는 정책 파일이 적용되지 않은 경우 이 CipherSpec을 사용하여 WebSphere MQ 탐색기에서 큐 관리자로의 연결을 보호할 수 없습니다.
5. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다.
6. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다. 이름 FIPS_WITH_DES_CBC_SHA은(는) 히스토리이며 이 CipherSpec이 이전에(더 이상) FIPS를 준수하지 않는다는 사실을 반영합니다. 이 CipherSpec은 더 이상 사용되지 않으므로 앞으로는 사용하지 않는 것이 좋습니다.
7. AMQ9288 오류로 인해 연결이 종료되기 전까지 이 CipherSpec을 사용하여 최대 32GB의 데이터를 전송할 수 있습니다. 이 오류를 방지하려면 3중 DES를 사용하지 않거나, 이 CipherSpec을 사용할 때 비밀 키 재설정을 사용으로 설정하십시오.

**플랫폼 지원:**

- a 지원되는 모든 플랫폼에서만 사용 가능합니다.
- b UNIX, Linux, and Windows 플랫폼에서만 사용 가능합니다.

**관련 개념**

32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 호환성』
이 토픽은 IBM WebSphere 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

## 더 이상 사용되지 않는 CipherSpec

필요한 경우 WebSphere MQ와 함께 사용할 수 있는 더 이상 사용되지 않는 CipherSpecs 목록입니다.

더 이상 사용되지 않는 CipherSpec을 사용 가능으로 설정하는 방법에 대한 자세한 정보는 36 페이지의 『IBM WebSphere MQ 에서 지원되는 CipherSpec 값』의 내용을 참조하십시오.

WebSphere MQ TLS 지원과 함께 사용할 수 있는 더 이상 사용되지 않는 CipherSpecs가 다음 표에 나열됩니다.

| 플랫폼 지원 203 페이지의 『1』 | CipherSpec 이름 | 사용되는 프로토콜 | 데이터 무결성 | 암호화 알고리즘 | 암호화 비트 | FIPS 203 페이지의 『2』 | 스위트 B | 더 이상 사용되지 않는 경우 업데이트 |
|---|---|---|---|---|---|---|---|---|
| 모두 | DES_SHA_EXPORT 203 페이지의 『3』 | SSL 3.0 | SHA-1 | DES | 56 | 아니오 | 아니오 | 7.5.0.6 |
| Windows UNIX Linux | DES_SHA_EXPORT1024 203 페이지의 『4』 | SSL 3.0 | SHA-1 | DES | 56 | 아니오 | 아니오 | 7.5.0.6 |

| 플랫폼 지원 203 페이지의 『1』 | CipherSpec 이름 | 사용되는 프로토콜 | 데이터 무결성 | 암호화 알고리즘 | 암호화 비트 | FIPS 203 페이지의 『2』 | 스위트 B | 더 이상 사용되지 않는 경우 업데이트 |
|---|---|---|---|---|---|---|---|---|
| Windows UNIX Linux | FIPS_WITH_DES_CBC_SHA | SSL 3.0 | SHA-1 | DES | 56 | 아니오 203 페이지의 『6』 | 아니오 | 7.5.0.6 |
| Windows UNIX Linux | FIPS_WITH_3DES_EDE_CBC_SHA | SSL 3.0 | SHA-1 | 3DES | 168 | 아니오 203 페이지의 『7』 | 아니오 | 7.5.0.8 |
| 모두 | NULL_MD5 | SSL 3.0 | MD5 | 없음 | 0 | 아니오 | 아니오 | 7.5.0.6 |
| 모두 | NULL_SHA | SSL 3.0 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 7.5.0.6 |
| 모두 | RC2_MD5_EXPORT 203 페이지의 『3』 | SSL 3.0 | MD5 | RC2 | 40 | 아니오 | 아니오 | 7.5.0.7 |
| 모두 | RC4_MD5_EXPORT 203 페이지의 『3』 | SSL 3.0 | MD5 | RC4 | 40 | 아니오 | 아니오 | 7.5.0.7 |
| 모두 | RC4_MD5_US | SSL 3.0 | MD5 | RC4 | 128 | 아니오 | 아니오 | 7.5.0.7 |
| 모두 | RC4_SHA_US | SSL 3.0 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 7.5.0.7 |
| Windows UNIX Linux | RC4_56_SHA_EXPORT1024 203 페이지의 『4』 | SSL 3.0 | SHA-1 | RC4 | 56 | 아니오 | 아니오 | 7.5.0.7 |
| 모두 | TRIPLE_DES_SHA_US | SSL 3.0 | SHA-1 | 3DES | 168 | 아니오 | 아니오 | 7.5.0.8 |
| 모두 | TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | SHA-1 | DES | 56 | 아니오 203 페이지의 『5』 | 아니오 | 7.5.0.6 |
| Windows UNIX Linux | ECDHE_ECDSA_NULL_SHA256 | TLS 1.2 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 7.5.0.6 |
| Windows UNIX Linux | ECDHE_ECDSA_RC4_128_SHA256 | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 7.5.0.7 |
| Windows UNIX Linux | ECDHE_RSA_NULL_SHA256 | TLS 1.2 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 7.5.0.6 |

| 플랫폼 지원 203 페이지의 『1』 | CipherSpec 이름 | 사용되는 프로토콜 | 데이터 무결성 | 암호화 알고리즘 | 암호화 비트 | FIPS 203 페이지의 『2』 | 스위트 B | 더 이상 사용되지 않는 경우 업데이트 |
|---|---|---|---|---|---|---|---|---|
| Windows UNIX Linux | ECDHE_RSA_RC4_128_SHA256 | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 7.5.0.7 |
| Windows UNIX Linux | TLS_RSA_WITH_NULL_NULL | TLS 1.2 | 없음 | 없음 | 0 | 아니오 | 아니오 | 7.5.0.6 |
| 모두 | TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | SHA-256 | 없음 | 0 | 아니오 | 아니오 | 7.5.0.6 |
| Windows UNIX Linux | TLS_RSA_WITH_RC4_128_SHA256 | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 7.5.0.7 |
| 모두 | TLS_RSA_WITH_3DES_EDE_CBC_SHA203 페이지의 『8』 | TLS 1.0 | SHA-1 | 3DES | 168 | 예 | 아니오 | 7.5.0.8 |
| Windows UNIX Linux | ECDHE_ECDSA_3DES_EDE_CBC_SHA256203 페이지의 『8』 | TLS 1.2 | SHA-1 | 3DES | 168 | 예 | 아니오 | 7.5.0.8 |
| Windows UNIX Linux | ECDHE_RSA_3DES_EDE_CBC_SHA256203 페이지의 『8』 | TLS 1.2 | SHA-1 | 3DES | 168 | 예 | 아니오 | 7.5.0.8 |

**참고사항:**

1. 특정 플랫폼이 명시되지 않은 경우 모든 플랫폼에서 CipherSpec을 사용할 수 있습니다.
2. FIPS 인증 플랫폼에서 CipherSpec이 FIPS 인증 CipherSpec인지 여부를 지정합니다. FIPS에 대한 설명은 FIPS(Federal Information Processing Standards)를 참조하십시오.
3. 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.
4. 데이터 교환 키 크기는 1024비트입니다.
5. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다.
6. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다. FIPS_WITH_DES_CBC_SHA라는 이름은 역사적인 의미의 이름으로, 이 CipherSpec이 이전에 FIPS를 준수했었다는 사실을 나타냅니다(현재는 더 이상 준수하지 않음). 이 CipherSpec은 더 이상 사용되지 않으므로 앞으로는 사용하지 않는 것이 좋습니다.
7. FIPS_WITH_3DES_EDE_CBC_SHA라는 이름은 역사적인 의미의 이름으로, 이 CipherSpec이 이전에 FIPS를 준수했었다는 사실을 나타냅니다(현재는 더 이상 준수하지 않음). 이 CipherSpec은 더 이상 사용되지 않습니다.
8. AMQ9288 오류로 인해 연결이 종료되기 전까지 이 CipherSpec을 사용하여 최대 32GB의 데이터를 전송할 수 있습니다. 이 오류를 방지하려면 3중 DES를 사용하지 않거나, 이 CipherSpec을 사용할 때 비밀 키 재설정을 사용으로 설정하십시오.

## IBM WebSphere MQ Explorer 를 사용하여 CipherSpecs 에 대한 정보 얻기

IBM WebSphere MQ Explorer를 사용하여 CipherSpec의 설명을 표시할 수 있습니다.

199 페이지의 『CipherSpec 지정』에 있는 CipherSpec에 대한 정보를 확보하려면 다음 프로시저를 사용하십시오.

1. **IBM WebSphere MQ 탐색기**를 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.

## CipherSpec을 지정하기 위한 대안

운영 체제가 SSL 지원을 제공하는 플랫폼의 경우 시스템이 새 CipherSpec을 지원할 수도 있습니다. 새 CipherSpec을 SSLCIPH 매개변수에 지정할 수 있으나, 제공하는 값은 플랫폼에 따라 다릅니다.

**참고:** 이 절은 CipherSpecs 가 WebSphere MQ 제품과 함께 제공되기 때문에 UNIX, Linux 또는 시스템 는 적용되지 않으므로 배송 후 새 CipherSpecs 가 사용 가능하지 않습니다.

운영 체제가 SSL 지원을 제공하는 플랫폼에서는, 199 페이지의 『CipherSpec 지정』에 포함되지 않은 새 CipherSpec을 시스템이 지원할 수도 있습니다. 새 CipherSpec을 SSLCIPH 매개변수에 지정할 수 있으나, 제공하는 값은 플랫폼에 따라 다릅니다. 모든 경우에 스펙은 유효하고 시스템이 실행 중인 SSL 버전에 의해 지원되는 SSL CipherSpec에 해당되어야 합니다.

**IBM i**

16진 값을 나타내는 2자의 문자열.

허용되는 값에 대한 자세한 정보는 해당 제품 문서를 참조하십시오 ( IBM i 제품 문서에서 암호 스펙 검색).

CHGMQMCHL 또는 CRTMQMCHL 명령을 사용하여 값을 지정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
CRTMQMCHL CHLNAME('channel name') SSLCIPH('hexadecimal value')
```

ALTER QMGR MQSC 명령을 사용하여 SSLCIPH 매개변수를 설정할 수도 있습니다.

**z/OS**

16진 값을 나타내는 2자의 문자열. 16진 코드는 SSL 프로토콜에 정의된 값에 해당합니다.

자세한 정보는 *z/OS Cryptographic Services System SSL Programming*, SC24-5901의 API 참조 장에서 gsk_environment_open()에 대한 설명을 참조하십시오. 여기에는 두 자릿수로 된 16진 코드 형식으로 지원되는 모든 SSL V3.0 및 TLS V1.0 암호 스펙 목록이 있습니다.

## WebSphere MQ 클러스터에 대한 고려사항

WebSphere MQ 클러스터에서는 199 페이지의 『CipherSpec 지정』에 있는 CipherSpec 이름을 사용하는 것이 가장 안전합니다. 대체 스펙을 사용하는 경우 스펙이 다른 플랫폼에서는 유효하지 않을 수도 있음을 유의하십시오. 자세한 정보는 228 페이지의 『SSL 및 클러스터』의 내용을 참조하십시오.

## IBM WebSphere MQ MQI 클라이언트에 대해 CipherSpec 지정

IBM WebSphere MQ MQI 클라이언트에 대해 CipherSpec 을 지정하는 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 Directory 사용(Active Directory가 지원되는 Windows 시스템에서)

## JMS의 Java및 IBM WebSphere MQ 클래스에 대한 IBM WebSphere MQ 클래스가 있는 CipherSuite 지정

Java용 IBM WebSphere MQ 클래스 및 JMS의 IBM WebSphere MQ 클래스는 다른 플랫폼과는 다르게 CipherSuites 를 지정합니다.

Java의 IBM WebSphere MQ 클래스를 포함하는 CipherSuite 지정에 대한 정보는 SSL (Secure Sockets Layer) 지원을 참조하십시오.

JMS의 IBM WebSphere MQ 클래스를 사용하여 CipherSuite 를 지정하는 방법에 대한 정보는 JMS에 대해 WebSphere MQ 클래스와 함께 SSL (Secure Sockets Layer) 사용의 내용을 참조하십시오.

# SSL 및 TLS 비밀 키 재설정

IBM WebSphere MQ는 큐 관리자 및 클라이언트에서 보안 키의 재설정을 지원합니다.

비밀 키는 암호화된 데이터 바이트의 지정된 수가 채널을 통해 흐를 때 또는 채널이 특정 기간 동안 유휴인 후에 재설정됩니다.

키 재설정 값은 항상 MQ 채널의 시작 측에 의해 설정됩니다.

## 큐 관리자

큐 관리자의 경우 **ALTER QMGR** 명령을 **SSLRKEYC** 매개변수와 함께 사용하여 키 재협상 동안에 사용된 값을 설정합니다.

## 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 구조에서 KeyResetCount 필드를 사용하여
2. 환경 변수 MQSSLRESET를 사용하여
3. MQI 클라이언트 구성 파일에 SSLKeyResetCount 속성을 설정하여

이러한 변수는 범위 0 - 999 999 999의 정수로 설정할 수 있으며, 이는 SSL 또는 TLS 비밀 키가 재협상되기 전에 SSL 또는 TLS 내에서 전송 및 수신된 암호화되지 않은 바이트의 수를 나타냅니다. 0 값을 지정하면 SSL 또는 TLS 비밀 키가 절대 재협상되지 않음을 나타냅니다. SSL 또는 TLS 비밀 키 재설정 카운트를 1바이트 - 32KB의 범위에 지정하면, SSL 또는 TLS 채널은 32KB의 비밀 키 재설정 카운트를 사용합니다. 이는 작은 SSL 또는 TLS 비밀 키 재설정 값에서 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.

MQSCO 구조에 대한 전체 세부사항은 KeyResetCount(MQLONG)를 참조하십시오. MQSSLRESET의 전체 세부사항은 MQSSLRESET의 내용을 참조하십시오. 클라이언트 구성 파일에서 SSL 또는 TLS 사용에 대한 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오.

## Java

IBM WebSphere 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.

sslReset계수 필드 또는 환경 특성 MQC.SSL_RESET_COUNT_PROPERTY 의 값은 비밀 키가 재조정되기 전에 Java 클라이언트 코드의 WebSphere MQ 클래스가 보내고 받은 총 바이트 수를 나타냅니다. 송신된 바이트 수는 암호화 전의 바이트 수이며, 수신된 바이트 수는 복호화 후의 바이트 수입니다. 바이트 수에는 또한 Java 클라이언트의 WebSphere MQ 클래스가 보내고 받은 제어 정보도 포함되어 있습니다.

재설정 수가 기본값인 0인 경우 비밀 키는 재협상되지 않습니다. CipherSuite가 지정되지 않으면 재설정 수가 무시됩니다.

### JMS

IBM WebSphere 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. 또한 바이트 수에는 IBM WebSphere MQ classes for JMS에서 송신 및 수신한 제어 정보도 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 관리되지 않는 클라이언트의 경우, 정수 특성 SSLKeyResetCount는 비밀 키가 재조정되기 전에 SSL 또는 TLS 대화에서 송신하거나 수신한 암호화되지 않은 총 바이트 수를 나타냅니다.

.NET용 IBM WebSphere MQ 클래스에서 오브젝트 특성을 사용하는 데 대한 정보는 오브젝트 값 가져오기 및 설정을 참조하십시오.

### XMS .NET

XMS .NET 관리되지 않는 클라이언트의 경우, IBM WebSphere MQ 큐 관리자에 보안 연결을 참조하십시오.

**관련 참조**
ALTER QMGR
DISPLAY QMGR

## 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 206 페이지의 『사용자 엑시트 프로그램에서 기밀성 구현』 을 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.

## 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**

SSL 또는 TLS를 사용할 때 선택한 CipherSpec은 엔터프라이즈에서 데이터 무결성의 레벨을 판별합니다. WebSphere MQ Advanced Message Service(AMS)를 사용하는 경우 고유한 메시지에 대한 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 완전히 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.

# SSL 또는 TLS를 사용하여 두 개의 큐 관리자 연결

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 확보하고 관리해야 합니다. 테스트 시스템에서는 자체 서명 인증서 또는 로컬 인증 기관(CA)에서 발행한 인증서를 사용할 수 있습니다. 프로덕션 시스템에서는 자체 서명 인증서를 사용하지 마십시오. 자세한 정보는 ../zs14140_.dita의 내용을 참조하십시오.

인증서 작성 및 관리에 대한 전체 정보는 104 페이지의 『UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 작업』을 참조하십시오.

이 주제 콜렉션에서는 SSL 통신 설정에 관련된 태스크를 소개하고 이러한 태스크를 완료하는 데 대한 단계별 내용을 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. WebSphere MQ 구현의 경우, SSL 또는 TLS 서버가 항상 클라이언트로부터 인증서를 요청합니다.

**참고사항:**

1. 이 컨텍스트에서는 SSL 클라이언트가 데이터 교환을 시작하는 연결을 의미합니다.
2. 자세한 내용은 용어집을 참조하십시오.

UNIX, Linux 및 Windows 시스템에서 SSL 또는 TLS 클라이언트는 올바른 WebSphere MQ 형식으로 레이블된 인증서가 있는 경우에만 인증서를 송신합니다. MQ형식은 `ibmwebspheremq` 다음에 소문자로 변경된 큐 관리자의 이름이 옵니다. 예를 들어, `QM1`의 경우 `ibmwebspheremqqm1`입니다.

WebSphere MQ는 다른 제품에 대한 인증서와의 혼동을 피하기 위해 레이블에 `ibmwebspheremq` 접두부를 사용합니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). 클라이언트가 인증서를 송신하지 않으면 SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 REQUIRED로 설정된 SSLCAUTH 매개변수 또는 SSLPEER 매개변수 값 세트로 정의된 경우에만 인증에 실패합니다. 익명으로 큐 관리자를 연결하는 데 대한 자세한 정보는(즉, SSL 또는 TLS이 인증서를 송신하지 않는 경우) 192 페이지의 『단방향 인증을 사용하여 두 개의 큐 관리자 연결』의 내용을 참조하십시오.

## 디지털 인증서 레이블, 요구사항 이해

디지털 인증서를 사용하기 위해 SSL 및 TLS를 설정할 때 사용된 플랫폼과 연결에 사용한 방법에 따라 반드시 따라야 하는 특정 레이블 요구사항이 있을 수 있습니다.

### 이 태스크 정보

**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. 주제 식별 이름과 주제 공용 이름이 인증서 자체 내의 필드라는 점에 주목하십시오. 이들은 인증서가 작성될 때 정의되고 변경될 수 없습니다. 그러나 필요한 경우 디지털 인증서와 관련된 레이블을 변경할 수 있습니다.

**인증서 레이블 사용 방법**

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

인증서 레이블은 이름 지정 규칙을 따르므로, 사용 중인 플랫폼에 해당하는 올바른 레이블 이름 지정 규칙을 사용하는지 확인해야 합니다.

이 컨텍스트에서 SSL 또는 TLS 클라이언트는 데이터 교환을 시작하는 연결 파트너를 의미하며, 이는 IBM WebSphere MQ 클라이언트 또는 다른 큐 관리자일 수 있습니다.

SSL 또는 TLS 데이터 교환 동안에 SSL 또는 TLS 클라이언트는 항상 서버로부터 디지털 인증서를 확보하고 유효성 검증합니다. IBM WebSphere MQ 구현의 경우, SSL 또는 TLS 서버는 항상 클라이언트에 인증서를 요청하고 클라이언트는 인증서가 있으면 항상 인증서를 서버에 제공합니다. 클라이언트가 개인 인증서를 찾을 수 없으면, 클라이언트는 서버에 `no certificate` 응답을 보냅니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). 클라이언트가 인증서를 송신하지 않을 경우, SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 REQUIRED로 설정된 SSLCAUTH 매개변수 또는 SSLPEER 매개변수 값 세트로 정의되어 있으면 인증에 실패합니다.

단방향 인증을 사용하여 큐 관리자를 연결(즉, SSL 또는 TLS 클라이언트가 인증서를 송신하지 않는 경우)하는 것에 대한 자세한 정보는 192 페이지의 『단방향 인증을 사용하여 두 개의 큐 관리자 연결』의 내용을 참조하십시오.

## *, UNIX, Linux, and Windows* 시스템

### 이 태스크 정보

, UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 서버는 서버가 올바른 IBM WebSphere MQ 형식으로 레이블된 인증서를 찾은 경우에만 클라이언트에 인증서를 보냅니다. 이러한 시스템에서 올바른 형식은 `ibmwebspheremq`이고, 그 뒤에 큐 관리자의 이름이 소문자로 변경됩니다.

예를 들어, 큐 관리자 이름이 `QM1`인 경우 인증서 레이블 요구사항은 다음과 같습니다.

```
ibmwebspheremqqm1
```

큐 관리자의 키 저장소에서 올바른 대소문자 및 형식의 필수 레이블과 일치하는 인증서를 찾을 수 없는 경우, 오류가 발생하고 SSL 또는 TLS 데이터 교환에 실패합니다.

## *IBM WebSphere MQ* 클라이언트

### 이 태스크 정보

IBM WebSphere MQ 클라이언트 애플리케이션에서 연결할 때 SSL 또는 TLS 클라이언트는 `ibmwebspheremq` 형식의 레이블과 그 뒤에 클라이언트 애플리케이션 프로세스를 실행하는 사용자의 사용자 이름이 오는 인증서가 있는 경우에만 인증서를 송신합니다.

예를 들어, 사용자 이름 `wasadmin`의 경우 인증서 레이블 요구사항은 다음과 같이 소문자로 폴딩됩니다.

```
  ibmwebspheremqwasadmin
```

위의 레이블 요구사항은 C 또는 C++ 및 .NET용 메시지 서비스 클라이언트에 적용됩니다.

### *IBM WebSphere MQJava* 또는 *IBM WebSphere MQJMS* 클라이언트

#### 이 태스크 정보

IBM WebSphere MQ Java 또는 IBM WebSphere MQ JMS 클라이언트는 SSL 또는 TLS 데이터 교환 중에 JSSE(Java Secure Socket Extension) 제공자의 기능을 사용하여 개인 인증서를 선택하므로 인증서 레이블 요구사항에 따르지 않습니다.

기본 작동은 JSSE 클라이언트가 키 저장소의 인증서를 통해 첫 번째로 찾은 허용 가능한 개인 인증서 선택을 반복하는 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 위해 또는 특정한 JSSE 제공자와 조합하여 IBM WebSphere MQ Java 클라이언트 애플리케이션이 수행하는 핸드쉐이크를 더 잘 이해하기 위해 설정하여 디버깅을 사용 가능하게 할 수 있습니다.

```
javax.net.debug=ssl
```

를 사용한다.

명령행에서 `-Djavax.net.debug=ssl`를 사용할 수도 있고 애플리케이션 내에서나 구성을 통해 변수를 설정할 수도 있습니다.

**관련 개념**

122 페이지의 『UNIX, Linux, and Windows 시스템에서 개인 인증서를 키 저장소에 가져오기』
개인 인증서를 가져오려면 이 프로시저를 따르십시오.

## 두 개의 큐 관리자의 상호 인증을 위해 자체 서명 인증서 사용

자체 서명 SSL 또는 TLS 인증서를 사용하여 두 큐 관리자 사이의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

#### 이 태스크 정보

시나리오:

- 안전하게 통신하는 데 필요한 두 개의 큐 관리자 QM1 및 QM2가 있습니다. QM1과 QM2 간에 상호 인증을 수행해야 합니다.
- 자체 서명 인증서를 사용하여 보안 통신을 테스트하기로 결정했습니다.

결과 구성은 다음과 유사합니다.

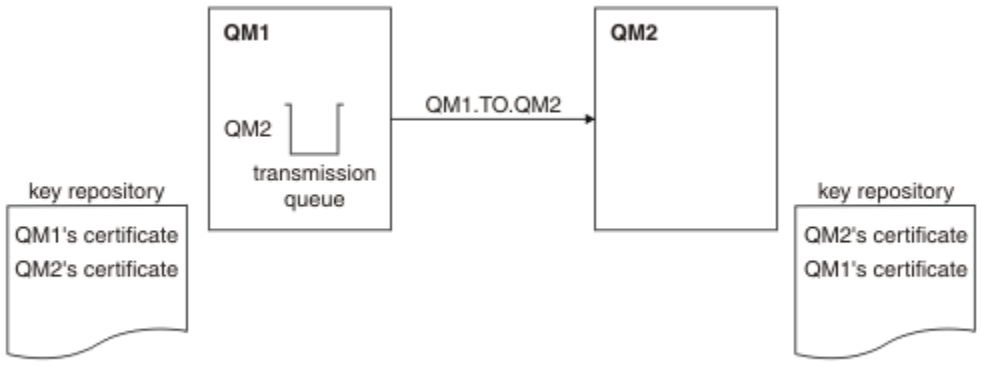



그림 *20.* 이 태스크로부터 나온 구성

189 페이지의 그림 14에서는 QM1용 키 저장소에 QM1에 대한 인증서와 QM2로부터의 공용 인증서가 포함됩니다. QM2용 키 저장소에는 QM2용 인증서와 QM1로부터의 공용 인증서가 포함됩니다.

## 프로시저

1. 운영 체제에 따라 각 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 큐 관리자마다 자체 서명 인증서를 작성하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
3. 각 인증서의 사본을 추출하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. FTP 등의 유틸리티를 사용하여 QM1 인증서의 공용 부분을 QM2 시스템에 전송하거나 반대로 전송하십시오.
5. 큐 관리자마다 파트너 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
6. QM1에서 다음 예와 같은 명령을 발행하여 송신자 채널 및 연관된 전송 큐를 정의하십시오.

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
   SSLCIPH(RC4_MD5_US) DESCR('Sender channel using SSL from QM1 to QM2')

   DEFINE QLOCAL(QM2) USAGE(XMITQ)
   ```

   이 예제에서는 CipherSpec RC4_MD5를 사용합니다. 채널의 각 끝에 있는 CipherSpec은 동일해야 합니다.
7. QM2에서 다음 예와 같은 명령을 발행하여 수신자 채널을 정의하십시오.

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   채널의 이름은 6단계에서 정의한 송신자 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.
8. 채널을 시작합니다.

## 결과

189 페이지의 그림 14에 설명된 대로 키 저장소 및 채널이 작성됩니다.

## 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

큐 관리자 QM2에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
```

```
    CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
    CONNAME(9.20.35.92)                     CURRENT
    RQMNAME(QM1)
    SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
    SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ
 Development,O=IBM,ST=Hampshire,C=UK")
    STATUS(RUNNING)                         SUBSTATE(RECEIVE)
    XMITQ( )
```

각 경우에, SSLPEER의 값은 2단계에서 작성된 파트너 인증서의 DN의 값과 일치해야 한다. 인증서가 자체 서명되었으므로 발행자 이름이 피어 이름과 일치합니다.

SSLPEER은 선택사항입니다. 지정되면, 파트너 인증서의 DN(2단계에서 지정됨)을 허용할 수 있도록 해당 값을 설정해야 합니다. SSLPEER의 사용에 대한 자세한 정보는 WebSphere MQ SSLPEER값에 대한 규칙 의 내용을 참조하십시오.

## 두 개의 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용

CA 서명 SSL 또는 TLS 인증서를 사용하여 두 큐 관리자 사이의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 안전하게 통신하는 데 필요한 두 개의 큐 관리자 QMA 및 QMB가 있습니다. QMA와 QMB 간에 상호 인증을 수행해야 합니다.
- 추후에 프로덕션 환경에서 이 네트워크를 사용할 계획이어서 시작부터 CA 서명 인증서를 사용하기로 결정했습니다.

결과 구성은 다음과 유사합니다.

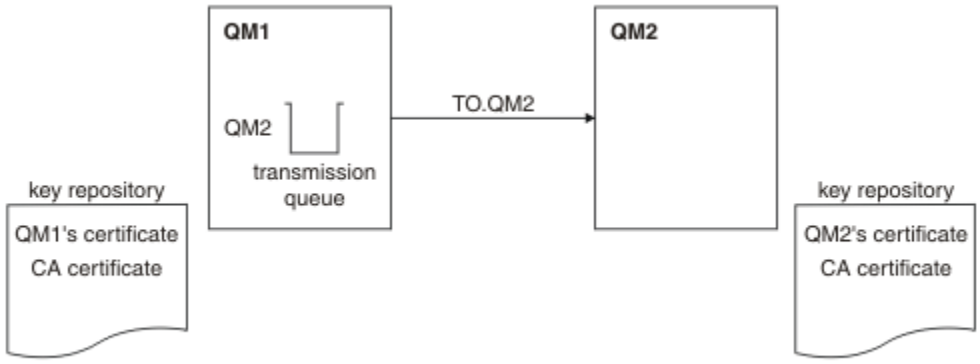



그림 *21.* 이 태스크로부터 나온 구성

191 페이지의 그림 15에서는 QMA용 키 저장소에 QMA의 인증서와 CA 인증서가 포함됩니다. QMB용 키 저장소에는 QMB의 인증서와 CA 인증서가 포함됩니다. 이 예에서는 QMA의 인증서와 QMB의 인증서가 둘 다 동일한 CA에 의해 발행되었습니다. QMA의 인증서와 QMB의 인증서가 다른 CA에 의해 발행된 경우 QMA 및 QMB에 대한 키 저장소에는 두 CA 인증서가 모두 포함되어야 합니다.

### 프로시저

1. 운영 체제에 따라 각 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 큐 관리자마다 CA 서명 인증서를 요청하십시오.
   두 개의 큐 관리자에 대해 서로 다른 CA를 사용할 수도 있습니다.
   - UNIX, Linux 및 Windows 시스템의 경우.

3. 큐 관리자마다 인증 기관 인증서를 키 저장소에 추가하십시오.
   큐 관리자가 다른 인증 기관을 사용 중인 경우, 각 인증 기관에 대한 CA 인증서를 키 저장소 둘 다에 추가해야 합니다.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. 큐 관리자마다 CA 서명 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
5. QMA에서 다음 예와 같은 명령을 발행하여 송신자 채널 및 연관된 전송 큐를 정의하십시오.

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(RC2_MD5_EXPORT) DESCR('Sender channel using SSL from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

   이 예제에서는 CipherSpec RC4_MD5를 사용합니다. 채널의 각 끝에 있는 CipherSpec은 동일해야 합니다.
6. QMB에서 다음 예와 같은 명령을 발행하여 수신자 채널을 정의하십시오.

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
```

   채널의 이름은 6단계에서 정의한 송신자 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.
7. 채널을 시작하십시오.

## 결과

191 페이지의 그림 15에 설명된 대로 키 저장소 및 채널이 작성됩니다.

## 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QMA에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QMB)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QMB)
```

큐 관리자 QMB에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

```
   XMITQ( )
```

각 경우에, SSLPEER의 값은 2단계에서 작성된 파트너 인증서의 식별 이름 (DN) 의 값과 일치해야 합니다. 발급자 이름은 4단계에서 추가된 개인 인증서를 서명한 CA 인증서의 주제 DN과 일치합니다.

## 단방향 인증을 사용하여 두 개의 큐 관리자 연결

큐 관리자가 다른 대상에 대해 단방향 인증을 사용하여 연결하도록 하기 위해(즉, SSL 또는 TLS 클라이언트가 인증서를 송신하지 않을 경우) 상호 인증으로 시스템을 수정하려면 이러한 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 두 개의 큐 관리자(QM1 및 QM2)가 190 페이지의 『두 개의 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용』에서와 같이 설정되었습니다.
- QM2에 대해 단방향 인증을 사용하여 연결되도록 QM1을 변경할 수 있습니다.

결과 구성은 다음과 유사합니다.

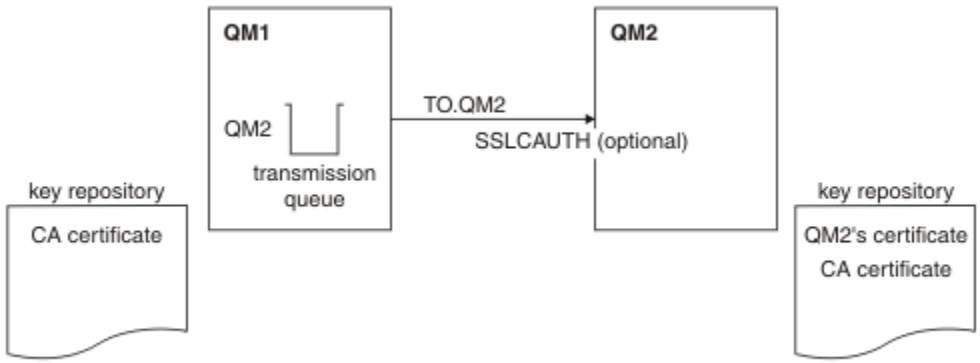



그림 *22.* 단방향 인증을 허용하는 큐 관리자

### 프로시저

1. 운영 체제에 따라 키 저장소에서 QM1의 개인 인증서를 제거하십시오.
   - UNIX, Linux 및 Windows 시스템. 인증서는 다음과 같이 레이블됩니다.
     - `ibmwebspheremq` 다음에 큐 관리자의 이름이 소문자로 접힙니다. 예를 들어, QM1의 경우 `ibmwebspheremqqm1`입니다.
2. 옵션: QM1에서 SSL 또는 TLS 채널이 이전에 실행된 경우 SSL 또는 TLS 환경을 새로 고치십시오.
3. 수신자에서 익명 연결을 허용합니다.

### 결과

193 페이지의 그림 16에 설명된 대로 키 저장소 및 채널이 변경됩니다.

### 다음에 수행할 작업

송신자 채널이 실행되었고 사용자가 REFRESH SECURITY TYPE(SSL) 명령을 발행한 경우(2단계에서), 채널이 자동으로 재시작됩니다. 송신자 채널이 실행되지 않은 경우, 이를 시작하십시오.

채널의 서버 끝에서 채널 상태 표시에 있는 피어 이름 매개변수 값의 존재는 클라이언트 인증서가 플로우되었음을 표시합니다.

일부 DISPLAY 명령을 실행하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

QM1 큐 관리자에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 유사합니다.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

QM2 큐 관리자에서 다음 명령을 입력하십시오.

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 유사합니다.

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

QM2에서는 SSLPEER 필드가 비어 있습니다. 이는 QM1이 인증서를 송신하지 않았음을 보여줍니다. QM1에서는 SSLPEER의 값이 QM2의 개인 인증서에 있는 DN의 값과 일치합니다.

## 클라이언트를 큐 관리자에 안전하게 연결

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 확보하고 관리해야 합니다. 테스트 시스템에서는 자체 서명 인증서 또는 로컬 인증 기관(CA)에서 발행한 인증서를 사용할 수 있습니다. 프로덕션 시스템에서는 자체 서명 인증서를 사용하지 마십시오. 자세한 정보는 ../zs14140_.dita의 내용을 참조하십시오.

인증서 작성 및 관리에 대한 전체 정보는 104 페이지의 『UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 작업』을 참조하십시오.

이 주제 콜렉션에서는 SSL 통신 설정에 관련된 태스크를 소개하고 이러한 태스크를 완료하는 데 대한 단계별 내용을 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. WebSphere MQ 구현의 경우, SSL 또는 TLS 서버가 항상 클라이언트로부터 인증서를 요청합니다.

UNIX, Linux, and Windows 시스템에서 SSL 또는 TLS 클라이언트는 올바른 WebSphere MQ 형식으로 레이블된 인증서가 있는 경우에만 인증서를 송신합니다. MQ형식은 `ibmwebspheremq` 다음에 소문자로 변경된 로그온 사용자 ID가 옵니다(예: `ibmwebspheremqmyuserid`).

WebSphere MQ는 다른 제품에 대한 인증서와의 혼동을 피하기 위해 레이블에 `ibmwebspheremq` 접두부를 사용합니다. 전체 인증서 레이블을 소문자로 지정해야 합니다.

SSL 또는 TLS 서버는 항상 클라이언트 인증서를 유효성 검증합니다(전송된 경우). 클라이언트가 인증서를 송신하지 않으면 SSL 또는 TLS 서버 역할을 수행하는 채널의 끝이 REQUIRED로 설정된 SSLCAUTH 매개변수 또는 SSLPEER 매개변수 값 세트로 정의된 경우에만 인증에 실패합니다. 익명으로 큐 관리자를 연결하는 데 대한 자세한 정보는 197 페이지의 『클라이언트를 큐 관리자에 익명으로 연결』의 내용을 참조하십시오.

## 클라이언트 및 큐 관리자의 상호 인증을 위해 자체 서명 인증서 사용

자체 서명 SSL 또는 TLS 인증서를 사용하여 클라이언트와 큐 관리자 간의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 현재 안전하게 통신하는 데 필요한 클라이언트 C1과 큐 관리자 QM1이 있습니다. C1과 QM1 간에 상호 인증을 수행해야 합니다.
- 자체 서명 인증서를 사용하여 보안 통신을 테스트하기로 결정했습니다.

IBM i의 DCM은 자체 서명 인증서를 지원하지 않으므로 이 태스크는 IBM i 시스템에서 적용 가능하지 않습니다.

결과 구성은 다음과 유사합니다.

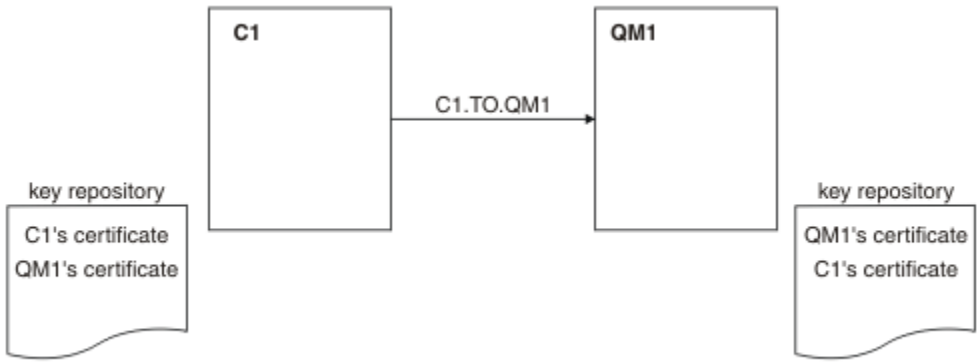



그림 *23.* 이 태스크로부터 나온 구성

195 페이지의 그림 17에서는 QM1용 키 저장소에 QM1에 대한 인증서와 C1로부터의 공용 인증서가 포함됩니다. C1용 키 저장소에는 C1용 인증서와 QM1로부터의 공용 인증서가 포함됩니다.

### 프로시저

1. 운영 체제에 따라 클라이언트 및 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 클라이언트 및 큐 관리자에 대해 자체 서명 인증서를 작성하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
3. 각 인증서의 사본을 추출하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. FTP와 같은 유틸리티를 사용하여 C1 인증서의 공용 부분을 QM1 시스템으로, 그리고 그 반대로 전송하십시오.
5. 클라이언트 및 큐 관리자에 대해 파트너 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
6. 큐 관리자에서 REFRESH SECURITY TYPE(SSL) 명령을 실행하십시오.

7. 다음 방법 중 하나로 클라이언트 연결 채널을 정의하십시오.
   - WebSphere MQ MQI 클라이언트에서 클라이언트 연결 채널 작성에 설명된 대로 C1의 MQSCO 구조와 함께 MQCONNX 호출 사용.
   - 서버 연결 및 클라이언트 연결 정의 작성 에 설명된 대로 클라이언트 채널 정의 테이블을 사용하십시오.
8. QM1에서 다음 예와 같은 명령을 발행하여 서버 연결 채널을 정의하십시오.

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   채널의 이름은 6단계에서 정의한 클라이언트 연결 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.

### 결과

195 페이지의 그림 17에 설명된 대로 키 저장소와 채널이 작성됩니다.

### 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력은 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

채널 정의의 SSLPEER 필터 속성을 설정하는 것은 선택사항입니다. 채널 정의 SSLPEER이 설정된 경우, 이 값은 2단계에서 작성된 파트너 인증서의 주제 DN과 일치해야 합니다. 연결이 성공적으로 완료되면 표시 CHSTATUS 출력의 SSLPEER 필드에 원격 클라이언트 인증서의 주제 DN이 표시된다.

## 클라이언트 및 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용

CA 서명 SSL 또는 TLS 인증서를 사용하여 클라이언트와 큐 관리자 간의 상호 인증을 구현하려면 다음 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 현재 안전하게 통신하는 데 필요한 클라이언트 C1과 큐 관리자 QM1이 있습니다. C1과 QM1 간에 상호 인증을 수행해야 합니다.
- 추후에 프로덕션 환경에서 이 네트워크를 사용할 계획이어서 시작부터 CA 서명 인증서를 사용하기로 결정했습니다.

결과 구성은 다음과 유사합니다.

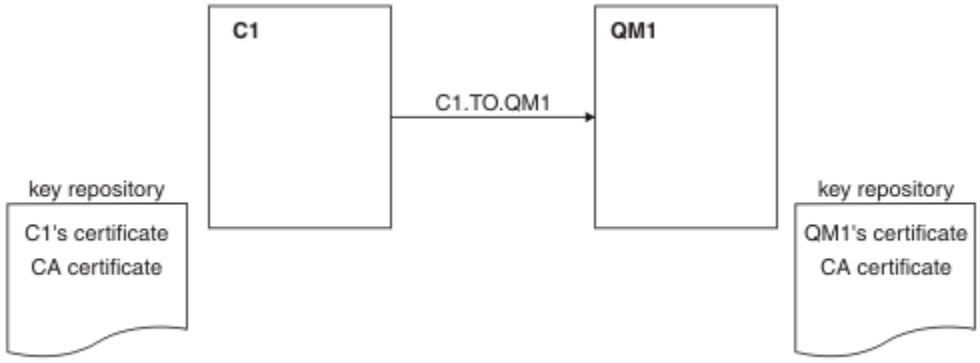



그림 *24.* 이 태스크로부터 나온 구성

196 페이지의 그림 18에서는 C1용 키 저장소에 C1용 인증서와 CA 인증서가 포함됩니다. QM1용 키 저장소에는 QM1용 인증서와 CA 인증서가 포함됩니다. 이 예에서는 C1의 인증서와 QM1의 인증서가 둘 다 동일한 CA에 의해 발행되었습니다. C1의 인증서와 QM1의 인증서가 다른 CA에 의해 발행된 경우 C1 및 QM1에 대한 키 저장소에 두 CA 인증서가 모두 포함되어야 합니다.

## 프로시저

1. 운영 체제에 따라 클라이언트 및 큐 관리자에서 키 저장소를 준비하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
2. 클라이언트 및 큐 관리자에 대해 CA 서명 인증서를 요청하십시오.
   클라이언트 및 큐 관리자에 대해 서로 다른 CA를 사용할 수도 있습니다.
   - UNIX, Linux 및 Windows 시스템의 경우.
3. 클라이언트 및 큐 관리자에 대해 인증 기관 인증서를 키 저장소에 추가하십시오.
   클라이언트 및 큐 관리자가 다른 인증 기관을 사용 중인 경우, 각 인증 기관에 대한 CA 인증서를 키 저장소 둘 다에 추가해야 합니다.
   - UNIX, Linux 및 Windows 시스템의 경우.
4. 클라이언트 및 큐 관리자에 대해 CA 서명 인증서를 키 저장소에 추가하십시오.
   - UNIX, Linux 및 Windows 시스템의 경우.
5. 다음 방법 중 하나로 클라이언트 연결 채널을 정의하십시오.
   - WebSphere MQ MQI 클라이언트에서 클라이언트 연결 채널 작성에 설명된 대로 C1의 MQSCO 구조와 함께 MQCONNX 호출 사용.
   - 서버 연결 및 클라이언트 연결 정의 작성 에 설명된 대로 클라이언트 채널 정의 테이블을 사용하십시오.
6. QM1에서 다음 예와 같은 명령을 발행하여 서버 연결 채널을 정의하십시오.

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   채널의 이름은 6단계에서 정의한 클라이언트 연결 채널과 동일해야 하며 동일한 CipherSpec을 사용합니다.

## 결과

196 페이지의 그림 18에 설명된 대로 키 저장소 및 채널이 작성됩니다.

## 다음에 수행할 작업

DISPLAY 명령을 사용하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 같습니다.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ
Development,O=IBM,ST=Hampshire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 출력의 SSLPEER 필드에서는 2단계에서 작성된 원격 클라이언트 인증서의 주체 DN을 보여줍니다. 발급자 이름은 4단계에서 추가된 개인 인증서를 서명한 CA 인증서의 주제 DN과 일치합니다.

## 클라이언트를 큐 관리자에 익명으로 연결

큐 관리자를 익명으로 다른 관리자에 연결하도록 허용하기 위해 상호 인증으로 시스템을 수정하려면 이러한 샘플 지시사항을 따르십시오.

### 이 태스크 정보

시나리오:

- 큐 관리자와 클라이언트(QM1 및 C1)가 196 페이지의 『클라이언트 및 큐 관리자의 상호 인증을 위해 CA 서명 인증서 사용』에서와 같이 설정되었습니다.
- C1가 익명으로 QM1에 연결되도록 이를 변경하려고 합니다.

결과 구성은 다음과 유사합니다.

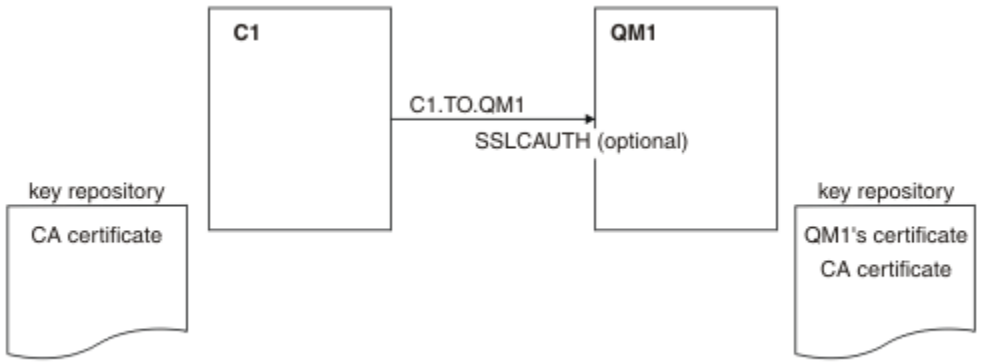



그림 *25.* 익명 연결을 허용하는 클라이언트 및 큐 관리자

### 프로시저

1. 운영 체제에 따라 C1의 키 저장소에서 개인 인증서를 제거하십시오.
   - UNIX, Linux 및 Windows 시스템. 인증서는 다음과 같이 레이블됩니다.
     - `ibmwebspheremq` 다음에 로그온 사용자 ID가 소문자로 접혀 있습니다 (예: `ibmwebspheremqmyuserid`).
2. 클라이언트 애플리케이션을 재시작하거나 클라이언트 애플리케이션을 닫도록 하고 모든 SSL 또는 TLS 연결을 다시 여십시오.
3. 다음 명령을 발행하여 큐 관리자에서 익명 연결을 허용하십시오.

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

### 결과

198 페이지의 그림 19에 설명된 대로 키 저장소 및 채널이 변경됩니다.

### 다음에 수행할 작업

채널의 서버 끝에서 채널 상태 표시에 있는 피어 이름 매개변수 값의 존재는 클라이언트 인증서가 플로우되었음을 표시합니다.

일부 DISPLAY 명령을 실행하여 태스크가 완료되었는지 확인하십시오. 태스크에 성공하면 결과 출력이 다음 예에 표시된 것과 유사합니다.

큐 관리자 QM1에서 다음 명령을 입력하십시오.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

결과 출력은 다음 예제와 유사합니다.

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

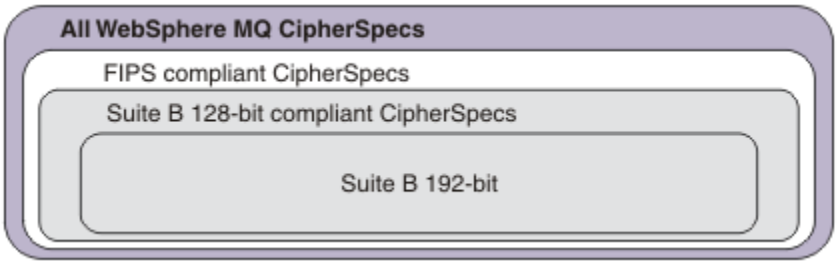
C1이 인증서를 보내지 않았음을 보여주는 SSLCERTI 및 SSLPEER 필드는 비어 있습니다.

## CipherSpec 지정

**DEFINE CHANNEL** MQSC 명령 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec 을 지정하십시오.

IBM WebSphere MQ에 사용할 수 있는 CipherSpec 중 일부는 FIPS를 준수합니다. NULL_MD5와 같이 다른 CipherSpec은 FIPS를 준수하지 않아도 됩니다. 마찬가지로, 일부 FIPS 준수 CipherSpec은 Suite B를 준수하지만 그렇지 않은 경우도 있습니다. 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.




다음 표에 IBM WebSphere MQ SSL 및 TLS 지원과 함께 사용할 수 있는 암호 스펙이 나와 있습니다. 개인 인증서를 요청할 때 공용 및 개인 키 쌍의 키 크기를 지정합니다. SSL 데이터 교환 동안에 사용되는 키 크기는 인증서에 저장된 크기입니다(표에 명시된 것처럼 CipherSpec으로 판별되는 경우는 제외).

| CipherSpec 이름 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 | 암호화 비트 | FIPS[1] | 스위트 B 128 비트 | 스위트 B 192 비트 |
|---|---|---|---|---|---|---|---|
| NULL_MD5 [a] | SSL 3.0 | MD5 | 없음 | 0 | 아니오 | 아니오 | 아니오 |

| CipherSpec 이름 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 | 암호화 비트 | FIPS[1] | 스위트 B 128 비트 | 스위트 B 192 비트 |
|---|---|---|---|---|---|---|---|
| NULL_SHA [a] | SSL 3.0 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| RC4_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC4 | 40 | 아니오 | 아니오 | 아니오 |
| RC4_MD5_US [a] | SSL 3.0 | MD5 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| RC4_SHA_US [a] | SSL 3.0 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| RC2_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC2 | 40 | 아니오 | 아니오 | 아니오 |
| DES_SHA_EXPORT [2] [a] | SSL 3.0 | SHA-1 | DES | 56 | 아니오 | 아니오 | 아니오 |
| RC4_56_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | RC4 | 56 | 아니오 | 아니오 | 아니오 |
| DES_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | DES | 56 | 아니오 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA [a] | TLS 1.0 | SHA-1 | AES | 128 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA [4] [a] | TLS 1.0 | SHA-1 | AES | 256 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_DES_CBC_SHA [a] | TLS 1.0 | SHA-1 | DES | 56 | No[5] | 아니오 | 아니오 |
| FIPS_WITH_DES_CBC_SHA [b] | SSL 3.0 | SHA-1 | DES | 56 | No[6] | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 [b] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 [b] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 128 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 256 | 예 | 아니오 | 아니오 |
| ECDHE_ECDSA_RC4_128_SHA256 [b] | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| ECDHE_RSA_RC4_128_SHA256 [b] | TLS 1.2 | SHA_1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |
| ECDHE_ECDSA_AES_128_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 128 | 예 | 아니오 | 아니오 |
| ECDHE_ECDSA_AES_256_CBC_SHA384 [b] | TLS 1.2 | SHA-384 | AES | 256 | 예 | 아니오 | 아니오 |
| ECDHE_RSA_AES_128_CBC_SHA256 [b] | TLS 1.2 | SHA-256 | AES | 128 | 예 | 아니오 | 아니오 |
| ECDHE_RSA_AES_256_CBC_SHA384 [b] | TLS 1.2 | SHA-384 | AES | 256 | 예 | 아니오 | 아니오 |
| ECDHE_ECDSA_AES_128_GCM_SHA256 [b] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | 예 | 예 | 아니오 |

| CipherSpec 이름 | 사용되는 프로토콜 | MAC 알고리즘 | 암호화 알고리즘 | 암호화 비트 | FIPS[1] | 스위트 B 128 비트 | 스위트 B 192 비트 |
|---|---|---|---|---|---|---|---|
| ECDHE_ECDSA_AES_256_GCM_SHA384 [b] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | 예 | 아니오 | 예 |
| ECDHE_RSA_AES_128_GCM_SHA256 [b] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | 예 | 아니오 | 아니오 |
| ECDHE_RSA_AES_256_GCM_SHA384 [b] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | 예 | 아니오 | 아니오 |
| TLS_RSA_WITH_NULL_SHA256 [b] | TLS 1.2 | SHA-256 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| ECDHE_RSA_NULL_SHA256 [b] | TLS 1.2 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| ECDHE_ECDSA_NULL_SHA256 [b] | TLS 1.2 | SHA-1 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| TLS_RSA_WITH_NULL_NULL [b] | TLS 1.2 | 없음 | 없음 | 0 | 아니오 | 아니오 | 아니오 |
| TLS_RSA_WITH_RC4_128_SHA256 [b] | TLS 1.2 | SHA-1 | RC4 | 128 | 아니오 | 아니오 | 아니오 |

**참고:**

1. FIPS 인증 플랫폼에서 CipherSpec이 FIPS 인증 CipherSpec인지 여부를 지정합니다. FIPS에 대한 설명은 FIPS(Federal Information Processing Standards)를 참조하십시오.
2. 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.
3. 데이터 교환 키 크기는 1024비트입니다.
4. 탐색기가 사용하는 JRE에 적절한 제한 없는 정책 파일이 적용되지 않은 경우 이 CipherSpec을 사용하여 WebSphere MQ 탐색기에서 큐 관리자로의 연결을 보호할 수 없습니다.
5. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다.
6. 이 CipherSpec은 2007년 5월 19일 이전에 FIPS 140-2 인증되었습니다. 이름 FIPS_WITH_DES_CBC_SHA은(는) 히스토리이며 이 CipherSpec이 이전에(더 이상) FIPS를 준수하지 않는다는 사실을 반영합니다. 이 CipherSpec은 더 이상 사용되지 않으므로 앞으로는 사용하지 않는 것이 좋습니다.
7. AMQ9288 오류로 인해 연결이 종료되기 전까지 이 CipherSpec을 사용하여 최대 32GB의 데이터를 전송할 수 있습니다. 이 오류를 방지하려면 3중 DES를 사용하지 않거나, 이 CipherSpec을 사용할 때 비밀 키 재설정을 사용으로 설정하십시오.

**플랫폼 지원:**

- a 지원되는 모든 플랫폼에서만 사용 가능합니다.
- b UNIX, Linux, and Windows 플랫폼에서만 사용 가능합니다.

**관련 개념**

32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 호환성』
이 토픽은 IBM WebSphere MQ에서 CipherSpec과 디지털 인증서 사이의 관계를 소개하여 보안 정책에 대해 적합한 CipherSpec 및 디지털 인증서를 선택하는 방법에 대한 정보를 제공합니다.

**관련 참조**

DEFINE CHANNEL

## IBM WebSphere MQ Explorer 를 사용하여 CipherSpecs 에 대한 정보 얻기

IBM WebSphere MQ Explorer를 사용하여 CipherSpec의 설명을 표시할 수 있습니다.

199 페이지의 『CipherSpec 지정』 에 있는 CipherSpec에 대한 정보를 확보하려면 다음 프로시저를 사용하십시오.

1. **IBM WebSphere MQ 탐색기**를 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.

## CipherSpec을 지정하기 위한 대안

운영 체제가 SSL 지원을 제공하는 플랫폼의 경우 시스템이 새 CipherSpec을 지원할 수도 있습니다. 새 CipherSpec을 SSLCIPH 매개변수에 지정할 수 있으나, 제공하는 값은 플랫폼에 따라 다릅니다.

**참고:** 이 절은 CipherSpecs 가 WebSphere MQ 제품과 함께 제공되기 때문에 UNIX, Linux 또는 시스템 는 적용되지 않으므로 배송 후 새 CipherSpecs 가 사용 가능하지 않습니다.

운영 체제가 SSL 지원을 제공하는 플랫폼에서는, 199 페이지의 『CipherSpec 지정』 에 포함되지 않은 새 CipherSpec을 시스템이 지원할 수도 있습니다. 새 CipherSpec을 SSLCIPH 매개변수에 지정할 수 있으나, 제공하는 값은 플랫폼에 따라 다릅니다. 모든 경우에 스펙은 유효하고 시스템이 실행 중인 SSL 버전에 의해 지원되는 SSL CipherSpec에 해당되어야 합니다.

**IBM i**

16진 값을 나타내는 2자의 문자열.

허용되는 값에 대한 자세한 정보는 해당 제품 문서를 참조하십시오 ( IBM i 제품 문서에서 암호 스펙 검색).

CHGMQMCHL 또는 CRTMQMCHL 명령을 사용하여 값을 지정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
CRTMQMCHL CHLNAME('channel name') SSLCIPH('hexadecimal value')
```

ALTER QMGR MQSC 명령을 사용하여 SSLCIPH 매개변수를 설정할 수도 있습니다.

**z/OS**

16진 값을 나타내는 2자의 문자열. 16진 코드는 SSL 프로토콜에 정의된 값에 해당합니다.

자세한 정보는 *z/OS Cryptographic Services System SSL Programming*, SC24-5901의 API 참조 장에서 gsk_environment_open()에 대한 설명을 참조하십시오. 여기에는 두 자릿수로 된 16진 코드 형식으로 지원되는 모든 SSL V3.0 및 TLS V1.0 암호 스펙 목록이 있습니다.

### WebSphere MQ 클러스터에 대한 고려사항

WebSphere MQ 클러스터에서는 199 페이지의 『CipherSpec 지정』 에 있는 CipherSpec 이름을 사용하는 것이 가장 안전합니다. 대체 스펙을 사용하는 경우 스펙이 다른 플랫폼에서는 유효하지 않을 수도 있음을 유의하십시오. 자세한 정보는 228 페이지의 『SSL 및 클러스터』 의 내용을 참조하십시오.

## IBM WebSphere MQ MQI 클라이언트에 대해 CipherSpec 지정

IBM WebSphere MQ MQI 클라이언트에 대해 CipherSpec 을 지정하는 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 Directory 사용(Active Directory가 지원되는 Windows 시스템에서)

### JMS의 Java및 IBM WebSphere MQ 클래스에 대한 IBM WebSphere MQ 클래스가 있는 CipherSuite 지정

Java용 IBM WebSphere MQ 클래스 및 JMS의 IBM WebSphere MQ 클래스는 다른 플랫폼과는 다르게 CipherSuites 를 지정합니다.

Java의 IBM WebSphere MQ 클래스를 포함하는 CipherSuite 지정에 대한 정보는 SSL (Secure Sockets Layer) 지원을 참조하십시오.

JMS의 IBM WebSphere MQ 클래스를 사용하여 CipherSuite 를 지정하는 방법에 대한 정보는 JMS에 대해 WebSphere MQ 클래스와 함께 SSL (Secure Sockets Layer) 사용의 내용을 참조하십시오.

## 감사

이벤트 메시지를 사용하여 보안 침입 또는 침입 시도를 확인할 수 있습니다. IBM WebSphere 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 참조하고, 일반적으로 이벤트 모니터링에 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 SSL을 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. z/OS의 RACF 또는 다른 플랫폼의 OAM을 사용하여 메시지에서 사용자 ID를 확인하여 리모트 큐 관리자에게 권한을 부여하십시오.

### 이 태스크 정보

플랫폼의 보안 기능 및 WebSphere MQ의 액세스 제어 메커니즘을 사용하여 큐에 대한 액세스를 제어할 수 있습니다.

### 프로시저

1. 특정 큐 관리자가 메시지를 큐에 넣는 것을 방지하려면 플랫폼에서 사용 가능한 보안 기능을 사용하십시오.

   예를 들면, 다음과 같습니다.

   - z/OS용 WebSphere MQ의 RACF 또는 기타 외부 보안 관리자
   - 다른 플랫폼의 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/OS에서는 검사할 때 MCAUSER에서 파생된 사용자 ID와 네트워크에서 수신한 사용자 ID를 모두 사용합니다.

   **CTX**
   메시지와 연관된 컨텍스트 정보에서 사용자 ID를 사용하십시오. z/OS에서는 검사할 때 MCAUSER에서 파생된 사용자 ID나 네트워크에서 수신한 사용자 ID, 또는 둘 다를 사용합니다. 링크가 신뢰되고 인증되는 경우 이 옵션을 사용하십시오.

   **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.

   **ALTMCA(z/OS 전용)**
   CTX에 대해서는 네트워크에서 수신한 모든 사용자 ID가 사용되지는 않습니다.

## 리모트 클러스터 큐에 메시지를 넣는 권한 부여

사용자 플랫폼에서 큐 관리자에 연결하고 해당 큐 관리자에서 큐에 넣을 수 있는 액세스 권한을 부여합니다.

### 이 태스크 정보

기본 작동은 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 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

- UNIX, Linux 및 시스템 경우 다음 명령을 실행하십시오.

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
  ```

  사용자는 다른 클러스터 큐가 아닌 지정된 클러스터 큐에만 메시지를 넣을 수 있습니다.

  변수 이름에는 다음과 같은 의미가 있습니다.

  **QMgrName**
  대기열 관리자의 이름입니다.

  **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. 176 페이지의 『메시지를 리모트 클러스터 큐에 넣기 위한 권한 부여』의 지시사항에 따라 이 권한을 설정하십시오.

**관련 정보**
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 리모트 IP 주소, 리모트 큐 관리자 이름 또는 원격 시스템이 제공하는 SSL/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.

   
   
   **UNIX and Linux 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.

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)
   ```

## 결과

강제로 제거된 큐 관리자는 변경되지 않습니다. 해당 로컬 클러스터 정의가 이러한 큐 관리자를 클러스터에 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 및 클러스터

클러스터를 위해 SSL을 구성할 때 CLUSRCVR 채널 정의가 자동 정의된 CLUSSDR 채널로서 다른 큐 관리자에 전파됨을 유의하십시오. CLUSRCVR 채널이 SSL을 사용하는 경우 채널을 사용하여 통신하는 모든 큐 관리자에서 SSL을 구성해야 합니다.

SSL에 대한 자세한 정보는 SSL 및 TLS에 대한 WebSphere MQ 지원을 참조하십시오. 이 충고는 일반적으로 클러스터 채널에 적용 가능하지만 다음에 일부 특수 고려사항을 제공하고 싶을 수도 있습니다.

IBM WebSphere 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 SSL 연결성을 위해 구성된 경우에는 다음 고려사항이 적용됩니다.

- 이 CLUSRCVR과 통신하려는 모든 큐 관리자에게는 SSL 지원에 대한 액세스가 있어야 합니다. 이 SSL 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 WebSphere 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=WebSphere 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 문자열 RC4_SHA_US 값 05는 IBM i에서 허용되지만 UNIX, Linux 또는 Windows 시스템에서는 올바른 스펙이 아닙니다. 따라서 사용자 정의 SSLCIPH 매개변수가 CLUSRCVR에서 사용되는 경우에는 결과로 나오는 모든 자동 정의된 클러스터 송신자 채널은 기본 SSL 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: RC4_MD5_US)을 사용하십시오.

SSLCRLNL 매개변수는 개별 큐 관리자에 적용되고 클러스터 내에 다른 큐 관리자에게 전파되지 않습니다.

### 클러스터된 큐 관리자 및 채널을 SSL로 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 WebSphere MQ 릴리스의 새로운 CipherSpec이며 이전 릴리스에서 지원되지 않을 수 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.

클러스터의 모든 큐 관리자를 WebSphere MQ V6 이상으로 업그레이드하십시오(큐 관리자가 이 레벨에 있지 않은 경우). SSL이 이들 각각으로부터 작동할 수 있도록 인증서 및 키를 분배하십시오.

### 이 태스크 정보

CLUSRCVR을 한 번에 하나씩 변경하고 다음 변경을 수행하기 전에 클러스터를 통해 변경사항이 플로우되도록 허용하십시오. 현재 채널이 변경사항이 클러스터 전체에 분배될 때까지 역방향 경로를 변경하지 않았는지 확인하십시오.

### 프로시저

1. 원하는 순서대로 CLUSRCVR 채널을 SSL로 전환하십시오.

   변경사항이 SSL로 변경되지 않는 채널과 반대 방향으로 플로우됩니다.
2. 모든 수동 CLUSSDR 채널을 SSL로 스위치하십시오.

   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 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에 자동으로 상태 업데이트를 보낸 이후 27일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

**관련 개념**

199 페이지의 『CipherSpec 지정』

**DEFINE CHANNEL** MQSC 명령 또는 **ALTER CHANNEL** MQSC 명령에서 **SSLCIPH** 매개변수를 사용하여 CipherSpec 을 지정하십시오.

32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 호환성』

이 토픽은 IBM WebSphere 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

## 클러스터 큐 관리자 및 채널에서 SSL 또는 TLS 사용 안함

SSL 또는 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. 슬러리형 매개변수의 값을 작은따옴표 ' ' 의 빈 문자열로 설정하십시오.

   원하는 순서대로 클러스터 수신자 채널에서 SSL 또는 TLS를 끌 수 있습니다.

   SSL 또는 TLS가 활성 상태를 유지하는 채널을 통해 반대 방향으로 변경사항이 플로우됨을 참조하십시오.
2. **DISPLAY CLUSQMGR(*)** ALL 명령을 사용하여 다른 큐 관리자 모두에서 새 값이 반영되는지 확인하십시오.
3. 모든 수동 클러스터 송신자 채널에서 SSL 또는 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 230 페이지의 그림 26에 표시되고 다음 예에 설명됩니다.

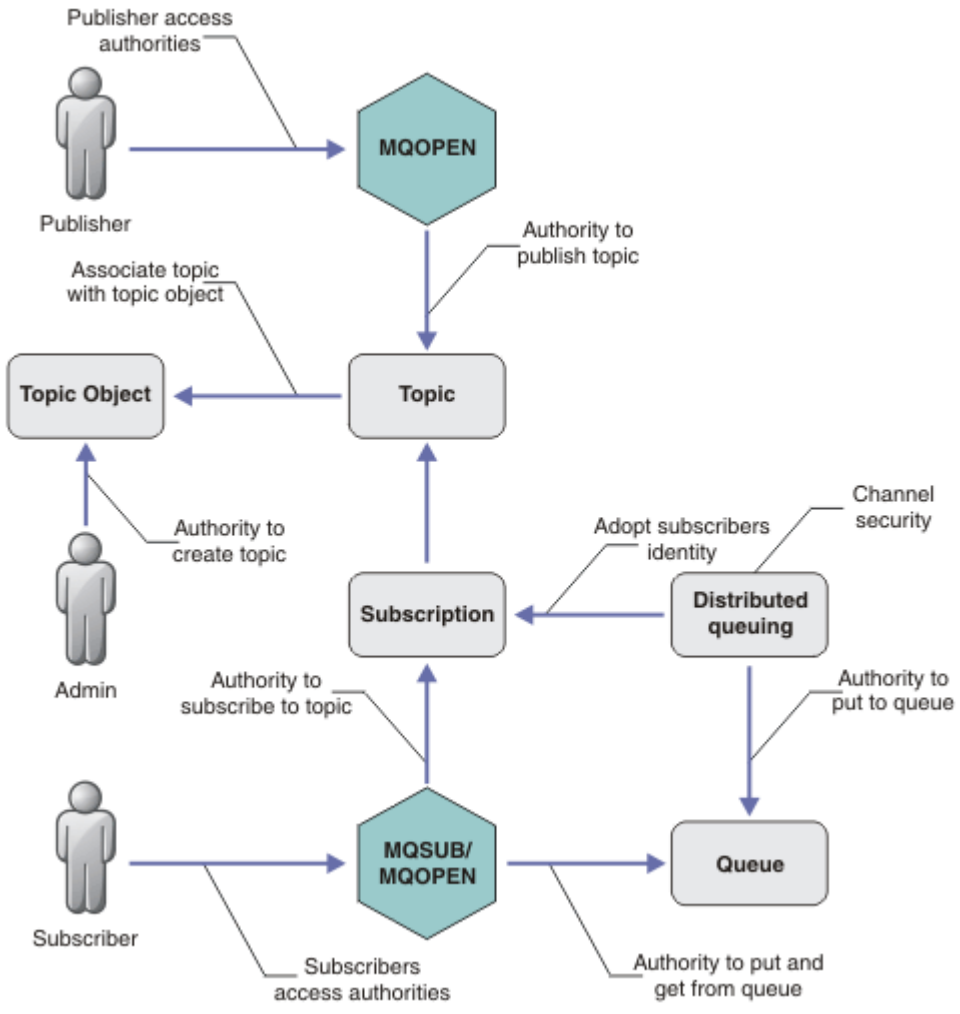



그림 *26.* 발행/구독 보안 관계

**토픽**

토픽은 토픽 문자열로 식별되며, 일반적으로 트리로 구성됩니다(토픽 트리참조). 토픽에 대한 액세스를 제어하려면 토픽을 토픽 오브젝트와 연관시켜야 합니다.232 페이지의 『토픽 보안 모델』 토픽 오브젝트를 사용하여 토픽을 보안하는 방법을 설명합니다.

**관리 토픽 오브젝트**

관리 토픽 오브젝트 목록과 함께 **setmqaut** 명령을 사용하여 누가 어떤 용도로 토픽에 대한 액세스를 가지는지를 제어할 수 있습니다. 예 236 페이지의 『토픽 구독을 위해 사용자에게 액세스 부여』 및 241 페이지의 『토픽에 발행을 위해 사용자에게 액세스 부여』를 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**
233 페이지의 『토픽 오브젝트 이름을 사용하여 구독』

**토픽을 사용하여 구독**
234 페이지의 『토픽 노드가 없는 토픽 문자열을 사용한 구독』

**와일드카드와 함께 토픽을 사용하여 구독**
234 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. 245 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. 234 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.

### 큐 관리자 간 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 WebSphere MQ 클러스터 구성에서 채널은 채널 프로세스의 권한을 사용하여 대상 큐에 메시지를 배치하도록 PUTAUT=DEF를 사용하여 구성됩니다. PUTAUT=CTX를 사용하여 구독 중인 사용자가 클러스터에 있는 또 다른 큐 관리자에 구독을 전달하는 권한을 갖도록 채널 구성을 수정할 수 있습니다.

큐 관리자 사이의 발행/구독 보안에서는 채널 정의를 변경하여 클러스터에 있는 기타 서버에 구독을 전파하도록 허용되는 사용자를 제어하는 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. WebSphere MQ 관리자 - mqm 그룹의 구성원

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.

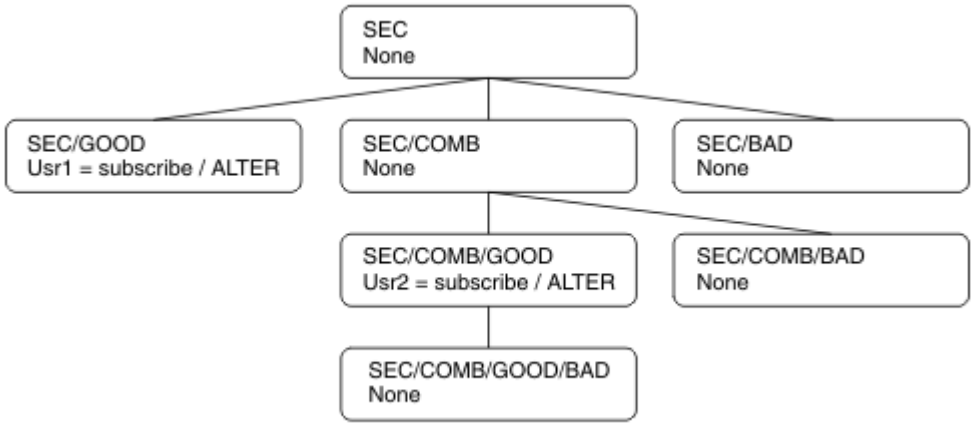
보안 속성은 특정 운영 체제 사용자 ID 또는 보안 그룹이 토픽 오브젝트에 대해 가진 권한을 나타내는 액세스 제어 목록입니다.

표시된 보안 속성 또는 권한으로 토픽 오브젝트를 정의한 다음 예를 고려하십시오.

표 *17.* 토픽 오브젝트 권한의 예

| **토픽 이름** | **토픽 문자열** | **권한 - z/OS가 아님** | **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.

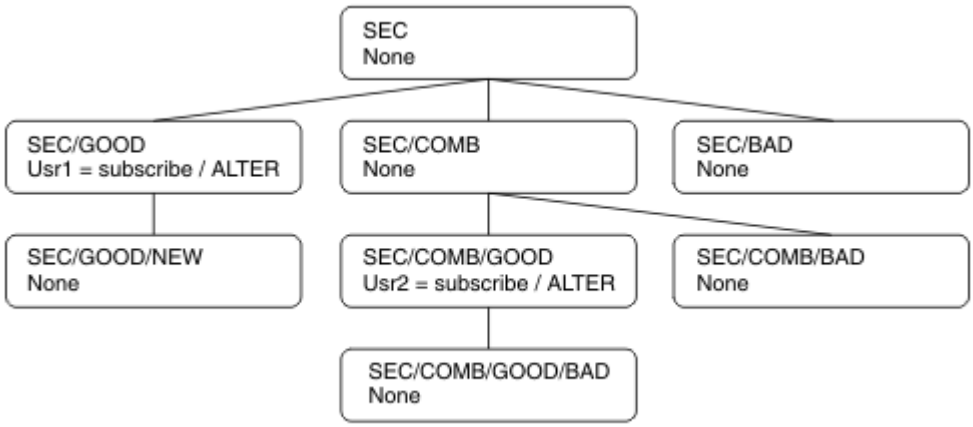
## 토픽 노드가 있는 토픽 문자열을 사용한 구독

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.

hobj를 지정하지 않았지만 공백인 경우, 다음 조건이 적용되면 관리 대상 큐가 작성됩니다.

- MQSO_MANAGED 옵션을 지정했습니다.
- 구독이 없습니다.
- 작성이 지정되어 있습니다.

hobj가 공백이고 기존 구독을 대체하거나 재개하는 경우, 이전에 제공된 목적지 큐가 관리되거나 관리되지 않습니다.

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 또 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, 236 페이지의 표 18에 표시된 대로 검사 수행 방식에 영향을 줍니다.

| 표 *18.* 명령에 대한 보안 검사에 사용되는 사용자 *ID* | | | |
|---|---|---|---|
| **명령** | **SUBUSER 지정 및 공백** | **SUBUSER 지정 및 완료** | **SUBUSER가 지정되지 않음** |
| | 관리자 ID 사용 | | 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 방법으로 토픽에 액세스 제어 설정이 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.

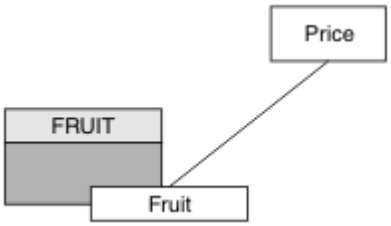



그림 *27.* 토픽 오브젝트 액세스의 예

| 표 *19.* 토픽 오브젝트 액세스의 예 | | |
|---|---|---|
| **주제** | **필요한 구독 액세스** | **토픽 오브젝트** |
| Price | 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.
   - 기타 플랫폼의 경우:

FRUIT 오브젝트에 대한 사용자 액세스를 부여하여 토픽 "Price/Fruit"를 구독할 수 있도록 USER1에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

Windows UNIX Linux **UNIX and Linux 시스템**

```
setmqaut -t topic -n FRUIT -p USER1 +sub
```

### 결과

USER1이 토픽 "Price/Fruit"를 구독하려는 경우 이에 성공합니다.

USER2가 토픽 "Price/Fruit"를 구독하려는 경우, 이에 실패하며 다음과 함께 MQRC_NOT_AUTHORIZED 메시지가 나타납니다.

- Windows UNIX Linux 기타 플랫폼에서는 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 236 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.

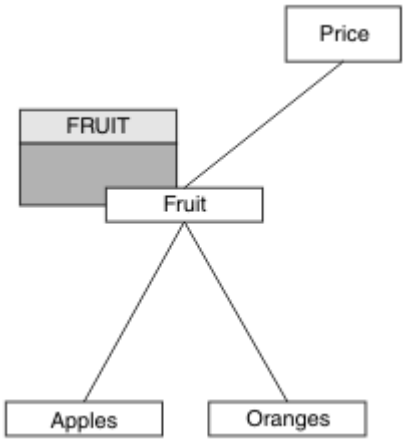



그림 *28.* 토픽 트리 내에서 토픽에 대한 액세스 부여의 예

표 *20.* 예제 토픽 및 토픽 오브젝트에 대한 액세스 요구사항

| 주제 | 필요한 구독 액세스 | 토픽 오브젝트 |
|---|---|---|
| Price | 사용자 없음 | 없음 |

표 *20.* 예제 토픽 및 토픽 오브젝트에 대한 액세스 요구사항 *(계속)*

| 주제 | 필요한 구독 액세스 | 토픽 오브젝트 |
|---|---|---|
| Price/Fruit | USER1 | 과일 |
| Price/Fruit/Apples | USER1 | |
| Price/Fruit/Oranges | USER1 | |

이전 태스크 USER1 에서 z/OS 의 hlq.SUBSCRIBE.FRUIT 프로파일에 대한 액세스 권한을 부여하고 다른 플랫폼의 FRUIT 프로파일에 대한 등록 액세스 권한을 부여하여 "Price/Fruit" 토픽에 대한 등록에 대한 액세스가 부여되었습니다. 이 단일 프로파일은 USER1에게 "Price/Fruit/Apples", "Price/Fruit/Oranges" 및 "Price/Fruit/#"의 구독 권한도 부여합니다.

USER1이 토픽 "Price/Fruit/Apples"를 구독하려는 경우 이에 성공합니다.

USER2가 토픽 "Price/Fruit/Apples"를 구독하려는 경우, 이에 실패하며 다음 내용과 함께 MQRC_NOT_AUTHORIZED 메시지가 나타납니다.

- z/OS에서 다음 메시지가 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 나타납니다.

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- 기타 플랫폼에서는 다음 권한 부여 이벤트가 제공됩니다.

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

다음에 유의하십시오.

- z/OS에서 수신하는 메시지는 동일한 토픽 오브젝트 및 프로파일이 액세스를 제어하므로 이전 태스크에서 수신한 메시지와 동일합니다.
- 다른 플랫폼에서 수신한 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 237 페이지의 『트리 내에서 더 깊은 토픽을 구독할 수 있는 액세스를 사용자에게 부여』에 설명된 설정을 사용합니다.

### 이 태스크 정보

이전 태스크에서는 "Price/Fruit/Apples" 토픽에 대한 USER2의 액세스가 거부되었습니다. 이 주제에서는 다른 토픽이 아닌 해당 토픽에 대한 액세스를 부여하는 방법을 알려줍니다.

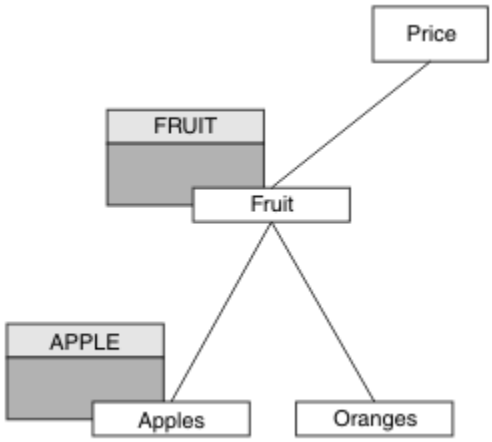



그림 *29.* 토픽 트리 내에서 특정 토픽에 대한 액세스 부여

표 *21.* 예제 토픽 및 토픽 오브젝트에 대한 액세스 요구사항

| 주제 | 필요한 구독 액세스 | 토픽 오브젝트 |
|---|---|---|
| Price | 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.
   - 기타 플랫폼의 경우:

     이전 태스크 USER1 에서 FRUIT 프로파일에 대한 사용자 등록 액세스 권한을 부여하여 "Price/Fruit/Apples" 토픽에 대한 등록에 대한 액세스가 부여되었습니다.

     또한 이 단일 프로파일은 "Price/Fruit/Oranges" 및 "Price/Fruit/#"에 대한 구독 액세스 권한이 USER1에게 부여되었고 이것은 새 토픽 오브젝트 및 이와 연관된 프로파일이 추가되어도 그대로 남아 있습니다.

     APPLE 프로파일에 대한 사용자 구독 액세스를 부여하여 토픽 "Price/Fruit/Apples"를 구독할 수 있도록 USER2에게 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

     Windows UNIX Linux **UNIX and Linux 시스템**

     ```
     setmqaut -t topic -n APPLE -p USER2 +sub
     ```

## 결과

z/OS에서, USER1이 토픽 "Price/Fruit/Apples"를 구독하려는 경우 hlq.SUBSCRIBE.APPLE 프로파일에 대한 첫 번째 보안 검사가 실패하지만 트리의 위로 이동하여 hlq.SUBSCRIBE.FRUIT 프로파일을 사용하면

USER1이 구독할 수 있으므로 구독이 성공하고 리턴 코드가 MQSUB 호출로 송신되지 않습니다. 하지만 첫 번째 검사에서 RACF ICH 메시지가 생성됩니다.

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

USER2가 토픽 "Price/Fruit/Apples"를 구독하려는 경우, 첫 번째 프로파일에서 보안 검사를 통과했으므로 이에 성공합니다.

USER2가 토픽 "Price/Fruit/Oranges"를 구독하려는 경우 이에 실패하며 다음 내용과 함께 MQRC_NOT_AUTHORIZED 메시지가 나타납니다.

- Windows UNIX Linux Windows, UNIX및 Linux 플랫폼에서 다음 권한 이벤트가 발생합니다.

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

이 설정을 사용하면 z/OS의 경우 콘솔에서 추가 ICH 메시지를 수신하게 됩니다. 토픽 트리를 다른 방법으로 보안 설정하면 이를 피할 수 있습니다.

## 추가 메시지를 방지하기 위한 액세스 제어 변경

이 주제는 둘 이상의 사용자가 토픽을 구독하기 위해 액세스를 부여하는 방법과 z/OS에서 추가 레이프 ICH408I 메시지를 피하는 방법을 설명하는 태스크 목록에서 네 번째입니다.

### 시작하기 전에

이 토픽에서는 추가 오류 메시지를 피할 수 있도록 238 페이지의 『트리 내에서 더 깊은 토픽만 구독할 수 있는 또 다른 사용자 액세스 부여』에 설명된 설정이 개선되었습니다.

### 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.

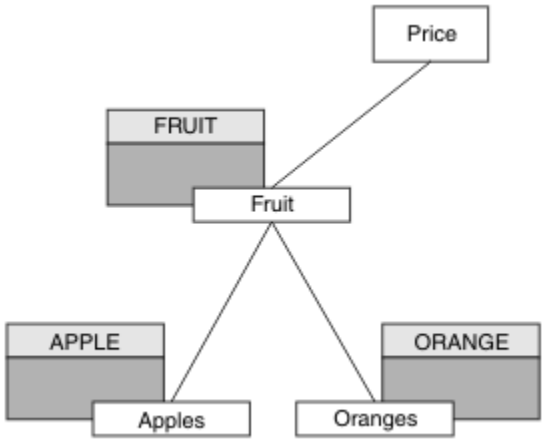



그림 *30.* 추가 메시지를 방지하기 위해 액세스 제어를 부여하는 예

다음과 같이 새 토픽 오브젝트를 정의하십시오.

### 프로시저

1. MQSC 명령 DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')을 실행하십시오.
2. 다음과 같이 액세스를 부여하십시오.

- 기타 플랫폼의 경우:

  플랫폼별 권한 부여 명령을 사용하여 동등한 액세스를 설정하십시오.

  Windows UNIX Linux **UNIX and Linux 시스템**

```
setmqaut -t topic -n ORANGE -p USER1 +sub
setmqaut -t topic -n APPLE -p USER1 +sub
```

### 결과

z/OS에서, USER1이 토픽 "Price/Fruit/Apples"를 구독하려는 경우 hlq.SUBSCRIBE.APPLE 프로파일에 대한 첫 번째 보안 검사가 성공합니다.

이와 유사하게 USER2가 토픽 "Price/Fruit/Apples"를 구독하려는 경우 첫 번째 프로파일에서 보안 검사를 통과했으므로 이에 성공합니다.

USER2가 토픽 "Price/Fruit/Oranges"를 구독하려는 경우 이에 실패하며 다음 내용과 함께 MQRC_NOT_AUTHORIZED 메시지가 나타납니다.

- Windows UNIX Linux 기타 플랫폼에서는 다음 권한 부여 이벤트가 제공됩니다.

```
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.

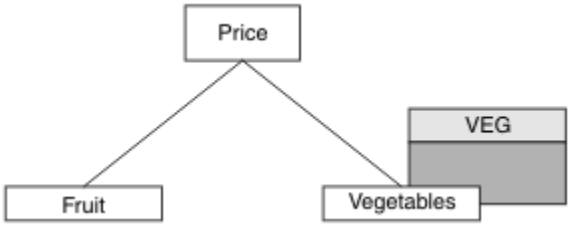



그림 *31.* 토픽에 대한 발행 액세스 권한 부여

표 *22.* 발행 액세스 요구사항의 예

| 주제 | 필요한 발행 액세스 | 토픽 오브젝트 |
|---|---|---|
| Price | 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.
   - 기타 플랫폼의 경우:

     VEG 프로파일에 대한 사용자 액세스를 부여하여 토픽 `"Price/Vegetables"`를 발행할 수 있도록 USER1에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

     Windows UNIX Linux **UNIX and Linux 시스템**

     ```
     setmqaut -t topic -n VEG -p USER1 +pub
     ```

### 결과

USER1이 토픽 `"Price/Vegetables"`를 발행하려는 경우 이에 성공합니다. 즉, MQOPEN 호출이 성공합니다.

USER2가 토픽 `"Price/Vegetables"`를 발행하려는 경우 MQOPEN 호출이 실패하고 다음과 함께 `MQRC_NOT_AUTHORIZED` 메시지가 나타납니다.

- Windows UNIX Linux 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"
```

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 241 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.

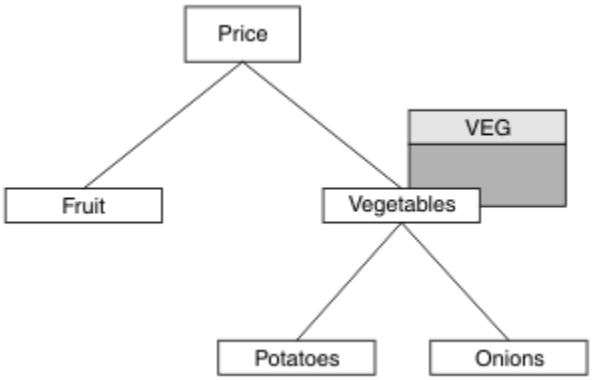



그림 *32.* 토픽 트리 내에서 토픽에 대한 발행 액세스 부여

| 표 *23.* 발행 액세스 요구사항의 예 | | |
|---|---|---|
| **주제** | **필요한 구독 액세스** | **토픽 오브젝트** |
| Price | 사용자 없음 | 없음 |
| Price/Vegetables | USER1 | VEG |
| Price/Vegetables/Potatoes | USER1 | |
| Price/Vegetables/Onions | USER1 | |

In the previous task USER1 was granted access to publish topic "Price/Vegetables/Potatoes" by granting it access to the hlq.PUBLISH.VEG profile on z/OS or publish access to the VEG profile on other platforms. 또한 이 단일 프로파일은 "Price/Vegetables/Onions"에 공개할 수 있는 USER1 액세스 권한을 부여합니다.

USER1이 토픽 "Price/Vegetables/Potatoes"에서 발행하려는 경우 이에 성공합니다. 즉, MQOPEN 호출이 성공합니다.

USER2가 토픽 "Price/Vegetables/Potatoes"를 구독하려는 경우 이에 실패합니다. 즉, MQOPEN 호출이 실패하며 MQRC_NOT_AUTHORIZED 메시지가 다음과 함께 나타납니다.

- z/OS에서 다음 메시지가 시도한 토픽 트리를 통해 전체 보안 경로를 표시하는 콘솔에 나타납니다.

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- 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/Potatoes"
```

다음에 유의하십시오.

- z/OS에서 수신하는 메시지는 동일한 토픽 오브젝트 및 프로파일이 액세스를 제어하므로 이전 태스크에서 수신한 메시지와 동일합니다.
- 다른 플랫폼에서 수신한 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 242 페이지의 『트리 내에서 더 깊은 토픽에 발행할 수 있는 액세스를 사용자에게 부여』에 설명된 설정을 사용합니다.

### 이 태스크 정보

이전 태스크에서는 USER1에게 토픽 "Price/Fruit"를 구독할 수 있는 액세스를 부여했습니다. 이 주제에서는 사용자에게 해당 토픽에 발행하기 위한 액세스를 부여하는 방법을 알려줍니다.

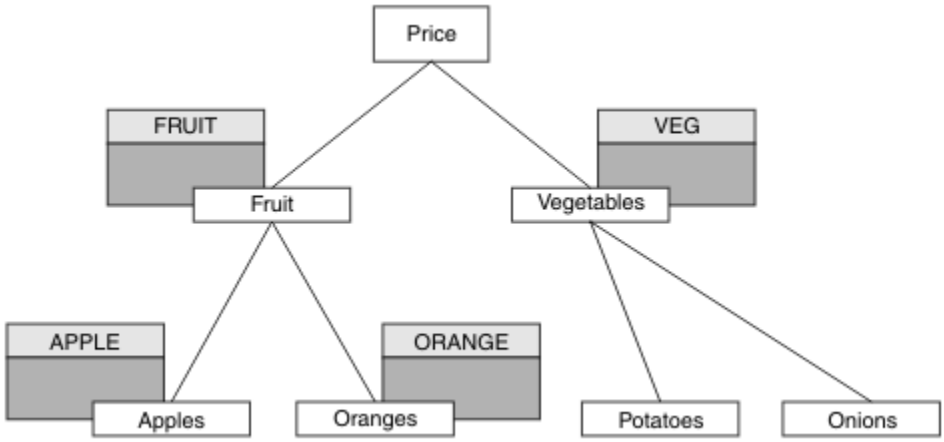



그림 *33.* 발행 및 구독을 위한 액세스 부여

표 *24.* 액세스 요구사항 발행 및 구독 예

| 주제 | 필요한 구독 액세스 | 필요한 발행 액세스 | 토픽 오브젝트 |
|---|---|---|---|
| Price | 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.

- 기타 플랫폼의 경우:

  FRUIT 프로파일에 대한 사용자 발행 액세스를 부여하여 토픽 "Price/Fruit"를 발행할 수 있도록 USER1에 액세스를 부여하십시오. 플랫폼에 대한 권한 부여 명령을 사용하여 다음을 수행하십시오.

  Windows UNIX Linux **UNIX and Linux 시스템**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

## 결과

z/OS에서 USER1이 토픽 "Price/Fruit"를 발행하려 하는 경우 MQOPEN 호출에 대한 보안 검사를 통과합니다.

USER2가 토픽 "Price/Fruit"에서 발행하려는 경우, 발행에 실패하며 다음 내용과 함께 MQRC_NOT_AUTHORIZED 메시지가 나타납니다.

- Windows UNIX Linux Windows, UNIX및 Linux 플랫폼에서 다음 권한 이벤트가 발생합니다.

  ```
  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 *25.* 보안에서 발생하는 액세스 권한의 전체 목록 예

| 주제 | 필요한 구독 액세스 | 필요한 발행 액세스 | 토픽 오브젝트 |
|---|---|---|---|
| Price | 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 | | | |

토픽 트리 내에 있는 여러 레벨의 보안 액세스에 대한 여러 요구사항을 갖고 있는 경우, z/OS 콘솔 로그에서 관련 없는 보안 경고를 수신하지 않도록 주의깊게 계획을 세워야 합니다. 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 *26.* 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. |

| 표 26. 기본 발행 컨텍스트 정보 (계속) | |
|---|---|
| **MQMD의 필드** | **사용된 값** |
| *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.

# IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security(AMS)은 종료 애플리케이션에 영향을 주지 않으면서 IBM WebSphere MQ Advanced Message Security 네트워크를 통한 민감한 데이터에 대한 상위 레벨 보호를 제공하는 별도로 라이센스가 있는 IBM WebSphere MQ Advanced Message Security의 컴포넌트입니다.

# IBM WebSphere MQ Advanced Message Security 개요

IBM WebSphere 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 IBM WebSphere MQ Advanced Message Security를 사용할 수 있습니다.

**관련 참조**

IBM WebSphere MQ AMS 메시지에서 사용되는 GSKit 리턴 코드

## 버전 7.0.1과 버전 7.5 사이에서 변경된 작동

IBM Advanced Message Security 가 WebSphere MQ 7.5의 구성요소가 되면서, IBM WebSphere MQ AMS 기능의 일부 측면이 변경되었으며, 기존 응용프로그램, 관리 스크립트 또는 관리 프로시저에 영향을 줄 수 있습니다.

큐 관리자를 버전 7.5로 업그레이드하기 전에 다음과 같은 변경사항 목록을 주의하여 검토하십시오. IBM WebSphere MQ 버전 7.5로의 시스템 마이그레이션을 시작하기 전에 기존 애플리케이션, 스크립트 및 프로시저 변경 계획을 세워야 하는지 여부를 결정하십시오.

- IBM WebSphere MQ AMS 설치는 WebSphere MQ 설치 프로세스의 일부입니다.
- IBM WebSphere MQ AMS 보안 기능은 설치를 통해 사용할 수 있고 보안 정책으로 제어합니다. IBM WebSphere MQ AMS가 데이터 인터셉트를 시작을 허용하기 위해 인터셉터를 사용으로 설정할 필요가 없습니다.
- IBM WebSphere MQ AMS in WebSphere MQ version 7.5 does not require the use of the **cfgmqs** command as in the stand-alone version of IBM WebSphere MQ AMS.

## IBM WebSphere MQ Advanced Message Security 의 기능 및 함수

Advanced Message Security는 메시지 레벨에서 데이터 서명 및 암호화를 제공하기 위해 WebSphere MQ 보안 서비스를 확장합니다. 확장된 서비스를 통해 메시지 데이터가 원래 큐에 배치된 시기와 검색된 시기 사이에서 수정되지 않도록 보장됩니다. 또한, IBM WebSphere MQ AMS는 메시지 데이터 송신자가 서명된 메시지를 대상 큐에 배치할 권한이 있는지 확인합니다.

다음은 IBM WebSphere MQ AMS 기능의 전체 목록입니다.

- WebSphere 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.
- WebSphere MQ에 대해 기존 독점 및 고객 작성 애플리케이션을 보안합니다.

## 오류 핸들링

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 - WebSphere MQ 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.

WebSphere MQ AMS는 SYSTEM.PROTECTION.ERROR.QUEUE 는 오류 처리 큐로 대기합니다. IBM WebSphere MQ AMS가 SYSTEM.PROTECTION.ERROR.QUEUE 앞에는 MQDLH 헤더가 있습니다.

WebSphere MQ 관리자도 SYSTEM.PROTECTION.ERROR.QUEUE 는 다른 큐를 가리키는 알리어스 큐로 설정됩니다.

## 키 개념

도구가 작동하는 방법 및 이를 효율적으로 사용하는 방법을 이해하려면 Advanced Message Security에서 키 개념에 대해 배우십시오.

### *PKI(Public Key Infrastructure)*

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 애플리케이션의 ID는 서명되거나 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**

270 페이지의 『키 저장소 및 인증서 사용』

WebSphere MQ 애플리케이션에 투명한 암호 보호를 제공하기 위해 Advanced Message Security는 공개 키 인증서 및 개인 키가 저장되는 키 저장소 파일을 사용합니다.

## 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 Security에서는 Java 및 고유 애플리케이션에 자체 서명된 인증서를 지원합니다.

**관련 개념**

7 페이지의 『암호화』

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

## 오브젝트 권한 관리자

OAM(Object Authority Manager)은 WebSphere MQ 제품에서 제공되는 권한 서비스 컴포넌트입니다.

Advanced Message Security 엔티티에 대한 액세스는 WebSphere 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.

- MQI를 통해 Advanced Message Security 오브젝트에 대한 액세스. 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

## 지원되는 기술

Advanced Message Security는 보안 인프라를 제공하기 위해 몇몇 기술 컴포넌트에 의존합니다.

Advanced Message Security는 다음 WebSphere MQ API(Application Programming Interface)를 지원합니다.

- 메시지 큐 인터페이스(MQI)
- WebSphere MQ JMS (Java Message Service) 1.0.2 및 1.1.
- WebSphere MQ Java의 기본 클래스
- 비관리 모드에서 .Net을 위한 WebSphere MQ 클래스

**참고:** Advanced Message Security는 X.509 준수 인증 기관을 지원합니다.

## 알려진 제한사항

IBM WebSphere MQ Advanced Message Security의 제한사항에 대해 학습하십시오.

- 다음 IBM WebSphere MQ 옵션은 지원되지 않습니다.
  - 발행/구독.
  - 채널 데이터 변환.
  - 유통목록.
  - 애플리케이션 메시지 세그먼트화.
  - HP-UX 플랫폼에서 API 엑시트를 사용하여 비스레드 애플리케이션의 사용.
  - 관리 모드에서 .NET에 대한 IBM WebSphere MQ 클래스(클라이언트 또는 바인딩 연결).
  - .NET에 대한 메시지 서비스 클라이언트(XMS) 애플리케이션
  - C/C++을 위한 메시지 서비스 클라이언트(XMS supportPac IA94) 애플리케이션.
- 모든 Java 애플리케이션은 IBM Java Runtime에 의존합니다.

  IBM WebSphere MQ Advanced Message Security에서는 다른 벤더가 제공하는 JRE를 지원하지 않습니다.
- 클라이언트 모드에서 IBM WebSphere MQ Advanced Message Security 를 사용하는 JMS및 Java 클라이언트 응용프로그램

  Any JMS, or Java, client application (including IBM WebSphere MQ Explorer and IBM WebSphere MQ Managed File Transfer agents) cannot use IBM WebSphere MQ Advanced Message Security in client mode with a WebSphere MQ queue manager earlier than Version 7.5.

  메시지 보호 정책을 사용하려면, 이러한 애플리케이션이 IBM WebSphere MQ Version 7.5 큐 관리자와 상호작용하거나 로컬 바인딩 모드에서 애플리케이션과 같은 시스템의 큐 관리자에 연결되어 있어야 합니다.

- 이러한 인증서를 사용한 IBM WebSphere MQ Advanced Message Security 인터셉터의 동작은 정의되지 않았으므로 단일 키 저장소 파일에 같은 식별 이름으로 둘 이상의 인증서를 넣지 않아야 합니다.
- IBM WebSphere MQ Version 7.5 자원 어댑터는 IBM WebSphere MQ Advanced Message Security를 지원하지 않습니다. 애플리케이션 서버 환경 내에서 실행 중인 IBM WebSphere MQ classes for JMS 또는 IBM WebSphere MQ classes for Java 애플리케이션과 함께 메시지 보호가 필요한 경우, 다음과 같습니다.
  - Version 8.0 또는 이후 자원 어댑터를 사용하도록 애플리케이션 서버를 구성해야 합니다.
  - 또는 MCA(Message Channel Agent) 인터셉션을 사용해야 합니다.

## 사용자 시나리오

Advanced Message Security로 이룰 수 있는 비즈니스 목적에 대해 이해하려면 가능한 시나리오에 친숙해지십시오.

### *Windows* 플랫폼용 빠른 시작 안내서

이 안내서를 사용하여 Windows 플랫폼을 위한 메시지 보안을 제공하기 위해 IBM Advanced Message Security을 빠르게 구성할 수 있습니다. 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.

- SERVER
- 개발 툴킷(샘플 프로그램용)
- Advanced Message Security

세부사항은 Windows 시스템의 IBM WebSphere MQ 기능 을 (를) 참조하십시오.

운영 체제에서 적절한 WebSphere MQ 명령을 찾아 실행할 수 있도록 **setmqenv** 명령을 사용하여 현재 환경을 초기화하는 방법에 대한 정보는 setmqenv를 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 Security 인터셉터는 표준 WebSphere MQ 인터페이스를 통해 WebSphere MQ 인프라를 입력하는 지점에서 메시지를 서명 및 암호화합니다. 이 기본 설정은 WebSphere MQ에서 수행되고 다음 단계에서 구성됩니다.

WebSphere MQ 탐색기를 사용하여 모든 기본 마법사 설정을 사용하여 큐 관리자 QM_VERIFY_AMS및 TEST.Q라는 로컬 큐를 작성하거나 \WebSphere 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 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 작성하고 이러한 두 사용자 모두에게 HOMEPATH 및 HOMEDRIVE가 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 2명의 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
   ```

## 결과

그러면 사용자가 작성되고 사용자에게 필요한 권한이 부여됩니다.

## 다음에 수행할 작업

단계가 올바르게 수행되었는지 확인하려면 254 페이지의 『7. 설정 테스트』 섹션에 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 alice와 bob을 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 JRE의 일부인 **keytool** 명령을 사용하여 JKS키 저장소 및 인증서를 작성해야 합니다 (자세한 내용은 260 페이지의 『Java 클라이언트를 위한 빠른 시작 안내서』 참조). 다른 모든 언어와 로컬 바인딩을 사용하는 자바 애플리케이션의 경우 이 안내서의 단계가 올바른 것입니다.

## 프로시저

1. IBM 키 관리 GUI(strmqikm.exe)를 사용하여 사용자 alice에 대한 새 키 데이터베이스를 작성하십시오.

   ```
   Type:       CMS
   Filename:   alicekey.kdb
   Location:   C:/Documents and Settings/alice/AMS
   ```

   **참고:**

- 데이터베이스를 보안 설정하기 위해 강력한 비밀번호를 사용하는 것이 권장됩니다.
- **파일에 비밀번호 숨김** 선택란이 선택되어 있는지 확인하십시오.

2. 키 데이터베이스 컨텐츠 보기를 **개인 인증서**로 변경하십시오.
3. **새 자체 서명**을 선택하십시오. 이 시나리오에서는 자체 서명된 인증서가 사용됩니다.
4. 다음 필드를 사용하여 암호화에 사용할 사용자 `alice`를 식별하는 인증서를 작성하십시오.

```
Key label: Alice_Cert
Common Name: alice
Organisation: IBM
Country: GB
```

**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.
- **Key label** 매개변수는 인터셉터가 필수 정보를 수신하기 위해 찾아볼 인증서의 이름을 지정합니다.
- **Common Name** 및 선택적 매개변수는 **식별 이름**(DN)의 세부사항을 지정하고 이는 각 사용자에게 고유해야 합니다.

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

키 데이터베이스 및 인증서가 located.This 인 디렉토리에 대한 Advanced Message Security 인터셉터를 가리켜 일반 텍스트 양식으로 해당 정보를 보유하는 `keystore.conf` 파일을 통해 수행해야 합니다. 각 사용자에게는 별도의 `keystore.conf` 파일이 있어야 합니다. `alice` 및 bob 모두에 대해 이 단계를 완료해야 합니다.

`keystore.conf`의 컨텐츠는 다음 양식이어야 합니다.

```
cms.keystore = <dir>/keystore_file
cms.certificate = certificate_label
```

### 예

이 시나리오의 경우 `keystore.conf`의 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 레이블은 공백을 포함할 수 있으므로 "Alice_Cert" 및 "Alice_Cert "가 두 개의 다른 인증서로 인식됩니다. 그러나 혼동을 피하기 위해 레이블의 이름에는 공백을 사용하지 않는 것이 좋습니다.
- 키 저장소 형식에는 CMS(Cryptographic Message Syntax), JKS(Java Keystore), JCEKS(Java Cryptographic Extension Keystore)가 있습니다. 자세한 정보는 270 페이지의 『키 저장소 구성 파일의 구조 (keystore.conf)』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`의 비기본 위치를 사용하는 방법에 대한 자세한 정보는 270 페이지의 『키 저장소 및 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. alice를 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/alice/AMS/bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm

runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/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

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_AMS에서 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. 예에서 메시지는 SHA1 알고리즘으로 서명되고 AES256 알고리즘으로 암호화됩니다. alice는 유일한 유효한 송신자이고 bob은 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 데이터베이스로부터 각각의 사용자의 인증서에 명시된 것과 정확하게 일치한다.

### 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

### 이 태스크 정보
다른 사용자 하에서 다른 프로그램을 실행하여 애플리케이션이 제대로 구성되었는지 확인할 수 있습니다.

### 프로시저
1. 사용자 alice로서 실행하도록 사용자를 전환하십시오.

   cmd.exe를 마우스 오른쪽 단추로 클릭하고 **실행 도구...**를 선택하십시오. 메시지가 표시되면 사용자 alice로 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 bob로서 실행하도록 사용자를 전환하십시오.

   cmd.exe를 마우스 오른쪽 단추로 클릭하고 **실행 도구...**를 선택하여 다른 창을 여십시오. 메시지가 표시되면 사용자 bob로 로그인하십시오.
5. 사용자 Bob으로서, 샘플 애플리케이션을 사용하여 메시지를 가져오십시오.

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

### 결과
두 사용자 모두에 대해 애플리케이션이 올바르게 구성된 경우, 사용자 alice의 메시지는 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.

## *UNIX* 플랫폼용 빠른 시작 안내서

이 안내서를 사용하여 플랫폼 메시지 보안을 제공하도록 IBM Advanced Message Security 를 신속하게 구성할 수 있습니다. 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
- SERVER
- 샘플 프로그램
- IBM Global Security Kit
- MQ Advanced Message Security

각 특정 플랫폼에서 컴포넌트 이름에 대해서는 다음 주제를 참조하십시오.

- Linux 시스템의IBM WebSphere MQ 구성요소
- HP-UX 시스템의IBM WebSphere MQ 구성요소
- AIX 시스템의IBM WebSphere MQ 구성요소
- Solaris 시스템의IBM WebSphere 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 Security 인터셉터는 표준 WebSphere MQ 인터페이스를 통해 WebSphere MQ 인프라를 입력하는 지점에서 메시지를 서명 및 암호화합니다. 이 기본 설정은 WebSphere MQ에서 수행되고 다음 단계에서 구성됩니다.

WebSphere MQ 탐색기를 사용하여 모든 기본 마법사 설정을 사용하여 큐 관리자 QM_VERIFY_AMS및 TEST.Q라는 로컬 큐를 작성하거나 <MQ_INSTALL_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** 을 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 2명의 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
```

### 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 260 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 alice와 bob을 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 JRE의 일부인 **keytool** 명령을 사용하여 JKS키 저장소 및 인증서를 작성해야 합니다 (자세한 내용은 260 페이지의 『Java 클라이언트를 위한 빠른 시작 안내서』 참조). 다른 모든 언어와 로컬 바인딩을 사용하는 자바 애플리케이션의 경우 이 안내서의 단계가 올바른 것입니다.

## 프로시저

1. 사용자 alice의 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 별개의 keystore.conf 파일이 있어야 합니다. alice 및 bob 모두에 대해 이 단계를 완료해야 합니다.

keystore.conf의 컨텐츠는 다음 양식이어야 합니다.

```
cms.keystore = <dir>/keystore_file
cms.certificate = certificate_label
```

### 예

이 시나리오의 경우 keystore.conf의 컨텐츠는 다음과 같습니다.

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

**참고:**

- 키 저장소 파일의 경로는 파일 확장자 없이 제공되어야 합니다.
- 키 저장소 형식에는 CMS(Cryptographic Message Syntax), JKS(Java Keystore), JCEKS(Java Cryptographic Extension Keystore)가 있습니다. 자세한 정보는 270 페이지의 『키 저장소 구성 파일의 구조 (keystore.conf)』의 내용을 참조하십시오.
- HOME/.mqs/keystore.conf는 Advanced Message Security가 keystore.conf 파일을 검색하는 기본 위치입니다. keystore.conf의 비기본 위치를 사용하는 방법에 대한 자세한 정보는 270 페이지의 『키 저장소 및 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. alice를 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_AMS에서 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 예에서 메시지는 SHA1 알고리즘을 사용하여 alice 사용자에 의해 서명되고 256비트 AES 알고리즘을 사용하여 암호화됩니다.alice는 유일한 유효한 송신자이고 bob은 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 데이터베이스로부터 각각의 사용자의 인증서에 명시된 것과 정확하게 일치한다.

## 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

## 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 alice로서 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

두 사용자 모두에 대해 애플리케이션이 올바르게 구성된 경우, 사용자 alice의 메시지는 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.

## *Java* 클라이언트를 위한 빠른 시작 안내서
클라이언트 바인딩을 사용하여 연결하는 Java 애플리케이션에 대한 메시지 보안을 제공하도록 IBM Advanced Message Security 를 신속하게 구성하려면 이 안내서를 사용하십시오. 이 안내서를 끝까지 착실하게 따라가면 사용자 ID를 확인하기 위한 키 저장소를 작성하고 큐 관리자에 대한 서명/암호화 정책을 정의하는 작업을 마치게 될 것입니다.

### 시작하기 전에
**빠른 시작 안내서**(Windows 또는 UNIX)에 설명된 것처럼 적절한 컴포넌트가 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 Security 인터셉터는 표준 WebSphere MQ 인터페이스를 통해 WebSphere MQ 인프라를 입력하는 지점에서 메시지를 서명 및 암호화합니다. 이 기본 설정은 WebSphere 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 송신자인 alice와 수신자인 bob이라는 두 사용자가 나옵니다. 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** 을 참조하십시오.

## 프로시저

1. 사용 중인 플랫폼에 대한 **빠른 시작 안내서**(Windows 또는 UNIX)에 설명된 바와 같이 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
```

3. 또한 2명의 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
```

## 결과

그러면 사용자가 작성되고 사용자에게 필요한 권한이 부여됩니다.

## 다음에 수행할 작업

단계가 올바르게 수행되었는지 확인하려면 264 페이지의 『7. 설정 테스트』 절에 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 시스

템에서는 각 사용자는 고유의 개인용 키 저장소를 가지고 있을 것입니다. 마찬가지로, 이 안내서에서는 alice와 bob을 위한 키 데이터베이스를 작성하고 이들 간에 사용자 인증서를 공유합니다.

**참고:** 이 안내서에서는 클라이언트 바인딩을 사용하여 연결하는 Java로 작성된 샘플 애플리케이션을 사용합니다. 로컬 바인딩이나 C 애플리케이션을 사용하는 Java 애플리케이션을 사용할 계획이라면, **runmqakm** 명령을 사용하여 CMS 키 저장소와 인증서를 작성해야 합니다. 이것은 **빠른 시작 안내서**(Windows 또는 UNIX)에 표시되어 있습니다.

## 프로시저

1. 키 저장소를 작성할 디렉토리를 작성하십시오. 예: /home/alice/.mqs. 사용자 플랫폼의 **빠른 시작 안내서**(Windows 또는 UNIX)에서 사용하는 것과 같은 디렉토리에서 작성하려 합니다.

   **참고:** 이 디렉토리는 다음 단계에서 *keystore-dir*입니다.
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 서명된 인증서를 사용하지 말고 인증 기관에서 서명한 인증서를 사용하는 것이 좋습니다.
   - 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. UNIX에서 키 저장소를 읽을 수 있는지 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 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.conf 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.conf 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** (250 페이지의 『Windows 플랫폼용 빠른 시작 안내서』 )따랐기 때문에 이미 keystore.conf 이 있는 경우, 위의 행에 추가할 기존 항목을 편집할 수 있습니다.
- 자세한 정보는 270 페이지의 『키 저장소 구성 파일의 구조 (keystore.conf)』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.

**참고:** 추출 및 내보내기라는 용어는 서로 다른 인증 도구에서 서로 다르게 사용됩니다. 예를 들어, IBM GSKit Keyman(ikeyman) 도구는 인증서(공개 키) 추출과 개인 키 내보내기를 구분합니다. 이 구분은 두 옵션을 모두 제공하는 도구에 아주 중요합니다. 실수로 내보내기를 사용하면 개인 키가 누출되어 애플리케이션이 완전히 훼손될 수 있기 때문입니다. 구분이 이렇게 중요하므로 WebSphere MQ 문서에서는 이들 용어를 일관되게 사용하도록 노력합니다. 그러나 Java keytool은 공개 키만 추출하는 *exportcert* 명령행 옵션을 제공합니다. 이런 이유로 다음 프로시저에서는 인증서 내보내기를 언급할 때 *exportcert* 옵션을 사용합니다.

## 프로시저

1. alice를 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. alice를 식별하는 인증서를 bob이 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. bob에 대해 단계를 반복하십시오.

## 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.

```
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 setmqspl의 내용을 참조하십시오. 각 정책 이름은 적용되는 큐 이름과 동일해야 합니다.

### 예

이는 `TEST.Q` 큐에 정의되고, `SHA1` 알고리즘을 사용하여 `alice` 사용자가 서명하고 `bob` 사용자에 대해 256비트 AES 알고리즘을 사용하여 암호화된 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 데이터베이스로부터 각각의 사용자의 인증서에 명시된 것과 정확하게 일치한다.

## 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 제한 없는 JCE 정책 파일이 설치되었는지 확인하십시오.

**참고:** WebSphere MQ에 제공된 Java의 버전에 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. 여러 사용자의 프로그램 실행에 관한 세부사항은 사용 중인 플랫폼의 **빠른 시작 안내서**(Windows 또는 UNIX)를 참조하십시오.

## 프로시저

1. 이 JMS 샘플 애플리케이션을 실행하려면, JSM용 IBM WebSphere MQ 클래스에서 사용된 환경 변수에 표시된 대로 사용자 플랫폼에 CLASSPATH 설정을 사용하여 샘플 디렉토리가 포함되는지 확인하십시오.
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

두 사용자 모두에 대해 애플리케이션이 올바르게 구성된 경우, 사용자 `alice`의 메시지는 `bob`이 가져오기 애플리케이션을 실행할 때 표시됩니다.

## 리모트 큐 보호

리모트 큐 연결을 완전히 보호하려면 메시지가 전송되는 리모트 큐와 로컬 큐에 동일한 정책이 설정되어야 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 WebSphere MQ에 전달되기 전에 암호화됩니다. Advanced Message Security가 리모트 큐에 놓인 메시지를 처리한 후에 WebSphere 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)

## *WebSphere Message Broker*를 사용하여 보호된 메시지 라우팅

IBM Advanced Message Security은 WebSphere Message Broker 버전 8.0.0.1 이상이 설치된 인프라에서 메시지를 보호할 수 있습니다. WebSphere Message Broker 환경에서 보안을 적용하기 전에 두 제품의 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 WebSphere Message Broker를 통해 전달되는 메시지를 보안하기 위해 WebSphere Message Broker가 해당 컨텐츠를 모르고 메시지를 처리하도록 허용하거나 ( 시나리오 1) 메시지를 수신 및 전송할 수 있는 권한이 있는 사용자로 설정할 수 있음을 의미합니다 ( 시나리오 2).

시나리오 *1 - Message Broker*가 메시지 컨텐츠를 볼 수 없음

## 시작하기 전에

WebSphere Message Broker가 기존 큐 관리자에 연결되어 있어야 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`에 기반하여 메시지는 *bob's* (`QBOB`) 로 라우트됩니다.[1](`QCECIL`) 또는 기본 (`QDEF`) 큐입니다. Advanced Message Security은 메시지 페이로드(payload)만 보호하고 해당 헤더 및 특성은 보호되지 않은 상태로 남으며, WebSphere Message Broker에서 읽을 수 있기 때문에 라우팅이 가능합니다. Advanced Message Security *alice*, *bob* 및 *cecil*에서만 사용됩니다. WebSphere Message Broker에 대해 이를 설치하거나 구성하지 않아도 됩니다.

WebSphere Message Broker는 메시지를 복호화하려는 시도를 피하기 위해 보호되지 않는 알리어스 큐에서 보호된 메시지를 수신합니다. 보호된 큐를 직접 사용할 계획이었으면 메시지는 복호화 불가능한 것으로서 DEAD LETTER 큐에 놓일 것입니다. 메시지는 WebSphere Message Broker에 의해 라우팅되어 대상 큐에 변경되지 않은 채로 도착합니다. 따라서 원래 작성자의 서명이 그대로 유지되고(*bob*과 *cecil*은 모두 *alice*가 보낸 메시지만 허용함) 전과 같이 보호됩니다(*bob*과 *cecil*만 메시지를 읽을 수 있음). WebSphere Message Broker는 라우팅된 메시지를 보호되지 않는 알리어스에 넣습니다. 수신인은 IBM WebSphere MQ AMS가 메시지를 투명하게 복호화하는 보호된 출력 큐로부터 메시지를 검색합니다.

[1] cecil

## 프로시저

1. **빠른 시작 안내서** (Windows 또는 UNIX) 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. **태스크 5. 빠른 시작 안내서 (Windows 또는 UNIX) 의 인증서 공유하기**
3. *alice*가 *bob* 및 *cecil*에 대해 암호화된 메시지를 전송할 수 있도록 *bob* 및 *cecil*의 인증서를 *alice*에게 제공하십시오.

   이전 단계에 지정된 방법을 사용하여 이를 수행하십시오.
4. 큐 관리자에서 QIN, QBOB, QCECIL 및 QDEF라는 로컬 큐를 정의하십시오.

   ```
   DEFINE QLOCAL(QIN)
   ```
5. QIN 큐의 보안 정책을 적합한 구성에 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. WebSphere Message Broker에서 메시지의 routeTo 특성에 따라 AIN 알리어스 큐에 도착하는 메시지를 BOB, CECIL 또는 DEF 노드로 라우팅하는 메시지 플로우를 작성하십시오. 이를 수행하려면, 다음 단계를 따르십시오.

   a) IN이라는 MQInput 노드를 작성하고 AIN 알리어스를 해당 큐 이름으로 지정하십시오.

   b) BOB, CECIL, DEF라는 MQOutput 노드를 작성하고 알리어스 큐 ABOB, ACECIL, 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. WebSphere Message Broker 런타임 컴포넌트에 메시지 플로우를 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. 그러면 AIN 알리어스 큐에서 WebSphere Message Broker에 의해 보호된 양식으로 메시지가 검색됩니다. WebSphere Message Broker는 모든 특성으로, 암호화되지 않은 routeTo 특성을 읽는 메시지를 라우트할 위치를 결정합니다. WebSphere Message Broker는 보호되지 않는 적절한 알리어스에 메시지를 배치하여 추가적인 보호를 회피합니다. 큐로부터 *bob* 또는 *cecil*에 의해 수신되면 메시지를 복호화되고 디지털 서명이 확인됩니다.

시나리오 *2 - Message Broker*가 메시지 컨텐츠를 볼 수 있음

## 이 태스크 정보

이 시나리오에서는 개인 사용자로 구성된 한 그룹이 WebSphere Message Broker로 메시지를 보낼 수 있도록 허용됩니다. 또 다른 그룹은 WebSphere Message Broker에서 작성된 메시지를 수신하도록 권한 부여됩니다. 파티와 WebSphere Message Broker 간 전송은 도청될 수 없습니다.

WebSphere Message Broker는 큐가 열릴 때만 보호 정책과 인증서를 읽으므로, 변경사항을 적용하려면 보호 정책을 업데이트한 후 실행 그룹을 다시 로드해야 합니다.

```
mqsireload execution-group-name
```

WebSphere Message Broker가 메시지 페이로드를 읽거나 서명할 수 있는 권한 부여된 당사자로 간주되는 경우, WebSphere Message Broker 서비스를 시작하는 사용자에 대해 Advanced Message Security 를 구성해야 합니다. 메시지를 큐에 넣거나 가져오는 동일한 사용자 또는 WebSphere Message Broker 애플리케이션을 작성 및 배치하는 사용자가 아니어도 됩니다.

## 프로시저

1. **빠른 시작 안내서** (Windows 또는 UNIX) 에 설명된 대로 Advanced Message Security 를 사용하도록 앨리, 귀찮게 하다, 세실 및 멍하다 및 WebSphere Message Broker 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 WebSphere Message Broker 서비스 사용자에 제공합니다.

   *alice*, *bob*, *cecil* 및 *dave*를 식별하는 인증서 각각을 외부 파일로 추출한 다음, 추출된 인증서를 WebSphere Message Broker 키 저장소에 추가하여 이를 수행하십시오. **태스크 5. 빠른 시작 안내서 (Windows 또는 UNIX) 의 인증서 공유하기**

3. WebSphere Message Broker 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 위해 WebSphere Message Broker 서비스 사용자의 인증서가 필요합니다. WebSphere Message Broker 서비스 사용자는 메시지의 작성자를 확인하기 위해 *alice*

와 *bob*의 인증서가 필요합니다. WebSphere Message Broker 서비스 사용자는 메시지를 암호화하기 위해 *cecil* 및 *dave*의 인증서가 필요합니다. *cecil*과 *dave*는 WebSphere Message Broker로부터 메시지가 오는지 확인하기 위해 WebSphere Message Broker 서비스 사용자의 인증서가 필요합니다.

4. IN으로 이름 지정된 로컬 큐를 정의하고 *alice* 및 *bob*이 작성자로, WebSphere Message Broker의 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. OUT으로 이름 지정된 로컬 큐를 정의하고 WebSphere Message Broker의 서비스 사용자가 작성자로, *cecil*과 *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. WebSphere Message Broker에서 MQInput 및 MQOutput 노드로 메시지 플로우를 작성하십시오. IN 큐를 사용하도록 MQInput 노드를 구성하고 OUT 큐를 사용하도록 MQOutput 노드를 구성하십시오.
7. WebSphere Message Broker 런타임 컴포넌트에 메시지 플로우를 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 IN으로 보낸 메시지는 암호화되어 WebSphereMessage Broker만 이를 읽을 수 있습니다. WebSphere Message Broker는 *alice* 및 *bob*에게서 온 메시지만 수락하고 다른 메시지는 모두 거부합니다. 수락된 메시지가 적절히 처리된 다음, 출력 큐 OUT에 넣기 전에 *cecil* 및 *dave*의 키로 서명되고 암호화됩니다. *cecil* 및 *dave*만 이를 읽을 수 있으며, WebSphere Message Broker가 서명하지 않은 메시지는 거부됩니다.

## *IBM WebSphere MQ Managed File Transfer*과(와) 함께 *IBM WebSphere MQ Advanced Message Security* 사용

이 시나리오에서는 IBM WebSphere MQ Managed File Transfer를 통해 전송되는 데이터에 대해 메시지 개인정보 보호정책을 제공하도록 Advanced Message Security 를 구성하는 방법에 대해 설명합니다.

## 시작하기 전에

보호하려는 IBM WebSphere MQ Managed File Transfer 에서 사용하는 큐를 호스트하는 WebSphere MQ 설치에 Advanced Message Security 구성요소가 설치되어 있는지 확인하십시오.

IBM WebSphere MQ Managed File Transfer 에이전트가 바인딩 모드에서 연결 중인 경우에는 GSKit 컴포넌트가 로컬 설치에 설치되어 있는지 또한 확인하십시오.

## 이 태스크 정보

데이터를 두 개의 IBM WebSphere MQ Managed File Transfer 에이전트 사이에서 전송할 때 중단되면 기밀 데이터가 전송을 관리하는 데 사용되는 기본 WebSphere MQ 큐에 보호되지 않은 상태로 남아 있을 수도 있습니다. 이 시나리오에서는 IBM WebSphere MQ Managed File Transfer 큐에서 이러한 데이터를 보호하기 위해 Advanced Message Security 를 구성하고 사용하는 방법에 대해 설명합니다.

이 시나리오에서는 스크립트를 사용하는 기본 파일 전송시나리오에 설명된 대로 단일 큐 관리자인 hubQM를 공유하는 두 개의 IBM WebSphere MQ Managed File Transfer 큐와 두 개의 에이전트 ( AGENT1 및 AGENT2) 가 있는 하나의 시스템을 포함하는 단순 토폴로지를 고려합니다. 두 에이전트 모두 바인딩 모드 또는 클라이언트 모드에서 동일한 방법으로 연결됩니다.

*1.* 인증서 작성

## 시작하기 전에

이 시나리오에서는 FTAGENTS 그룹의 사용자 `ftagent`를 사용하여 IBM WebSphere MQ Managed File Transfer 에이전트 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:**

- IBM WebSphere MQ Managed File Transfer 에이전트가 바인딩 모드에서 실행 중인 경우, CMS(Cryptographic Message Syntax) 키 저장소를 작성하는 데 사용하는 명령에 대해서는 사용자 플랫폼의 **빠른 시작 안내서**(Windows 또는 UNIX)에서 자세히 설명합니다.
- IBM WebSphere MQ Managed File Transfer 에이전트가 클라이언트 모드에서 실행 중인 경우 Java 키 저장소(JKS)를 작성하는 데 필요한 명령은 260 페이지의 『Java 클라이언트를 위한 빠른 시작 안내서』에 자세히 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=IBM, L=Hursley, ST=Hampshire, C=GB
   ```

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 위해 IBM WebSphere MQ 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=IBM,
   L=Hursley, ST=Hampshire, C=GB"
   -e AES128 -r "CN=ftagent, OU=MFT, O=IBM, L=Hursley, ST=Hampshire, C=GB"
   ```

3. IBM WebSphere MQ Managed File Transfer 에이전트 프로세스를 실행 중인 사용자에게 시스템 정책 큐를 찾아보고 메시지를 오류 큐에 넣기 위한 액세스가 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 IBM WebSphere MQ 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.

## 설치 중IBM WebSphere MQ Advanced Message Security
다양한 플랫폼에 IBM WebSphere MQ Advanced Message Security 컴포넌트를 설치합니다.

### 이 태스크 정보
전체 설치 프로시저에 대해서는 IBM WebSphere MQ Advanced Message Security 설치를 참조하십시오.
**관련 태스크**
설치 제거IBM WebSphere MQ Advanced Message Security

## 키 저장소 및 인증서 사용
WebSphere MQ 애플리케이션에 투명한 암호 보호를 제공하기 위해 Advanced Message Security는 공개 키 인증서 및 개인 키가 저장되는 키 저장소 파일을 사용합니다.

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.

키 저장소 위치는 키 저장소 구성 파일(기본적으로, `keystore.conf`)에 제공됩니다. 각 Advanced Message Security 사용자에게는 키 저장소 파일을 가리키는 키 저장소 구성 파일이 있어야 합니다. Advanced Message Security 키 저장소 파일의 형식은 `.kdb`, `.jceks`, `.jks`입니다.

`keystore.conf` 파일의 기본 위치는 다음과 같습니다.

- 유닉스 플랫폼: `$HOME/.mqs/keystore.conf`
- Windows 플랫폼의 경우: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

지정된 키 저장소 파일 이름 및 위치를 사용 중인 경우에는 다음 명령을 사용해야 합니다.

- Java의 경우: `java -D` *`MQS_KEYSTORE_CONF=path/filename app_name`*
- C 클라이언트 및 서버의 경우
  - UNIX and Linux: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windows: `set MQS_KEYSTORE_CONF=`*`path\filename`*

    **참고:** Windows의 경로는 둘 이상의 드라이브 문자가 사용 가능한 경우 드라이브 이름을 지정할 수 있습니다.

**관련 개념**
283 페이지의 『송신자 식별 이름』
송신자 식별 이름(DN)은 메시지를 큐에 배치할 수 있는 권한을 가진 사용자를 식별합니다.

283 페이지의 『수신자 식별 이름』
수신자 식별 이름(DN)은 큐로부터 메시지를 수신할 권한이 있는 사용자를 식별합니다.

### 키 저장소 구성 파일의 구조 (keystore.conf)
키 저장소 구성 파일 (`keystore.conf`) 은 Advanced Message Security 를 해당 키 저장소의 위치로 지정합니다.

두 가지 유형의 구성 CMS와 Java (JKS및 JCEKS) 가 있다. CMS 구성 항목 앞에는 `cms.` 접두부가 있으며 Java는 키 저장소의 유형에 따라 `jks.` 또는 `jceks.` 접두부가 붙습니다.

구성 파일은 유형에 따라 다음 구조 중 하나에 해당됩니다.

```
cms.keystore = /<dir>/<keystore_file>
cms.certificate = certificate_label

jceks.keystore = <dir>/Keystore
```

```
jceks.certificate = <certificate_label>
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE

jks.keystore = <dir>/Keystore
jks.certificate = <certificate_label>
jks.encrypted = no
jks.keystore_pass = <password>
jks.key_pass = <password>
jks.provider = IBMJCE
```

구성 파일 매개변수는 다음과 같이 정의됩니다.

**keystore**
키 저장소 파일에 대한 경로.

**중요사항:**

- 키 저장소 파일에 대한 경로는 파일 확장자를 포함하지 않아야 합니다.
- Java키 저장소 파일의 경우 IBM WebSphere MQ AMS 는 `.jks`, `.jceks`, `.jck`파일 형식을 지원합니다.

**certificate**
인증서 레이블.

**encrypted**
비밀번호의 상태입니다.

**keystore_pass**
키 저장소 파일의 비밀번호.

**참고:**

- CMS 키 저장소의 경우 IBM WebSphere MQ AMS는 스태쉬 파일 ( `.sth`)에 의존하는 반면, JKS 및 JCEKS에는 인증서 및 사용자의 개인 키 둘 모두에 대한 비밀번호가 필요할 수도 있습니다.
- 일반 텍스트 양식으로 비밀번호를 저장하면 보안 위험이 있습니다.

**key_pass**
사용자의 개인 키의 비밀번호입니다.

**중요사항:** 일반 텍스트 양식으로 비밀번호를 저장하면 보안 위험이 존재할 수 있습니다.

**provider**
키 저장소 인증서에 필요한 암호화 알고리즘을 구현하는 자바 보안 제공자입니다.

**참고:** 현재 IBMJCE는 Advanced Message Security에서 제공되는 유일한 제공자입니다.

**중요사항:** 키 저장소에 저장된 정보는 WebSphere MQ를 사용하여 송신한 데이터의 보안 플로우에 매우 중요합니다. 이러한 파일에 파일 권한을 지정할 때 보안 관리자가 특별히 주의해야 하는 이유이기도 합니다.

다음은 `keystore.conf` 파일에 대한 예입니다.

```
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE
```

**관련 태스크**

280 페이지의 『Java에서 비밀번호 보호』
키 저장소 및 개인 키 비밀번호를 일반 텍스트로 저장하면 보안 위험이 제기되므로 Advanced Message Security는 키 저장소 파일에서 사용 가능한 사용자 키를 사용하여 이러한 비밀번호를 긁어 모을 수 있는 도구를 제공합니다.

## 메시지 채널 에이전트(MCA) 인터셉션

MCA 인터셉트를 사용하면 IBM WebSphere MQ 에서 실행되는 큐 관리자가 선택적으로 서버 연결 채널에 정책을 적용할 수 있습니다.

IBM WebSphere MQ AMS 외부에 남아 있는 클라이언트는 MCA 인터셉션을 통해 큐 관리자에 계속 연결되어 있고 메시지를 암호화하고 복호화할 수 있습니다.

MCA 인터셉션은 클라이언트에서 IBM WebSphere MQ AMS를 사용할 수 없을 때 IBM WebSphere MQ AMS 기능을 제공하기 위한 것입니다. MCA 인터셉션과 IBM WebSphere MQ AMS 가능 클라이언트를 사용하면 메시지가 이중으로 보호되어 애플리케이션 수신에 장애가 될 수 있습니다.

Version 7.5 또는 이후 클라이언트를 사용하여 제품의 이전 버전에서 큐 관리자에 연결하는 경우 2085 (MQRC_UNKNOWN_OBJECT_NAME) 오류가 보고되면 클라이언트에서 IBM WebSphere MQ Advanced Message Security 를 사용 안함으로 설정해야 합니다. 추가 정보는 274 페이지의 『클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함』의 내용을 참조하십시오.

### 키 저장소 구성 파일

기본적으로 MCA 인터셉션의 키 저장소 구성 파일은 keystore.conf이고 큐 관리자 또는 리스너를 시작한 사용자의 HOME 디렉토리 경로에 .mqs 디렉토리에 있습니다. 키 저장소는 MQS_KEYSTORE_CONF 환경 변수를 사용하여 구성될 수도 있습니다. IBM WebSphere MQ AMS 키 저장소 구성에 대한 자세한 정보는 270 페이지의 『키 저장소 및 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.

MCA 인터셉션 설정 예는 272 페이지의 『IBM WebSphere MQ AMS MCA 인터셉션 예』의 내용를 참조하십시오.

**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 WebSphere MQ AMS MCA* 인터셉션 예

IBM WebSphere MQ AMS MCA 인터셉션 설정 방법에 대한 예 태스크.

### 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.

### 이 태스크 정보

이 태스크는 MCA 인터셉션을 사용하도록 시스템을 설정하고 설정을 확인하는 프로세스를 설명합니다.

**참고:** IBM WebSphere MQ Version 7.5 이전의 IBM WebSphere MQ AMS는 별도로 설치하여, 애플리케이션을 보호하도록 인터셉터를 구성해야 하는 추가 기능 제품이었습니다. Version 7.5부터는 인터셉터가 기본으로 포함되오 MQ 클라이언트 및 서버 런타임 환경에서 동적으로 사용됩니다. 이 MCA 인터셉션 예에서는 채널의 서버측에 인터셉터가 제공되며, (12단계에서는) 이전의 클라이언트 런타임을 사용하여 보호되지 않는 메시지가 채널에 배치되게 한 후 MCA 인터셉터가 메시지를 보호하는 과정을 확인할 수 있도록 합니다. 이 예에서 Version 7.5 이후의 클라이언트를 사용하게 되면 MQ 클라이언트 런타임 인터셉터와 MCA 인터셉터가 MQ로 들어오는 메시지를 보호하므로 메시지가 이중으로 보호됩니다.

**주의:** 코드에서 userID를 사용자의 ID로 바꾸십시오.

### 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/mq75
KEYSTORELOC=/home/testusr/ssl/ams1
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

gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
gsk8capicmd_64 -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.

   **태스크 5. 빠른 시작 안내서 (Windows 또는 UNIX) 의 인증서 공유하기**

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
```

4. AMSQMGR1 큐 관리자를 작성하고 시작하십시오.
5. *port* 14567 및 *control* QMGR로 리스너를 정의하십시오.
6. 채널 권한을 사용 안함으로 설정하거나 채널 권한의 규칙을 설정하십시오.
   자세한 정보는 SET CHLAUTH의 내용을 참조하십시오.
7. 큐 관리자를 중지합니다.
8. 키 저장소를 설정하십시오.

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. 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. 채널 구성을 설정하십시오.

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. 자동으로 MCA 인터셉터를 사용하지 않는 MQ 클라이언트에서 **amqsputc**를 실행하십시오(예: IBM WebSphere MQ Version 7.1 또는 이전 클라이언트). 다음 두 메시지를 배치하십시오.

```
/opt/mqm/samp/bin/amqsputc TESTQ TESTQMGR
```

13. 보안 정책을 제거하고 결과를 확인하십시오.

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

14. IBM WebSphere MQ Version 7.5 설치에서 큐를 찾아보십시오.

```
/opt/mq75/samp/bin/amqsbcg TESTQ AMSQMGR1
```

    찾아보기 출력은 암호화된 형식으로 메시지를 표시합니다.
15. 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
```

16. IBM WebSphere MQ Version 7.5 설치에서 **amqsgetc** 를 실행하십시오.

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

**관련 태스크**

260 페이지의 『Java 클라이언트를 위한 빠른 시작 안내서』
클라이언트 바인딩을 사용하여 연결하는 Java 애플리케이션에 대한 메시지 보안을 제공하도록 IBM Advanced Message Security 를 신속하게 구성하려면 이 안내서를 사용하십시오. 이 안내서를 끝까지 착실하게 따라가면 사용자 ID를 확인하기 위한 키 저장소를 작성하고 큐 관리자에 대한 서명/암호화 정책을 정의하는 작업을 마치게 될 것입니다.

**관련 참조**

249 페이지의 『알려진 제한사항』
IBM WebSphere MQ Advanced Message Security의 제한사항에 대해 학습하십시오.

## 클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함

Version 7.5 또는 이후 클라이언트를 사용하여 제품의 이전 버전에서 큐 관리자에 연결하고 2085 (MQRC_UNKNOWN_OBJECT_NAME) 오류가 보고되는 경우 클라이언트에서 IBM WebSphere MQ Advanced Message Security (AMS) 를 사용 안함으로 설정해야 합니다.

### 이 태스크 정보

Version 7.5에서 IBM WebSphere MQ Advanced Message Security (AMS) 는 IBM WebSphere MQ 클라이언트에서 자동으로 사용 가능하므로, 기본적으로 클라이언트는 큐 관리자의 오브젝트에 대한 보안 정책을 확인하려고 합니다. 그러나 제품의 이전 버전 (예: Version 7.1) 에 있는 서버에는 AMS 를 사용할 수 없으며, 이로 인해 2085 (MQRC_UNKNOWN_OBJECT_NAME) 오류가 보고됩니다.

이 오류가 보고되면 제품의 이전 버전에서 큐 관리자에 연결하려고 할 때 다음과 같이 클라이언트에서 AMS 를 사용 안함으로 설정할 수 있습니다.

- Java 클라이언트의 경우, 다음 방법 중 하나를 사용하십시오.
  - V7.5.0.4 환경 변수 AMQ_DISABLE_CLIENT_AMS 설정.
  - V7.5.0.4 Java 시스템 특성 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 설정.
  - V7.5.0.5 mqclient.ini 파일의 **Security** 스탠자 아래에서 DisableClientAMS 특성 사용.
- C 클라이언트의 경우, 다음 방법 중 하나를 사용하십시오.
  - V7.5.0.4 환경 변수 AMQ_DISABLE_CLIENT_AMS 설정.
  - V7.5.0.5 mqclient.ini 파일의 **Security** 스탠자 아래에서 DisableClientAMS 특성 사용.

### 프로시저

- 클라이언트에서 AMS를 사용 안함으로 설정하려면, 다음 옵션 중 하나를 사용하십시오.

  **V7.5.0.4 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
  - Version 7.5 이상의 IBM WebSphere MQ classes for Java 또는 IBM WebSphere MQ classes for JMS 클라이언트를 사용 중인 경우.

  AMQ_DISABLE_CLIENT_AMS를 사용하여 C 클라이언트의 AMS 기능을 사용 불가능하게 할 수도 있습니다.

AMQ_DISABLE_CLIENT_AMS 환경 변수를 작성하고 애플리케이션이 실행 중인 환경에서 TRUE로 설정하십시오. 예를 들면, 다음과 같습니다.

```
export AMQ_DISABLE_CLIENT_AMS=TRUE
```

V 7.5.0.4 **com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 시스템 특성**

IBM WebSphere MQ classes for JMS 및 IBM WebSphere MQ classes for Java 클라이언트의 경우 Java 시스템 특성 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 를 Java 애플리케이션의 값 TRUE 로 설정하십시오.

예를 들어, Java 명령이 호출될 때 Java 시스템 특성을 -D 옵션으로 설정할 수 있습니다.

```
java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/java/lib/
com.ibm.mqjms.jar  my.java.applicationClass
```

또는 애플리케이션이 이 파일을 사용하는 경우 JMS 구성 파일 `jms.config` 내에 Java 시스템 특성을 지정할 수 있습니다.

V 7.5.0.5 **`mqclient.ini` 파일의 `DisableClientAMS` 특성**

IBM WebSphere MQ classes for JMS 및 IBM WebSphere MQ classes for Java 클라이언트 및 C 클라이언트의 경우 다음 예제에 표시된 대로 **`Security`** 스탠자 아래의 특성 이름 `DisableClientAMS` 를 `mqclient.ini` 파일에 추가하십시오.

```
Security:
DisableClientAMS=Yes
```

다음 예제에 표시된 대로 AMS를 사용으로 설정할 수도 있습니다.

```
Security:
DisableClientAMS=No
```

### 다음에 수행할 작업

AMS 보호 큐 열기와 관련한 문제점에 대한 자세한 정보는 297 페이지의 『JMS를 사용할 때 보호된 큐를 여는 중 문제』의 내용을 참조하십시오.

**관련 개념**

272 페이지의 『메시지 채널 에이전트(MCA) 인터셉션』
MCA 인터셉트를 사용하면 IBM WebSphere MQ 에서 실행되는 큐 관리자가 선택적으로 서버 연결 채널에 정책을 적용할 수 있습니다.

**관련 태스크**

구성 파일을 사용하여 클라이언트 구성

**관련 참조**

IBM WebSphere 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 32 페이지의 『IBM WebSphere MQ의 디지털 인증서 및 CipherSpec 호환성』의 내용을 참조하십시오.

### 키 사용 확장

키 사용 확장은 인증서를 사용할 수 있는 방법에 추가 제한사항을 둡니다.

Advanced Message Security에서 키 사용법은 다음과 같이 설정되어야 합니다. for certificates in X.509 V3 이상에서 인증서의 경우 보호 품질 무결성을 위해 사용되는 표준의 경우 키 사용법 확장이 설정된 경우 이들은 적어도 다음 둘 중 하나를 포함해야 합니다.

- **`nonRepudiation`**

- **digitalSignature**

보호 품질 개인정보의 경우 키 사용법 확장자가 설정된 경우 이들은 또한 **keyEncipherment** 확장을 포함해야 합니다.

**관련 개념**

285 페이지의 『보호 품질』

Advanced Message Security 데이터 보호 정책은 QoP(Quality of Protection)를 의미합니다.

## IBM WebSphere MQ Advanced Message Security에서 인증서 유효성 검증 메소드

IBM WebSphere MQ Advanced Message Security를 사용하여 큐의 메시지가 보안 표준을 충족하지 않는 인증서를 사용하여 보호되지 않도록 폐기된 인증서를 감지하고 거부할 수 있습니다.

IBM WebSphere MQ AMS를 사용하면 온라인 인증서 상태 프로토콜(OCSP) 또는 인증서 폐기 목록(CRL)을 사용하여 인증서 유효성을 확인할 수 있습니다.

IBM WebSphere MQ AMS는 OCSP 또는 CRL 검사 또는 둘 모두에 대해 구성될 수 있습니다. 두 메소드 모두가 사용으로 설정되면 성능상의 이유로 IBM WebSphere MQ AMS는 먼저 폐기 상태를 위해 OCSP를 사용합니다. 인증서의 폐기 상태가 OCSP 검사 후에도 미해결이면 IBM WebSphere MQ AMS는 CRL 검사를 사용합니다.

**관련 개념**

276 페이지의 『OCSP(Online Certificate Status Protocol)』

OCSP(Online Certificate Status Protocol)는 인증서의 폐기 여부를 판별하므로 인증서의 신뢰 여부를 판별하는 데 도움이 됩니다.

278 페이지의 『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.

### *OCSP(Online Certificate Status Protocol)*

OCSP(Online Certificate Status Protocol)는 인증서의 폐기 여부를 판별하므로 인증서의 신뢰 여부를 판별하는 데 도움이 됩니다.

기본 인터셉터에서 *OCSP* 검사 사용

Advanced Message Security에서 OCSP(Online Certificate Status Protocol) 검사를 사용 가능하게 하려면 키 저장소 구성 파일을 수정해야 합니다.

### 프로시저

키 저장소 구성 파일에 다음 옵션을 추가하십시오.

**참고:** 테이블에 제공된 개별 옵션 값은 기본값입니다.

다음 값 중 하나를 지정해야 합니다.

- `ocsp.enable=on`
- `ocsp.url=<`*`reposnder_URL`*`>`
- `ocsp.http.proxy.host=<`*`OCSP_proxy`*`>`

| 옵션 | 설명 |
|---|---|
| `ocsp.enable=off` | 검사하는 인증서에 OCSP 응답자가 있는 URI를 포함하는 PKIX_AD_OCSP 액세스 방법의 권한 정보 액세스 확장자가 있는 경우 OCSP 검사를 사용합니다.<br>가능한 값: `on`/`off`. |
| `ocsp.url=<`*`reposnder_URL`*`>` | OCSP 응답자의 URL 주소입니다. |
| `ocsp.http.proxy.host=<`*`OCSP_proxy`*`>` | OCSP 프록시 서버의 URL 주소입니다. |
| `ocsp.http.proxy.port=<`*`port_number`*`>` | OCSP 프록시 서버의 포트 번호입니다. |

| 옵션 | 설명 |
| --- | --- |
| ocsp.nonce.generation=on/off | OCSP를 조회할 때 임시값을 생성합니다.<br>기본값: off. |
| ocsp.nonce.check=on/off | OCSP로부터 응답을 받은 후 임시값을 검사합니다.<br>기본값: off. |
| ocsp.nonce.size=8 | 임시값 크기(바이트)입니다. |
| ocsp.http.get=on/off | 요청 메소드로서 HTTP GET을 지정합니다. 이 옵션이 off로 설정되면 HTTP POST가 사용됩니다. |
| 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. |

*Java* 에서 *OCSP* 확인 사용하기
Advanced Message Security에서 Java에 대해 OCSP 체크인을 사용하려면 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
인증서에서 권한 정보 액세스(AIA)가 설정되었는지 여부를 확인합니다.

## 프로시저

1. AIA가 설정되지 않거나 인증서를 대체하려면 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 다음 행으로 $JAVA_HOME/lib/security/java.security 파일을 편집하여 OCSP 검사를 사용으로 설정하십시오.

   ```
   ocsp.enable=true
   ```

2. AIA가 설정되면 다음 행으로 $JAVA_HOME/lib/security/java.security 파일을 편집하여 OCSP 검사를 사용으로 설정하십시오.

   ```
   ocsp.enable=true
   ```

## 다음에 수행할 작업

Java Security Manager를 사용하는 경우에는 구성을 완료하고 다음 Java 권한을 lib/security/java.policy 에 추가하십시오.

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

### 프로시저

다음 속성을 구성 파일에 추가하십시오.

```
ocsp.enable=true
```

**중요사항:** 구성 파일에서 이 속성을 설정하면 java.security 설정이 대체됩니다.

### 다음에 수행할 작업

구성을 완료하려면 `lib/security/java.policy`에 다음 Java 권한을 추가하십시오.

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

## 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.IBM WebSphere MQ AMS PKIX 검증 알고리즘을 사용하여 인증서 경로를 검증한다. 체인이 작성되고 확인되면, IBM WebSphere MQ 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 IBM WebSphere MQ AMS는 **digitalSignature** 또는 **nonRepudiation** 또한 설정되었는지 여부를 확인합니다. 이들이 설정되지 않은 경우에는 `MQRC_SECURITY_ERROR`가 보고되고 로그됩니다. 그런 다음 IBM WebSphere MQ AMS는 구성 파일에 지정된 내용에 따라서 파일 또는 LDAP에서 CRL을 다운로드합니다. IBM WebSphere MQ AMS에서는 DER 형식으로 인코딩된 CRL만이 지원됩니다. 키 저장소 구성 파일에 CRL 관련 정보를 찾을 수 없는 경우 IBM WebSphere MQ AMS는 CRL 유효성 검증 검사를 수행하지 않습니다. 각 CA 인증서의 경우 IBM WebSphere MQ 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 Security에서 LDAP(Lightweight Directory Access Protocol) 서버로부터 CLR을 다운로드할 수 있도록 키 저장소 구성 파일을 수정해야 합니다.

### 프로시저

다음 옵션을 구성 파일에 추가하십시오.

**참고:** 테이블에 제공된 개별 옵션 값은 기본값입니다.

| 옵션 | 설명 |
|---|---|
| `crl.ldap.host=<`*`host_name`*`>` | LDAP 서버 호스트 이름 |

| 옵션 | 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. IBM WebSphere MQ AMS Java 인터셉터가 LDAP 서버에 연결되면 제공된 나머지 서버에서 CRL을 다운로드하지 않습니다. |
| 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 대한 로그인. 이 값이 지정되지 않으면 LDAP의 CRL은 읽기 쉬워야 합니다. |
| crl.ldap.pass=<*password*> | LDAP 서버에 대한 비밀번호. |
| 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 0보다 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 | GSKit이 허용하는 HTTP 서버에서 검색할 수 있는 CRL의 최대 크기 (바이트) 입니다. |
| crl.http.timeout=30 | IBM WebSphere MQ AMS 제한시간을 초과한 후 서버 응답의 대기 시간(초). |
| crl.http.cache_size=0 | HTTP 캐시 크기(바이트 단위). |

*Java* 에서 인증서 폐기 목록 지원 사용하기
Advanced Message Security에서 CRL 지원을 사용으로 설정하려면 LDAP(Lightweight Directory Access Protocol) 서버로부터 CRL을 다운로드하고 java.security 파일을 구성하도록 IBM WebSphere MQ AMS에 허용하기 위해 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. IBM WebSphere MQ AMS Java 인터셉터가 LDAP 서버에 연결되면 제공된 나머지 서버에서 CRL을 다운로드하지 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. JRE/lib/security/java.security 파일을 다음 특성으로 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. |

## Java에서 비밀번호 보호

키 저장소 및 개인 키 비밀번호를 일반 텍스트로 저장하면 보안 위험이 제기되므로 Advanced Message Security는 키 저장소 파일에서 사용 가능한 사용자 키를 사용하여 이러한 비밀번호를 긁어 모을 수 있는 도구를 제공합니다.

### 시작하기 전에

keystore.conf 파일 소유자는 파일 소유자만이 파일을 읽을 권한이 있음을 확실히 해야 합니다. 이 장에서 설명된 비밀번호 보호는 보호의 추가적 측정입니다.

### 프로시저

1. 키 저장소에 대한 경로 및 사용자 레이블을 포함하도록 keystore.conf 파일을 편집하십시오.

```
jceks.keystore = c:/Documents and Setting/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.provider = IBMJCE
```

2. 도구를 실행하려면 다음을 실행하십시오.

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password
```

암호화된 비밀번호가 있는 출력이 생성되고 keystore.conf 파일에 복사될 수 있습니다.

출력을 keystore.conf 파일에 자동으로 복사하려면 다음을 실행하십시오.

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password >> ~/<path_to_keystore>/keystore.conf
```

**참고:**

다양한 플랫폼에서 keystore.conf의 기본 위치의 목록은 270 페이지의 『키 저장소 및 인증서 사용』의 내용을 참조하십시오.

#### 예

다음은 이러한 출력의 예입니다.

```
#Fri Jul 30 15:20:29 CEST 2010
jceks.key_pass=MMXh997n5ZOr8uRlJmc5qity9MN2CggGBMKCDxdbn1AyPklvdgTsOLG6X3C1YT7oDzwaqZFlOR4t\r\nm
Zsc7JGAx8nqqxLnAucdGn0NWo6xnjZB1n501YGol2k/
PhaQHhFXKMAU9dKg0f8djOtCAOlX4ETe\r\nfYl9LBUt2wk87uM7dSs\=
jceks.keystore_pass=OIdeayBnSCfLG4cFuxEVrk6SYyAsdSPpDqgPf16s9s1M04cqZjNbhgjoA2EXonudHZHH+4s2drvQ
\r\nCUvQgu9GuaBMJK2F2OjtHJJ1Y4BVeLW2c2okgawo/
W2J1AdUYKkJ0raYTkDouLaTYTQeulyG0xIl\r\niD2si1xUCxhYvvyhbbY\=
jceks.encrypted=yes
```

# IBM WebSphere MQ Advanced Message Security 보안 정책 관리

IBM WebSphere MQ Advanced Message Security는 보안 정책을 사용하여 큐를 통해 플로우하는 메시지를 암호화 및 인증하기 위한 암호화 및 서명 알고리즘을 지정합니다.

## 보안 정책 개요

IBM Advanced Message Security 보안 정책은 메시지의 암호화 및 서명 방식을 설명하는 개념적 오브젝트입니다.

보안 정책 속성의 세부사항은 다음 하위 주제를 참조하십시오.

#### 관련 개념

285 페이지의 『보호 품질』
Advanced Message Security 데이터 보호 정책은 QoP(Quality of Protection)를 의미합니다.

284 페이지의 『보안 정책 속성』
Advanced Message Security를 사용하여 데이터를 보호하기 위한 특정 알고리즘 또는 방법을 선택할 수 있습니다.

## 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 (IBM WebSphere MQ AMS) 정책과 대기열 사이에 일대일 맵핑이 있습니다.

큐와 이름이 같은 정책을 작성함으로써 해당 큐의 정책을 활성화합니다. 일치하는 정책 이름이 없는 큐는 IBM WebSphere MQ 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..

## 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.

## 암호화 알고리즘

암호화 알고리즘은 암호화 데이터 메시지가 정책과 연관된 큐에 배치될 때 사용되어야 하는 알고리즘을 나타냅니다.

유효한 값은 다음을 포함합니다.

- RC2
- DES
- 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.

## 허용

허용 속성은 IBM Advanced Message Security이 보안 정책이 지정되지 않은 메시지의 허용 여부를 표시합니다.

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.

## 송신자 식별 이름

송신자 식별 이름(DN)은 메시지를 큐에 배치할 수 있는 권한을 가진 사용자를 식별합니다.

IBM Advanced Message Security (IBM WebSphere MQ AMS) 는 메시지가 검색될 때까지 유효한 사용자가 데이터 보호 큐에 메시지를 넣었는지 여부를 확인하지 않습니다. 이때 정책에 올바른 송신자가 하나 이상 지정되어 있고 메시지를 큐에 배치한 사용자가 올바른 송신자 목록에 없는 경우, IBM WebSphere MQ AMS에서는 애플리케이션에 오류를 리턴하고 메시지를 오류 큐에 배치합니다.

정책에는 0개 이상의 송신자 DN이 지정될 수 있습니다. 송신자 DN이 정책에 지정되지 않은 경우 사용자의 인증서를 신뢰할 수 있다면 사용자가 데이터 보호 메시지를 큐에 배치할 수 있습니다.

송신자 식별 이름의 형식은 다음과 같습니다.

```
CN=Common Name,O=Organization,C=Country
```

**중요사항:**

- 모든 DN은 대문자로 되어 있어야 합니다. 다음 표에 표시된 순서대로 DN에 있는 모든 구성요소 이름 ID를 지정해야 합니다.

| 컴포넌트 이름 | 가치 |
|---|---|
| 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.
- IBM WebSphere MQ AMS는 라틴-1 문자 세트의 값만 있는 DN을 지원합니다. 문자 세트로 DN을 작성하려면 먼저 iKeyman 유틸리티 또는 UTF-8 코딩을 켠 UNIX 플랫폼을 사용하여 UTF-8 코딩으로 작성된 DN으로 인증서를 작성해야 합니다. 그런 다음, UTF-8 코딩을 켠 UNIX 플랫폼에서 정책을 작성하거나 WebSphere MQ에 대한 IBM WebSphere MQ AMS 플러그인을 사용해야 합니다.

**관련 개념**

283 페이지의 『수신자 식별 이름』
수신자 식별 이름(DN)은 큐로부터 메시지를 수신할 권한이 있는 사용자를 식별합니다.

## 수신자 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 DN은 대문자로 되어 있어야 합니다. 다음 표에 표시된 순서대로 DN에 있는 모든 구성요소 이름 ID를 지정해야 합니다.

| 컴포넌트 이름 | 가치 |
|---|---|
| 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. |

- 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. 문자 세트로 DN을 작성하려면 먼저 iKeyman 유틸리티 또는 UTF-8 코딩을 켠 UNIX 플랫폼을 사용하여 UTF-8 코딩으로 작성된 DN으로 인증서를 작성해야 합니다. 그런 다음, UTF-8 코딩을 켠 UNIX 플랫폼에서 정책을 작성하거나 WebSphere MQ에 대한 Advanced Message Security 플러그인을 사용해야 합니다.

**관련 개념**

283 페이지의 『송신자 식별 이름』

송신자 식별 이름(DN)은 메시지를 큐에 배치할 수 있는 권한을 가진 사용자를 식별합니다.

## 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. 다음 테이블은 Advanced Message Security에서 보안 정책 속성을 제공합니다.

| 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.

- RC2
- DES

- 3DES
- AES128
- AES256

Advanced Message Security는 또한 다음과 같은 암호화 해시 함수를 지원합니다.

- MD5
- 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.

### 보호 품질

Advanced Message Security 데이터 보호 정책은 QoP(Quality of Protection)를 의미합니다.

Advanced Message Security에서 세 개의 QoP(Quality of Protection) 레벨은 메시지에 서명하고 암호화하는 데 사용되는 암호화 알고리즘에 따라 다릅니다.

- 개인정보 보호 - 큐에 배치된 메시지는 서명되고 암호화되어야 합니다.
- 무결성 - 큐에 배치된 메시지는 서명자에 의해 서명되어야 합니다.
- 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.

서명 알고리즘 또는 암호화 알고리즘을 규정하지 않는 정책에는 NONE의 QOP가 있습니다. Advanced Message Security 는 QOP가 없음이 있는 정책을 가진 큐에 대해 데이터 보호를 제공하지 않는다.

## 보안 정책 관리

보안 정책은 메시지가 암호학적으로 암호화되고 서명되는 방법을 설명하는 개념적 오브젝트입니다.

보안 정책와 관련된 모든 관리 태스크는 다음 위치에서 실행됩니다.

- 유닉스 플랫폼: `<MQInstallRoot>/bin`
- Windows 플랫폼에서는 설치 시 PATH 환경 변수가 업데이트되는 모든 위치에서 관리 태스크를 실행할 수 있습니다.

**관련 태스크**

286 페이지의 『보안 정책 작성』
보안 정책은 메시지를 놓일 때 메시지가 보호되는 방법 또는 메시지를 수신할 때 메시지를 보호해야 하는 방법을 정의합니다.

286 페이지의 『보안 정책 변경』
Advanced Message Security를 사용하여 이미 정의한 보안 정책의 세부사항을 변경할 수 있습니다.

287 페이지의 『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.

288 페이지의 『보안 정책 제거』

Advanced Message Security에서 보안 정책을 제거하려면 setmqspl 명령을 사용해야 합니다.

## 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 WebSphere MQ 오브젝트의 이름 지정 규칙을 따라야 합니다.
- 보안 정책을 작성하기 위해 필요한 +connect +inq +chg 권한이 있어야 합니다. 권한 변경 명령의 전체 구문은 **setmqaut** 을 (를) 참조하십시오.
- WebSphere MQ 큐와 큐 관리자에서 조작하기 위해 필요한 권한이 있는지 확인하십시오. 추가 정보는 290 페이지의 『OAM 권한 부여』의 내용을 참조하십시오.

### 예

다음은 QMGR 큐 관리자에서 정책 작성의 예입니다. 정책은 DN: CN=joe,O=IBM,C=US 및 DN: CN=jane,O=IBM,C=US와 같은 인증서에 대해 SHA1 알고리즘을 사용하여 메시지에 서명하고 AES256 알고리즘을 사용하여 메시지를 암호화하도록 지정합니다. 이 정책은 MY.QUEUE에 첨부됩니다.

```
$ setmqspl -m QMGR -p MY.QUEUE -s SHA1 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

다음은 QMGR 큐 관리자에서 정책 작성의 에입니다. 정책은 DNs: CN=john,O=IBM,C=US 및 CN=jeff,O=IBM,C=US와 같은 인증서에 대해 DES 알고리즘을 사용하여 메시지를 암호화하고 DN: CN=phil,O=IBM,C=US와 같은 인증서에서 MD5 알고리즘으로 서명하도록 지정합니다.

```
$ setmqspl -m QMGR -p MY.OTHER.QUEUE -s MD5 -e DES -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

## 보안 정책 변경

Advanced Message Security를 사용하여 이미 정의한 보안 정책의 세부사항을 변경할 수 있습니다.

## 시작하기 전에

- 운영하려는 큐 관리자가 실행 중이어야 합니다.
- 보안 정책을 작성하려면 +connect +inq +chg 권한이 있어야 합니다. 권한 변경 명령의 전체 구문은 **setmqaut** 을 (를) 참조하십시오.

## 이 태스크 정보

보안 정책을 변경하려면 setmqspl 명령을 이미 존재하는 정책에 적용하여 새 속성을 제공하십시오.

### 예

다음은 메시지가 DN:CN=bob,O=IBM,C=US가 포함된 인증서에 대한 RC2 알고리즘을 사용하여 암호화되고 DN:CN=jeff,O=IBM,C=US가 포함된 인증서에 대한 SHA1 알고리즘으로 서명될 것임을 지정하는 QMGR로 이름 지정된 큐 관리자에서 MYQUEUE라는 이름의 정책을 작성하는 예제입니다.

```
setmqspl -m QMGR -p MYQUEUE -e RC2 -s SHA1 -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 SHA1 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

### 관련 참조

setmqspl

## 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.
- 보안 정책을 표시하고 덤프하려면 필수 +connect +inq +dsp 권한이 큐 관리자에 적용되어야 합니다. 권한 변경 명령의 전체 구문은 **setmqaut** 을 (를) 참조하십시오.

## 이 태스크 정보

다음은 dspmqspl 명령 플래그의 목록입니다.

표 *27. 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

이 예에서는 venus.queue.manager에 대해 두 개의 보안 정책을 작성합니다.

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s MD5 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s MD5 -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: MD5
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: MD5
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: MD5
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 MD5 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

보안 정책을 가져오려면:

- Windows 플랫폼의 경우, policies.bat를 실행하십시오.
- UNIX 플랫폼의 경우,
  1. mqm WebSphere MQ 관리 그룹에 속하는 사용자로서 로그온하십시오.
  2. . policies.sh를 실행하십시오.

**관련 참조**
dspmqspl 명령 속성의 전체 목록

## 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.
- 보안 정책을 작성하려면 +connect +inq +chg 권한이 있어야 합니다. 권한 변경 명령의 전체 구문은 **setmqaut** 을 (를) 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

## 시스템 큐 보호

시스템 큐를 사용하면 WebSphere MQ와 해당 보조 애플리케이션 사이의 통신이 가능합니다. 큐 관리자가 작성될 때마다 시스템 큐 또한 WebSphere 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. 286 페이지의 『보안 정책 작성』의 내용을 참조하십시오.

Windows 플랫폼에서 시스템 큐 보호를 사용하려면 `keystore.conf` 파일을 다음 디렉토리에 복사하십시오.

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

`SYSTEM.ADMIN.COMMAND.QUEUE`에 대한 보호를 제공하려면 명령 서버가 키 및 인증서에 액세스할 수 있도록 키 및 구성을 포함하는 `keystore` 및 `keystore.conf`에 대한 액세스 권한을 명령 서버가 보유해야 합니다. `SYSTEM.ADMIN.COMMAND.QUEUE`의 보안 정책에 대한 모든 변경사항을 적용하려면 명령 서버를 재시작해야 합니다.

명령 큐로부터 전송되고 수신된 모든 메시지는 정책 설정에 따라 서명되거나 서명된 후 암호화됩니다. 관리자가 허가된 서명자를 정의하는 경우 서명자 식별 이름(DN) 검사에 통과하지 못한 명령 메시지는 명령 서버에서 실행되지 않으며 Advanced Message Security 오류 핸들링 큐로 라우트되지 않습니다. WebSphere MQ Explorer 임시 동적 큐에 대한 응답으로 송신된 메시지는 WebSphere MQ AMS에서 보호되지 않습니다.

Advanced Message Security 보안 정책의 변경을 적용하려면 WebSphere MQ 명령 서버를 재시작해야 합니다.

보안 정책에는 다음 SYSTEM 큐에 영향을 미치지 않습니다.

- SYSTEM.ADMIN.ACCOUNTING.QUEUE
- SYSTEM.ADMIN.ACTIVITY.QUEUE
- SYSTEM.ADMIN.CHANNEL.EVENT
- SYSTEM.ADMIN.COMMAND.EVENT
- 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
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.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
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

## OAM 권한 부여

파일 권한은 모든 사용자가 `setmqspl` 및 `dspmqspl` 명령을 실행하기 위한 권한을 제공합니다. 하지만, IBM Advanced Message Security은 오브젝트 권한 관리자(OAM)에 의존하며, WebSphere MQ 관리 그룹인 mqm 그룹에 속하지 않거나 권한 부여된 보안 정책 설정을 읽을 권한이 없는 사용자가 이런 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
```

## 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.

WebSphere MQ에서 생성되는 명령 및 구성 이벤트는 전용 큐로 송신되는 PCF 형식의 메시지입니다.

구성 이벤트 메시지는 이벤트가 발생하는 큐 관리자의 SYSTEM.ADMIN.CONFIG.EVENT 큐로 송신됩니다.

명령 이벤트 메시지는 이벤트가 발생하는 큐 관리자의 SYSTEM.ADMIN.COMMAND.EVENT 큐로 송신됩니다.

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.

- 286 페이지의 『보안 정책 작성』 - 두 개의 WebSphere MQ 이벤트 메시지를 생성:
  - 구성 이벤트
  - 명령 이벤트
- 286 페이지의 『보안 정책 변경』 - 세 개의 WebSphere MQ 이벤트 메시지를 생성:
  - 오래된 보안 구성 값을 포함하는 구성 이벤트
  - 새 보안 정책 값을 포함하는 구성 이벤트
  - 명령 이벤트
- 287 페이지의 『보안 정책 표시 및 덤프』 - 한 개의 WebSphere MQ 이벤트 메시지를 생성:
  - 명령 이벤트
- 288 페이지의 『보안 정책 제거』 - 두 개의 WebSphere MQ 이벤트 메시지를 생성:
  - 구성 이벤트
  - 명령 이벤트

## 이벤트 로깅 사용 가능 및 사용 불가능

큐 관리자 속성 CONFIGEV 및 CMDEV를 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 CONFIGEV를 ENABLED로 설정하십시오. 구성 이벤트를 사용 불가능하게 설정하려면 CONFIGEV를 DISABLED로 설정하십시오. 예를 들어, 다음 MQSC 명령을 사용하여 구성 이벤트를 사용 가능으로 설정할 수 있습니다.

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

**명령 이벤트**

명령 이벤트를 사용 가능하게 설정하려면 CMDEV를 ENABLED로 설정하십시오. DISPLAY MQSC 명령 및 Inquire PCF 명령을 제외하고 명령에 명령 이벤트를 사용 가능으로 설정하려면 NODISPLAY에 CMDEV를 설정하십시오. 명령 이벤트를 사용 불가능하게 설정하려면 CMDEV를 DISABLED로 설정하십시오. 예를 들어, 다음 MQSC 명령을 사용하여 명령 이벤트를 사용 가능으로 설정할 수 있습니다.

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

**관련 태스크**

Websphere MQ에서 구성, 명령 및 로거 이벤트 제어

## 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**
WebSphere 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.

***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.

***Command***

설명: 명령 코드.

ID: MQIACF_COMMAND.

데이터 유형: MQCFIN.

값: **MQCMD_INQUIRE_PROT_POLICY 숫자 값 205**
**MQCMD_CREATE_PROT_POLICY 숫자 값 206**
**MQCMD_DELETE_PROT_POLICY 숫자 값 207**
**MQCMD_CHANGE_PROT_POLICY 숫자 값 208**
WebSphere MQ 7.5 `cmqcfc.h`에 정의되어 있음

리턴됨: 항상.

CommandData 에는 PCF 명령을 포함하는 PCF 요소가 포함되어 있습니다.

## 구성 이벤트 메시지 형식

구성 이벤트는 표준 Advanced Message Security 형식의 PCF 메시지입니다.

MQMD 메시지 디스크립터의 가능한 값은 이벤트 메시지 MQMD (메시지 설명자)를 참조하십시오.

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 구조와 이 구조를 따르는 매개변수 구조로 구성된다. 가능한 MQCFH값은 이벤트 메시지 MQCFH (PCF 헤더)를 참조하십시오.

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 메시지의 경우 MQMD.AccountingToken 값 또는 로컬 명령의 경우 Windows SID.

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**
WebSphere 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** - WebSphere MQ 7.5 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** - WebSphere MQ 7.5 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** - WebSphere MQ 7.5 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** - WebSphere MQ 7.5 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** - WebSphere MQ 7.5 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** - WebSphere MQ 7.5 또는 cmqc.h 파일에 정의된 숫자 값.

리턴됨: WebSphere MQ 정책에 정의된 암호화 알고리즘이 있을 때마다

***SignerDNs***

설명: 허용된 서명자의 제목 식별 이름.

ID: **MQCA_SIGNER_DN**

데이터 유형: MQCFSL

값: **2113** - WebSphere MQ 7.5 또는 `cmqc.h` 파일에 정의된 숫자 값.

최대 길이: 정책에서 가장 긴 서명자 DN이지만 MQ_DISTINGUISHED_NAME_LENGTH보다 길지는 않음

리턴됨: WebSphere MQ 정책에 정의될 때마다.

***RecipientDNs***

설명: 허용된 서명자의 제목 식별 이름.

ID: **MQCA_RECIPIENT_DN**

데이터 유형: MQCFSL

값: **2114** - WebSphere MQ 7.5 또는 `cmqc.h` 파일에 정의된 숫자 값.

최대 길이: 정책에서 가장 긴 수신인 DN이지만 MQ_DISTINGUISHED_NAME_LENGTH보다 길지는 않음

리턴됨: WebSphere MQ 정책에 정의될 때마다.

# 문제점 및 솔루션

이 절에서는 IBM 제품 설치와 관련하여 발생할 수 있는 문제점을 해결하는 방법을 설명합니다. 여기서 설명하는 정보를 바탕으로 Advanced Message Security과 관련된 문제점을 식별하고 해결하십시오.

## com.ibm.security.pkcsutil.PKCSException: 컨텐츠 암호화 중 오류

오류 `com.ibm.security.pkcsutil.PKCSException: Error encrypting contents` 에서는 IBM Advanced Message Security 에 암호화 알고리즘에 액세스하는 데 문제가 있음을 나타냅니다.

다음 오류가 Advanced Message Security로부터 리턴되면:

```
DRQJP0103E   The IBM WebSphere MQ Advanced Message Security Java interceptor failed to protect
message.
  com.ibm.security.pkcsutil.PKCSException: Error encrypting contents
(java.security.InvalidKeyException: Illegal key size or default parameters)
```

`JAVA_HOME/lib/security/local_policy.jar/*.policy`의 JCE 보안 정책이 MQ AMS 정책에 사용된 서명 알고리즘에 대한 액세스를 부여하는지를 확인하십시오.

사용하려는 서명 알고리즘이 현재 보안 정책에 지정되지 않은 경우 다음 위치에서 올바른 Java 정책 파일을 다운로드하십시오.

- IBM Java 1.4.2의 SDK 정책 파일.
- IBM Java 5.0의 SDK 정책 파일.
- IBM Java 6.0의 SDK 정책 파일.
- IBM Java 7.0의 SDK 정책 파일.

## OSGi 지원

IBM Advanced Message Security과 함께 OSGi 번들을 사용하려면 추가 매개변수가 필요합니다.

OSGi 번들 시작 중에 다음 매개변수를 실행하십시오.

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7
```

keystore.conf에서 암호화된 비밀번호를 사용할 때 OSGi 번들이 실행 중일 때 다음 명령문을 추가해야 합니다.

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7,com.ibm.misc
```

**제한사항:** IBM WebSphere MQ AMS에서는 OSGi 번들 내에서 보호되는 큐에 대해 MQ 기본 Java 클래스만 사용하는 통신을 지원합니다.

## JMS를 사용할 때 보호된 큐를 여는 중 문제

IBM WebSphere MQ Advanced Message Security를 사용하여 보호된 큐를 열 때 다양한 문제점이 발생할 수 있습니다.

JMS를 실행 중이고 JMS MQ2008과 함께 오류 2085(MQRC_UNKNOWN_OBJECT_NAME)를 수신합니다.

260 페이지의 『Java 클라이언트를 위한 빠른 시작 안내서』 에 설명된 대로 IBM WebSphere MQ Advanced Message Security 를 설정했는지 확인했습니다.

가능한 원인은 비IBM JRE(Java Runtime Environment)를 사용 중이기 때문입니다. 이는 249 페이지의 『알려진 제한사항』 에 설명된 알려진 제한사항입니다.

AMQ_DISABLE_CLIENT_AMS 환경 변수를 설정하지 않았습니다.

### 문제점 해결

이 문제점을 해결하기 위해서는 네 가지 옵션이 있습니다.

1. JMS 애플리케이션을 지원되는 IBM Java Runtime Environment(JRE) 하에서 시작하십시오.
2. 애플리케이션을 큐 관리자가 실행 중인 시스템과 같은 시스템을 이동하고 바인딩 모드 연결을 사용하여 연결하십시오.

   바인딩 모드 연결은 플랫폼 고유 라이브러리를 사용하여 IBM WebSphere MQ API 호출을 수행합니다. 이에 따라, 고유 AMS 인터셉터는 AMS 조작을 수행하는 데 사용되고 JRE의 기능에는 신뢰성이 없습니다.
3. MCA 인터셉터는 클라이언트가 AMS 처리를 수행할 필요 없이 큐 관리자에 도착하자마자 메시지의 서명 및 암호화를 허용하므로 이를 사용하십시오.

   보호가 큐 관리자에서 적용되는 점을 고려할 때 클라이언트에서 큐 관리자로 전송 중인 메시지를 보호하기 위해 대체 메커니즘이 사용되어야 합니다. 가장 일반적으로 이는 애플리케이션이 사용하는 서버 연결 채널에서 SSL/TLS 암호화를 구성하여 달성됩니다.
4. IBM WebSphere MQ Advanced Message Security를 사용하고 싶지 않은 경우 AMQ_DISABLE_CLIENT_AMS 환경 변수를 설정하십시오.

자세한 정보는 272 페이지의 『메시지 채널 에이전트(MCA) 인터셉션』의 내용을 참조하십시오.

**참고:** MCA 인터셉터가 메시지를 전달할 각 큐에 대해 보안 정책이 있어야 합니다. 즉, 대상 큐는 MCA 인터셉터에 지정된 인증서의 식별 이름과 일치하는 서명자 및 수신자의 식별 이름(DN)과 함께 AMS 보안 정책이 있어야 합니다. 즉, 큐 관리자가 사용하는 `keystore.conf`에서 `cms.certificate.channel.SYSTEM.DEF.SVRCONN` 특성이 지정한 인증서의 DN입니다.

# 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

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

150-945
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.

지적 재산권 라이센스 부여
2-31 Roppongi 3-chome, Minato-Ku
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

**다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.** IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 명시적 또는 묵시적인 일체의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 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

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 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다.

이 정보를 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.

This book contains information on intended programming interfaces that allow the customer to write programs to obtain the services of IBM WebSphere MQ.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버그를 돕기 위해 제공된 것입니다.

**중요사항:** 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

## 상표

IBM, IBM 로고, ibm.com®는 전세계 여러 국가에 등록된 IBM Corporation의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/copytrade.shtml) 에 있습니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

이 제품에는 Eclipse 프로젝트 (http://www.eclipse.org/) 에서 개발한 소프트웨어가 포함되어 있습니다.

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

부품 번호:

(1P) P/N: