7.5

# *IBM WebSphere MQ* の保護

**注記**

本書および本書で紹介する製品をご使用になる前に、327 ページの『特記事項』に記載されている情報をお読みください。

本書は、 IBM® WebSphere® MQ バージョン 7 リリース 5、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様が IBM に情報を送信する場合、お客様は IBM に対し、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で情報を使用または配布する非独占的な権利を付与します。

# 目次

# 機密保護

セキュリティーは、IBM WebSphere MQ アプリケーションの開発者と IBM WebSphere MQ 権限を構成するシステム管理者の両方にとって重要な考慮事項です。

# セキュリティーの概要

このトピック集では、IBM WebSphere MQ セキュリティー概念について説明します。

コンピューター・システムに適用されるときに、まずセキュリティー概念およびメカニズムが表示され、続いて IBM WebSphere MQ に実装されるときに、それらのセキュリティー・メカニズムの説明が表示されます。

## セキュリティーの概念とメカニズム

このトピック集では、IBM WebSphere MQ のインストールで考慮する必要があるセキュリティーの側面について説明します。

一般に受け入れられているセキュリティーの側面は以下のとおりです。

- 5 ページの『識別と認証』
- 6 ページの『許可』
- 6 ページの『監査』
- 7 ページの『機密性』
- 7 ページの『データ整合性』

セキュリティー・メカニズム は、セキュリティー・サービスをインプリメントするために使用される、技術的なツールと技術です。 特定のサービスを提供するために単独で動作するメカニズムもあれば、他のメカニズムと連携して動作するメカニズムもあります。 一般的なセキュリティー・メカニズムの例を挙げれば、以下のようになります。

- 7 ページの『暗号化方式』
- 9 ページの『メッセージ・ダイジェストとデジタル署名』
- 9 ページの『デジタル証明書』
- 13 ページの『公開鍵インフラストラクチャー (PKI)』

IBM WebSphere MQ の実装を計画している場合は、重要なそれらのセキュリティーの側面を実装するにはどのセキュリティー・メカニズムが必要かを検討してください。 これらのトピックを読んだ後に検討しなければならない事柄については、45 ページの『セキュリティー要件の計画』を参照してください。

**関連概念**

206 ページの『SSL または TLS による 2 つのキュー・マネージャーの接続』
SSL または TLS 暗号セキュリティー・プロトコルを使用するセキュア通信では、 通信チャネルをセットアップし、認証に使用するデジタル証明書を管理する必要があります。

111 ページの『SSL または TLS の取り扱い』
これらのトピックでは、IBM WebSphere MQ での SSL または TLS の使用に関連した単一タスクを実行する方法について説明します。

### 識別と認証

識別 とは、システムのユーザー、またはシステムで実行するアプリケーションを一意的に識別する機能のことをいいます。 認証 とは、ユーザーまたはアプリケーションが本人または本物であることを証明する機能のことをいいます。

例えば、ユーザーが、ユーザー ID とパスワードを入力してシステムにログオンする場合を考えてみましょう。 システムは、ユーザー ID を使用してユーザーを識別します。 さらにシステムは、ログオン時に指定されたパスワードが正しいかどうかを確認してユーザーを認証します。

## 否認防止

否認防止 サービスは、識別と認証サービスの拡張版と見なすことができます。 一般に、否認防止が適用されるのは、データが電子的に送信される場合です。例えば、株式の仲買人が株を売買する注文や、口座間で資金を振り替えるための銀行への注文などです。

否認防止サービスの全体的な目標は、特定のメッセージが特定の個人に関連していることを証明する、ということです。

否認防止サービスには、複数のコンポーネントが組み込まれ、各コンポーネントが別々の機能を提供します。 メッセージの送信側が、メッセージを送信したことを否認する場合、発信証明 を持つ否認防止サービスは、その特定の個人によってメッセージが送信されたことを証明する、否認できない証拠を、受信側に提供できます。 メッセージの受信側が、メッセージを受信したことを否認する場合、送達証明 を持つ否認防止サービスは、その特定の個人によってメッセージが受信されたことを証明する、否認できない証拠を、送信側に提供できます。

実際に、ほぼ 100% の確実性のある証明、すなわち否認できない証拠は、達成が難しい目標です。 実際の世界では、完全に安全なものはありません。 セキュリティーの管理では、ビジネスに許容可能なレベルまでリスクを管理することに関心が高まっています。 このような環境では、許容でき、裁判で認められる証拠を提供できることが、否認防止サービスに対して、より現実的に求められることです。

IBM WebSphere MQ は、データを電子的に送信する手段であるので、否認防止は、IBM WebSphere MQ 環境における適切なセキュリティー・サービスです。 例えば、特定の個人に関連したアプリケーションによって、特定のメッセージが送信または受信されたという、同時証拠が必要な場合があります。

IBM WebSphere MQ Advanced Message Security を備えた IBM WebSphere MQ には、基本機能の一部として否認防止サービスが用意されていません。 しかし、この製品資料には、独自の出口プログラムを作成することによって、WebSphere MQ 環境で独自の否認防止サービスを用意する方法に関する情報が含まれています。

**関連概念**

20 ページの『IBM WebSphere MQ による識別と認証』
IBM WebSphere MQ では、メッセージ・コンテキスト情報および相互認証を使用して識別と認証を実装できます。

## 許可

許可は、許可ユーザーとそのアプリケーションだけにアクセスを制限することによって、システム内のクリティカル・リソースを保護します。 これにより、リソースの無許可の使用、または無許可の方法によるリソースの使用を防止します。

**関連概念**

21 ページの『IBM WebSphere MQ での許可』
許可を使用して、特定の個人またはアプリケーションが IBM WebSphere MQ 環境で実行できることを制限できます。

## 監査

監査とは、予期しないまたは許可されていないアクティビティーが実行されたかどうか、あるいはこうしたアクティビティーを実行しようとする試みがなされたかどうかを検出するために、イベントを記録および検査するプロセスのことです。

許可をセットアップする方法について詳しくは、48 ページの『許可の計画』および関連するサブトピックを参照してください。

**関連概念**

21 ページの『IBM WebSphere MQ での監査』
IBM WebSphere MQ は、イベント・メッセージを実行して異常なアクティビティーが行われたことを記録できます。

## 機密性

機密性 サービスは、重要な機密情報が無許可で開示されることを防止します。

データにアクセスできなければ、データが読み取られないことを前提とすると、機密データがローカル側に保管されている場合は、アクセス制御メカニズムで機密データを保護できます。 これより高いレベルのセキュリティーが必要である場合は、データを暗号化することができます。

通信ネットワーク (特にインターネットなどの危険性の高いネットワーク) で機密データを送信する場合は、その機密データを暗号化します。 ネットワーキング環境では、アクセス制御メカニズムは、盗聴などのデータの代行受信に対しては無効です。

## データ整合性

データ保全性 サービスは、データに無許可の変更が加えられたかどうかを検出します。

データの変更には 2 とおりあります。つまり、ハードウェアや伝送のエラーによる偶発的なものと、意図的な攻撃によるものです。 多くのハードウェア製品や伝送プロトコルには、ハードウェア・エラーや伝送エラーを検出して修正するメカニズムがあります。 データ保全性サービスの目的は、意図的な攻撃を検出することです。

データ保全性サービスは、データが変更されたかどうかの検出だけを目的とします。 このサービスは、データが変更された場合に、それを元の状態に戻すことを目的としてはいません。

アクセスが拒否されれば、データを変更できないことを前提とすれば、アクセス制御メカニズムが、データ保全性の確保に役立ちます。 しかし、機密性の場合と同様に、アクセス制御メカニズムは、ネットワーキング環境では無効です。

## 暗号の概念

このトピック集では、WebSphere MQ に該当する暗号方式の概念を取り上げます。

ここで使用するエンティティー という語は、キュー・マネージャー、WebSphere MQ MQI クライアント、個々のユーザー、メッセージを交換できる他のシステムのいずれかを指します。

**関連概念**

22 ページの『IBM WebSphere MQ での暗号化』

IBM WebSphere MQ は、Secure sockets Layer (SSL) および Transport Security Layer (TLS) プロトコルを使用して暗号化を提供します。

### 暗号化方式

暗号化方式とは、平文 と呼ばれる可読テキストと、暗号文 と呼ばれる非可読形式との間で変換を行うプロセスです。

以下のような流れになります。

1. 送信側が、plaintext のメッセージを ciphertext に変換する。 プロセスのこの部分は、暗号化 (場合によっては 暗号化方式) と呼ばれます。
2. ciphertext が受信側に送信される。
3. 受信側が、ciphertext のメッセージを plaintext 形式に戻す。 プロセスのこの部分は、復号 (場合によっては、暗号化解除) と呼ばれます。

暗号化の定義については、用語集を参照してください。

この変換には、伝送中のメッセージの外観を変えるが内容には影響を与えない、一連の数学的な演算が含まれています。 暗号化したメッセージは理解不能になるので、暗号化の手法を使用すれば、機密性を確保し、無許可の表示 (盗聴) からメッセージを保護できます。 メッセージの保全性を確保するデジタル署名でも、暗号化の手法を使用します。 詳細については、18 ページの『SSL および TLS でのデジタル署名』を参照してください。

暗号化の手法には、鍵の使用により固有のものにされる、一般的なアルゴリズムが使用されます。 次の 2 つのクラスのアルゴリズムがあります。

- 送信側と受信側の両方が同じ秘密鍵 (secret key) を使用することを必要とするアルゴリズム。 共有鍵を使用するアルゴリズムは、対称 アルゴリズムと呼ばれます。8 ページの図 1 対称鍵暗号方式を示しています。
- 暗号化と復号に別々の鍵を使用するアルゴリズム。 どちらかの鍵を秘密にする必要があります。もう一方の鍵は公開できます。 公開鍵と秘密鍵のペアを使用するアルゴリズムは、非対称 アルゴリズムと呼ばれます。8 ページの図 2 は、 公開鍵暗号方式とも呼ばれる非対称鍵暗号方式を示しています。

使用する暗号化と復号のアルゴリズムは、公開できますが、共有秘密鍵と秘密鍵は秘密にしておく必要があります。

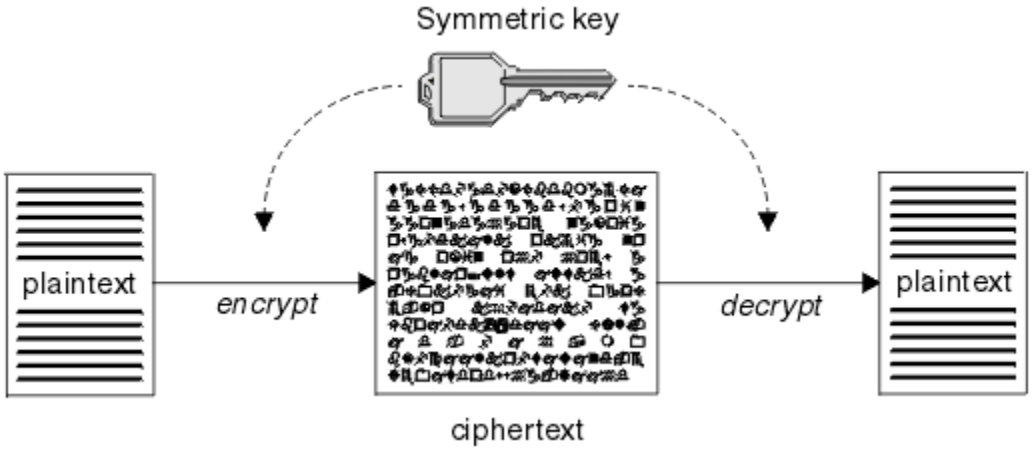



図 *1.* 対称鍵暗号化方式

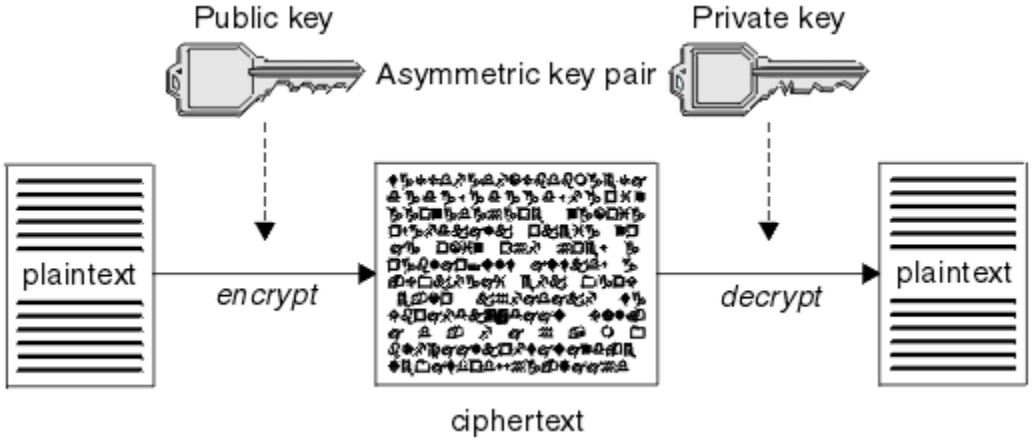



図 *2.* 非対称鍵暗号化方式

8 ページの図 2 は、受信側の公開鍵を使用して暗号化され、受信側の秘密鍵を使用して復号される plaintext を示しています。 所定の受信側だけが、ciphertext を復号するための秘密鍵を保持します。 送信側が、秘密鍵を使用してメッセージを暗号化することも可能であることに注意してください。秘密鍵を使用して暗号化すると、送信側の公開鍵を持っている任意の人物が、メッセージを復号できるようになり、メッセージがその送信側から送信されたものであることが保証されます。

非対称アルゴリズムでは、メッセージは、公開鍵または秘密鍵のどちらかで暗号化されますが、復号するには、もう一方の鍵しか使用できません。 秘密鍵だけが秘密であり、公開鍵はだれでも知ることができます。 対称アルゴリズムでは、共有鍵を知っているのが、送信側と受信側だけでなければなりません。 これは、鍵配布の問題 と呼ばれます。 非対称アルゴリズムの方が、低速ですが、鍵配布の問題がないという利点があります。

暗号化方式に関連したその他の用語は、次のとおりです。

**強度**
暗号化の強度は、鍵のサイズによって決まります。 非対称アルゴリズムには、大きな鍵が必要です。 例えば、次のようにします。

| | |
|---|---|
| 1024 ビット | 低強度の非対称鍵 |
| 2048 ビット | 中強度の非対称鍵 |
| 4096 ビット | 高強度の非対称鍵 |

対称鍵はこれより小さく、256 ビット・キーで強い暗号化機能が得られます。

**ブロック暗号化アルゴリズム**
このアルゴリズムは、データをブロックごとに暗号化します。 例えば、RSA Data Security Inc. の RC2 アルゴリズムは、8 バイト長のブロックを使用します。 通常、ブロック・アルゴリズムは、ストリーム・アルゴリズムよりも低速です。

**ストリーム暗号化アルゴリズム**
このアルゴリズムは、データの各バイトを暗号化の対象にします。 通常、ストリーム・アルゴリズムは、ブロック・アルゴリズムよりも高速です。

## メッセージ・ダイジェストとデジタル署名

メッセージ・ダイジェストは、ハッシュ関数によって計算される、メッセージの内容に相当する固定サイズの数値表現です。 メッセージ・ダイジェストは暗号化して、デジタル署名を作成することができます。

本来、メッセージのサイズは一定ではありません。 メッセージ・ダイジェストは、メッセージの内容に相当する固定サイズの数値表現です。 メッセージ・ダイジェストは、次の 2 つの基準を満たす変換を行うハッシュ関数によって計算されます。

- ハッシュ関数は単方向でなければならない。 関数の方向を逆にして、特定のメッセージ・ダイジェストに対応するメッセージを見つけることが不可能でなければなりません (可能性のあるメッセージをすべてテストする場合は除きます)。
- 同じダイジェストにハッシュされる 2 つのメッセージを見つけることは、計算上不可能でなければならない。

メッセージ・ダイジェストは、メッセージ自体と一緒に送信されます。 受信側は、メッセージ用のダイジェストを生成して、送信側のダイジェストと比較することができます。 メッセージの保全性は、2 つのメッセージ・ダイジェストが同じ場合に検証されます。 伝送中にメッセージに改ざんが行われると、ほぼ確実に、メッセージ・ダイジェストが異なります。

秘密対称鍵を使用して作成されるメッセージ・ダイジェストは、メッセージが変更されていないことを保証することができるので、メッセージ認証コード (MAC) とも呼ばれます。

送信側は、メッセージ・ダイジェストを生成してから、非対称秘密鍵のペアを使用してダイジェストを暗号化し、デジタル署名を作成することもできます。 署名は、ローカルに生成されたダイジェストと比較する前に、受信側で暗号化解除される必要があります。

**関連概念**
18 ページの『SSL および TLS でのデジタル署名』
デジタル署名は、メッセージの表記を暗号化することによって作成されます。 この暗号化は、署名者の秘密鍵を使用し、通常、効率を上げるために、メッセージ自体ではなく、メッセージ・ダイジェストを対象とし行われます。

## デジタル証明書

デジタル証明書を使用すると、ある公開鍵が指定されたエンティティーに属することが認証され、偽名の使用による被害を防ぐことができます。 デジタル証明書は認証局によって発行されます。

デジタル証明書は、偽名の使用を防止します。これは、公開鍵の所有者が個人であるか、キュー・マネージャーであるか、その他のエンティティーであるかに関係なく、デジタル証明書は公開鍵をその所有者にバインドするからです。 デジタル証明書は、非対称鍵体系を使用する場合に公開鍵の所有権を保証するので、公開鍵証明書とも呼ばれます。 デジタル証明書には、エンティティーの公開鍵が含まれ、公開鍵がそのエンティティーに属していることを表明します。

- 証明書が個人エンティティーの証明書である場合、個人用証明書 またはユーザー証明書 と呼ばれます。

- 証明書が認証局の証明書である場合、*CA* 証明書 または署名者証明書 と呼ばれます。

公開鍵が、所有者によって別のエンティティーに直接送信される場合、メッセージが傍受され、公開鍵が別のものに置き換えられる危険性があります。 これは、中間一致攻撃 *(man in the middle attack)* と呼ばれます。 この問題の解決法は、公開鍵が通信相手のエンティティーに本当に属していることを確実に保証してくれる信頼のおける三者機関を通じて公開鍵を交換するというものです。 公開鍵を直接送信する代わりに、公開鍵をデジタル証明書に組み込むように、信頼のおける第三者機関に依頼します。 デジタル証明書を発行する、信頼のおける第三者機関は、認証局 (CA) と呼ばれます。CA については、11 ページの『認証局』を参照してください。

### デジタル証明書の内容

デジタル証明書には、X.509 標準で規定された、特定の情報が含まれています。

WebSphere MQ によって使用されるデジタル証明書は、X.509 標準に準拠します。この標準は、必要な情報と、その情報を送信するための形式を指定します。 X.509 は、X.500 シリーズの標準の Authentication フレームワーク部分です。

デジタル証明書には、少なくとも、認証されるエンティティーについて次の情報が含まれています。

- 所有者の公開鍵
- 所有者の識別名
- 証明書を発行した CA の識別名
- 証明書の発効日
- 証明書の有効期限日
- X.509 で定義された証明書データ形式のバージョン番号。 X.509 標準の現行バージョンはバージョン 3 であり、ほとんどの証明書はそのバージョンに準拠しています。
- シリアル番号。 これは、証明書を発行した CA によって割り当てられる固有 ID です。 シリアル番号は、証明書を発行した CA 内で固有のものです。つまり、同じ CA 証明書によって署名された 2 つの証明書が同じシリアル番号を持つことはありません。

X.509 バージョン 2 証明書には発行者 ID とサブジェクト ID も含まれ、X.509 バージョン 3 証明書にはいくつかの拡張情報を含めることができます。 証明書の拡張には、基本制約拡張のように標準のものと、実装に特有のものがあります。 拡張はクリティカルな場合があります。その場合、システムがそのフィールドを認識できる必要があります。フィールドを認識できない場合、システムは証明書を拒否する必要があります。 拡張がクリティカルでない場合、システムは、そのフィールドを認識できない場合はそれを無視することができます。

個人証明書のデジタル署名は、その証明書を署名した CA の秘密鍵を使用して生成されます。 個人証明書を検証する必要があるユーザーは、CA の公開鍵を使用してこれを行うことができます。 CA の証明書には、その公開鍵が含まれています。

デジタル証明書には、秘密鍵は入っていません。 秘密鍵は秘密にしておく必要があります。

### 個人用証明書の要件

WebSphere MQ は、X.509 規格に準拠したデジタル証明書をサポートしています。 そのためには、クライアント認証オプションが必要です。

IBM WebSphere MQ はピアツーピア・システムであるため、SSL 用語では、これはクライアント認証と見なされます。 したがって、SSL 認証に使用される個人証明書が、クライアント認証の鍵使用を許可する必要があります。 すべてのサーバー証明書でこのオプションが使用可能になっているわけではないので、証明書の提供者は、場合によっては、安全な証明書のためルート CA でクライアント認証を使用可能にする必要があります。

デジタル証明書のデータ形式を指定する標準に加えて、証明書が有効であるかどうかを判別するための標準もあります。 これらの標準は、特定の種類のセキュリティー・ブリーチ (抜け穴) を防ぐために、時間の経過とともに更新されます。 例えば、旧来の X.509 バージョン 1 および 2 証明書には、その証明書が他の証明書を署名するために正当に使用可能であるかどうかが示されませんでした。 そのため、悪意あるユーザーが正当な提供元から個人証明書を入手し、他のユーザーの偽名を使用する目的で使用する新たに証明書を作成することが可能でした。

X.509 バージョン 3 証明書を使用すると、BasicConstraints および KeyUsage 証明書拡張によって、どの証明書が他の証明書を署名するために正当に使用可能であるかを指定することができます。 IETF RFC 5280 標準には一連の証明書妥当性検査のルールが規定されており、偽名攻撃を予防するために準拠アプリケーション・ソフトウェアはこのルールを実装する必要があります。 証明書ルール一式は、証明書妥当性検査ポリシーとして知られています。

IBM WebSphere MQ での証明書妥当性検査ポリシーの詳細については、33 ページの『IBM WebSphere MQ での証明書妥当性検査ポリシー』を参照してください。

### 認証局

認証局 (CA) とは、エンティティーの公開鍵が本当にそのエンティティーに属するものであることの保証を与えてくれるデジタル証明書を発行する、信頼のおける第三者機関です。

CA の役割は、次のとおりです。

- デジタル証明書に対する要求を受け取った後、要求側の ID を確認してから、個人用証明書の作成、署名、返送を行う
- CA 証明書内で CA 自身の公開鍵を提供する
- 証明書取り消しリスト (CRL) 内で、信頼されなくなった証明書のリストを公開する。 詳しくは、150 ページの『取り消された証明書の取り扱い』を参照してください。
- OCSP 応答側サーバーを操作して、証明書の失効状況にアクセスする

### 識別名

識別名 (DN) は、X.509 証明書内のエンティティーを固有に識別します。

一般に、DN では次の属性タイプが使用されます。

| | |
|---|---|
| SERIALNUMBER | 証明書のシリアル番号 |
| MAIL | メール・アドレス |
| E | E メール・アドレス (MAIL の方が好ましいため非推奨) |
| UID または USERID | ユーザー ID |
| CN | 共通名 |
| T | 役職 |
| OU | 部門名 |
| DC | ドメイン・コンポーネント |
| O | 組織名 |
| STREET | 通り/住所の 1 行目 |
| L | 地域名 |
| ST (または SP もしくは S) | 都道府県名 |
| 「PC」 | 郵便番号 |
| C | 国名 |
| UNSTRUCTUREDNAME | ホスト名 |
| UNSTRUCTUREDADDRESS | IP アドレス |
| DNQ | 識別名修飾子 |

X.509 標準は、通常は DN に含まれないが、デジタル証明書にオプショナルの拡張機能を提供できるその他の属性を定義します。

X.509 標準は、DN がストリング形式で指定されることを定めています。 以下に例を示します。

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

コモン・ネーム (CN) は、個々のユーザー、またはその他の任意のエンティティー (例えば、Web サーバー) を記述できます。

DN には、複数の OU および DC 属性を含めることができます。 他の属性の場合は、それぞれ 1 つのインスタンスのみが許可されます。 OU 項目の順序が重要です。この順序は、(最高レベルの部門を先頭とする) 部門名の階層を指定します。 DC 項目の順序も重要です。

IBM WebSphere MQ は、特定の誤った形式の DN を許容します。 詳細については、SSLPEER 値についての WebSphere MQ の規則を参照してください。

**関連概念**

10 ページの『デジタル証明書の内容』
デジタル証明書には、X.509 標準で規定された、特定の情報が含まれています。

### 認証局からの個人用証明書の取得

信頼できる外部の認証局 (CA) から証明書を取得することができます。

デジタル証明書を取得するには、認証要求の形式で CA に情報を送信します。 X.509 標準は、この情報の形式を定義しますが、CA の中には独自の形式を持つものがあります。 証明書要求は通常、システムが使用する証明書管理ツールによって生成されます。例えば、 UNIX、 Linux®、および Windows システム上の iKeyman ツールと、 z/OS®上の RACF® などです。 この情報には、識別名および公開鍵が含まれます。 証明書管理ツールが証明書要求を生成するときに、秘密鍵も生成します。この秘密鍵は、秘密にしておく必要があります。 秘密鍵を配布しないでください。

CA がユーザーの要求を受け取ると、CA は、ユーザーの ID を検証した後、証明書を作成し、個人用証明書としてユーザーに返送します。

12 ページの図 3 は、CA からデジタル証明書を取得するプロセスを示しています。

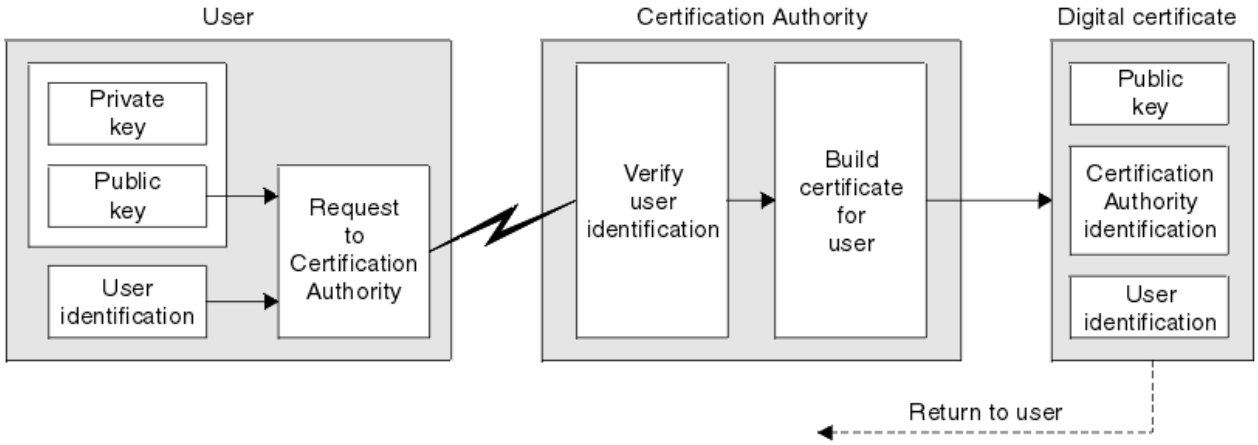



図 *3.* デジタル証明書の取得

図の説明:

- 「ユーザー識別」には、サブジェクト識別名が含まれます。
- 「認証局の識別」には、証明書を発行している CA の識別名が含まれます。
-

デジタル証明書には、図で示した以外にも追加フィールドが含まれます。 デジタル証明書内のその他のフィールドについては、10 ページの『デジタル証明書の内容』を参照してください。

### 証明書チェーンの働き

別のエンティティー用の証明書を受け取る場合、ルート *CA* 証明書を取得するために、証明書チェーン の使用が必要になる場合があります。

証明書チェーンは、認証パス とも呼ばれ、エンティティーの認証に使用される証明書のリストです。 このチェーンまたはパスは、そのエンティティーの証明書から始まり、チェーン内の各証明書は、チェーン内の次の証明書によって指定されるエンティティーによって署名されます。 チェーンは、ルート CA 証明書

で終了します。 ルート CA 証明書は、常に、認証局 (CA) 自体によって署名されます。 ルート CA 証明書に到達するまで、チェーン内のすべての証明書の署名が検証されなければなりません。

13 ページの図 4 は、証明書の所有者から、ルート CA までの認証パスを示しています。トラストのチェーンは、ルート CA から始まります。

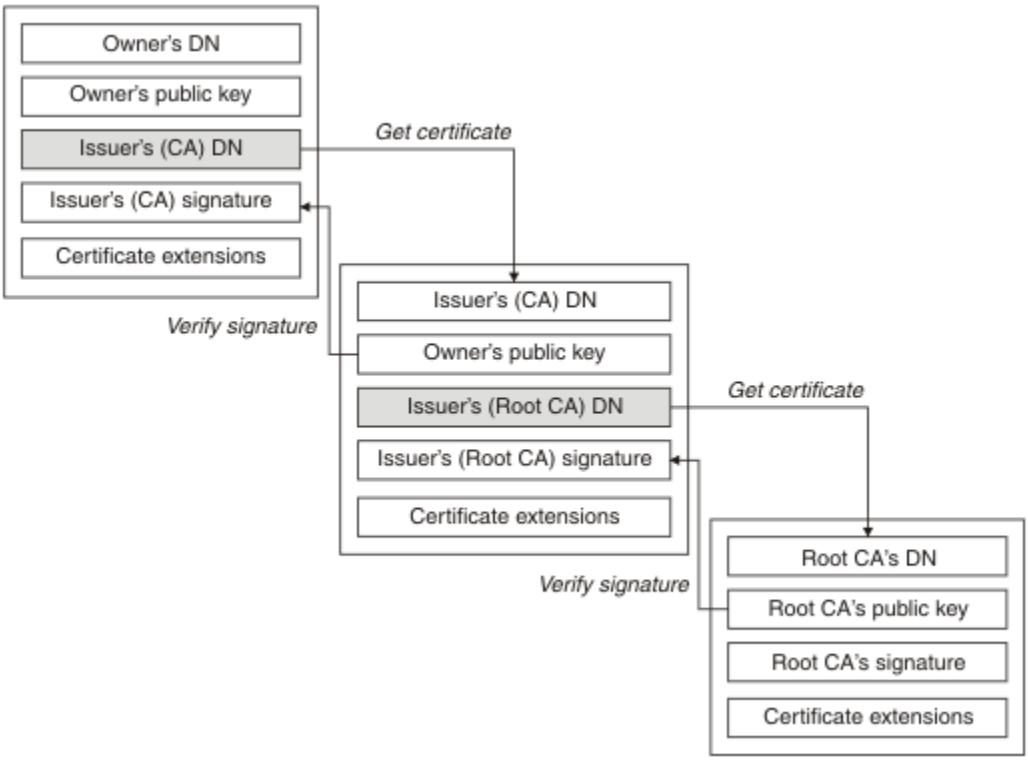



図 *4.* トラストのチェーン

それぞれの証明書には 1 つ以上の拡張が含まれることがあります。 CA に属する証明書には、通常、他の証明書に署名できることを示す isCA フラグが設定された BasicConstraints 拡張が含まれます。

### 証明書が無効になる場合

デジタル証明書は、有効期限が切れたり、取り消されたりすることがあります。

デジタル証明書は、一定の期間について発行され、有効期限日以降は無効になります。

証明書の有効期限の定義については、用語集を参照してください。

証明書は、次のような様々な理由で取り消される場合があります。

- 所有者が別の組織に移動した。
- 秘密鍵が秘密でなくなった。

WebSphere MQ は、Online Certificate Status Protocol (OCSP) 応答側に要求を送信することにより、ある証明書が取り消されているかどうかを確認できます (UNIX、Linux および Windows の場合のみ)。 あるいは、LDAP サーバーで CRL にアクセスすることもできます。 OCSP の失効情報および CRL 情報は、認証局によって公表されます。 詳しくは、150 ページの『取り消された証明書の取り扱い』を参照してください。

## 公開鍵インフラストラクチャー *(PKI)*

公開鍵インフラストラクチャー (PKI) は、トランザクションの当事者の認証に公開鍵暗号化方式の使用をサポートするシステムであり、機能、ポリシー、およびサービスから構成されます。

公開鍵インフラストラクチャー (PKI) のコンポーネントを定義する単一の標準があるのではなく、PKI は、通常、認証局 (CA) と登録局 (RA) から構成されています。 CA は、次のサービスを提供します。

- デジタル証明書を発行する
- デジタル証明書を検証する

- デジタル証明書を取り消す
- 公開鍵を配布する

X.509 標準は、業界標準の公開鍵インフラストラクチャー (PKI) の基礎を提供します。

デジタル証明書と認証局 (CA) の詳細については、9 ページの『デジタル証明書』を参照してください。 RA は、デジタル証明書が要求されるときに提供される情報を検証します。 RA がその情報を検証すると、CA はデジタル証明書を要求側に発行することができます。

PKI は、デジタル証明書と公開鍵を管理するためのツールも提供することができます。 場合によっては、PKI は、デジタル証明書を管理するためのトラスト階層 と呼ばれますが、大部分の定義には、追加サービスが含まれます。 一部の定義には、暗号化サービスとデジタル署名サービスが含まれますが、これらのサービスは、PKI の運用にとって不可欠ではありません。

## 暗号セキュリティー・プロトコル: SSL および TLS

暗号プロトコルは、2 者間の通信のプライバシーとデータ保全性を確保できる セキュア接続を提供します。 Transport Layer Security (TLS) プロトコルは Secure Sockets Layer (SSL) が進化したものです。 IBM WebSphere MQ は、SSL と TLS の両方をサポートしています。

両方のプロトコルの基本的な目標は、機密性 (プライバシー と呼ばれることもある)、データ保全性、識別、および認証を、デジタル証明書を使用して提供することです。

2 つのプロトコルは、似たところもありますが、SSL 3.0 と TLS のさまざまなバージョンは相互運用しないことから 分かるように両者は大きく異なります。

**関連概念**

22 ページの『IBM WebSphere MQ のセキュリティー・プロトコル』
IBM WebSphere MQ は、Transport Layer Security (TLS) プロトコルと Secure Sockets Layer (SSL) プロトコルの両方をサポートして、メッセージ・チャネルと MQI チャネルにリンク・レベルのセキュリティーを提供します。

### *Secure Sockets Layer (SSL)* および *Transport Layer Security (TLS)* の概念

SSL および TLS プロトコルを使用すると、2 者間で、相互に識別および認証したり、機密性とデータ保全性を確保しながら通信したりすることができます。 TLS プロトコルは、Netscape の SSL 3.0 プロトコルが進化したものですが、TLS と SSL 間に相互運用性はありません。

SSL および TLS プロトコルは、インターネット上の通信セキュリティーを提供します。 クライアント/サーバー・アプリケーションはこれらのプロトコルを使用することで、機密が保護された高信頼の通信を行えます。 プロトコルには、Record Protocol と Handshake Protocol の 2 つの層があります。 これらは、TCP/IP などのトランスポート・プロトコルの上の層になります。 これらは両方とも、非対称と対称の暗号化手法を使用します。

SSL または TLS 接続はアプリケーションによって開始され、このアプリケーションが SSL または TLS クライアントになります。 接続を受け取るアプリケーションが、SSL または TLS サーバーになります。 新たに開始されたどのセッションも、SSL または TLS プロトコルによって定義されるハンドシェークから始まります。

IBM WebSphere MQ でサポートされる CipherSpecs の全リストは、217 ページの『CipherSpec の指定』に記載されています。

SSL プロトコルの詳細については、https://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt で提供されている情報を参照してください。 TLS プロトコルの詳細については、Internet Engineering Task Force の Web サイト (https://www.ietf.org) で TLS Working Group によって提供されている情報を参照してください。

### *SSL* または *TLS* ハンドシェークの概要

SSL または TLS ハンドシェークにより、SSL または TLS クライアントと SSL または TLS サーバーは 通信に使用する秘密鍵を設定できます。

このセクションでは、SSL または TLS クライアントとサーバーが相互に通信できるようにするステップの要約を示します。

- 使用するプロトコルのバージョンについて合意する。
- 暗号アルゴリズムを選択する。
- デジタル証明書を交換し、検証して、互いを認証する。
- 非対称暗号化手法を使用して、共有秘密鍵を生成する。これにより、鍵配布の問題が避けられます。 その後、SSL または TLS は、この共有鍵を使用してメッセージの対称暗号化の処理を実行します。対称暗号化は、非対称暗号化より高速です。

暗号アルゴリズムとデジタル証明書の詳細については、関連情報を参照してください。

SSL ハンドシェークに必要な手順の概要は、次のとおりです。

1. SSL または TLS クライアントは、 "client hello" メッセージを送信します。このメッセージには、SSL または TLS バージョンなどの暗号情報と、クライアントによってサポートされる CipherSuites が優先順にリストされます。 また、このメッセージには、以降の計算で使用されるランダム・バイト・ストリングも入っています。 このプロトコルにより、"クライアント・ハロー" には、クライアントがサポートするデータ圧縮メソッドを組み込むことができます。
2. SSL または TLS サーバーは、 "server hello" メッセージで応答します。このメッセージには、クライアントによって提供されたリストからサーバーが選択した CipherSuite 、セッション ID、および別のランダム・バイト・ストリングが含まれています。 また、サーバーは、そのデジタル証明書も送信します。サーバーがクライアント認証用のデジタル証明書を必要とする場合、サーバーは、サポートされる証明書のタイプのリストと、受け入れ可能な認証局 (CAs)の識別名を含む"クライアント証明書要求"を送信します。
3. SSL または TLS クライアントはサーバーのデジタル証明書を検証します。 詳しくは、16 ページの『SSL および TLS による識別、認証、機密性、保全性』を参照してください。
4. SSL または TLS クライアントは、クライアントとサーバーの両方が以降のメッセージ・データの暗号化に使用する秘密鍵を計算できるようにする、ランダム・バイト・ストリングを送信する。 このランダム・バイト・ストリング自体は、サーバーの公開鍵を使用して暗号化されます。
5. SSL または TLS サーバーが "クライアント証明書要求"を送信した場合、クライアントは、クライアントの秘密鍵で暗号化されたランダム・バイト・ストリングを、クライアントのデジタル証明書とともに送信するか、または "デジタル証明書なしアラート"を送信します。 このアラートは警告にすぎませんが、一部のインプリメンテーションでは、クライアント認証が必須である場合、ハンドシェークは失敗します。
6. SSL または TLS サーバーはクライアントの証明書を検査します。 詳しくは、16 ページの『SSL および TLS による識別、認証、機密性、保全性』を参照してください。
7. SSL または TLS クライアントは、サーバーに "終了" メッセージを送信します。このメッセージは秘密鍵で暗号化され、ハンドシェークのクライアント部分が完了したことを示します。
8. SSL または TLS サーバーは、クライアントに "終了" メッセージを送信します。このメッセージは秘密鍵で暗号化され、ハンドシェークのサーバー部分が完了したことを示します。
9. SSL または TLS セッションの間、サーバーとクライアントは、共有秘密鍵を使用して対称的に暗号化されるメッセージを交換できるようになる。

16 ページの図 5 は、SSL または TLS ハンドシェークを示しています。

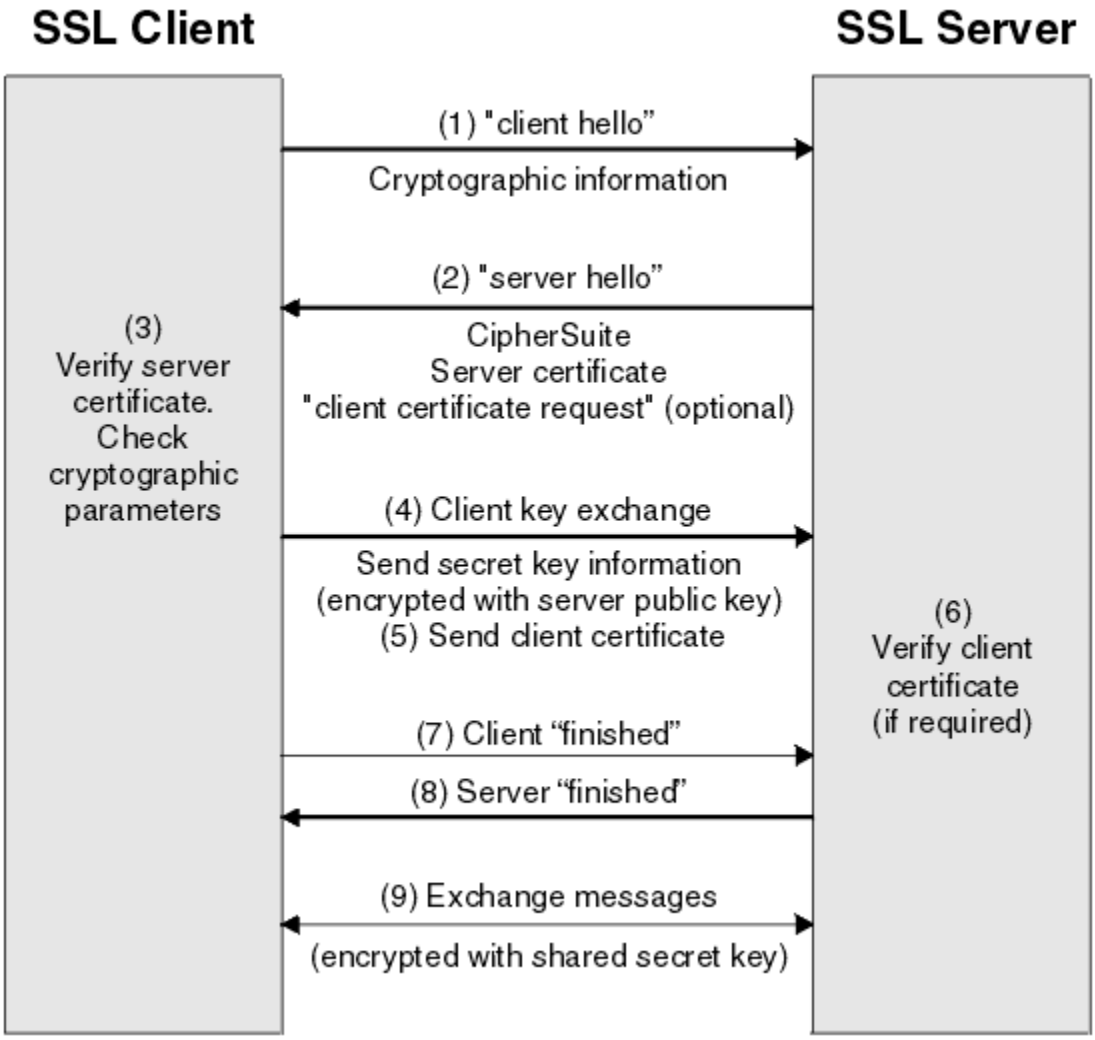



図 *5. SSL* または *TLS* ハンドシェークの概要

### *SSL* および *TLS* による識別、認証、機密性、保全性

クライアントとサーバーの両方の認証時に、非対称鍵のペアで鍵のどちらかを使用してデータを暗号化し、ペアのもう一方の鍵を使用して復号することが必要な手順があります。 保全性のためには、メッセージ・ダイジェストを使用します。

TLS ハンドシェークに関連するステップの概要については、14 ページの『SSL または TLS ハンドシェークの概要』を参照してください。

### SSL および TLS での認証

サーバーの認証の場合、クライアントはサーバーの公開鍵を使用して、秘密鍵の計算に使用されるデータを暗号化します。 サーバーは、正しい秘密鍵を使用してそのデータを復号する場合だけ、秘密鍵を生成することができます。

クライアント認証の場合、サーバーは、クライアント証明書内の公開鍵を使用して、ハンドシェークのステップ 15 ページの『5』 でクライアントが送信するデータを復号します。 秘密鍵を使用して暗号化される終了メッセージの交換 (概要のステップ 15 ページの『7』 と 15 ページの『8』) により、認証が完了したことが確認されます。

認証ステップのいずれかが失敗すると、ハンドシェークが失敗し、セッションは終了します。

SSL または TLS ハンドシェーク時のデジタル証明書の交換は、認証プロセスの一環です。 証明書が偽名の使用をどのように防止するかについては、 関連情報を参照してください。 必要な証明書は、以下のとおり

です。ここで、CA X は、SSL または TLS クライアントに証明書を発行し、CA Y は、SSL または TLS サーバーに証明書を発行します。

サーバー認証のみの場合、SSL または TLS サーバーは次のものが必要です。

- CA Y によってサーバーに発行される個人用証明書
- サーバーの秘密鍵

SSL または TLS クライアントは次のものが必要です。

- CA Y の CA 証明書

SSL または TLS サーバーがクライアント認証を必要とする場合、サーバーは、クライアントに個人用証明書を発行した CA (この場合は CA X) の公開鍵を使用して、クライアントのデジタル証明書を検証することによって、クライアントの ID を検証します。 サーバーとクライアント認証のいずれの場合も、サーバーは次のものを必要とします。

- CA Y によってサーバーに発行される個人用証明書
- サーバーの秘密鍵
- CA X の CA 証明書

クライアントは次のものが必要です。

- CA X によってクライアントに発行される個人用証明書
- クライアントの秘密鍵
- CA Y の CA 証明書

SSL または TLS サーバーとクライアントの両方で、ルート CA 証明書までの証明書チェーンを作成するために、他の CA 証明書が必要になる場合があります。 証明書チェーンの詳細については、 関連情報を参照してください。

## 証明書の検査時に行われること

概要のステップ 15 ページの『3』 および 15 ページの『6』 で述べたように、SSL または TLS クライアントはサーバーの証明書を検査し、SSL または TLS サーバーはクライアントの証明書を検査します。 この検査には、次の 4 つの側面があります。

1. デジタル署名が検査されます (18 ページの『SSL および TLS でのデジタル署名』を参照)。
2. 証明書チェーンが検査されます。中間 CA 証明書が必要です (12 ページの『証明書チェーンの働き』を参照)。
3. 有効期限とアクティブ化の日付、および有効期間が検査されます。
4. 証明書の失効状況が検査されます (150 ページの『取り消された証明書の取り扱い』を参照)。

## 秘密鍵の再設定

SSL または TLS ハンドシェーク中に、SSL または TLS のクライアントとサーバー間のデータを暗号化するために秘密鍵 が生成されます。 秘密鍵は数式の中で使用され、その数式がデータに適用されて、平文を読み取り不能な暗号文に変換したり、暗号文を平文に変換したりします。

秘密鍵は、ハンドシェークの一部として送信されたランダム・テキストから生成され、平文を暗号文に暗号化するために使用されます。 秘密鍵は MAC (メッセージ確認コード) アルゴリズムでも使用されます。このアルゴリズムは、メッセージが変更されたかどうかの判断に使用されます。 詳細については、9 ページの『メッセージ・ダイジェストとデジタル署名』を参照してください。

秘密鍵が発見されれば、メッセージの平文を暗号文から復号したり、メッセージ・ダイジェストを計算したりできるので、メッセージを検出せずに変更できます。 複雑なアルゴリズムでも、可能なすべての数学的変換を暗号文に適用すれば、最後には平文を発見できます。 秘密鍵が破損している場合の復号または変更可能なデータ量を最小化するために、秘密鍵を定期的に再調整できます。 秘密鍵が再調整されると、前の秘密鍵は、新規の秘密鍵で暗号化されたデータの復号には使用できなくなります。

## SSL での機密性

SSL および TLS は、対称暗号化と非対称暗号化の組み合わせを使用して、メッセージのプライバシーを確保します。 SSL または TLS ハンドシェーク時に、SSL または TLS クライアントとサーバーは、1 つのセッションだけに使用される暗号化アルゴリズムと共有秘密鍵を一致させます。 SSL または TLS クライアントとサーバー間で伝送されるすべてのメッセージは、そのアルゴリズムと鍵を使用して暗号化され、メッセージが傍受された場合であっても、秘密のままであることを確実にします。 SSL は、さまざまな暗号アルゴリズムをサポートします。 SSL および TLS は、共有秘密鍵のトランスポート時に非対称暗号化を使用するので、鍵配布の問題はありません。 暗号化手法の詳細については、7 ページの『暗号化方式』を参照してください。

## SSL および TLS での保全性

SSL および TLS は、メッセージ・ダイジェストを計算してデータ保全性を提供します。 詳細については、227 ページの『メッセージのデータ保全性』を参照してください。

SSL または TLS を使用するとデータ保全性が確保されます (ただし、217 ページの『CipherSpec の指定』の表に示されているようにチャネル定義の CipherSpec でハッシュ・アルゴリズムが使用されている場合)。

特に、データ保全性が重要となる場合には、ハッシュ・アルゴリズム「なし」とリストされる CipherSpec を選択しないでください。 また、MD5 は非常に古くなり、ほとんどの場合、実用目的では安全と言えなくなったため、これを使用しないよう強くお勧めします。

### *CipherSpec* および *CipherSuite*

暗号セキュリティー・プロトコルは、セキュア接続で使用されるアルゴリズムと一致しなければなりません。 CipherSpec および CipherSuite は、アルゴリズムの特定の組み合わせを定義します。

CipherSpec は、暗号化アルゴリズムとメッセージ認証コード (MAC) アルゴリズムの組み合わせを指定します。 TLS 接続または SSL 接続の両端が通信できるようになるには、両端で CipherSpec が一致する必要があります。

**重要 :** IBM WebSphere MQ チャネルを扱う場合は、CipherSpec を使用します。 Java チャネル、JMS チャネル、または MQTT チャネルを扱う場合は、CipherSuite を指定します。

CipherSpecs について詳しくは、217 ページの『CipherSpec の指定』 を参照してください。

CipherSuite は、SSL 接続または TLS 接続で使用される 1 組の暗号アルゴリズムです。 1 組の暗号アルゴリズムは、次の 3 つの別々のアルゴリズムから構成されます。

- ハンドシェーク時に使用される鍵交換と認証のアルゴリズム
- データの暗号化に使用される暗号化アルゴリズム
- メッセージ・ダイジェストの生成に使用される MAC (メッセージ認証コード) アルゴリズム

1 組の中に含まれているコンポーネント (アルゴリズム) ごとにいくつかのオプションがありますが、TLS 接続または SSL 接続でアルゴリズムを指定する場合は、特定の組み合わせだけが有効になります。 有効な CipherSuite の名前で、使用されるアルゴリズムの組み合わせが指定されます。 例えば、CipherSuite SSL_RSA_WITH_RC4_128_MD5 は、次の組み合わせを指定します。

- RSA 鍵交換と認証のアルゴリズム
- 128 ビット鍵を使用する RC4 暗号化アルゴリズム
- MD5 MAC アルゴリズム

鍵交換と認証には複数のアルゴリズムが使用できますが、現在は RSA アルゴリズムが最も広く使用されています。 使用される暗号化アルゴリズムと MAC アルゴリズムには、さらに多くの種類があります。

### *SSL* および *TLS* でのデジタル署名

デジタル署名は、メッセージの表記を暗号化することによって作成されます。 この暗号化は、署名者の秘密鍵を使用し、通常、効率を上げるために、メッセージ自体ではなく、メッセージ・ダイジェストを対象とし行われます。

署名される文書の内容に依存しない手書きの署名とは異なり、デジタル署名は、署名されるデータに応じて変わります。 2 つの別々のメッセージが、同じエンティティーによってデジタル署名される場合、2 つ

の署名は異なりますが、両方の署名を同じ公開鍵、つまり、メッセージを署名したエンティティーの公開鍵で検証することができます。

デジタル署名プロセスのステップは、次のとおりです。

1. 送信側は、メッセージ・ダイジェストを計算した後、送信側の秘密鍵を使用してそのメッセージ・ダイジェストを暗号化して、デジタル署名を作成する。
2. 送信側は、メッセージと一緒にデジタル署名を送信する。
3. 受信側は、送信側の公開鍵を使用してデジタル署名を復号し、送信側のメッセージ・ダイジェストを再生成する。
4. 受信側は、受信したメッセージ・データからメッセージ・ダイジェストを計算し、この 2 つのダイジェストが同一であるかどうかを検証する。

19 ページの図 6 には、このプロセスが図示されています。

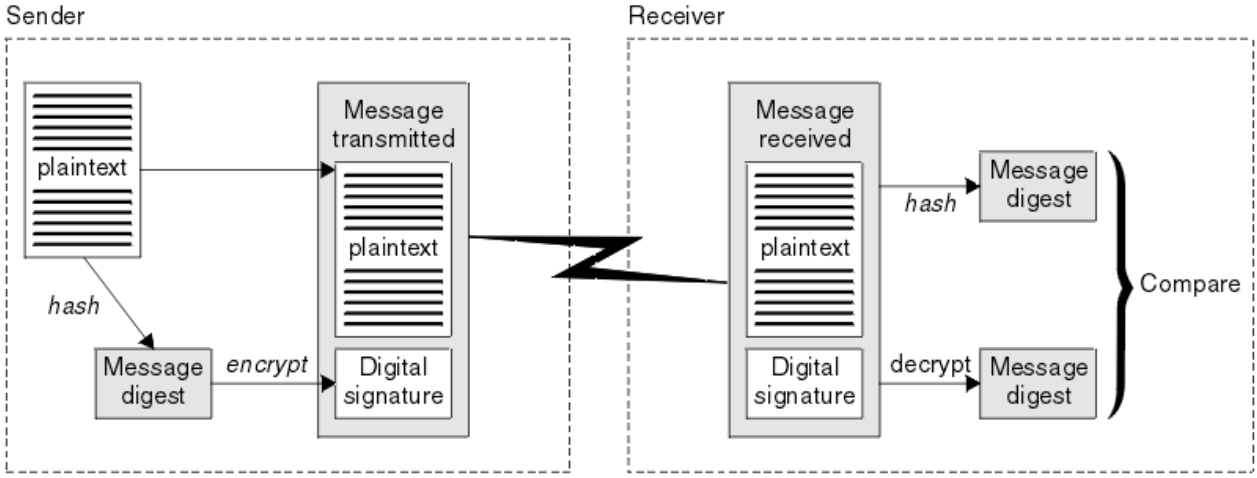



図 6. デジタル署名のプロセス

デジタル署名が検証されると、受信側は次のことを知ることができます。

- メッセージが伝送中に変更されていないこと
- メッセージが、そのメッセージを送信したと表明するエンティティーによって送信されたこと

デジタル署名は、保全性および認証サービスの一部分をなしています。 また、デジタル署名は、発信証明も提供します。 送信側だけが秘密鍵を知っているので、送信側がメッセージの発信元であるという強固な証拠になります。

**注 :** メッセージ自体も暗号化できます。メッセージを暗号化すると、メッセージ内の情報の機密性が保護されます。

## 連邦情報処理標準

米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。 米国国立標準技術研究所 (NIST) は、IT システムおよびセキュリティーに関する重要な機関です。 NIST は、連邦情報処理標準 (FIPS) などの勧告や規格を策定しています。

これらの規格のうちでも重要なのは、強力な暗号アルゴリズムの使用を必須とする FIPS 140-2 です。 FIPS 140-2 では、転送中のパケットが変更されることを防ぐためにハッシュ・アルゴリズムを使用することも規定しています。

IBM WebSphere MQ は、FIPS 140-2 サポートを提供します (そのように構成されている場合)。

時間の経過とともに、アナリストは既存の暗号化およびハッシュ・アルゴリズムに対する攻撃を開発します。 こうした攻撃に対抗するために、新しいアルゴリズムが採用されます。 FIPS 140-2 は、こうした変更を反映するために定期的に更新されます。

### アメリカ国家安全保障局 *(NSA) Suite B* 暗号方式

米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。 アメリカ国家安全保障局 (NSA) は、Suite B 規格の中で、相互運用可能な一連の暗号化アルゴリズムを推奨しています。

Suite B 規格では、特定のセキュアな暗号アルゴリズムのセットのみを使用する運用方式が指定されています。 Suite B 規格では、次の事柄が指定されています。

- 暗号化アルゴリズム (AES)
- 鍵交換アルゴリズム (ECDH (Elliptic Curve Diffie-Hellman))
- デジタル署名アルゴリズム (ECDSA (Elliptic Curve Digital Signature Algorithm))
- ハッシュ・アルゴリズム (SHA-256 または SHA-384)

さらに、IETF RFC 6460 規格によって、Suite B 規格に準拠するために必要な詳細なアプリケーションの構成および動作を定義する Suite B 準拠プロファイルが指定されています。 次の 2 つのプロファイルが定義されています。

1. TLS バージョン 1.2 で使用する Suite B 準拠プロファイル。 Suite B 準拠操作用に構成された場合、上でリストされている暗号アルゴリズムの中の限られたセットのみが使用されます。
2. TLS バージョン 1.0 または TLS バージョン 1.1 で使用する暫定プロファイル。 このプロファイルによって、Suite B 非準拠サーバーとの相互運用が可能になります。 Suite B 暫定操作用に構成された場合、追加の暗号アルゴリズムおよびハッシュ・アルゴリズムが使用できます。

Suite B 規格は、確実なセキュリティーのレベルを提供するために、使用可能な暗号アルゴリズムのセットを制限するという点で、FIPS 140-2 と概念的に似ています。

Windows、 UNIX 、および Linux システムでは、 WebSphere MQ は、Suite B 準拠の TLS 1.2 プロファイルに準拠するように構成できますが、Suite B 移行プロファイルをサポートしません。 詳しくは、30 ページの『IBM WebSphere MQ における NSA Suite B 暗号方式』を参照してください。

**関連情報**

19 ページの『連邦情報処理標準』
米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。 米国国立標準技術研究所 (NIST) は、IT システムおよびセキュリティーに関する重要な機関です。 NIST は、連邦情報処理標準 (FIPS) などの勧告や規格を策定しています。

## IBM WebSphere MQ セキュリティー・メカニズム

このトピック集では、IBM WebSphere MQ においてさまざまなセキュリティー概念を実装する方法について説明します。

IBM WebSphere MQ には、 5 ページの『セキュリティーの概念とメカニズム』で紹介されているすべてのセキュリティー概念を実装するためのメカニズムが用意されています。 これらについての詳細は、以下のセクションで説明されています。

### IBM WebSphere MQ による識別と認証

IBM WebSphere MQ では、メッセージ・コンテキスト情報および相互認証を使用して識別と認証を実装できます。

IBM WebSphere MQ 環境における識別と認証の例を、以下で説明します。

- どのメッセージにも、メッセージ・コンテキスト 情報を入れることができます。 その情報は、メッセージ記述子に格納されます。 キュー・マネージャーは、アプリケーションによってメッセージがキューに書き込まれるときにその情報を生成できます。 あるいは、アプリケーションに関連したユーザー ID に、この情報を提供する許可が与えられている場合は、アプリケーションがこの情報を提供できます。

  メッセージ内のコンテキスト情報により、受信側アプリケーションは、メッセージの発信元についての情報を得ることができます。 例えば、コンテキスト情報には、メッセージを書き込んだアプリケーションの名前、およびそのアプリケーションに関連したユーザー ID が入っています。

- メッセージ・チャネルが開始すると、チャネルの両端にあるメッセージ・チャネル・エージェント (MCA) が、その相手側を相互に認証できます。 この手法のことを相互認証 といいます。 相互認証は、送信側の MCA に対して、メッセージの送信先のパートナーが本物であることを保証します。 受信側 MCA も、同様に送信元のパートナーが本物であることの保証が得られます。

**関連概念**

5 ページの『識別と認証』
識別 とは、システムのユーザー、またはシステムで実行するアプリケーションを一意的に識別する機能のことをいいます。 認証 とは、ユーザーまたはアプリケーションが本人または本物であることを証明する機能のことをいいます。

## IBM WebSphere MQ での許可

許可を使用して、特定の個人またはアプリケーションが IBM WebSphere MQ 環境で実行できることを制限できます。

以下に、 IBM WebSphere MQ 環境での許可の例をいくつか示します。

- 許可された管理者のみが IBM WebSphere MQ リソースを管理するためのコマンドを発行できるようにします。
- アプリケーションに関連付けられているユーザー ID がキュー・マネージャーへの接続を許可されている場合だけ、そのアプリケーションがキュー・マネージャーに接続することを許可する。
- アプリケーションが、その機能で必要なキューだけを開くことを許可する。
- アプリケーションが、その機能に必要なトピックだけをサブスクライブすることを許可する。
- アプリケーションが、その機能に必要な操作だけをキューで実行することを許可する。 例えば、アプリケーションが、特定のキュー上のメッセージをブラウズすることだけが必要であり、メッセージの書き込みや取得が必要ない場合があります。

許可をセットアップする方法について詳しくは、48 ページの『許可の計画』および関連するサブトピックを参照してください。

**関連概念**

6 ページの『許可』
許可は、許可ユーザーとそのアプリケーションだけにアクセスを制限することによって、システム内のクリティカル・リソースを保護します。 これにより、リソースの無許可の使用、または無許可の方法によるリソースの使用を防止します。

## IBM WebSphere MQ での監査

IBM WebSphere MQ は、イベント・メッセージを実行して異常なアクティビティーが行われたことを記録できます。

IBM WebSphere MQ 環境における監査の例を、以下で説明します。

- アプリケーションはオープンする権限がないキューをオープンしようとします。 計測イベント・メッセージが出されます。 イベント・メッセージを検査することによって、この試行が行われたことを知り、どのアクションが必要かを判別することができます。
- アプリケーションはチャネルをオープンしようとしますが、SSL が接続を許可しないため、試行は失敗します。 計測イベント・メッセージが出されます。 イベント・メッセージを検査することによって、この試行が行われたことを知り、どのアクションが必要かを判別することができます。

**関連概念**

6 ページの『監査』
監査とは、予期しないまたは許可されていないアクティビティーが実行されたかどうか、あるいはこうしたアクティビティーを実行しようとする試みがなされたかどうかを検出するために、イベントを記録および検査するプロセスのことです。

## IBM WebSphere MQ での機密性

メッセージを暗号化することによって、IBM WebSphere MQ で機密性を実装することができます。

IBM WebSphere MQ 環境で機密性がどのように確保されるかの例を、以下に説明します。

- 送信側の MCA が伝送キューからメッセージを取得した後、IBM WebSphere MQ が SSL または TLS を使用してメッセージを暗号化してから、メッセージはネットワークを介して受信側の MCA に送信されます。チャネルの相手側で、このメッセージは復号されてから、受信側の MCA がそのメッセージを宛先キューに入れます。
- メッセージがローカル・キューに保管されている間、メッセージの内容を無許可の開示から保護するには、IBM WebSphere MQ によって提供されるアクセス制御メカニズムで十分です。 しかし、より高いレベルのセキュリティーを確保するために、IBM WebSphere MQ Advanced Message Security を使用して、キューに格納されているメッセージを暗号化することができます。

**関連概念**

7 ページの『機密性』
機密性 サービスは、重要な機密情報が無許可で開示されることを防止します。

## IBM WebSphere MQ でのデータ保全性

データ保全性サービスを使用して、メッセージが変更されたかどうかを検出できます。

IBM WebSphere MQ 環境でデータ保全性が確保される方法についての例を、以下に示します。

- SSL または TLS を使用して、メッセージがネットワークを介して伝送されている間に、メッセージの内容が意図的に変更されたかどうかを検出することができます。 SSL および TLS では、メッセージ・ダイジェスト・アルゴリズムは転送中に変更されたメッセージを検出します。 すべての IBM WebSphere MQ CipherSpecs は、メッセージ・ダイジェスト・アルゴリズムを提供します (メッセージ・データ保全性を提供しない TLS_RSA_WITH_NULL_NULL を除く)。
- メッセージがローカル・キューに保管されている間、メッセージの内容を意図的に変更できないようにするには、IBM WebSphere MQ によって提供されるアクセス制御メカニズムで十分です。 しかし、より高いレベルのセキュリティーを確保するために、IBM WebSphere MQ Advanced Message Security を使用して、メッセージがキューに書き込まれた時間から、メッセージがキューから取り出された時間までの間に、メッセージの内容が意図的に変更されたかどうかを検出することができます。

**関連概念**

7 ページの『データ整合性』
データ保全性 サービスは、データに無許可の変更が加えられたかどうかを検出します。

## IBM WebSphere MQ での暗号化

IBM WebSphere MQ は、Secure sockets Layer (SSL) および Transport Security Layer (TLS) プロトコルを使用して暗号化を提供します。

詳細は、22 ページの『IBM WebSphere MQ のセキュリティー・プロトコル』を参照してください

**関連概念**

7 ページの『暗号の概念』
このトピック集では、WebSphere MQ に該当する暗号方式の概念を取り上げます。

## IBM WebSphere MQ のセキュリティー・プロトコル

IBM WebSphere MQ は、Transport Layer Security (TLS) プロトコルと Secure Sockets Layer (SSL) プロトコルの両方をサポートして、メッセージ・チャネルと MQI チャネルにリンク・レベルのセキュリティーを提供します。

メッセージ・チャネルと MQI チャネルは、SSL または TLS プロトコルを使用してリンク・レベル・セキュリティーを提供できます。呼び出し側 MCA が SSL または TLS クライアントであり、応答側 MCA が SSL または TLS サーバーです。 WebSphere MQ は、SSL プロトコルのバージョン 3.0、および Transport Layer Security (TLS) プロトコルのバージョン 1.0 とバージョン 1.2 をサポートしています。 チャネル定義の一環として CipherSpec によって提供される SSL または TLS プロトコルによって使用される暗号アルゴリズムを指定します。

メッセージ・チャネルの両端、および MQI チャネルのサーバー側で、MCA は、接続しているキュー・マネージャーの代理をします。 SSL または TLS ハンドシェーク時に、この MCA は、キュー・マネージャーのデ

ジタル証明書を、チャネルの相手側にあるパートナーの MCA に送信します。 MQI チャネルのクライアント側にある WebSphere MQ コードは、WebSphere MQ クライアント・アプリケーションのユーザーの代理をします。 SSL または TLS ハンドシェーク時に、この WebSphere MQ コードは、ユーザーのデジタル証明書を、MQI チャネルのサーバー側にある MCA に送信します。

キュー・マネージャーおよび WebSphere MQ クライアント・ユーザーは、チャネルのサーバー・サイドで SSLCAUTH (REQUIRED) が指定されていない限り、SSL または TLS クライアントとして機能するときに個人デジタル証明書を関連付ける必要はありません。

デジタル証明書は、鍵リポジトリー に保管されます。 キュー・マネージャー属性 *SSLKeyRepository* は、キュー・ マネージャーの デジタル証明書を保持するキー・リポジトリーの場所を指定します。 WebSphere MQ クライアント・システムでは、MQSSLKEYR 環境変数は、 ユーザーの デジタル証明書を保持する鍵リポジトリーの場所を指定します。 あるいは、 WebSphere MQ クライアント・アプリケーションは、MQCONNX 呼び出しで、SSL および TLS 構成オプション構造 MQSCO の *KeyRepository* フィールドにその位置を指定することができます。 鍵リポジトリー、および鍵リポジトリーの場所の指定方法の詳細については、関連トピックを参照してください。

**関連概念**

14 ページの『暗号セキュリティー・プロトコル: SSL および TLS』
暗号プロトコルは、2 者間の通信のプライバシーとデータ保全性を確保できる セキュア接続を提供します。 Transport Layer Security (TLS) プロトコルは Secure Sockets Layer (SSL) が進化したものです。 IBM WebSphere MQ は、SSL と TLS の両方をサポートしています。

## *IBM WebSphere MQ での SSL および TLS のサポート*

IBM WebSphere MQ は、Secure Sockets Layer (SSL) プロトコルと Transport Layer Security (TLS) プロトコルの両方をサポートします。

SSL プロトコルと TLS プロトコルの詳細については、 関連情報を参照してください。

IBM WebSphere MQ は、SSL バージョン 3.0 および TLS 1.0 と TLS 1.2 を次のようにサポートしています。

**Java および JMS クライアント**
これらのクライアントは、JVM を使用して SSL および TLS サポートを提供します。

**UNIX, Linux, and Windows、および HP Integrity NonStop Server システム**
UNIX, Linux, and Windows、および HP Integrity NonStop Server システムでは、IBM WebSphere MQ と共に SSL および TLS サポートがインストールされます。

IBM WebSphere MQ SSL および TLS サポートの前提条件については、IBM WebSphere MQ のシステム要件を参照してください。

*SSL* または *TLS* 鍵リポジトリー
相互認証される SSL 接続または TLS 接続では、接続の両端で鍵リポジトリーが必要です (プラットフォームによって鍵リポジトリーの呼び名が異なります)。 鍵リポジトリーには、デジタル証明書と秘密鍵が含まれます。

ここでは、デジタル証明書とそれに関連した秘密鍵のストア (格納場所) を指して鍵リポジトリー という一般的な用語を使用しています。 SSL および TLS をサポートするプラットフォームおよび環境で使用される個々のストア名は、次のとおりです。

| | |
|---|---|
| Java および JMS | 鍵ストアおよびトラストストア |
|  Windows、UNIX and Linux システム | 鍵データベース・ファイル |

詳細については、9 ページの『デジタル証明書』および 14 ページの『Secure Sockets Layer (SSL) および Transport Layer Security (TLS) の概念』を参照してください。

相互認証される SSL または TLS 接続では、接続の両端で鍵リポジトリーが必要です。 鍵リポジトリーには、次のものが入っています。

- さまざまな認証局から受け取るいくつかの CA 証明書。キュー・マネージャーまたはクライアントは、その CA 証明書に基づいて、接続のリモート側にあるパートナーから受け取る証明書を検証します。 個々の証明書は、証明書チェーンに入っている場合があります。
- 認証局から受信する 1 つ以上の個人用証明書。 個別の個人用証明書を各キュー・マネージャーまたは WebSphere MQ MQI クライアントと関連付けます。 個人用証明書は、相互認証が必要な場合に SSL または TLS クライアントで不可欠です。 相互認証が必要ない場合、クライアントで個人用証明書は必要ありません。 鍵リポジトリーには、各個人用証明書に対応する秘密鍵が含まれている場合もあります。
- 信頼できる CA 証明書によって署名されるのを待っている認証要求。

鍵リポジトリーの保護についての詳細は、24 ページの『IBM WebSphere MQ の鍵リポジトリーの保護』を参照してください。

鍵リポジトリーの位置は、ご使用のプラットフォームに応じて異なります。


**Windows、 UNIX and Linux システム**

Windows、 UNIX and Linux システムでは、鍵リポジトリーは鍵データベース・ファイルです。 鍵データベース・ファイルの名前には、ファイル拡張子 `.kdb` が必要です。 例えば、 UNIX and Linux では、キュー・マネージャー `QM1` のデフォルトの鍵データベース・ファイルは `/var/mqm/qmgrs/QM1/ssl/key.kdb` です。 IBM WebSphere MQ がデフォルトの場所にインストールされている場合、 Windows 上の同等のパスは `C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\key.kdb` です。

Windows、UNIX and Linux システムでは、各鍵データベース・ファイルにパスワード・スタッシュ・ファイルが関連付けられています。 このファイルは、プログラムが鍵データベースにアクセスできるようにする、暗号化されたパスワードを保持しています。 パスワード・スタッシュ・ファイルは、鍵データベースと同じディレクトリー内にあり、同じファイル語幹を持つ必要があります。また、サフィックス `.sth` で終わる必要があります。例えば、 `/var/mqm/qmgrs/QM1/ssl/key.sth` です。

**注 :** Windows、 UNIX and Linux システムでは、PKCS #11 暗号ハードウェア・カードには、鍵データベース・ファイルに保持されている証明書と鍵を含めることができます。 証明書と鍵が PKCS #11 カード上に保持される場合、WebSphere MQ は、鍵データベース・ファイルとパスワード・スタッシュ・ファイルの両方へのアクセス権が、引き続き必要です。

Windows および UNIX システムでは、キュー・マネージャーまたは WebSphere MQ MQI クライアントに関連付けられた個人用証明書の秘密鍵も鍵データベースに含まれています。

*IBM WebSphere MQ* の鍵リポジトリーの保護

IBM WebSphere MQ 用の鍵リポジトリーは、1 つのファイルです。 所定のユーザーだけが、鍵リポジトリーにアクセスできるようにしてください。 所定のユーザーだけがアクセスできるようにすると、侵入者やその他の無許可のユーザーが、鍵リポジトリー・ファイルを別のシステムにコピーし、そのシステム上に同一のユーザー ID を設定して、所定のユーザーに成りすますことを防ぐことができます。

ファイルに対する許可は、ユーザーの umask および使用されるツールに応じて異なります。 Windows では、IBM WebSphere MQ アカウントは `BypassTraverseChecking` 許可を必要とします。この場合、ファイル・パス内のフォルダーの許可は影響を与えません。

鍵リポジトリー・ファイルのファイル許可を調べて、ファイルとその格納場所のフォルダーが誰でも読み取れる状態にはならないように (さらに、できればグループ読み取りも許可されないように) 設定してください。

使用するすべてのシステムで鍵ストアを読み取り専用にし、保守目的で管理者だけに書き込み操作を許可することをお勧めします。

実際には、どこに存在するか、またパスワード保護されているかどうかに関わらず、すべての鍵ストアを保護する必要があります。鍵リポジトリーを保護してください。

キュー・マネージャーの鍵リポジトリーのリフレッシュ

鍵リポジトリーの内容を変更した場合、新しい内容はキュー・マネージャーにすぐには反映されません。キュー・マネージャーで鍵リポジトリーの新しい内容を使用するには、REFRESH SECURITY TYPE(SSL) コマンドを実行する必要があります。

これは意図的な操作です。実行中の複数のチャネルが互いに異なるバージョンの鍵リポジトリーを使用するという状態を避けることができます。 セキュリティー管理上、どの時点でもただ 1 つのバージョンの鍵リポジトリーだけをキュー・マネージャーでロードできます。

REFRESH SECURITY TYPE(SSL) コマンドの詳細については、REFRESH SECURITY を参照してください。

PCF コマンドまたは WebSphere MQ エクスプローラーを使用して鍵リポジトリーを最新表示することもできます。 詳細については、MQCMD_REFRESH_SECURITY コマンドおよびこの製品資料の WebSphere MQ エクスプローラーのセクションでトピック『*SSL* または *TLS* セキュリティーのリフレッシュ』を参照してください。

**関連概念**

25 ページの『SSL キー・リポジトリーの内容と SSL 設定のクライアント・ビューの最新表示』
リフレッシュしたキー・リポジトリーの内容を使用してクライアント・アプリケーションを更新するには、クライアント・アプリケーションを停止してから再始動する必要があります。

*SSL* キー・リポジトリーの内容と *SSL* 設定のクライアント・ビューの最新表示

リフレッシュしたキー・リポジトリーの内容を使用してクライアント・アプリケーションを更新するには、クライアント・アプリケーションを停止してから再始動する必要があります。

WebSphere MQ クライアント上でセキュリティーをリフレッシュすることはできません。クライアントには REFRESH SECURITY TYPE(SSL) コマンドと同等のコマンドはありません (詳しくは REFRESH SECURITY を参照)。

セキュリティー証明書を変更した場合、リフレッシュしたキー・リポジトリーの内容を使用してクライアント・アプリケーションを更新するためには、必ずそのアプリケーションを停止してから再始動する必要があります。

チャネルの再始動によって構成がリフレッシュされ、アプリケーションに再接続ロジックがある場合、STOP CHL STATUS(INACTIVE) コマンドを発行することによって、クライアントでセキュリティーをリフレッシュできます。

**関連概念**

24 ページの『キュー・マネージャーの鍵リポジトリーのリフレッシュ』
鍵リポジトリーの内容を変更した場合、新しい内容はキュー・マネージャーにすぐには反映されません。キュー・マネージャーで鍵リポジトリーの新しい内容を使用するには、REFRESH SECURITY TYPE(SSL) コマンドを実行する必要があります。

連邦情報処理標準 *(FIPS)*

このトピックでは、米国連邦情報・技術局の連邦情報処理標準 (FIPS) 暗号モジュール評価プログラムについて紹介し、さらに Windows、UNIX and Linux、および z/OS システムの SSL チャネルまたは TLS チャネルで使用できる暗号機能について紹介します。

UNIX、Linux、および Windows システムでの IBM WebSphere MQ SSL または TLS 接続の FIPS 140-2 準拠については、 26 ページの『 UNIX、 Linux、および Windows の連邦情報処理標準 (FIPS)』を参照してください。

暗号ハードウェアが存在する場合は、IBM WebSphere MQ で使用される暗号モジュールが、ハードウェア製造メーカーによって提供される暗号モジュールになるように構成できます。 この場合、これらの暗号モジュールが FIPS 認定済みの場合にのみ、構成は FIPS 準拠です。

連邦情報処理標準は、時間の経過とともに、暗号化アルゴリズムおよびプロトコルに対する新たなアタックを反映して更新されてきました。 例えば、一部の CipherSpec は FIPS による認証を中止する可能性があります。 そのような変更が生じたら、最新の標準を実装するために、IBM WebSphere MQ も更新されます。 その結果、 メンテナンスの適用後に動作が変わることがあります。

**関連概念**

108 ページの『MQI クライアントでの実行時に FIPS 認定の CipherSpec のみを使用するように指定する』
FIPS 準拠のソフトウェアを使用して鍵リポジトリーを作成し、次にチャネルで FIPS 認定の CipherSpec を使用しなければならないことを指定します。

114 ページの『iKeyman、iKeycmd、runmqakm、および runmqckm の使用』

UNIX、Linux 、および Windows システムでは、iKeyman GUI を使用して、またはコマンド行から iKeycmd または runmqakm を使用して、鍵とデジタル証明書を管理します。

**関連タスク**

WebSphere MQ classes for Java で SSL を有効にする

WebSphere MQ classes for JMS での Secure Sockets Layer (SSL) の使用

**関連資料**

JMS オブジェクトの SSL プロパティー

**関連情報**

19 ページの『連邦情報処理標準』

米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。米国国立標準技術研究所 (NIST) は、IT システムおよびセキュリティーに関する重要な機関です。 NIST は、連邦情報処理標準 (FIPS) などの勧告や規格を策定しています。

*UNIX*、 *Linux*、および *Windows* の連邦情報処理標準 *(FIPS)*

Windows、UNIX and Linux システム上の SSL または TLS チャネルで暗号化が必要な場合、WebSphere MQ は IBM Crypto for C (ICC) という名前の暗号化パッケージを使用します。 Windows、 UNIX and Linux プラットフォームでは、ICC ソフトウェアは、米国連邦情報・技術局の連邦情報処理標準 (FIPS) 暗号モジュール評価プログラム (レベル 140-2) に合格しました。

Windows、 UNIX and Linux システムでの WebSphere MQ SSL または TLS 接続の FIPS 140-2 準拠は、以下のとおりです。

- すべての IBM WebSphere MQ メッセージ・チャネルの場合 (CLNTCONN チャネル・タイプを除く)、以下の条件が満たされているなら、接続は FIPS 準拠です。
  - インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーにおいて、インストールされている GSKit ICC バージョンの FIPS 140-2 準拠が認定されている。
  - キュー・マネージャーの SSLFIPS 属性が YES に設定されている。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。
- すべての IBM WebSphere MQ MQI クライアント・アプリケーションの場合、以下の条件が満たされているなら、接続は GSKit を使用し、FIPS 準拠です。
  - インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーにおいて、インストールされている GSKit ICC バージョンの FIPS 140-2 準拠が認定されている。
  - MQI クライアントの関連トピックで説明されているように、FIPS 認定済み暗号方式のみを使用することを指定している。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。
- クライアント・モードを使用する IBM WebSphere MQ classes for Java アプリケーションの場合、以下の条件が満たされると、接続は JRE の SSL および TLS 実装を使用し、FIPS 準拠になります。
  - アプリケーションの実行に使用される Java ランタイム環境は、インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーでは FIPS 準拠です。
  - Java クライアントの関連トピックで説明されているように、FIPS 認定の暗号化のみを使用することを指定しました。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。
- クライアント・モードを使用する IBM WebSphere MQ classes for JMS アプリケーションの場合、以下の条件が満たされているなら、接続は JRE の SSL 実装および TLS 実装を使用し、FIPS 準拠です。
  - アプリケーションの実行に使用される Java ランタイム環境は、インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーでは FIPS 準拠です。

  - JMS クライアントの関連トピックで説明されているように、FIPS 認定済み暗号方式のみを使用することを指定している。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。
- 非管理対象 .NET クライアント・アプリケーションの場合、以下の条件が満たされているなら、接続は GSKit を使用し、FIPS 準拠です。
  - インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーにおいて、インストールされている GSKit ICC バージョンの FIPS 140-2 準拠が認定されている。
  - .NET クライアントの関連トピックで説明されているように、FIPS 認定済み暗号方式のみを使用することを指定している。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。
- 非管理対象 XMS .NET クライアント・アプリケーションの場合、以下の条件が満たされているなら、接続は GSKit を使用し、FIPS 準拠です。
  - インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーにおいて、インストールされている GSKit ICC バージョンの FIPS 140-2 準拠が認定されている。
  - XMS .NET の資料で説明されているように、FIPS 認定済み暗号方式のみを使用することを指定している。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。

サポートされるすべての AIX®、 Linux、 HP-UX、Solaris、 Windows、および z/OS プラットフォームは、各フィックスパックまたはリフレッシュ・パックに含まれている README ファイルに記載されている場合を除き、FIPS 140-2 認定です。

GSkit を使用する SSL および TLS 接続の場合、FIPS 140-2 認定のコンポーネントの名前は *ICC* です。 どのプラットフォームについても、GSKit FIPS 準拠は、このコンポーネントのバージョンによって決まります。 現在インストールされている ICC バージョンを判別するには、 **dspmqver -p 64 -v** コマンドを実行します。

**dspmqver -p 64 -v** の出力のうち ICC に関連する部分の例を以下に抜粋します。

```
  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 8 に含まれる) の NIST 認証ステートメントについては、アドレス Cryptographic Module Validation Program を参照してください。

暗号ハードウェアが存在する場合は、IBM WebSphere MQ で使用される暗号モジュールが、ハードウェア製造メーカーによって提供される暗号モジュールになるように構成できます。 この場合、これらの暗号モジュールが FIPS 認定済みの場合にのみ、構成は FIPS 準拠です。

**注 :** FIPS 140-2 準拠操作用に構成された 32 ビット Solaris x86 SSL および TLS クライアントでは、 Intel システムでの稼働時に障害が起きます。 この障害は、FIPS 140-2 準拠の GSKit-Crypto Solaris x86 32 ビット・ライブラリー・ファイルが Intel チップ・セットをロードしないことが原因で発生します。 影響を受けたシステムでは、クライアント・エラー・ログにエラー AMQ9655 が記録されます。 この問題を解決する

には、FIPS 140-2 準拠を無効にするか、またはクライアント・アプリケーション 64 ビットを再コンパイルします (64 ビット・コードは影響を受けないため)。

## FIPS 140-2 準拠での運用時に適用される Triple-DES 制約事項

WebSphere MQ を FIPS 140-2 に準拠して運用するように構成すると、Triple-DES (3DES) CipherSpecs に関連する追加の制約事項が適用されます。 それらの制約事項を適用することにより、US NIST SP800-67 勧告に準拠します。

1. Triple-DES キーはすべての部分が固有でなければなりません。
2. Triple-DES キーのどの部分も、NIST SP800-67 の定義による Weak、Semi-Weak、または Possibly-Weak にすることはできません。
3. 秘密鍵をリセットするまでは、接続を介して 32 GB までしかデータを転送することができません。 デフォルトでは、WebSphere MQ は秘密セッション鍵をリセットしないので、このリセットを構成する必要があります。 Triple-DES CipherSpec を使用し FIPS 140-2 に準拠した状態で、秘密鍵リセットを有効にしないと、最大バイト・カウントを超過した後に、エラー AMQ9288 を出して接続が閉じてしまいます。 秘密鍵リセットの構成方法については、224 ページの『SSL および TLS 秘密鍵のリセット』を参照してください。

WebSphere MQ は、規則 1 および 2 に既に準拠している Triple-DES セッション鍵を生成します。 ただし、3 番目の制限を満たすには、FIPS 140-2 構成で Triple DES CipherSpec を使用する場合に、秘密鍵のリセットを使用可能にする必要があります。 あるいは、Triple-DES を使用しないという方法もあります。

**関連概念**

108 ページの『MQI クライアントでの実行時に FIPS 認定の CipherSpec のみを使用するように指定する』
FIPS 準拠のソフトウェアを使用して鍵リポジトリーを作成し、次にチャネルで FIPS 認定の CipherSpec を使用しなければならないことを指定します。

114 ページの『iKeyman、iKeycmd、runmqakm、および runmqckm の使用』
UNIX、Linux、および Windows システムでは、iKeyman GUI を使用して、またはコマンド行から iKeycmd または runmqakm を使用して、鍵とデジタル証明書を管理します。

**関連タスク**

WebSphere MQ classes for Java で SSL を有効にする

WebSphere MQ classes for JMS での Secure Sockets Layer (SSL) の使用

**関連資料**

JMS オブジェクトの SSL プロパティー

**関連情報**

19 ページの『連邦情報処理標準』
米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。 米国国立標準技術研究所 (NIST) は、IT システムおよびセキュリティーに関する重要な機関です。 NIST は、連邦情報処理標準 (FIPS) などの勧告や規格を策定しています。

*IBM WebSphere MQ MQI クライアントでの SSL と TLS*
IBM WebSphere MQ では、クライアントで SSL と TLS を使用できるようになっています。 SSL または TLS の使用については、さまざまな調整方法があります。

IBM WebSphere MQ は、 Windows、 UNIX and Linux システム上の IBM WebSphere MQ MQI クライアントに対して SSL および TLS サポートを提供します。 IBM WebSphere MQ classes for Java を使用している場合は、 WebSphere MQ classes for Java の使用 を参照してください。 IBM WebSphere MQ classes for JMS を使用している場合は、WebSphere MQ classes for JMS の使用を参照してください。 このセクションの残りの部分は、Java 環境または JMS 環境には適用されません。

IBM WebSphere MQ MQI クライアントの鍵リポジトリーは、IBM WebSphere MQ クライアント構成ファイルの MQSSLKEYR 値で指定するか、アプリケーションで MQCONNX 呼び出しを実行するときに指定できます。 チャネルが SSL を使用することを指定するには、次の 3 つのオプションがあります。

- チャネル定義テーブルを使用する
- MQCONNX 呼び出しで SSL 構成オプション構造体 MQSCO を使用する

- Active Directory を使用する (Windows システム上)

チャネルが SSL を使用することを指定するのに、MQSERVER 環境変数を使用することはできません。

チャネルの相手側で SSL が指定されていない限り、SSL なしで既存の IBM WebSphere MQ MQI クライアント・アプリケーションをそのまま実行できます。

SSL 鍵リポジトリーの内容、SSL 鍵リポジトリーの位置、認証情報、暗号ハードウェアのパラメーターをクライアント・マシンで変更した場合は、アプリケーションがキュー・マネージャーに接続するために使用しているクライアント接続チャネルでその変更を有効にするために、すべての SSL 接続を終了する必要があります。 すべての接続を終了したら、SSL チャネルを再始動します。 新規 SSL 設定がすべて使用されます。 これらの設定は、キュー・マネージャー・システムで REFRESH SECURITY TYPE(SSL) コマンドによって最新表示される設定に似ています。

IBM WebSphere MQ MQI クライアントが暗号ハードウェアを使用する Windows UNIX and Linux システム上で実行されている場合は、MQSSLCRYP 環境変数を使用してそのハードウェアを構成します。 この変数は、ALTER QMGR MQSC コマンドの SSLCRYP パラメーターと同じ意味を持ちます。 ALTER QMGR MQSC コマンドの SSLCRYP パラメーターについては、ALTER QMGR を参照してください。 SSLCRYP パラメーターの GSK_PCS11 バージョンを使用する場合は、PKCS #11 トークン・ラベル全体を小文字で指定する必要があります。

SSL 秘密鍵リセットおよび FIPS は、IBM WebSphere MQ MQI クライアントでサポートされています。 詳しくは、224 ページの『SSL および TLS 秘密鍵のリセット』 および 26 ページの『 UNIX、 Linux、および Windows の連邦情報処理標準 (FIPS)』 を参照してください。

IBM WebSphere MQ MQI クライアントの SSL サポートについて詳しくは、107 ページの『IBM WebSphere MQ MQI クライアント・セキュリティーのセットアップ』 を参照してください。

**関連タスク**
構成ファイルを使用したクライアントの構成

*MQI* チャネルで *SSL* を使用するよう指定する
MQI チャネルで SSL を使用するには、クライアント接続チャネルの *SSLCipherSpec* 属性の値を、クライアント・プラットフォーム上で IBM WebSphere MQ によってサポートされている CipherSpec の名前にする必要があります。

クライアント接続チャネルは、この属性の値を使用して以下の方法で定義できます。 以下に、高い優先順位のものから示していきます。

1. PreConnect 出口が、使用するチャネル定義構造体を指定する場合。

   チャネル定義は、チャネル定義構造体 MQCD の *SSLCipherSpec* フィールド で CipherSpec の名前を指定できます。 この構造体は、PreConnect 出口が使用する MQNXP パラメーター構造体の **ppMQCDArrayPtr** フィールドで返されます。

2. WebSphere MQ MQI クライアント・アプリケーションが MQCONNX 呼び出しを発行する場合。

   アプリケーションは、チャネル定義構造体 MQCD の *SSLCipherSpec* フィールド で CipherSpec の名前を指定できます。この構造体は、MQCONNX 呼び出しのパラメーターである 接続オプション構造体 MQCNO によって参照されます。

3. クライアント・チャネル定義テーブル (CCDT) を使用する。

   クライアント・チャネル定義テーブル内の 1 つ以上のエントリーで、CipherSpec の名前を指定できます。例えば、DEFINE CHANNEL MQSC コマンドを使用してエントリーを作成する場合は、コマンドで SSLCIPH パラメーターを使用して CipherSpec の名前を指定することができます。

4. Windows で Active Directory を使用する。

   Windows システムで **setmqscp** 制御コマンドを使用して Active Directory でクライアント接続チャネル定義を公開することができます。 これらの定義の 1 つ以上で、CipherSpec の名前を指定できます。

例えば、MQCONNX 呼び出しの MQCD 構造体でクライアント・アプリケーションがクライアント接続チャネル定義を提供している場合、この定義は、WebSphere MQ クライアントがアクセスするクライアント・チャネル定義テーブル内のどのエントリーよりも優先されます。

MQSERVER 環境変数を使用して、SSL を使用する MQI チャネルのクライアント側でチャネル定義を提供することはできません。

クライアント証明書が流れたかどうかを確認するには、チャネルのサーバー側にあるチャネル・ステータスを表示して、ピア名パラメーター値が存在することを確認します。

**関連概念**

224 ページの『IBM WebSphere MQ MQI クライアント用の CipherSpec の指定』
IBM WebSphere MQ MQI クライアントの CipherSpec を指定するには、3 つのオプションがあります。

*IBM WebSphere MQ の CipherSpecs および CipherSuites*
IBM WebSphere MQ は、SSL と TLS の両方の CipherSpec、RSA、および Diffie-Hellman アルゴリズムをサポートしています。

WebSphere MQ は、SSL V3 および TLS V1.0 と V1.2 の CipherSpec をサポートしています。

WebSphere MQ は、RSA と Diffie-Hellman の鍵交換および認証のアルゴリズムをサポートしています。 SSL ハンドシェーク中に使用される鍵のサイズは、使用するデジタル証明書によって決まりますが、一部の CipherSpec には、ハンドシェーク時の鍵サイズの仕様が含まれています。 ハンドシェークの鍵サイズが大きいほど、認証は強力になります。 鍵のサイズが小さいほど、ハンドシェークは高速になります。

**関連概念**

18 ページの『CipherSpec および CipherSuite』
暗号セキュリティー・プロトコルは、セキュア接続で使用されるアルゴリズムと一致しなければなりません。 CipherSpec および CipherSuite は、アルゴリズムの特定の組み合わせを定義します。

*IBM WebSphere MQ* における *NSA Suite B* 暗号方式
このトピックでは、Suite B 準拠 TLS 1.2 プロファイルに準拠するように Windows、 Linux、および UNIX システムで IBM WebSphere MQ を構成する方法について説明します。

NSA Cryptography Suite B 標準は、時間の経過とともに、暗号化アルゴリズムおよびプロトコルに対する新たなアタックを反映して更新されてきました。 例えば、一部の CipherSpec は Suite B による認証を中止する可能性があります。 そのような変更が生じたら、最新の標準を実装するために、IBM WebSphere MQ も更新されます。 その結果、 メンテナンスの適用後に動作が変わることがあります。 IBM WebSphere MQ Version 7.5 README ファイルには、製品の保守レベルごとに強制された Suite B のバージョンがリストされています。 Suite B 準拠を強制するように IBM WebSphere MQ を構成した場合は、保守適用の計画を立てるときに、必ず README ファイルをお読みください (IBM MQ、WebSphere MQ、および MQSeries 製品の README を参照してください)。

Windows、 UNIX、および Linux システムでは、 IBM WebSphere MQ は、表 1 に示されているセキュリティー・レベルで Suite B 準拠 TLS 1.2 プロファイルに準拠するように構成できます。

表 *1. Suite B* のセキュリティー・レベルと許可される *CipherSpec* およびデジタル署名アルゴリズム

| **安全レベル** | **許可される CipherSpec** | **許可されるデジタル署名アルゴリズム** |
|---|---|---|
| 128 ビット | ECDHE_ECDSA_AES_128_GCM_SHA256<br>ECDHE_ECDSA_AES_256_GCM_SHA384 | ECDSA と SHA-256<br>ECDSA と SHA-384 |
| 192 ビット | ECDHE_ECDSA_AES_256_GCM_SHA384 | ECDSA と SHA-384 |
| 両方[1] | ECDHE_ECDSA_AES_128_GCM_SHA256<br>ECDHE_ECDSA_AES_256_GCM_SHA384 | ECDSA と SHA-256<br>ECDSA と SHA-384 |

1. 128 ビットと 192 ビット両方のセキュリティー・レベルを同時に構成することができます。 Suite B の構成により、最小許容レベルの暗号アルゴリズムが決まるため、両方のセキュリティー・レベルを構成することは、128 ビット・セキュリティー・レベルのみを構成することに相当します。 192 ビット・セキュリティー・レベルの暗号アルゴリズムは、128 ビット・セキュリティー・レベルに最小限必要な暗号アルゴリズムよりも強力です。そのため、192 ビット・セキュリティー・レベルが有効にされていな

いとしても、192 ビット・セキュリティー・レベルの暗号アルゴリズムが 128 ビット・セキュリティー・レベルに許可されます。

**注 :** セキュリティー・レベル で使用する命名規則は、必ずしも楕円曲線のサイズや AES 暗号アルゴリズムの鍵サイズを表してはいません。

## CipherSpec の Suite B への準拠

IBM WebSphere MQ のデフォルトの動作は Suite B 標準に準拠しませんが、 IBM WebSphere MQ は、Windows、 UNIX 、および Linux システムのいずれかまたは両方のセキュリティー・レベルに準拠するように構成できます。 Suite B を使用するように IBM WebSphere MQ が正しく構成された後は、CipherSpec を使用して、Suite B に適合しない方法でアウトバウンド・チャネルを開始しようとすると、エラー AMQ9282 が発生します。 また、このアクティビティーの結果、MQI クライアントが理由コード MQRC_CIPHER_SPEC_NOT_SUITE_B を返します。 同様に、Suite B 構成に準拠していない CipherSpec を使用してインバウンド・チャネルを開始しようとすると、結果としてエラー AMQ9616 が出されます。

WebSphere MQ CipherSpec の詳細については、217 ページの『CipherSpec の指定』を参照してください。

## Suite B とデジタル証明書

Suite B は、デジタル証明書への署名に使用するデジタル署名アルゴリズムを制限します。 Suite B はまた、証明書に格納することができる公開鍵のタイプを制限します。 したがって、リモート・パートナーで構成されている Suite B セキュリティー・レベルによって許可されるデジタル署名アルゴリズムおよび公開鍵タイプを使用する証明書を使用するように、WebSphere MQ を構成する必要があります。 このセキュリティー・レベル要件に準拠しないデジタル証明書は拒否され、その接続はエラー AMQ9633 または AMQ9285 で失敗します。

128 ビット Suite B セキュリティー・レベルでは、証明書のサブジェクトの公開鍵は NIST P-256 楕円曲線または NIST P-384 楕円曲線のいずれかを使用し、NIST P-256 楕円曲線または NIST P-384 楕円曲線のいずれかによって署名される必要があります。 192 ビット Suite B セキュリティー・レベルでは、証明書のサブジェクトの公開鍵は NIST P-384 楕円曲線を使用し、NIST P-384 楕円曲線によって署名される必要があります。

Suite B 準拠操作に適した証明書を取得する場合は、**runmqakm** コマンドに **-sig_alg** パラメーターを指定して使用することにより、適切なデジタル署名アルゴリズムを要求します。 EC_ecdsa_with_SHA256 および EC_ecdsa_with_SHA384 の **-sig_alg** パラメーターの値は、許可されている Suite B デジタル署名アルゴリズムによって署名される楕円曲線鍵に対応します。

**runmqakm** コマンドの詳細については、runmqckm および runmqakm のオプションを参照してください。

**注 : iKeycmd** および **iKeyman** ツールは、Suite B 準拠操作に対するデジタル証明書の作成をサポートしません。

## デジタル証明書の作成および要求

Suite B のテスト用に自己署名デジタル証明書を作成する場合は、122 ページの『UNIX, Linux, and Windows システムでの自己署名個人証明書の作成』を参照してください。

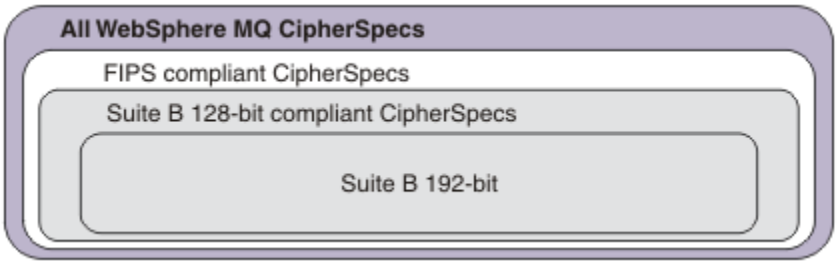
Suite B の実動用に CA 署名デジタル証明書を要求する場合は、124 ページの『UNIX, Linux, and Windows システムでの個人証明書の要求』を参照してください。

**注 :** 使用される認証局は、IETF RFC 6460 に記載されている要件を満たすデジタル証明書を生成しなければなりません。

## FIPS 140-2 と Suite B

Suite B 規格は、確実なセキュリティーのレベルを提供するために、使用可能な暗号アルゴリズムのセットを制限するという点で、FIPS 140-2 と概念的に似ています。 現在サポートされている Suite B CipherSpec は、IBM WebSphere MQ が FIPS 140-2 準拠操作用に構成されている場合に使用可能です。 そのため、WebSphere MQ を FIPS と Suite B の両方に同時に準拠するよう構成することが可能です。その場合、両方の制限のセットが適用されます。

次の図は、これらのサブセット間の関係を示しています。




## WebSphere MQ を Suite B 準拠操作用に構成する

Windows、UNIX、および Linux で Suite B 準拠操作用に IBM WebSphere MQ を構成する方法については、32 ページの『Suite B 用 IBM WebSphere MQ の構成』を参照してください。

IBM WebSphere MQ は、IBM i および z/OS プラットフォーム上では、Suite B 準拠操作をサポートしていません。 WebSphere MQ の Java および JMS クライアントも Suite B 準拠操作をサポートしていません。

**関連概念**
108 ページの『MQI クライアントでの実行時に FIPS 認定の CipherSpec のみを使用するように指定する』
FIPS 準拠のソフトウェアを使用して鍵リポジトリーを作成し、次にチャネルで FIPS 認定の CipherSpec を使用しなければならないことを指定します。

*Suite B* 用 *IBM WebSphere MQ* の構成
IBM WebSphere MQ は、UNIX, Linux, and Windows システム上で、NSA Suite B 規格に準拠して動作するよう構成することができます。

Suite B は、確実なセキュリティーのレベルを提供するために、使用可能な暗号アルゴリズムのセットを制限します。IBM WebSphere MQ スイート B に準拠して動作するように構成することで、セキュリティー・レベルを向上させることができます。 Suite B の詳細については、20 ページの『アメリカ国家安全保障局 (NSA) Suite B 暗号方式』を参照してください。 Suite B の構成と SSL および TLS チャネルでのその影響についての詳細は、30 ページの『IBM WebSphere MQ における NSA Suite B 暗号方式』を参照してください。

## キュー・マネージャー

キュー・マネージャーについては、コマンド **ALTER QMGR** にパラメーター **SUITEB** を指定して使用し、必要なセキュリティー・レベルに適した値を設定してください。 詳しくは、ALTER QMGR を参照してください。

PCF コマンド **MQCMD_CHANGE_Q_MGR** にパラメーター **MQIA_SUITE_B_STRENGTH** を指定することによっても、Suite B 準拠操作用にキュー・マネージャーを構成できます。

## MQI クライアント

デフォルトでは、MQI クライアントは Suite B 準拠を適用しません。 以下のいずれかのオプションを実行することにより、MQI クライアントの Suite B 準拠を有効にできます。

1. MQCONNX 呼び出しで、MQSCO 構造体の **EncryptionPolicySuiteB** フィールドを、以下の 1 つ以上の値に設定する。
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

   MQ_SUITE_B_NONE にその他の値を指定して使用するのは無効です。
2. MQSUITEB 環境変数を、以下の 1 つ以上の値に設定する。
   - NONE

- 128_BIT
- 192_BIT

コンマ区切りリストを使用して、複数の値を指定することができます。 値 NONE をその他の値と一緒に使用するのは無効です。

3. MQI クライアント構成ファイルの SSL スタンザ内で、 **EncryptionPolicySuiteB** 属性を以下の 1 つ以上の値に設定する。
   - NONE
   - 128_BIT
   - 192_BIT

   コンマ区切りリストを使用して、複数の値を指定することができます。 NONE をその他の値と一緒に使用するのは無効です。

**注 :** MQI クライアントの設定は、優先度の順にリストされています。 MQCONNX 呼び出しの MSCO 構造体は、MQSUITEB 環境変数の設定をオーバーライドし、それによって SSL スタンザ内の属性がオーバーライドされます。

MQSCO 構造体の詳細については、MQSCO - SSL 構成オプションを参照してください。

クライアント構成ファイルでの Suite B の使用について詳しくは、 クライアント構成ファイルの SSL スタンザ を参照してください。

MQSUITEB 環境変数の使い方の詳細については、環境変数を参照してください。

## .NET

.NET の非管理対象クライアントの場合、プロパティー **MQC.ENCRYPTION_POLICY_SUITE_B** は、必要な Suite B セキュリティーのタイプを示します。

IBM WebSphere MQ classes for .NET での Suite B の使い方の詳細については、『MQEnvironment .NET クラス』を参照してください。

### *IBM WebSphere MQ* での証明書妥当性検査ポリシー

証明書妥当性検査ポリシーは、証明書チェーン妥当性検査においてセキュリティーに関する業界の標準規格にどの程度厳密に準拠するかを決定します。

以下のように、証明書妥当性検査ポリシーはプラットフォームおよび環境に応じて異なります。

- すべてのプラットフォームについて、Java および JMS アプリケーションの場合、証明書妥当性検査ポリシーは、Java ランタイム環境の JSSE コンポーネントに応じて異なります。 証明書妥当性検査ポリシーについて詳しくは、JRE のドキュメンテーションを参照してください。
- UNIX, Linux, and Windows システムの場合、証明書妥当性検査ポリシーは、GSKit によって提供され、構成可能です。 以下の 2 つの異なる証明書妥当性検査ポリシーがサポートされています。
  - レガシー証明書妥当性検査ポリシー。 これは、現行の IETF 証明書妥当性検査標準規格に準拠していない古いデジタル証明書との後方互換性および相互運用性を最大限確保するために使用されます。 このポリシーは、基本ポリシーと呼ばれます。
  - RFC 5280 規格に厳格に準拠した証明書妥当性検査ポリシー。 このポリシーは、標準ポリシーと呼ばれます。

UNIX, Linux, and Windows システムでの証明書妥当性検査ポリシーの構成方法については、33 ページの『IBM WebSphere MQ での証明書妥当性検査ポリシーの構成』を参照してください。 基本証明書検証ポリシーと標準証明書検証ポリシーの違いについて詳しくは、 UNIX、 Linux および Windows システムでの証明書検証とトラスト・ポリシーの設計 を参照してください。

### *IBM WebSphere MQ* での証明書妥当性検査ポリシーの構成

リモート・パートナー・システムから受け取ったデジタル証明書を妥当性検査するために、どの SSL/TLS 証明書妥当性検査ポリシーを使用するかを 4 通りの方法で指定できます。

キュー・マネージャー上では、証明書妥当性検査ポリシーを 4 通りの方法で設定できます。

- キュー・マネージャー属性 *CERTVPOL* を使用する。 この属性の設定について詳しくは、 ALTER QMGR を参照してください。

クライアント上では、いくつかの方法で証明書妥当性検査ポリシーを設定することができます。 複数の方法を併用してポリシーを設定する場合、クライアントは次の優先順位で設定を使用します。

1. クライアント MQSCO 構造の *CertificateValPolicy* フィールドを使用する。 このフィールドの使用について詳しくは、MQSCO - SSL 構成オプションを参照してください。
2. クライアント環境変数 *MQCERTVPOL* を使用する。 この変数の使用について詳しくは、 MQCERTVPOL を参照してください。
3. クライアント SSL スタンザ調整パラメーター設定 *CertificateValPolicy* を使用する。 この設定の使用について詳しくは、 クライアント構成ファイルの SSL スタンザ を参照してください。

証明書妥当性検査ポリシーの詳細については、33 ページの『IBM WebSphere MQ での証明書妥当性検査ポリシー』を参照してください。

*IBM WebSphere MQ* におけるデジタル証明書と *CipherSpec* の互換性
このトピックでは、IBM WebSphere MQ における CipherSpec とデジタル証明書の関係を概説することにより、個々のセキュリティー・ポリシーに適した CipherSpec とデジタル証明書を選択する方法を説明します。

IBM WebSphere MQ の以前のリリースでサポートされていたすべての SSL および TLS CipherSpec では、デジタル署名および鍵の共有に RSA アルゴリズムを使用していました。 サポートされていたデジタル証明書のすべてのタイプは、サポートされていた CipherSpec のすべてのタイプと互換性があったため、デジタル証明書を変更することなく任意のチャネルの CipherSpec を変更することが可能でした。

IBM WebSphere MQ v7.5 では、サポートされている CipherSpec のサブセットのみが、サポートされているすべてのタイプのデジタル証明書で使用可能です。 そのため、使用するデジタル証明書に適した CipherSpec を選択する必要があります。 同様に、組織のセキュリティー・ポリシーで、特定の CipherSpec の使用が求められている場合は、その CipherSpec に適したデジタル証明書を取得しなければなりません。

## MD5 デジタル署名アルゴリズムと TLS 1.2

TLS 1.2 プロトコルを使用する場合、MD5 アルゴリズムを使用して署名されたデジタル証明書は拒否されます。 これは、現在多くの暗号のアナリストが MD5 アルゴリズムを脆弱と見なしていて、このアルゴリズムの使用が一般に推奨されていないためです。 TLS 1.2 プロトコルに基づく新しい CipherSpec を使用する場合は、デジタル証明書のデジタル署名に MD5 アルゴリズムが使用されていないことを確認してください。 SSL 3.0 および TLS 1.0 プロトコルを使用する古い CipherSpec にはこの制限が適用されないため、MD5 デジタル署名を使用した証明書を引き続き使用することができます。

特定の証明書のデジタル署名アルゴリズムを表示するには、 **runmqakm** コマンドを使用できます。

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

ここで、cert_label は表示したいデジタル署名アルゴリズムの証明書ラベルです。

**注 : iKeycmd (runmqckm)** ツールおよび **iKeyman (strmqikm)** GUI を使用して一連のデジタル署名アルゴリズムを表示することもできますが、**runmqakm** ツールの方が広い範囲に対応しています。

**runmqakm** コマンドを実行すると、指定された署名アルゴリズムの使用を示す出力が以下のように表示されます。

```
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` 行は、`MD5WithRSASignature` アルゴリズムが使用されていることを示しています。 このアルゴリズムは MD5 に基づいているため、このデジタル証明書を TLS 1.2 CipherSpec と一緒に使用することはできません。

## 楕円曲線と RSA CipherSpec の相互運用性

すべての CipherSpec がすべてのデジタル証明書と共に使用できるわけではありません。 CipherSpec には 3 つのタイプがあり、それらは CipherSpec 名の接頭部によって示されます。 CipherSpec のタイプごとに、使用できるデジタル証明書のタイプに関する制限があります。 これらの制限は、WebSphere MQ のすべての SSL および TLS 接続に適用されますが、楕円曲線暗号のユーザーにとっては、特に重要です。

CipherSpec とデジタル証明書の関係の要約を、以下の表に示します。

表 *2. CipherSpec* とデジタル証明書の関係

| タイプ | CipherSpec 名の接頭部 | 説明 | 必要な公開鍵のタイプ | デジタル署名の暗号化アルゴリズム | 秘密鍵の設定方法 |
|---|---|---|---|---|---|
| 1 | ECDHE_ECDSA_ | 楕円曲線公開鍵、楕円曲線秘密鍵、および楕円曲線デジタル署名アルゴリズムを使用する CipherSpec。 | 楕円曲線 | ECDSA | ECDHE |
| 2 | ECDHE_RSA_ | RSA 公開鍵、楕円曲線秘密鍵、および楕円曲線デジタル署名アルゴリズムを使用する CipherSpec。 | RSA | RSA | ECDHE |
| 3 | (その他すべて) | RSA 公開鍵および RSA デジタル署名アルゴリズムを使用する CipherSpec。 | RSA | RSA | RSA |

**注 :** タイプ 1 および 2 の CipherSpec は、UNIX, Linux, and Windows プラットフォーム上の WebSphere MQ キュー・マネージャーおよび MQI クライアントでのみサポートされます。

必要な公開鍵タイプの列は、CipherSpec のそれぞれのタイプを使用する場合に個人証明書が持っていなければならない公開鍵のタイプを示します。 個人証明書は、リモート・パートナーに対してキュー・マネージャーまたはクライアントを識別するエンド・エンティティー証明書です。

デジタル署名暗号化アルゴリズムは、ピアを検証するために使用する暗号化アルゴリズムです。 この暗号化アルゴリズムは、デジタル署名を計算するために、MD5、SHA-1、SHA-256 などのハッシュ・アルゴリズムと共に使用されます。 使用可能なデジタル署名アルゴリズムには、「RSA with MD5」や「ECDSA with SHA-256」など、さまざまなものがあります。 表で、ECDSA は ECDSA を使用するデジタル署名アルゴリズムのセットを指し、RSA は RSA を使用するデジタル署名アルゴリズムのセットを指します。 指定の暗号化アルゴリズムに基づいている限り、セット内の任意のサポート対象デジタル署名アルゴリズムを使用できます。

タイプ 1 の CipherSpec では、個人証明書が楕円曲線公開鍵を持つことが必要です。 これらの CipherSpec を使用する場合、接続の秘密鍵の設定に楕円曲線 Diffie Hellman 短期鍵共有が使用されます。

タイプ 2 の CipherSpec では、個人証明書が RSA 公開鍵を持つことが必要です。 これらの CipherSpec を使用する場合、接続の秘密鍵の設定に楕円曲線 Diffie Hellman 短期鍵共有が使用されます。

タイプ 3 の CipherSpec では、個人証明書が RSA 公開鍵を持つことが必要です。 これらの CipherSpec を使用する場合、接続の秘密鍵の設定に RSA 鍵交換が使用されます。

この制限リストは完全なものではありません。構成によっては、相互運用に影響を与える追加の制限が生じる場合があります。 例えば、FIPS 140-2 または NSA Suite B 規格に準拠するように WebSphere MQ が構成されている場合、選択可能な構成の範囲は更に制限されます。 詳しくは、以下のセクションを参照してください。

WebSphere MQ キュー・マネージャーは、それ自身を識別するための個人証明書を 1 つしか使用できません。 これは、キュー・マネージャー上のすべてのチャネルが同じデジタル証明書を使用することを意味し、そのため、各キュー・マネージャーは同時に 1 つのタイプの CipherSpec しか使用できないことになります。 同様に、WebSphere MQ クライアント・アプリケーションは、それ自身を識別するための個人証明書を 1 つしか使用できません。 これは、1 つのアプリケーション・プロセス内のすべての SSL および TLS 接続が同じデジタル証明書を使用することを意味し、そのため、各クライアント・アプリケーション・プロセスは同時に 1 つのタイプの CipherSpec しか使用できないことになります。

これら 3 つのタイプの CipherSpec を直接相互運用することはできません。これは現在の SSL 規格および TLS 規格の制限です。 例えば、QM1 という名前のキュー・マネージャー上で、TO.QM1 という名前の受信側チャネルに ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec を使用することにしたと仮定します。 ECDHE_ECDSA_AES_128_CBC_SHA256 はタイプ 1 の CipherSpec なので、QM1 は楕円曲線鍵と ECDSA ベースのデジタル署名を使用する個人証明書を持つ必要があります。 したがって、QM1 と直接通信するすべてのクライアントおよび他のキュー・マネージャーは、タイプ 1 CipherSpec 要件を満たすデジタル証明書を持たなければなりません。 キュー・マネージャー QM1 に接続する他のチャネルは、その他の CipherSpec (例えば、ECDHE_ECDSA_3DES_EDE_CBC_SHA256) も使用できますが、QM1 と通信するためには、タイプ 1 CipherSpec を使用しなくてはなりません。

使用する WebSphere MQ ネットワークを計画する際は、どのチャネルで SSL または TLS が必要かを慎重に検討し、相互運用が必要なすべてのクライアントおよびキュー・マネージャーで、同じタイプの CipherSpec および適切なデジタル証明書を使用することを確認してください。 IETF 規格の RFC 4492、RFC 5246、および RFC 6460 では、TLS 1.2 における楕円曲線 CipherSpec の詳細な使用法について説明しています。

デジタル証明書のデジタル署名アルゴリズムおよび公開鍵タイプを表示するには、**runmqakm** コマンドを以下のように使用します。

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

ここで、`cert_label` はデジタル署名アルゴリズムを表示したい証明書のラベルです。

**runmqakm** コマンドを実行すると、公開鍵のタイプを示す出力が以下のように表示されます。

```
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
```

Public Key Type 行は、この場合、証明書が楕円曲線公開鍵を持つことを示しています。 Signature Algorithm 行は、この場合、EC_ecdsa_with_SHA384 アルゴリズムが使用されていることを示しています。これは、ECDSA アルゴリズムに基づいています。 したがって、この証明書の使用に適した CipherSpec タイプは、タイプ 1 のみです。

同じパラメーターを指定して **iKeycmd (runmqckm)** ツールを使用することもできます。 **iKeyman (strmqikm)** GUI を使用して、デジタル署名アルゴリズムを表示することもできます。それには、鍵リポジトリーをオープンして、証明書のラベルをダブルクリックします。 ただし、**runmqakm** ツールの方が広範囲のアルゴリズムをサポートしているため、これを使用してデジタル証明書を表示する方法をお勧めします。

## 楕円曲線 CipherSpec と NSA Suite B

WebSphere MQ を Suite B 準拠 TLS 1.2 プロファイルに適合するように構成すると、30 ページの『IBM WebSphere MQ における NSA Suite B 暗号方式』で説明されているように、許可される CipherSpec およびデジタル署名アルゴリズムが制限されます。 さらに、構成されているセキュリティー・レベルに応じて、許容される楕円曲線鍵の範囲が小さくなります。

128 ビット Suite B セキュリティー・レベルでは、証明書のサブジェクトの公開鍵は NIST P-256 楕円曲線または NIST P-384 楕円曲線のいずれかを使用し、NIST P-256 楕円曲線または NIST P-384 楕円曲線のいずれかによって署名される必要があります。 **runmqakm** コマンドを使用して、このセキュリティー・レベルのデジタル証明書を要求することができます。この場合、-sig_alg パラメーターに EC_ecdsa_with_SHA256 または EC_ecdsa_with_SHA384 を使用します。

192 ビット Suite B セキュリティー・レベルでは、証明書のサブジェクトの公開鍵は NIST P-384 楕円曲線を使用し、NIST P-384 楕円曲線によって署名される必要があります。 **runmqakm** コマンドを使用して、このセキュリティー・レベルのデジタル証明書を要求することができます。この場合、-sig_alg パラメーターに EC_ecdsa_with_SHA384 を使用します。

サポートされる NIST 楕円曲線は次のとおりです。

表 *3.* サポートされる *NIST* 楕円曲線

| **NIST FIPS 186-3 曲線の名前** | **RFC 4492 曲線の名前** | **楕円曲線鍵のサイズ (ビット)** |
|---|---|---|
| P-256 | secp256r1 | 256 |

| 表 3. サポートされる *NIST* 楕円曲線 (続き) | | |
|---|---|---|
| **NIST FIPS 186-3 曲線の名前** | **RFC 4492 曲線の名前** | **楕円曲線鍵のサイズ (ビット)** |
| P-384 | secp384r1 | 384 |
| P-521 | secp521r1 | 521 |

**注 :** NIST P-521 楕円曲線は、Suite B 準拠操作には使用できません。

**関連概念**

217 ページの『CipherSpec の指定』
**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのいずれかで **SSLCIPH** パラメーターを使用して、 CipherSpec を指定します。

108 ページの『MQI クライアントでの実行時に FIPS 認定の CipherSpec のみを使用するように指定する』
FIPS 準拠のソフトウェアを使用して鍵リポジトリーを作成し、次にチャネルで FIPS 認定の CipherSpec を使用しなければならないことを指定します。

30 ページの『IBM WebSphere MQ における NSA Suite B 暗号方式』
このトピックでは、Suite B 準拠 TLS 1.2 プロファイルに準拠するように Windows、 Linux、および UNIX システムで IBM WebSphere MQ を構成する方法について説明します。

20 ページの『アメリカ国家安全保障局 (NSA) Suite B 暗号方式』
米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。アメリカ国家安全保障局 (NSA) は、Suite B 規格の中で、相互運用可能な一連の暗号化アルゴリズムを推奨しています。

## *IBM WebSphere MQ* でサポートされる *CipherSpec* 値

デフォルトの CipherSpec のセットでは、以下の値のみ使用できます。

**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

## 推奨されない CipherSpec の有効化

デフォルトでは、推奨されない CipherSpec をチャネル定義上に指定できません。 推奨されない CipherSpec を指定しようとすると、メッセージ AMQ9788 がキュー・マネージャーのエラー・ログに表示されます。

qm.ini ファイルを編集して、推奨されない CipherSpecs を再度有効にすることができます。 qm.ini ファイルの SSL スタンザ内に、以下の行を追加します。

```
SSL:
AllowWeakCipherSpec=Yes
```

環境変数 *AMQ_SSL_WEAK_CIPHER_ENABLE* を任意の値に設定して、非推奨の 1 つ以上の CipherSpecs をサーバーで実行時に再度使用可能にすることができます。 この環境変数は、qm.ini ファイルで指定されている値にかかわらず、CipherSpecs を有効にします。

## チャネル認証レコード

チャネル認証レコードを使用すれば、接続システムに与えるアクセス権限をチャネル・レベルで細かく制御できるようになります。

キュー・マネージャーに接続してくるクライアントのなかには、ブランクのユーザー ID や、望ましくないアクションの実行権限を備えたハイレベルなユーザー ID で接続しようとするものもあります。 チャネル認証レコードを使用すれば、こうしたクライアントからのアクセスをブロックできるようになります。 また、クライアントが表明するユーザー ID には、クライアントのプラットフォームでは 有効であっても、サーバーのプラットフォームでは不明または無効な形式の ID もあります。 チャネル認証レコードを使用すれば、表明されたユーザー ID を有効なユーザー ID にマッピングできるようになります。

何らかの方法でキュー・マネージャーに接続して悪事を働くクライアント・アプリケーションも存在する場合もあります。 こうしたアプリケーションの引き起こす問題から サーバーを保護するには、ファイアウォールのルールのアップデートまたはクライアント・アプリ ケーションの訂正が完了するまで、IP アドレスを使用して問題のクライアント・アプリケーション からの接続を一時的にブロックしておく必要があります。 チャネル認証レコードを使用すれば、こうしたクライアント・アプリケーションの IP アドレスからの接続をブロックできるようになります。

IBM WebSphere MQ Explorer などの管理ツールと、そのツールへの接続チャネルが すでにセットアップされている場合、そのチャネルの使用権が特定のクライアント・コンピューターにのみ与えら れるようにしておきたいと思うこともあります。 チャネル認証レコードを使用して、特定の IP アドレスだけからチャネルを使用できるようにすることができます。

クライアントとして実行されるサンプル・アプリケーションを開始しようとしている場合は、サンプル・プログラムの作成と実行で、チャネル認証レコードを使ってキュー・マネージャーを安全にセットアップする例を参照してください。

チャネル認証レコードでインバウンド・チャネルを制御するには、MQSC コマンド **ALTER QMGR CHLAUTH(ENABLED)** を使用します。

新規インバウンド接続への応答で作成されたチャネル MCA には、**CHLAUTH** ルールが適用されています。 ローカルで始動されているチャネルへの応答で作成されたチャネル MCA の場合は、**CHLAUTH** ルールは適用されません。

表 *4.* 各種チャネル・ペアにおける *CHLAUTH* ルールの適用対象

| **チャネル・タイプ** | **CHLAUTH ルールが適用される MCA** |
|---|---|
| SDR-RCVR | RCVR |
| RQSTR-SVR (SVR で始動) | RQSTR |
| RQSTR-SVR (RQSTR で始動) | SVR |
| RQSTR-SDR (SDR で始動) | RQSTR |
| RQSTR-SDR (RQSTR で始動) | 初期接続の場合は SDR。 コールバック接続の場合は RQSTR。 |

チャネル認証レコードの作成により、以下の機能の実行が可能になります。

- 特定の IP アドレスからの接続をブロックする。
- 特定のユーザー ID からの接続をブロックする。

- 特定の IP アドレスから接続する任意のチャネルで使用する MCAUSER 値を設定する。
- 特定のユーザー ID を表明する任意のチャネルで使用する MCAUSER 値を設定する。
- 特定の SSL または TLS 識別名 (DN) を持つ任意のチャネルで使用する MCAUSER 値を設定する。
- 特定のキュー・マネージャーから接続する任意のチャネルに使用される MCAUSER 値を設定する。
- 特定のキュー・マネージャーから出されていても特定の IP アドレスから出されたものでない接続要求をブロックする。
- 特定の SSL/TLS 証明書を提示していても特定の IP アドレスから出されたものでない接続要求をブロックする。

以下の各セクションでは、上記の各用法について詳しく説明します。

MQSC コマンド **SET CHLAUTH** または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを作成、変更、または削除します。

**注 :** チャネル認証レコードが多数あると、キュー・マネージャーのパフォーマンスに悪影響が及ぶ可能性があります。

## IP アドレスのブロッキング

特定の IP アドレスからのアクセスをブロックするという役割はファイアウォールが果たすのが普通です。 しかし、WebSphere MQ システムにアクセスすべきでない IP アドレスからの接続試行があり、ファイアウォールが更新できるようになる前に一時的にそのアドレスをブロックしなければならない場合があるかもしれません。 これらの接続試行は WebSphere MQ チャネルからではなく、WebSphere MQ リスナーをターゲットとするように誤って構成された他のソケット・アプリケーションからも行われる可能性もあります。 このような場合には、BLOCKADDR タイプのチャネル認証レコードを設定することによって IP アドレ スのブロッキングを行います。 1 つ以上の単一アドレス、アドレス範囲、またはワイルドカードを含むパターンを指定できます。

このような方法で IP アドレスがブロックされたためにインバウンド接続が拒否された時はいつでも、チャネル・イベントが有効でキュー・マネージャーが実行中であれば、イベント・メッセージ MQRC_CHANNEL_BLOCKED が理由修飾子 MQRQ_CHANNEL_BLOCKED_ADDRESS 付きで発行されます。 さらに、エラーを戻す前に接続を 30 秒間オープンしたままにすることで、ブロックされた接続の試みがリスナーに対して過剰に繰り返されることがないようにされます。

IP アドレスを特定のチャネルでのみブロックしたり、エラーの報告に遅延が起きないようにしたりするには、タイプ ADDRESSMAP のチャネル認証レコードを USERSRC(NOACCESS) パラメーター付きで設定します。

この理由でインバウンド接続が拒否されたときはいつでも、チャネル・イベントが有効でキュー・マネージャーが実行していれば、イベント・メッセージ MQRC_CHANNEL_BLOCKED が理由修飾子 MQRQ_CHANNEL_BLOCKED_NOACCESS 付きで発行されます。

例については、182 ページの『特定の IP アドレスのブロッキング』を参照してください。

## ユーザー ID のブロッキング

特定のユーザー ID がクライアント・チャネルにより接続しないようにするには、タイプ BLOCKUSER のチャネル認証レコードを設定します。 このタイプのチャネル認証レコードはクライアント・チャ ネルにのみ適用され、メッセージ・チャネルには適用されません。 ブロックする 1 つ以上のユーザー ID を指定できますが、ワイルドカードを使用することはできません。

このためにインバウンド接続が拒否された時はいつでも、チャネル・イベントが有効であれば、イベント・メッセージ MQRC_CHANNEL_BLOCKED が理由修飾子 MQRQ_CHANNEL_BLOCKED_USERID 付で発行されます。

例については、184 ページの『特定のユーザー ID のブロッキング』を参照してください。

USERSRC(NOACCESS) パラメーターを付けて USERMAP タイプのチャネル認証レコードを設定 すれば、ユーザー ID を指定して特定のチャンネルからのアクセスをブロックすることもできます。

この理由でインバウンド接続が拒否されたときはいつでも、チャネル・イベントが有効でキュー・マネージャーが実行していれば、イベント・メッセージ MQRC_CHANNEL_BLOCKED が理由修飾子 MQRQ_CHANNEL_BLOCKED_NOACCESS 付きで発行されます。

例については、187 ページの『クライアントが表明したユーザー ID のアクセスのブロック化』を参照してください。

### キュー・マネージャー名のブロッキング

指定したキュー・マネージャーからのチャネル接続には アクセス権限を一切与えないようにするには、USERSRC(NOACCESS) パラメーターを付けて QMGRMAP タイ プのチャネル認証レコードを設定します。 単一のキュー・マネージャー名またはワイルドカードを含むパターンを指定できます。 キュー・マネージャーからのアクセスをブロックする、BLOCKUSER 機能と同等のものはありません。

この理由でインバウンド接続が拒否されたときはいつでも、チャネル・イベントが有効でキュー・マネージャーが実行していれば、イベント・メッセージ MQRC_CHANNEL_BLOCKED が理由修飾子 MQRQ_CHANNEL_BLOCKED_NOACCESS 付きで発行されます。

例については、186 ページの『リモート・キュー・マネージャーからのアクセスのブロック化』を参照してください。

### SSL または TLS 識別名のブロッキング

指定された識別名 (DN) を含む SSL または TLS 個人証明書を提示するユーザーがアクセスしないように指定するには、タイプ SSLPEERMAP のチャネル認証レコードを USERSRC(NOACCESS) パラメーター付で設定します。 単一の識別名またはワイルドカードを含むパターンを指定できます。 識別名 (DN) へのアクセスをブロックする、BLOCKUSER 機能と同等のものはありません。

この理由でインバウンド接続が拒否されたときはいつでも、チャネル・イベントが有効でキュー・マネージャーが実行していれば、イベント・メッセージ MQRC_CHANNEL_BLOCKED が理由修飾子 MQRQ_CHANNEL_BLOCKED_NOACCESS 付きで発行されます。

例については、187 ページの『SSL 識別名のアクセスのブロック化』を参照してください。

### IP アドレスと使用を義務づけるユーザー ID とのマッピング

特定の IP アドレスからのチャネル接 続に対して所定の MCAUSER の使用を義務づけるようにするには、ADDRESSMAP タイプのチャネル認証レコードを設定します。 単一のアドレス、アドレスの範囲、またはワイルドカードを含むパターンを指定できます。

ポート転送機能の使用、DMZ セッションの切断、またはキュー・マネージャーに IP アドレスに示されている IP アドレスを変更する任意の他のセットアップ、いずれかが行われた場合、マッピング IP アドレスは使用するのに必ずしも適切ではなくなっています。

例については、187 ページの『MCAUSER ユーザー ID への IP アドレスのマッピング』を参照してください。

### キュー・マネージャー名と使用を義務づけるユーザー ID とのマッピング

特定のキュー・マネージャーか らのチャネル接続に対して所定の MCAUSER の使用を義務づけるようにするには、QMGRMAP タイ プのチャネル認証レコードを設定します。 単一のキュー・マネージャー名またはワイルドカードを含むパターンを指定できます。

例については、184 ページの『MCAUSER ユーザー ID へのリモート・キュー・マネージャーのマッピング』を参照してください。

### クライアントによって表明されたユーザー ID と使用を義務づけるユーザー ID とのマッピング

WebSphere MQ MQI クライアントからの接続により特定のユーザー ID を使用するのか、それとは異なる指定された MCAUSER を使用するのかを指定するには、タイプ USERMAP のチャネル認証レコードを設定します。 ユーザー ID のマッピングにはワイルドカードは使用されません。

例については、185 ページの『MCAUSER ユーザー ID への表明済みユーザー ID のマッピング』を参照してください。

## SSL/TLS 識別名と使用を義務づけるユーザー ID とのマッピング

特定の識別名を含む SSL/TLS 個 人証明書を提示したユーザーに対して所定の MCAUSER の使用を義務づけるようにするには、SSLPEERMAP タイプのチャネル認証レコードを設定します。 単一の識別名またはワイルドカードを含むパターンを指定できます。

例については、186 ページの『MCAUSER ユーザー ID への SSL または TLS 識別名のマッピング』を参照してください。

## IP アドレスに応じて、キュー・マネージャー、クライアント、または SSL または TLS 識別名をマップする

場合によっては、第三者がキュー・マネージャー名を偽装するという可能性もあります。 SSL/TLS 証明書や鍵データベース・ファイルが盗用または再利用される恐れもあります。 こうした脅威に対抗する目的から、特定のキュー・マネージャーまたはクライアントからの接続や、特定の識別名を使用した接続に対して所定の IP アドレスの使用を義務づけるように指定しておくことができます。 タイプ USERMAP、QMGRMAP、または SSLPEERMAP のチャネル認証レコードを設定し、許可される IP アドレス、または IP アドレスのパターンを ADDRESS パラメーターを使用して指定します。

例については、184 ページの『MCAUSER ユーザー ID へのリモート・キュー・マネージャーのマッピング』を参照してください。

## チャネル認証レコードの相互作用

接続しようとしているチャネルに一致するチャネル認証レコードが複数存在し、それぞれのレコードが矛盾した結果を伴うものであるという可能性もあります。 例えば、あるチャネルで表明されたユーザー ID が BLOCKUSER タイプのチャネル認証レコード でブロックされているユーザー ID であっても、同じチャネルで提示されている SSL/TLS 証明書は 、別のユーザー ID をブロックキングの対象としている SSLPEERMAP レコードに一致するものあるため、 ブロッキングの対象にはなっていないという場合もあります。 さらに、チャネル認証レコードでワイルドカードが使用されている場合 、1 つの IP アドレス、キュー・マネージャー名、SSL または TLS 識別名が複数のパターンに一致 するという場合もあります。 例えば、IP アドレス 192.0.2.6 はパターン 192.0.2.0-24、192.0.2. *、および 192.0.*.6 に一致します。 以下では、このような場合に実行されるアクションについて説明します。

- どのチャネル認証レコードを優先するかは、次のような規則に基づいて決定されます。
  - 個々のチャネル名を明示的に指定しているチャネル認証レコードは、チャネル名をワイルドカ ードで指定しているチャネル認証レコードよりも優先されます。
  - SSL/TLS 識別名を使用しているチャネル認証レコードは、ユーザー ID、キュー・マネージャー名、または IP アドレスを使用しているレコードよりも優先されます。
  - ユーザー ID またはキュー・マネージャー名を使用しているチャネル認証レコードは、IP アド レスを使用しているレコードよりも優先されます。
- 一致するチャネル認証レコードが見つかり、そのレコードで指定されている MCAUSER がある場合には、その MCAUSER が当該のチャネルに割り当てられます。
- 一致するチャネル認証レコードが見つかり、そのレコードで当該のチャネルにアクセス権限がないと指定されている場合には、*NOACCESS という MCAUSER 値が当該のチャネルに割り当てられます。 この値は、あとでセキュリティー出口 プログラムによって変更されることもあります。
- 一致するチャネル認証レコードが見つからない場合、あるいは一致するチャネル認証レコード が見つかって、そのレコードで当該のチャネルにユーザー ID の使用を義務づけることが指定されている場合は、「MCAUSER」フィールドが調べられます。
  - 「MCAUSER」フィールドがブランクである場合は、クライアントのユーザー ID が当該のチャネルに割り当てられます。
  - 「MCAUSER」フィールドがブランクでない場合は、その値が当該のチャネルに割り当てられます。
- 任意のセキュリティー出口プログラムが実行されます。 この出口プログラムがチャネル・ユーザー ID を設定する場合や、 そのアクセスをブロックするかどうかを決定する場合もあります。
- その接続がブロックされるか、MCAUSER が *NOACCESS に設定された場合には、そのチャネルを終了します。

- クライアント・チャネル以外のチャネルについて接続がブロックされなかった場合、それ以前 のステップで決定されたチャネル・ユーザー ID がブロック対象ユーザーのリストと照合されます。
  - そのユーザー ID がブロック対象ユーザーのリストに含まれている場合には、そのチャネルを終了します。
  - そのユーザー ID がブロック対象ユーザーのリストに含まれていない場合には、そのチャネルを実行します。

いくつかのチャネル認証レコードが、チャネル名、IP アドレス、キュー・マネージャー名、または SSL または TLS 識別名 (DN) と一致する場合は、最も具体的な一致が使用されます。 最も具体的な一致と見なされるものは、次のようにして決定されます。

- チャネル名の場合:
  - 最も具体的な一致は、ワイルドカードのない名前、例えば、A.B.C。
  - 最も総称的な一致は単一のアスタリスク (*) で、これはすべてのチャネル名に一致します。
  - 左端の位置にアスタリスクがあるパターンの方が、左端の位置に定義値があるパターンよりも総称的です。 従って、*.B.C の方が A.* よりも総称的です。
  - 2 番目の位置にアスタリスクがあるパターンの方が、2 番目の位置に定義値があるパターンよりも総称的で、後ろの各位置についても同様です。 従って、A.*.C の方が A.B.* よりも総称的です。
  - 複数のパターンの同じ位置にアスタリスクがある場合には、アスタリスクの後ろに続くノードが少ない方がより総称的です。 したがって A. * は A*.C より総称的です
- IP アドレスの場合:
  - 最も具体的な一致は、ワイルドカードのない名前、例えば、192.0.2.6。
  - 最も総称的な一致は単一のアスタリスク (*) で、これはすべてのチャネル名に一致します。
  - 左端の位置にアスタリスクがあるパターンの方が、左端の位置に定義値があるパターンよりも総称的です。 従って、*.0.2.6 の方が 192.* よりも総称的です。
  - 2 番目の位置にアスタリスクがあるパターンの方が、2 番目の位置に定義値があるパターンよりも総称的で、後ろの各位置についても同様です。 従って、192.*.2.6 の方が 192.0.* よりも総称的です。
  - 複数のパターンの同じ位置にアスタリスクがある場合には、アスタリスクの後ろに続くノードが少ない方がより総称的です。 したがって、192 .* は 192.*.2.* より総称的です。
  - ハイフン (-) で示された範囲はアスタリスクよりも具体的なものとみなされます。 したがって、「192.0.2.0-24」は「192.0.2.*」よりも具体的なものと判定されます。
  - 別のサブセットに包含される範囲は、それを包含する範囲よりも具体的なものとみなされます。 したがって、「192.0.2.5-15」は「192.0.2.0-24」よりも具体的なものと判定されます。
  - 範囲の重複は認められません。 例えば、チャネル認証レコードを「192.0.2.0-15」と「192.0.2.10-20」の両方に設定しておくことはできません。
  - 末尾に単一のアスタリスクを付けたパターンでない限り、パターンを構成するパートの数を所定の必須パート数よりも少なくすることはできません。 例えば、192.0.2 は無効ですが、192.0.2.* は有効です。
  - 末尾のアスタリスクは、適切なパート分離文字 (IPv4 の場合はドット (.)、IPv6 の場合はコ ロン (:)) でアドレスの他の部分から切り離しておく必要があります。 例えば、「192.0*」というパターンは、アスタリスクが他のパートと分けられていないため無効です。
  - 末尾のアスタリスクに隣接していないかぎり、パターンに追加のアスタリスクを含めることができます。 例えば、192.*.2.* は有効ですが、192.0.*.* 無効です。
  - IPv6 アドレス・パターンには、二重のコロンと末尾のアスタリスクを含めることはできません。 結果アドレスがあいまいになるためです。 例えば、2001::* は、2001:0000:*、2001:0000:0000:* などと拡張解釈することができます。
- キュー・マネージャー名の場合:
  - 最も具体的な一致は、ワイルドカードのない名前、例えば、192.0.2.6。
  - 最も総称的な一致は単一のアスタリスク (*) で、これはすべてのチャネル名に一致します。

– 左端の位置にアスタリスクがあるパターンの方が、左端の位置に定義値があるパターンよりも総称的です。 従って、*QUEUEMANAGER の方が QUEUEMANAGER* よりも総称的です。
– 2 番目の位置にアスタリスクがあるパターンの方が、2 番目の位置に定義値があるパターンよりも総称的で、後ろの各位置についても同様です。 従って、Q*MANAGER の方が QUEUE* よりも総称的です。
– 複数のパターンの同じ位置にアスタリスクがある場合には、アスタリスクの後ろに続く文字が少ない方がより総称的です。 従って、Q* の方が Q*MGR よりも総称的です。

- SSL または TLS 識別名 (DN) の場合、サブストリングの優先順位は以下のようになります。

表 5. サブストリングの優先順位

| 順序 | 識別名のサブストリング | 名前 |
|---|---|---|
| 1 | SERIALNUMBER= | 証明書のシリアル番号 |
| 2 | MAIL= | メール・アドレス |
| 3 | E= | E メール・アドレス (MAIL の方が好ましいため非推奨) |
| 4 | UID=、USERID= | ユーザー ID |
| 5 | CN= | 共通名 |
| 6 | T = (T) | 役職 |
| 7 | OU= | 組織単位 |
| 8 | DC= | ドメイン・コンポーネント |
| 9 | O= | 組織 |
| 10 | STREET= | 通り/住所の 1 行目 |
| 11 | L= | 市町村 |
| 12 | ST=, SP=, S= | 都道府県 |
| 13 | PC = | 郵便番号 |
| 14 | C = (C) | 国名 |
| 15 | UNSTRUCTUREDNAME= | ホスト名 |
| 16 | UNSTRUCTUREDADDRESS= | IP アドレス |
| 17 | DNQ= | 識別名修飾子 |

従って、SSL または TLS 証明書にサブストリング O=IBM と C=UK を含む識別名 (DN) があり、O=IBM と C=UK の両方のチャネル認証レコードがある場合、WebSphere MQ は O=IBM の方を優先して使用します。

1 つの識別名には複数の組織単位を指定することができます。最も大きな組織単位を最初に指定して各組織単位を階層 順に指定していく必要があります。 2 つの識別名が組織単位値を除いたすべての点で等しい場合、どちらの識別名がより具体的なものであるかは以下の規則に従って判定されます。

1. 組織単位属性の数が異なる場合、組織単位値の数の多い方が、より具体的な識別名とみなされます。組織単位の数が多くなるほど、識別名を細かく限定できるようになり、使用できる突き合わせ条件の数も多くなるためです。 たとえ最上位の組織単位がワイルドカード (OU=*) であっても、指定されている組織単位数の多い識別名がより具体的な名前とみなされることに変わりありません。
2. 組織単位属性の数が同じである場合、対応する組織単位値のペアが、以下のルールに従って左 (具体性の低い上位の組織単位) から右 (具体性の高い下位の組織単位) に向かって順番に比較されていきます。
   a. 最も具体的なものと判定されるのは、ワイルドカードを含まない組織単位値です。正確に一致するストリングが 1 つしかないためです。

b. その次に具体的なものと判定されるのは、先頭または末尾にワイルドカードを 1 つ含む組織単位です (例えば、「OU=ABC*」や「OU=*ABC」)。
c. その次は、2 つのワイルドカードを含む組織単位です (例えば、「OU=*ABC*」)。
d. 最も具体性の低いものとみなされるのは、アスタリスク (OU=*) のみで構成される組織単位です。

3. 同じ具体性レベルを備えた 2 つの属性値についてストリングの比較が行えるようになっている 場合、長い方の属性ストリングがより具体的なものとみなされます。
4. 具体性レベルおよびストリングの長さの等しい 2 つの属性値についてストリング比較が行われる場合には、識別名のストリングからワイルドカー ドを除いた長さが比較されます (このストリング比較では大文字小文字は区別されません)。

DC 値以外のすべての点で 2 つの DN が等しい場合は、OU の場合と同じ突き合わせ規則が適用されます。ただし、DC 値の左端の DC は最下位レベル (最も特定レベル) であり、比較順序はそれに応じて異なります。

### チャネル認証レコードの表示

チャネル認証レコードを表示するには、MQSC コマンド **DISPLAY CHLAUTH** または PCF コマンド **Inquire Channel Authentication Records** を使用します。 指定したチャネル名に一致するすべてのレコードを返させるか、または明示的な一致レコードを返させるかを選択できます。 明示的な一致レコードを表示すると、特定の IP アドレス/キュー・マネージャー/ユーザー ID を 使用して接続を試みるチャネルや、特定の識別名を含む SSL/TLS 個人証明書を提示して接続しよう とするチャネルがあった場合に使用されるチャネル認証レコードを特定できるようになります。

**関連概念**

55 ページの『リモート・メッセージングのセキュリティー』
このセクションでは、リモート・メッセージングにおけるセキュリティーについて説明します。

## IBM WebSphere MQ でのメッセージ・セキュリティー

IBM WebSphere MQ インフラスト ラクチャー内のメッセージ・セキュリティーは、別個にライセンス交付される コンポーネントの IBM WebSphere MQ Advanced Message Security によっ て提供されます。

IBM WebSphere MQ Advanced Message Security (AMS) は、IBM WebSphere MQ セキュリティー・サービスを拡張して、データの署名および暗号化をメッセージ・レベルで提供します。 拡張されたサービスは、メッセージ・データが最初にキューに入れられてから取り出されるまでの間にメッセージ・データが変更されていないことを保証します。 さらに、AMS は、メッセージ・データの送信者が、署名されたメッセージをターゲット・キューに入れる権限を持っていることを確認します。

**関連概念**

272 ページの『IBM WebSphere MQ Advanced Message Security』
IBM WebSphere MQ Advanced Message Security (AMS) は、別個にライセンス交付される IBM WebSphere MQ Advanced Message Security のコンポーネントです。これを使用すると、末端のアプリケーションに影響を与えずに、IBM WebSphere MQ Advanced Message Security ネットワーク経由で流れる機密データを高水準で保護できます。

# セキュリティー要件の計画

このトピック集では、IBM WebSphere MQ 環境でセキュリティーに関する計画を立てる場合の注意点を取り上げます。

IBM WebSphere MQ は、さまざまなプラットフォーム上で、各種アプリケーションに使用できます。 しかし、アプリケーションごとに、セキュリティー要件が異なる場合があります。 アプリケーションの中には、セキュリティーが非常に重要な考慮事項であるものがあります。

WebSphere MQ は、Secure Sockets Layer (SSL) と Transport Layer Security (TLS) のサポートを含むさまざまなリンク・レベル・セキュリティー・サービスを 提供します。

WebSphere の実装時に、セキュリティーの特定の局面を考慮する必要があります。 UNIX、Linux、および Windows システムでは、これらの局面を無視して何も対処しない場合、WebSphere MQ を使用できません。

セキュリティーに関する考慮事項を以下に説明します。

### WebSphere MQ を管理する権限

WebSphere MQ 管理者には、次の権限が必要です。

- WebSphere MQ を管理するためのコマンドを実行する権限
- IBM WebSphere MQ エクスプローラーの使用

詳細については、次の章を参照してください。

- 197 ページの『UNIX, Linux, and Windows システム上の IBM WebSphere MQ を管理する権限』

### WebSphere MQ オブジェクトを処理する権限

アプリケーションは、MQI 呼び出しを発行して、次の WebSphere MQ オブジェクトにアクセスできます。

- キュー・マネージャー
- キュー
- Processes
- 名前リスト
- トピック

アプリケーションは、プログラマブル・コマンド・フォーマット (PCF) コマンドを使用して、これらの WebSphere MQ オブジェクトにアクセスできます。さらに、チャネルや認証情報オブジェクトにアクセスすることも可能です。 これらのオブジェクトは WebSphere MQ によって保護することができ、アプリケーションに関連付けられているユーザー ID には、これらのオブジェクトにアクセスするための権限が必要です。

詳細については、49 ページの『アプリケーションが IBM WebSphere MQ を使用するための許可』を参照してください。

### チャネル・セキュリティー

メッセージ・チャネル・エージェント (MCA) に関連付けられているユーザー ID には、さまざまな WebSphere MQ リソースにアクセスするための権限が必要です。 例えば、MCA は、キュー・マネージャーに接続できなければなりません。 MCA が送信側 MCA である場合、チャネル用の伝送キューを開くことができなければなりません。 MCA が受信側 MCA である場合は、宛先キューを開くことができなければなりません。 チャネル、チャネル・イニシエーター、およびリスナーを管理する必要がある、アプリケーションに関連付けられているユーザー ID には、関連の PCF コマンドを使用する権限が必要です。 ただし、ほとんどのアプリケーションでは、そのようなアクセス権限は必要ありません。

詳細については、69 ページの『チャネル許可』を参照してください。

### その他の考慮事項

セキュリティーに関する以下の側面を検討する必要があるのは、WebSphere MQ の特定の機能または基本製品の拡張機能を使用する場合に限られます。

- 78 ページの『キュー・マネージャー・クラスターのセキュリティー』
- 79 ページの『IBM WebSphere MQ Publish/Subscribe のセキュリティー』
- 80 ページの『IBM WebSphere MQ Internet Pass-Thru のセキュリティー』

## 識別と認証の計画

使用するユーザー ID と、認証制御を適用する方法およびレベルを決定します。

オペレーティング・システムによってさまざまな長さのユーザー ID がサポートされることを念頭において、IBM WebSphere MQ アプリケーションのユーザーを識別する方法を決定する必要があります。 チャネル認証レコードを使用して、あるユーザー ID から別のユーザー ID にマップしたり、接続の一部の属性に

基づいてユーザー ID を指定したりできます。 SSL または TLS を使用する IBM WebSphere MQ チャネルは、識別および認証のメカニズムとしてデジタル証明書を使用します。 各デジタル証明書はサブジェクト識別名を持っています。 この名前は、チャネル認証レコードを使用して特定の ID にマッピングできます。 さらに、鍵リポジトリー内の CA 証明書によって、IBM WebSphere MQ に対する認証に使用できるデジタル証明書が決まります。 詳しくは、以下を参照してください。

- 184 ページの『MCAUSER ユーザー ID へのリモート・キュー・マネージャーのマッピング』
- 185 ページの『MCAUSER ユーザー ID への表明済みユーザー ID のマッピング』
- 186 ページの『MCAUSER ユーザー ID への SSL または TLS 識別名のマッピング』
- 187 ページの『MCAUSER ユーザー ID への IP アドレスのマッピング』

## クライアント・アプリケーションの認証の計画

通信レベル、セキュリティー出口、チャネル認証レコード、およびセキュリティー出口に渡される ID の 4 つのレベルで認証コントロールを適用できます。

検討するセキュリティーのレベルには、次の 4 つがあります。 この図は、サーバーに接続されている IBM WebSphere MQ MQI クライアントを示しています。 以下の説明文にあるとおり、セキュリティーは 4 つのレベルで適用されます。 MCA は、メッセージ・チャネル・エージェントです。

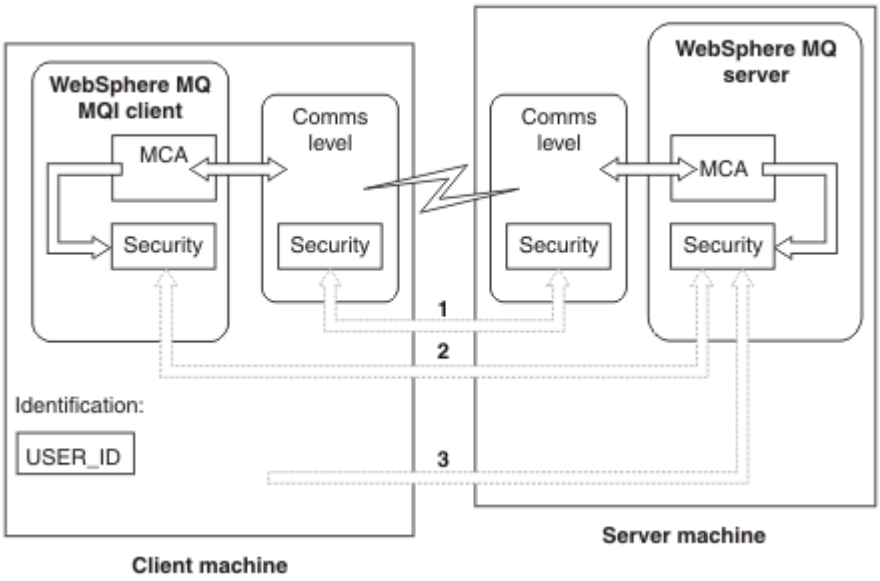



図 7. クライアント/サーバー間接続のセキュリティー

1. 通信レベル

   矢印 1 を参照。 通信レベルでセキュリティーを実装するには、SSL または TLS を使用します。 詳しくは、14 ページの『暗号セキュリティー・プロトコル: SSL および TLS』を参照してください。

2. チャネル認証レコード

   矢印 2 & 3 を参照してください。 認証は、セキュリティー・レベルで IP アドレスまたは SSL/TLS 識別名を使用して制御できます。 ユーザー ID をブロックしたり、表明されたユーザー ID を有効なユーザー ID にマップしたりすることもできます。 詳しい説明は、39 ページの『チャネル認証レコード』にあります。

3. チャネル・セキュリティー出口

   矢印 2 を参照。 クライアントからサーバーへの通信のためのチャネル・セキュリティー出口は、サーバー間通信の場合と同じ方法で機能します。 クライアントとサーバーの両方の相互認証を提供するために、プロトコルに依存しない一対の出口を書くことができます。 詳しい説明は、チャネル・セキュリティー出口プログラムにあります。

4. チャネル・セキュリティー出口に渡される ID

矢印 3 を参照。 クライアントからサーバーへの通信の場合、チャネル・セキュリティー出口はペアとして作動する必要はありません。 IBM WebSphere MQ クライアント側の出口は省略することができます。 この場合、ユーザー ID はチャネル記述子 (MQCD) に保管され、必要な場合はサーバー・サイド・セキュリティー出口によって変更することができます。

Windows クライアントは、識別を補助するための追加情報も送信します。

- サーバーに渡されるユーザー ID は、現在、クライアントにログオンしているユーザー ID です。
- 現在ログオンしているユーザーのセキュリティー ID。

IBM WebSphere MQ Client for HP Integrity NonStop Server における識別を補助するために、クライアントは、クライアント・アプリケーションを実行する OSS SAFEGUARD 別名を渡します。 この ID の形式は通常、<PRIMARYGROUP>.<ALIAS> です。 必要に応じて、チャネル認証レコードまたはセキュリティー出口を使用して、このユーザー ID をキュー・マネージャー上の代替ユーザー ID にマップすることができます。 このメッセージ出口について詳しくは、149 ページの『メッセージ出口による識別マッピング』 を参照してください。 チャネル認証レコードの定義について詳しくは、185 ページの『MCAUSER ユーザー ID への表明済みユーザー ID のマッピング』 を参照してください。

ユーザー ID の値、および使用可能ならセキュリティー ID の値は、IBM WebSphere MQ MQI クライアントの ID を確立するために、サーバー・セキュリティー出口で使用することができます。

### ユーザー *ID*

IBM WebSphere MQ MQI クライアントが Windows 上にあり、 IBM WebSphere MQ サーバーも Windows 上にあり、クライアント・ユーザー ID が定義されているドメインにアクセスできる場合、IBM WebSphere MQ は最大 20 文字のユーザー ID をサポートします。 UNIX and Linux のプラットフォームおよび構成では、最大長は 12 文字です。

WebSphere MQ for Windows サーバーは、クライアントが @ 文字を含むユーザー ID (例えば、 abc@d) で実行されている場合、Windows クライアントの接続をサポートしません。 クライアントの MQCONN 呼び出しへの戻りコードは、MQRC_NOT_AUTHORIZED になります。

ただし、2 つの @ 文字を使用してユーザー ID を指定できます (例: abc@@d)。 ユーザー ID が正しいドメインで一貫して解決されるようにするために、 id@domain 形式を使用することをお勧めします。これにより、 abc@@d@domain のようになります。

UNKNOWN は予約済みのユーザー ID で、NOBODY ユーザー ID も WebSphere MQ では特別な意味があることに注意してください。 オペレーティング・システムで UNKNOWN または NOBODY というユーザー ID を作成すると、意図しない結果になることがあります。

Windows を除いて、ユーザー ID は認証に使用しますが、グループは許可に使用します。

グループを考慮に入れずにサービス・アカウントを作成し、すべてのユーザー ID を個別に許可すると、すべてのユーザーが、他のすべてのユーザーの情報にアクセスできるようになってしまいます。

## 許可の計画

管理権限を持つユーザーや、IBM WebSphere MQ オブジェクトを適切に使用するアプリケーションのユーザー (IBM WebSphere MQ MQI クライアントから接続するユーザーを含む) を許可する方法を計画します。

IBM WebSphere MQ を使用するには、個々のユーザーまたはアプリケーションにアクセス権限を付与する必要があります。 必要なアクセス権限は、ユーザーまたはアプリケーションが受け持つ役割や、それらが実行する必要があるタスクによって異なります。 IBM WebSphere MQ での許可は、次の 2 つの主要なカテゴリーに分けることができます。

- 管理操作を実行する許可
- アプリケーションが IBM WebSphere MQ を使用するための許可

これらの操作のクラスはどちらも同じコンポーネントによって制御され、操作のカテゴリーを両方とも実行する権限を個々のユーザーまたはアプリケーションに付与することができます。

考慮する必要がある許可の特定の領域については、以下のトピックを参照してください。

## IBM WebSphere MQ を管理する権限

IBM WebSphere MQ 管理者には、さまざまな機能を実行するための権限が必要です。 その権限を取得する方法は、プラットフォームによって異なります。

IBM WebSphere MQ 管理者には、以下の権限が必要です。

- IBM WebSphere MQ を管理するためのコマンドの発行
- 代わりに、IBM WebSphere MQ Explorer

詳しくは、ご使用のオペレーティング・システムに該当するトピックを参照してください。

### *UNIX* および *Windows* システム上の *IBM WebSphere MQ* を管理する権限

IBM WebSphere MQ 管理者は、mqm グループのメンバーです。 このグループは、すべての IBM WebSphere MQ リソースにアクセスして、IBM WebSphere MQ 制御コマンドを実行することができます。 管理者は、他のユーザーに特定の権限を付与することができます。

UNIX および Windows システムで IBM WebSphere MQ 管理者になるには、ユーザーは *mqm* グループのメンバーでなければなりません。 このグループは、WebSphere MQ のインストール時に自動的に作成されます。 ユーザーが制御コマンドを発行できるようにするには、そのユーザーを mqm グループに追加する必要があります。 これには、UNIX システム上の root ユーザーが含まれます。

mqm グループのメンバーではないユーザーに管理特権を付与することができますが、それらのユーザーは IBM WebSphere MQ 制御コマンドを実行することはできません。アクセスが付与されたコマンドのみを実行することが許可されています。

さらに、Windows システムでは、SYSTEM アカウントと管理者アカウントが IBM WebSphere MQ リソースに対する全アクセス権限を持っています。

mqm グループのすべてのメンバーは、システム上で実行されている任意のキュー・マネージャーを管理できる権限を含めて、すべてのシステム上のすべての WebSphere MQ リソースにアクセスする権限を持っています。 このアクセス権は、ユーザーを mqm グループから除去することだけで、取り消すことができます。 Windows システム上では、Administrators グループのメンバーにも、すべての WebSphere MQ リソースへのアクセス権があります。

管理者は、**runmqsc** 制御コマンドを使用して、WebSphere MQ Script (MQSC) コマンドを発行することができます。 MQSC コマンドをリモート・キュー・マネージャーに送信するために **runmqsc** が間接モードで使用される場合、各 MQSC コマンドは、Escape PCF コマンド内にカプセル化されます。 管理者は、MQSC コマンドがリモート・キュー・マネージャーによって処理されるのに必要な権限を持っていなければなりません。

WebSphere MQ エクスプローラーでは、PCF コマンドによって管理タスクを実行します。 管理者には、WebSphere MQ エクスプローラーを使用してローカル・システム上のキュー・マネージャーを管理するための追加の権限は必要ありません。 WebSphere MQ エクスプローラーが別のシステム上のキュー・マネージャーの管理に使用される場合、管理者には、PCF コマンドがリモート・キュー・マネージャーによって処理されるのに必要な権限が必要です。

PCF コマンドと MQSC コマンドの処理時に実行される権限検査の詳細については、以下のトピックを参照してください。

- キュー・マネージャー、キュー、チャネル、プロセス、名前リスト、および認証情報オブジェクトを対象として実行されるコマンドについては、49 ページの『アプリケーションが IBM WebSphere MQ を使用するための許可』を参照してください。
- チャネル、チャネル・イニシエーター、リスナー、クラスターに対して実行するコマンドについては、『チャネル・セキュリティー』を参照してください。

UNIX および Windows システム上で WebSphere MQ の管理に必要な権限の詳細については、関連情報を参照してください。

## アプリケーションが IBM WebSphere MQ を使用するための許可

アプリケーションがオブジェクトにアクセスするときには、そのアプリケーションに関連するユーザー ID に適切な権限が必要です。

アプリケーションは、MQI 呼び出しを発行して、次の IBM WebSphere MQ オブジェクトにアクセスできます。

- キュー・マネージャー
- キュー
- Processes
- 名前リスト
- トピック

アプリケーションでは PCF コマンドを使用して、IBM WebSphere MQ オブジェクトを管理することもできます。 PCF コマンドが処理されるとき、ユーザー ID の権限コンテキストを使用して PCF メッセージが書き込まれます。

この場合、アプリケーションには、ユーザーとベンダーによって作成されるアプリケーションが含まれます。

IBM WebSphere MQ classes for Java、 IBM WebSphere MQ classes for JMS、 IBM WebSphere MQ classes for .NET、または Message Service Clients for C/C++ and .NET を使用するアプリケーションは、MQI を間接的に使用します。

また、MCA も、MQI 呼び出しを発行します。この MCA に関連したユーザー ID には、これらの WebSphere MQ オブジェクトにアクセスする権限が必要です。 これらのユーザー ID と、そのユーザー ID が必要とする権限の詳細については、69 ページの『チャネル許可』を参照してください。

## 権限検査が実行される場合

権限検査が実行されるのは、アプリケーションがキュー・マネージャー、キュー、プロセス、名前リストのいずれかにアクセスしようとするときです。

権限検査は、次の状況のもとで実行されます。

**アプリケーションが、MQCONN または MQCONNX 呼び出しを使用してキュー・マネージャーに接続するとき**

キュー・マネージャーは、アプリケーションに関連したユーザー ID を、オペレーティング・システムに要求します。 次に、キュー・マネージャーは、そのユーザー ID がそのキュー・マネージャーに接続する権限があるかどうかを調べ、今後の検査用にそのユーザー ID を保持します。

ユーザーは IBM WebSphere MQ にサインオンする必要はありません。 IBM WebSphere MQ では、ユーザーが基礎となるオペレーティング・システムにサインオンしていて、認証されているものと想定しています。

**アプリケーションが MQOPEN または MQPUT1 呼び出しを使用して IBM WebSphere MQ オブジェクトを開くとき**

すべての権限検査は、オブジェクトを開くときに実行され、その後そのオブジェクトにアクセスするときには実行されません。 例えば、権限検査はアプリケーションが名前リスト・オブジェクトを開くときに実行されます。 この検査は、アプリケーションがメッセージをキューに入れるとき、またはメッセージをキューから取得するときには、実行されません。

アプリケーションは、オブジェクトを開くときに、そのオブジェクトを対象として実行する必要がある操作のタイプを指定します。 例えば、アプリケーションが、キューを開いて、そのキュー上のメッセージをブラウズし、そのキューからメッセージを取得することはできても、そのキューにメッセージを入れることができない場合があります。 操作のタイプごとに、キュー・マネージャーは、アプリケーションに関連したユーザー ID に、その操作を実行する権限があるかどうかを調べます。

アプリケーションがキューを開くと、オブジェクト記述子の ObjectName フィールドで指定されたオブジェクトに対して、権限検査が実行されます。 ObjectName フィールドは、MQOPEN または MQPUT1 呼び出しで使用します。 このオブジェクトが別名キューまたはリモート・キュー定義である場合は、オブジェクトそのものに対して権限検査が実行されます。 検査は、別名キューまたはリモート・キューの定義が解決されるキューでは実行されません。 そのため、ユーザーはアクセスするための許可を必要としません。 キューを作成する権限は、特権ユーザーに限定してください。 限定しないと、一部のユーザーが単に別名を作成して 通常のアクセス管理を逃れる事態になりかねません。

アプリケーションはリモート・キューを明示的に参照できます。 アプリケーションは、オブジェクト記述子の ObjectName フィールドと ObjectQMgrName フィールドに、リモート・キューの名前とリ

モート・キュー・マネージャーの名前を設定します。 権限検査は、リモート・キュー・マネージャーと同じ名前の伝送キューに対して実行されます。 UNIX, Linux, and Windows では、クラスタリングが使用されていれば、リモート・キュー・マネージャー名と一致する RQMNAME プロファイルに対して検査が行われます。 アプリケーションは、オブジェクト記述子の ObjectName フィールドにクラスター・キューの名前を設定することによって、クラスター・キューを明示的に参照できます。 権限検査は、クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE に対して実行されます。

動的キューに対する権限は、それが派生したモデル・キューに基づきますが、必ずしも同じではありません (注 1 を参照)。

キュー・マネージャーが権限検査に使用するユーザー ID は、オペレーティング・システムから取得されます。 このユーザー ID は、アプリケーションがキュー・マネージャーに接続されるときに取得されます。 適切に許可されたアプリケーションは、代替ユーザー ID を指定した MQOPEN 呼び出しを発行できます。続いて、その代替ユーザー ID に対してアクセス制御検査が行われます。 代替ユーザー ID を使用しても、アプリケーションに関連付けられたユーザー ID は変更されず、単にアクセス制御検査にのみ使用されます。

**アプリケーションが MQSUB 呼び出しを使用してトピックをサブスクライブするとき**

アプリケーションがトピックをサブスクライブする際、アプリケーションは、実行する必要がある操作のタイプを指定します。 サブスクリプションを作成するか、既存のサブスクリプションを変更するか、既存のサブスクリプションを変更なしで再開するかのいずれかになります。 それぞれのタイプの操作についてキュー・マネージャーは、その操作を実行するための権限が、アプリケーションに関連付けられたユーザー ID に付与されていることを確認します。

アプリケーションがトピックにサブスクライブするとき、トピック・ツリーで見つかったトピック・オブジェクトに対して権限検査が実行されます。 実行対象のトピック・オブジェクトは、アプリケーションがサブスクライブしたトピック・ツリー内の位置またはその上位にあるトピック・オブジェクトです。 権限検査には、複数のトピック・オブジェクトに対する検査が関係する場合があります。 キュー・マネージャーが権限検査に使用するユーザー ID は、オペレーティング・システムから取得されます。このユーザー ID は、アプリケーションがキュー・マネージャーに接続されるときに取得されます。

キュー・マネージャーは、サブスクライバーのキューに対して権限検査を 実行しますが、管理対象キューに対しては実行しません。

**アプリケーションが MQCLOSE 呼び出しを使用して永続動的キューを削除するとき**

MQCLOSE 呼び出しで指定されたオブジェクト処理は、必ずしも永続動的キューを作成した MQOPEN 呼び出しから返されたオブジェクト処理と同じではありません。 これが異なる場合は、キュー・マネージャーが、MQCLOSE 呼び出しを発行したアプリケーションに関連付けられているユーザー ID を検査します。 この検査では、ユーザー ID がキューを削除する権限を持っているかどうかが調べられます。

サブスクリプションを閉じて削除するアプリケーションが、サブスクリプションを作成したアプリケーションでない場合は、削除するための適切な権限が必要です。

**WebSphere MQ オブジェクトを対象として実行される PCF コマンドが、コマンド・サーバーによって処理されるとき**

このルールには、PCF コマンドが認証情報オブジェクトに対して実行される場合も含まれます。

権限検査に使用されるユーザー ID は、PCF コマンドのメッセージ記述子内 の UserIdentifier フィールドにあるユーザー ID です。 このユーザー ID には、このコマンドが処理されるキュー・マネージャー上で必須の権限が必要です。 Escape PCF コマンド内にカプセル化された、同等の MQSC コマンドも、同じように扱われます。 UserIdentifier フィールドとその設定方法の詳細については、52 ページの『メッセージ・コンテキスト』を参照してください。

## 代替ユーザー権限

アプリケーションは、オブジェクトを開いたりトピックにサブスクライブしたりするときに、MQOPEN、MQPUT1、MQSUB の各呼び出しでユーザー ID を指定できます。 さらに、キュー・マネージャーに対して、アプリケーションに関連したユーザー ID ではなく、そのユーザー ID を権限検査で使用するように指定できます。

アプリケーションがオブジェクトを正常に開くことができるのは、次の両方の条件が満たされる場合だけです。

- アプリケーションに関連したユーザー ID に、権限検査用に別のユーザー ID を指定する権限がある。 この場合、アプリケーションには、代替ユーザー権限 があると表現します。
- アプリケーションが指定するユーザー ID に、要求された操作のタイプのオブジェクトを開く権限、またはトピックをサブスクライブする権限がある。

## メッセージ・コンテキスト

メッセージ・コンテキスト 情報により、メッセージを受信するアプリケーションは、そのメッセージの発信元についての情報を得ることができます。 その情報は、メッセージ記述子の各フィールドに格納されます。それらのフィールドは、3 つの論理部分に分けられています。

以下の部分があります。

**identity コンテキスト (identity context)**
これらのフィールドには、メッセージをキューに入れたアプリケーションのユーザーについての情報が入っています。

**origin コンテキスト**
これらのフィールドには、アプリケーション自体の情報と、メッセージがキューに入れられた時間についての情報が入っています。

**user コンテキスト**
これらのフィールドには、アプリケーションがキュー・マネージャーの送達するメッセージを選択するために使用できるメッセージ・プロパティーが入っています。

アプリケーションがメッセージをキューに入れるときに、そのアプリケーションは、メッセージ内にコンテキスト情報を生成するように、キュー・マネージャーに依頼することができます。 これが、デフォルトのアクションです。 アプリケーションはまた、コンテキスト・フィールドに情報を入れないように指定することもできます。 アプリケーションに関連したユーザー ID には、これらのどちらかを実行するためにも、特殊権限は必要ありません。

アプリケーションは、メッセージ内の identity コンテキスト・フィールドを設定して、キュー・マネージャーが origin コンテキストを生成できるようにするか、またはすべてのコンテキスト・フィールドを設定することができます。 また、アプリケーションは、取り出したメッセージから、キューに入れるメッセージに、identity コンテキスト・フィールドを渡したり、すべてのコンテキスト・フィールドを渡したりすることもできます。 ただし、アプリケーションに関連したユーザー ID には、コンテキスト情報を設定したり、渡すための権限が必要です。 アプリケーションは、メッセージを入れるキューを開くときに、コンテキスト情報を設定するか、渡すことを指定し、この時点で権限が検査されます。

次に、各コンテキスト・フィールドを簡単に説明します。

**identity コンテキスト**

**UserIdentifier**
メッセージを入れたアプリケーションに関連したユーザー ID。 キュー・マネージャーがこのフィールドを設定する場合、このフィールドは、アプリケーションがキュー・マネージャーに接続されるときに、オペレーティング・システムから取得されるユーザー ID に設定されます。

**AccountingToken**
メッセージの結果実行された作業料の請求に使用できる情報。

**ApplIdentityData**
アプリケーションに関連したユーザー ID に、identity コンテキスト・フィールドを設定する権限、またはすべてのコンテキスト・フィールドを設定する権限がある場合、そのアプリケーションは、identity に関連した任意の値にこのフィールドを設定することができます。 キュー・マネージャーがこのフィールドを設定する場合、このフィールドはブランクに設定されます。

**origin コンテキスト**

**PutApplType**
メッセージを入れたアプリケーションのタイプ。例えば、CICS® トランザクション。

**PutApplName**
メッセージを書き込むアプリケーションの名前。

**PutDate**
メッセージが入れられた日付。

**PutTime**
メッセージが入れられた時刻。

**ApplOriginData**
アプリケーションに関連したユーザー ID に、すべてのコンテキスト・フィールドを設定する権限がある場合、そのアプリケーションは、origin に関連した任意の値にこのフィールドを設定することができます。 キュー・マネージャーがこのフィールドを設定する場合、このフィールドはブランクに設定されます。

**user コンテキスト**

**MQINQMP** または **MQSETMP** に関して、次の値がサポートされています。

**MQPD_USER _CONTEXT**

プロパティーは user コンテキストに関連付けられます。

MQSETMP 呼び出しを使用してユーザー・コンテキストと関連付けたプロパティーを設定するのに、特別な権限は必要ありません。

V7.0 以降のキュー・マネージャーの場合、user コンテキストに関連したプロパティーは、MQOO_SAVE_ALL_CONTEXT の説明どおりに保存されます。 MQOO_PASS_ALL_CONTEXT を指定して MQPUT を実行すると、保存されたコンテキストから新しいメッセージへプロパティーがコピーされることになります。

**MQPD_NO_CONTEXT**

プロパティーはメッセージ・コンテキストに関連付けられません。

認識されない値は拒否されて、MQRC_PD_ERROR になります。 このフィールドの初期値は **MQPD_NO_CONTEXT** です。

各コンテキスト・フィールドの詳細については、MQMD - メッセージ記述子を参照してください。 メッセージ・コンテキストを使用する方法の詳細については、メッセージ・コンテキストを参照してください。

## *UNIX*、 *Linux* 、および *Windows* システムで *IBM WebSphere MQ* オブジェクトを処理する権限

IBM WebSphere MQ に付属の許可サービス・コンポーネントは、オブジェクト権限マネージャー *(OAM)* と呼ばれます。 このコンポーネントでは、認証検査および許可検査によるアクセス制御が提供されます。

1. 認証。

   IBM WebSphere MQ に付属の OAM で実行される認証検査は、基本的なものであり、特定の状況でのみ実行されます。 高度なセキュア環境で要求される厳密な要件に適合することは意図されていません。

   OAM では、アプリケーションがキュー・マネージャーに接続されるときに、次のような条件に適合する場合、認証検査が実行されます。

   接続中のアプリケーションで MQCSP 構造体が提供されていて、MQCSP 構造体の *AuthenticationType* 属性に値 MQCSP_AUTH_USER_ID_AND_PWD が指定されている場合、OAM の MQZID_AUTHENTICATE_USER 機能によって検査が実行されます。 この検査は、MQCSP 構造体のユーザー ID と *IdentityContext* (MQZIC) のユーザー ID とを比較して、それらが一致するかどうかを判別するというものです。 それらが一致しない場合は、検査は不合格となります。

   この基本的な検査は、ユーザーの完全認証を意図するものではありません。 例えば、MQCSP 構造体で提供されるパスワードの検査によってユーザーの認証性を検査することはしません。 また、アプリケーションで MQCSP 構造体が省略されている場合、検査は実行されません。

   許可サービス・コンポーネントによる、より完全な認証サービスがキュー・マネージャーで要求される場合でも、IBM WebSphere MQ に付属の OAM では、このサービスは提供されません。 許可サービス・コンポーネントを新規に作成するか、ベンダーから入手する必要があります。

2. 許可。

   許可検査は包括的なものであり、ほとんどの標準的な要件に適合することが意図されています。

アプリケーションが MQI 呼び出しを実行して、キュー・マネージャー、キュー、プロセス、トピック、名前リストのいずれかにアクセスするときに、許可検査が実行されます。 その他にも、例えば、コマンド・サーバーによってコマンドが実行されているときに、許可検査が実行されます。

UNIX、 Linux および Windows システムでは、キュー・マネージャー、キュー、プロセス、トピック、または名前リストである IBM WebSphere MQ オブジェクトにアクセスするためにアプリケーションが MQI 呼び出しを発行すると、許可サービス がアクセス制御を提供します。 このアクセス制御には、代替ユーザー権限、およびコンテキスト情報を設定または渡す権限の検査が含まれます。

Windows では、OAM は、UAC が有効になっている場合でも、Administrators グループのメンバーにすべての IBM WebSphere MQ オブジェクトにアクセスする権限を付与します。

さらに、 Windows システムでは、SYSTEM アカウントに IBM WebSphere MQ リソースへの全アクセス権限があります。

許可サービスは、これらの IBM WebSphere MQ オブジェクトのいずれか、または認証情報オブジェクトを対象として PCF コマンドが実行されるときにも、権限検査を提供します。 Escape PCF コマンド内にカプセル化された、同等の MQSC コマンドも、同じように扱われます。

許可サービスは、インストール可能なサービス です。これは、許可サービスは、1 つ以上のインストール可能サービスのコンポーネント によってインプリメントされることを意味しています。 各コンポーネントは、文書化されたインターフェースを使用して起動されます。 これにより、ユーザーとベンダーは、IBM WebSphere MQ MQ 製品によって提供されるコンポーネントを拡充したり、交換するためのコンポーネントを提供できるようになります。

IBM WebSphere MQ に付属の許可サービス・コンポーネントは、オブジェクト権限マネージャー *(OAM)* と呼ばれます。 作成するキュー・マネージャーごとに、OAM は自動的に使用可能になります。

OAM は、OAM がアクセス権を制御する IBM WebSphere MQ オブジェクトごとに、アクセス制御リスト (ACL) を保持します。 UNIX and Linux システム上では、グループ ID だけを、ACL 内に表示することができます。 これは、グループのすべてのメンバーは、同じ権限を持っていることを意味しています。 Windows システム上では、ユーザー ID とグループ ID の両方を、ACL に表示することができます。 これは、権限は、個々のユーザーおよびグループに対して付与できることを意味しています。

グループおよびユーザー ID のいずれにも、12 文字までという制限が当てはまります。 UNIX プラットフォームは通常、ユーザー ID の長さを 12 文字までと制限しています。AIX また、 Linux はこの制限を引き上げましたが、 IBM WebSphere MQ は引き続きすべての UNIX プラットフォームで 12 文字の制限を順守します。 12 文字を超えるユーザー ID を使用すると、IBM WebSphere MQ はその ID を "UNKNOWN" という値に置き換えます。 「"UNKNOWN"」という値でユーザー ID を定義しないでください。

OAM はユーザーを認証し、該当するアイデンティティー・コンテキスト・フィールドを変更します。 これを使用可能にするには、MQCONNX 呼び出しで接続セキュリティー・パラメーター構造 (MQCSP) を指定します。 構造は OAM Authenticate User 機能 (MQZ_AUTHENTICATE_USER) に渡され、それによって該当するアイデンティティー・コンテキスト・フィールドが設定されます。 IBM WebSphere MQ クライアントからの MQCONNX 接続の場合、MQCSP 内の情報は、クライアントがクライアント接続およびサーバー接続チャネルを介して接続しているキュー・マネージャーに流れます。 セキュリティー出口がそのチャネルで定義されている場合、MQCSP は各セキュリティー出口に渡され、出口がそれを変更することができます。 セキュリティー出口は MQCSP を作成することもできます。 このコンテキストでセキュリティー出口を使用するための詳細については、『チャネル・セキュリティー出口プログラム』を参照してください。

UNIX、Linux、および Windows システムでは、制御コマンド **setmqaut** は、権限の付与と取り消しを行い、ACL の保持に使用されます。 例えば、次のコマンドを入力するとします。

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

このコマンドにより、グループ VOYAGER のメンバーは、キュー・マネージャー JUPITER によって所有される MOON.EUROPA キュー上で、メッセージをブラウズできるようになります。 このコマンドは、メンバーがキューからメッセージの取得もできるようにします。 それらの権限を後で取り消す場合は、以下のコマンドを入力します。

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

また、次のコマンドがあるとします。

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

グループ VOYAGER のメンバーが、文字 MOON. で始まる名前を持つ任意のキューにメッセージを書き込むことができるようにします。 変更されました。* 総称プロファイルの名前です。総称プロファイル を使用すると、単一の **setmqaut** コマンドを使用して、オブジェクトのセットに対する権限を付与することができます。

制御コマンド **dspmqaut** は、指定されたオブジェクトに対するユーザーまたはグループの 現行の権限を表示するために使用できます。制御コマンド **dmpmqaut** も、総称プロファイルに関連した現行の権限を表示するのに使用できます。

権限検査が必要ない場合 (例えば、テスト環境)、OAM を使用不可にすることができます。

*PCF* を使用しての *OAM* コマンドへのアクセス
UNIX、Linux、および Windows システムでは、PCF コマンドを使用して OAM 管理コマンドにアクセスできます。

PCF コマンドおよびそれらと同等の OAM コマンドを次に示します。

表 *6. PCF* コマンドおよびそれらと同等の *OAM* コマンド

| **PCF コマンド** | **OAM コマンド** |
|---|---|
| Inquire Authority Records | dmpmqaut |
| Inquire Entity Authority | dspmqaut |
| Set Authority Record | setmqaut |
| Delete Authority Record | setmqaut with -remove option |

**setmqaut** および **dmpmqaut** コマンドは、mqm グループのメンバーに制限されます。 キュー・マネージャー上で dsp および chg 権限を付与された任意のグループのユーザーは、同等の PCF コマンドを実行することはできます。

これらのコマンドの使用について詳しくは、「プログラマブル・コマンド・フォーマットの概要」を参照してください。

## リモート・メッセージングのセキュリティー

このセクションでは、リモート・メッセージングにおけるセキュリティーについて説明します。

IBM WebSphere MQ の機能を使用する権限をユーザーに提供する必要があります。 これは、オブジェクトおよび定義に関して行う操作に応じて編成されます。 以下に例を示します。

- キュー・マネージャーは、許可されたユーザーが開始および停止できる。
- アプリケーションは、キュー・マネージャーに接続される必要があり、キューを使用する権限を持つ。
- メッセージ・チャネルは、許可されたユーザーが作成および制御する必要がある。
- オブジェクトはライブラリーに保管され、これらのライブラリーへのアクセスは制限できる。

リモート・サイトでのメッセージ・チャネル・エージェントは、送達されるメッセージがそのリモート・サイトでメッセージ送達を行う権限をもつユーザーから送られたものであることを確認する必要があります。 さらに、MCA はリモートで開始することが可能であるため、MCA を開始しようとするリモート・プロセスがそのための権限をもっていることを検査する必要があります。 その検査方法には、次の 4 つがあります。

1. RCVR、RQSTR、または CLUSRCVR チャネル定義の PutAuthority 属性を適切に使用して、着信メッセージがキューに書き込まれるときに、どのユーザーが許可検査に使用されるかを制御する。 MQSC コマンド・リファレンスの DEFINE CHANNEL コマンドの説明を参照してください。
2. チャネル認証レコードを実装し、不要な接続試行を拒否するか、リモート IP アドレス、リモート・ユーザー ID、提供されている SSL または TLS のサブジェクトの識別名 (DN)、またはリモート・キュー・マネージャー名に基づいて MCAUSER 値を設定する。

3. ユーザー出口 セキュリティー検査を実施して、対応するメッセージ・チャネルが認可されていることを確認する。 対応するチャネルのホストとなるシステムのセキュリティー機能を使用して、すべてのユーザーが適切な権限をもつことを確認し、個々のメッセージの検査を行わずに済むようにする。
4. ユーザー出口 メッセージ処理を実施し、個々のメッセージが許可を得ているかどうかを調べるようにする。

### *UNIX and Linux* システムでのオブジェクトのセキュリティー

管理ユーザーがその ID で IBM WebSphere MQ 管理コマンドを使用しようとする場合、その管理ユーザーは、使用しているシステムの mqm グループ (ルートを含む) に属している必要があります。

必ずユーザー ID「mqm」で amqcrsta を実行してください。

### UNIX and Linux システムでのユーザー ID

キュー・マネージャーは、大文字または大/小文字混合から成るすべてのユーザー ID を小文字に変換します。 その後、キュー・マネージャーは、ユーザー ID をメッセージのコンテキスト部分に挿入したり、権限を調べたりします。 したがって、権限は小文字の ID にのみ基づいています。

### *Windows* システムでのオブジェクトのセキュリティー

Windows システムでこの ID が IBM WebSphere MQ 管理コマンドを使用する場合、管理ユーザーは mqm グループと管理者グループの両方に属している必要があります。

### Windows システムのユーザー ID

Windows システムでは、メッセージ出口がインストールされていない場合は、キュー・マネージャーが、大文字または大/小文字混合から成るすべてのユーザー ID を小文字に変換します。 その後、キュー・マネージャーは、ユーザー ID をメッセージのコンテキスト部分に挿入したり、権限を調べたりします。 したがって、権限は小文字の ID にのみ基づいています。

### システム間のユーザー *ID*

Windows システムおよび UNIX and Linux システム以外のプラットフォームでは、メッセージ内のユーザー ID に大文字を使用します。

Windows システムおよび UNIX and Linux システムでメッセージ内のユーザー ID に小文字を使用できるようにするために、これらのプラットフォームでは、メッセージ・チャネル・エージェント (MCA) により以下の変換が行われます。

**送信側で**
メッセージ出口がインストールされていない場合は、すべてのユーザー ID 中の英字を大文字に変換します。

**受信側で**
メッセージ出口がインストールされていない場合は、すべてのユーザー ID 中の英字を小文字に変換します。

これ以外の何らかの理由で UNIX、Linux および Windows システムにメッセージ出口を提供した場合、自動的な変換は行われません。

## カスタム許可サービスの使用

IBM WebSphere MQ は、インストール可能な許可サービスを提供します。 代替サービスのインストールを選択することもできます。

IBM WebSphere MQ と共に提供される許可サービス・コンポーネントは、 オブジェクト権限マネージャー (OAM) と呼ばれます。 必要な許可機能が OAM によって提供されない場合は、独自の許可サービス・コンポーネントを作成することができます。 許可サービス・コンポーネントが実装する必要があるインストール可能なサービス機能については、インストール可能サービス・インターフェースの参照情報で説明されています。

## クライアントへのアクセス制御

アクセス制御は、ユーザー ID に基づいて行われます。 管理するユーザー ID が多く存在する場合もあり、ユーザー ID は異なる形式になる場合もあります。 サーバー接続のチャネル・プロパティーである MCAUSER をクライアントが使用する特別なユーザー ID 値に設定することができます。

IBM WebSphere MQ でのアクセス制御は、ユーザー ID に基づいて行われます。 通常は、MQI 呼び出しを発行するプロセスのユーザー ID が使用されます。 MQ MQI クライアントの場合、サーバー接続の MCA が、MQ MQI クライアントの代わりに MQI 呼び出しを発行します。 MQI 呼び出しを発行するために使用するサーバー接続 MCA の代替のユーザー ID を選択できます。 代替のユーザー ID は、クライアント・ワークステーションに関連付けられたものにするか、クライアントのアクセスを編成して制御するために選択した任意のものに関連付けられたものにすることができます。 そのユーザー ID には、サーバーで MQI 呼び出しを発行するために必要な権限が割り振られていなければなりません。 サーバー接続 MCA の権限で MQI 呼び出しを発行するのをクライアントに許可するよりも、代替ユーザー ID を選択することが推奨されています。

表 7. サーバー接続チャネルで使用されるユーザー *ID*

| ユーザー ID | いつ使用するか |
|---|---|
| セキュリティー出口によって設定されるユーザー ID | **CHLAUTH TYPE(BLOCKUSER)** 規則でブロックされない限り使用。 詳しくは、下記の 58 ページの『セキュリティー出口でのユーザー ID の設定』 セクションを参照してください。 |
| CHLAUTH 規則によって設定されるユーザー ID | セキュリティー出口によってオーバーライドされない限り使用。 詳しくは、 チャネル認証レコード を参照してください。 |
| SVRCONN チャネル定義の **MCAUSER** 属性で定義されるユーザー ID | セキュリティー出口または CHLAUTH 規則によってオーバーライドされない限り使用。 |
| クライアント・マシンから流れてくるユーザー ID | これ以外の手段ではユーザー ID が設定されない場合に使用。 |
| サーバー接続チャネルを開始したユーザー ID | これ以外の手段ではユーザー ID が設定されず、クライアント・ユーザー ID が流れてこない場合に使用。 詳しくは、下記の 58 ページの『チャネル・プログラムを実行するユーザー ID』 セクションを参照してください。 |

サーバー接続 MCA はリモート・ユーザーに代わって MQI 呼び出しを発行するため、リモート・クライアントの代わりにサーバー接続 MCA が MQI 呼び出しを発行することによるセキュリティーへの影響や、ユーザーが多くなった場合のアクセスの管理の方法について考慮しておくことは重要です。

- 1 つのアプローチは、サーバー接続の MCA 自体の権限で MQI 呼び出しを発行することです。 しかし、非常に大きなアクセス権限を持つサーバー接続の MCA が、クライアント・ユーザーの代わりに MQI 呼び出しを発行することは、通常望ましくありません。
- 別のアプローチは、クライアントから流れるユーザー ID を使用することです。 サーバー接続の MCA は、このクライアント・ユーザー ID のアクセス権限を使用して MQI 呼び出しを発行できます。 このアプローチの場合、以下に示すいくつかの点を考慮する必要があります。

  1. ユーザー ID は、プラットフォームが異なると形式も異なります。 クライアントのユーザー ID の形式がサーバーで受け入れられる形式と異なる場合に、問題が発生する場合があります。
  2. クライアントの数が多くなる可能性があり、ユーザー ID が異なっていたり、変更されたりする場合もあります。 ID はサーバーで定義され管理される必要があります。
  3. ユーザー ID が信頼できるものかどうかを考慮する必要があります。 クライアントからはどのようなユーザー ID でも送信でき、必ずしもログオン・ユーザーの ID が送信されるとは限りません。 例えば、クライアントは、セキュリティー上の理由で意図的にサーバー上でのみ定義された、mqm の完全な権限を持った ID を送信することができます。

- 推奨されるアプローチは、サーバーでクライアントを識別するトークンを定義して、クライアントに接続するアプリケーションの機能を制限することです。 これを行うには、通常、サーバー接続のチャネル・プロパティーである MCAUSER をクライアントによって使用される特別なユーザー ID 値に設定して、サーバー上で異なるレベルの権限を持つクライアントが使用するための ID をわずかに定義します。

## セキュリティー出口でのユーザー ID の設定

IBM WebSphere MQ MQI クライアントの場合、MQI 呼び出しを発行するプロセスはサーバー接続 MCA です。 サーバー接続の MCA によって使用されるユーザー ID は、MQCD の `MCAUserIdentifier` または `LongMCAUserIdentifier` フィールドに入っています。 これらのフィールドの内容は、以下によって設定されます。

- セキュリティー出口によって設定される任意の値
- クライアントからのユーザー ID
- MCAUSER (サーバー接続チャネル定義内)

セキュリティー出口は、呼び出されるときに表示される値をオーバーライドすることができます。

- サーバー接続チャネル MCAUSER の属性が非ブランクに設定される場合は、MCAUSER 値が使用されます。
- サーバー接続チャネル MCAUSER の属性がブランクの場合は、クライアントから受信されたユーザー ID が使用されます。
- サーバー接続チャネル MCAUSER の属性がブランクであり、クライアントから受信したユーザー ID がない場合は、サーバー接続チャネルを開始したユーザー ID が使用されます。

Windows プラットフォームでは MCAUSER フィールドを 12 文字に制限してください。 これを超える文字は切り捨てられ、許可エラーが発生する可能性があるためです。

クライアント・サイド・セキュリティー出口が使用されている場合は、IBM WebSphere MQ クライアントは、表明されたユーザー ID をサーバーに送信しません。

## チャネル・プログラムを実行するユーザー ID

ユーザー ID フィールドがサーバー接続チャネルを開始したユーザー ID から取得される場合は、以下の値が使用されます。

- z/OS の場合、z/OS 開始済みプロシージャー・テーブルによってチャネル開始プログラムの開始済みタスクに割り当てられたユーザー ID。
- TCP/IP (z/OS 以外) の場合、`inetd.conf` エントリーのユーザー ID、またはリスナーを始動したユーザー ID。
- SNA (z/OS 以外) の場合、SNA サーバーのエントリーか、(それがない場合は) 着信接続要求からのユーザー ID、あるいはリスナーを始動したユーザー ID。
- NetBIOS または SPX の場合、リスナーを始動したユーザー ID。

MCAUSER の属性をブランクに設定しているサーバー接続チャネル定義が存在する場合は、クライアントはこのチャネル定義を使用し、クライアントから提供されたユーザー ID によって決められたアクセス権限で、キュー・マネージャーに接続することができます。 したがって、キュー・マネージャーが実行されているシステムが、無許可のネットワーク接続を許可していると、セキュリティー上の問題 (機密漏れ) が発生する場合があります。 IBM WebSphere MQ デフォルト・サーバー接続チャネル (SYSTEM.DEF.SVRCONN) の MCAUSER 属性がブランクに設定されています。 無許可アクセスを防ぐには、IBM WebSphere MQ MQ オブジェクトにアクセスできないユーザー ID で、デフォルト定義の MCAUSER の属性を更新してください。

## ユーザー ID の大/小文字

`runmqsc` を使用してチャネルを定義すると、MCAUSER の属性は、ユーザー ID が単一引用符で囲まれていない場合に限り、大文字に変更されます。

UNIX、Linux および Windows システム上のサーバーの場合、クライアントから受信した MCAUserIdentifier フィールドの内容は小文字に変換されます。

IBM i サーバーの場合、クライアントから受信した LongMCAUserIdentifier フィールドの内容は大文字に変更されます。

UNIX and Linux システム上のサーバーの場合、クライアントから受信した LongMCAUserIdentifier フィールドの内容は小文字に変換されます。

デフォルトでは、MQ JMS バインディング・アプリケーションが使用されるときに渡されるユーザー ID は、アプリケーションを実行中の JVM のユーザー ID です。

また、createQueueConnection メソッドでユーザー ID を受け渡すこともできます。

# 機密性の計画

データの機密性を保持する方法を計画します。

機密性は、アプリケーション・レベルまたはリンク・レベルで実装できます。 SSL または TLS の使用を選択することもできます。この場合、デジタル証明書の使用を計画する必要があります。 標準の機能が要件を満たさない場合、チャネル出口プログラムを使用することもできます。

**関連概念**

59 ページの『リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーの比較』
このトピックでは、リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーのさまざまな側面について説明し、この 2 つのレベルのセキュリティーを比較します。

64 ページの『チャネル出口プログラム』
チャネル出口プログラム は、MCA の処理シーケンス内で、指定された場所で呼び出されるプログラムです。 ユーザーとベンダーは、独自のチャネル出口プログラムを作成することができます。 いくつかのチャネル出口プログラムが、IBM によって提供されています。

71 ページの『SSL を使用したチャネルの保護』
IBM WebSphere MQ の SSL サポートは、キュー・マネージャー認証情報オブジェクトや、さまざまな MQSC コマンドを使用します。 また、デジタル証明書の使用についても検討する必要があります。

## リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーの比較

このトピックでは、リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーのさまざまな側面について説明し、この 2 つのレベルのセキュリティーを比較します。

リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーを図にまとめたのが、60 ページの図 8 です。

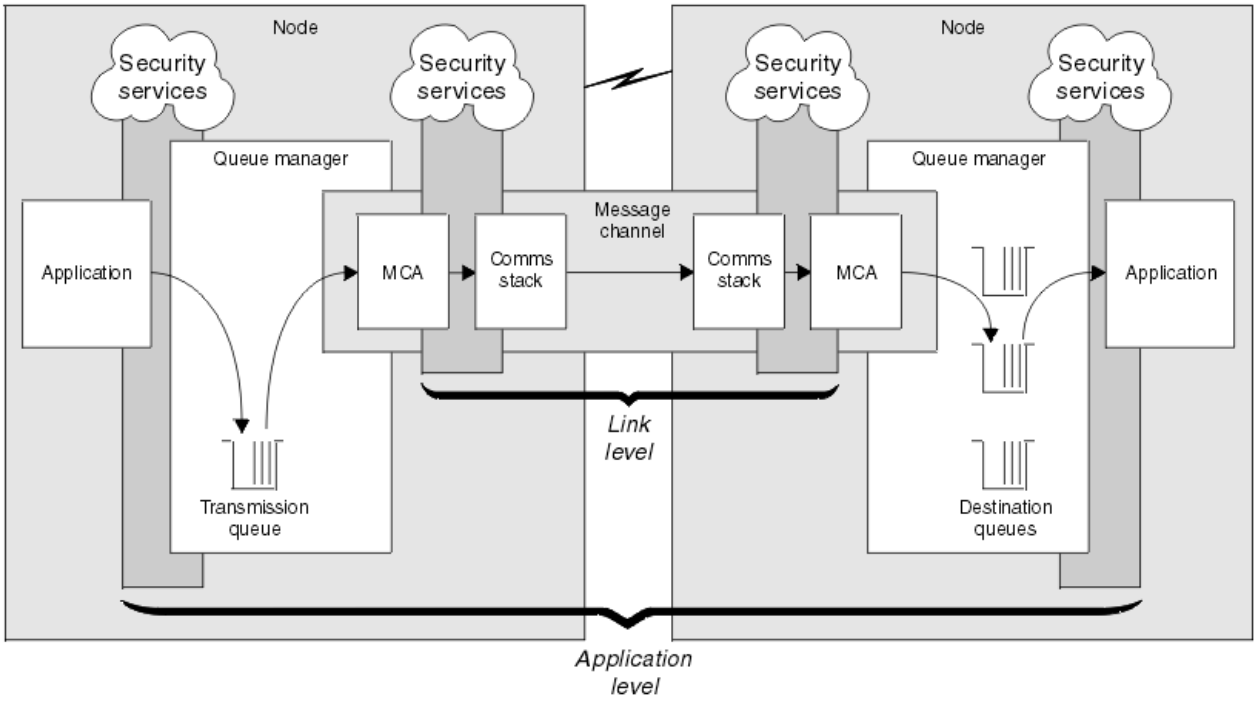



図 *8.* リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティー

## キュー内のメッセージの保護

リンク・レベル・セキュリティーは、メッセージがキュー・マネージャー間で転送されるときにそのメッセージを保護します。 メッセージが無保護ネットワークを介して伝送されるときに、リンク・レベル・セキュリティーは特に重要です。 しかし、メッセージがソース・キュー・マネージャー、宛先キュー・マネージャー、または中間キュー・マネージャーのいずれかのキューに保管されているときは、メッセージを保護できません。

これと比べると、アプリケーション・レベル・セキュリティーは、メッセージがキューに保管されている間もメッセージを保護できます。また、分散キューイングが使用されていないときであっても、アプリケーション・レベル・セキュリティーは適用されます。 この点が、リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーの大きな相違点であり、60 ページの図 8 に示されています。

## 制御されたトラステッド環境で動作していないキュー・マネージャー

キュー・マネージャーが、制御されたトラステッド環境で作動している場合、キューに保管されているメッセージを保護するには、WebSphere MQ によって提供されるアクセス制御メカニズムで十分です。 これは、特に、ローカル・キューイングだけが行われ、メッセージがキュー・マネージャー内にある場合に当てはまります。 この場合、アプリケーション・レベル・セキュリティーは必要ないと考えられます。

また、制御されたトラステッド環境で稼働している別のキュー・マネージャーにメッセージが転送されるか、このようなキュー・マネージャーから受信される場合も、アプリケーション・レベル・セキュリティーは不必要であると考えられます。 制御されたトラステッド環境で稼働していないキュー・マネージャーにメッセージを転送したり、そのようなキュー・マネージャーからメッセージを受信したりする場合は、アプリケーション・レベル・セキュリティーの必要性が大きくなります。

## コストの差

アプリケーション・レベル・セキュリティーは、管理とパフォーマンスの面で、リンク・レベル・セキュリティーよりコストがかかる場合があります。

設定と管理に関する制約が増える可能性が高いので、管理のコストは大きくなると考えられます。 例えば、特定のユーザーが、特定タイプのメッセージだけを送信し、特定のあて先だけにメッセージを送信することを確実に行う必要がある場合があります。 逆に、特定のユーザーが、特定タイプのメッセージだけを受

信し、特定の送信元からだけメッセージを受信することを確実に行う必要がある場合もあります。 1 つのメッセージ・チャネル上でリンク・レベル・セキュリティー・サービスを管理するのではなく、そのチャネル全体でメッセージを交換するすべてのユーザー・ペア用の規則を設定し、維持する必要が生じる場合があります。

アプリケーションがメッセージを書き込んだり取得したりするたびに、セキュリティー・サービスが起動する場合は、パフォーマンスに影響が及ぶ可能性があります。

企業は、リンク・レベル・セキュリティーの方がインプリメントが簡単なので、まず、リンク・レベル・セキュリティーのインプリメントを検討する傾向があります。 リンク・レベル・セキュリティーではすべての要件が満たされないことがわかると、アプリケーション・レベル・セキュリティーのインプリメントが検討されます。

## コンポーネントの可用性

一般に分散環境では、2 つ以上のシステムでセキュリティー・サービスのコンポーネントが必要になります。 例えば、メッセージの暗号化と復号が、別々のシステム上で行われることがあります。 これは、リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティーの両方に当てはまります。

異なるプラットフォームを使用し、それぞれのプラットフォームが別々のレベルのセキュリティー機能を備えている異種環境では、セキュリティー・サービスに必要なコンポーネントが、そのコンポーネントを必要とするすべてのプラットフォームでは入手できない場合があり、また、使いやすい形式では使用できない場合があります。 これは、コンポーネントをさまざまなソースから購入して、独自のアプリケーション・レベル・セキュリティーを提供しようとする場合は特に、リンク・レベル・セキュリティーよりも、アプリケーション・レベル・セキュリティーで起こる問題です。

## 送達不能キュー内のメッセージ

メッセージがアプリケーション・レベル・セキュリティーによって保護されているときに、なんらかの理由により、このメッセージがあて先に到達せず、送達不能キューに入れられる場合は、問題が発生する可能性があります。 メッセージ記述子と送達不能見出し内の情報から、メッセージを処理する方法がわからない場合、アプリケーション・データの内容の検査が必要な場合があります。 アプリケーション・データが暗号化されていて、所定の受信側だけが復号できる場合は、この検査は実行できません。

## アプリケーション・レベル・セキュリティーでは行えないこと

アプリケーション・レベル・セキュリティーは、完全なソリューションではありません。 アプリケーション・レベル・セキュリティーをインプリメントした場合であっても、一部のリンク・レベル・セキュリティー・サービスが引き続き必要な場合があります。 以下に例を示します。

- チャネルが開始するときに、2 つの MCA の相互認証が、引き続き必要な場合があります。 この相互認証は、リンク・レベル・セキュリティー・サービスでしか行えません。
- アプリケーション・レベル・セキュリティーは、組み込みメッセージ記述子を含む、伝送キュー見出し MQXQH を保護できません。 また、メッセージ・データ以外の、WebSphere MQ チャネル・プロトコル・フロー内のデータも保護できません。 この保護を提供できるのは、リンク・レベル・セキュリティーだけです。
- アプリケーション・レベル・セキュリティー・サービスが、MQI チャネルのサーバー側で起動される場合、このサービスは、チャネルを介して送信される MQI 呼び出しのパラメーターを保護できません。 特に、MQPUT、MQPUT1、または MQGET 呼び出し内のアプリケーション・データは、保護されません。 この場合、保護を提供できるのは、リンク・レベル・セキュリティーだけです。

## リンク・レベル・セキュリティー

リンク・レベル・セキュリティー とは、MCA、通信サブシステム、またはその両方の組み合わせによって、直接、または間接に起動されるセキュリティー・サービスを指します。

リンク・レベル・セキュリティーは、60 ページの図 8 に図示されています。

次にリンク・レベル・セキュリティー・サービスの例をいくつか挙げます。

- メッセージ・チャネルの両端にある MCA は、相手側を相互に認証することができます。 この相互の認証は、チャネルが開始し、通信接続が確立された後で、メッセージが流れ始める前に行われます。 どちらかの側で認証が失敗すると、チャネルはクローズされ、メッセージは転送されません。 これは、識別と認証サービスの例です。
- メッセージは、チャネルの送信側で暗号化され、受信側で復号されます。 これは、機密性サービスの例です。
- メッセージがネットワークを介して伝送されていたときに、そのメッセージの内容が意図的に変更されたかどうかを判別するために、チャネルの受信側でそのメッセージをチェックできます。 これは、データ保全性サービスの例です。

## IBM WebSphere MQ によって提供されるリンク・レベル・セキュリティー

IBM WebSphere MQ において機密性とデータ保全性を提供する主要な手段は、SSL または TLS を使用することです。 IBM WebSphere MQ での SSL または TLS の使い方の詳細については、23 ページの『IBM WebSphere MQ での SSL および TLS のサポート』を参照してください。 認証を行うために、IBM WebSphere MQ はチャネル認証レコードを使用する機能を提供します。 チャネル認証レコードは、個々のチャネルまたはチャネル・グループのレベルで、接続システムに付与されているアクセス権限を正確に制御します。 詳しくは、39 ページの『チャネル認証レコード』を参照してください。

### 独自のリンク・レベル・セキュリティーの提供

このトピック集では、独自のリンク・レベル・セキュリティー・サービスを用意する方法を取り上げます。独自のリンク・レベル・セキュリティー・サービスを提供するための主要な方法は、独自のチャネル出口プログラムを作成するというものです。

64 ページの『チャネル出口プログラム』では、チャネル出口プログラムの概要を紹介します。 その同じトピックで、IBM WebSphere MQ for Windows に用意されているチャネル出口プログラム (SSPI チャネル出口プログラム) についても説明します。 このチャネル出口プログラムは、ソース形式で提供されているので、ご自分の要件に合わせてソース・コードを変更することができます。 このチャネル出口プログラム、またはその他のベンダーから入手可能なチャネル出口プログラムがいずれも要件を満たさない場合は、独自のチャネル出口プログラムを設計し、作成することができます。 このトピックでは、チャネル出口プログラムでセキュリティー・サービスを用意する方法に関するヒントを取り上げます。 チャネル出口プログラムの作成方法については、チャネル出口プログラムの作成を参照してください。

### セキュリティー出口を使用したリンク・レベル・セキュリティー

セキュリティー出口は、通常、チャネルの両端に 1 つずつあって、ペアで機能します。 チャネルの始動時に初期のデータ・ネゴシエーションが完了した直後に、セキュリティー出口は呼び出されます。

セキュリティー出口は、識別と認証、アクセス制御、機密性を実装するために使用できます。

### メッセージ出口を使用したリンク・レベル・セキュリティー

メッセージ出口は、メッセージ・チャネル上でのみ使用でき、MQI チャネル上では使用できません。 メッセージ出口は、メッセージ内の伝送キュー見出し MQXQH (組み込みメッセージ記述子を含む) と、アプリケーション・データの両方にアクセスできます。 メッセージ出口は、メッセージの内容を変更し、メッセージの長さを変えることができます。

メッセージ出口は、メッセージの一部へのアクセスではなくメッセージ全体へのアクセスを必要とする任意の目的のために使用できます。

メッセージ出口は、識別と認証、アクセス制御、機密性、データ保全性、否認防止を実装するために使用できますが、セキュリティー以外の理由でも使用できます。

### 送信出口と受信出口を使用したリンク・レベル・セキュリティー

送信出口と受信出口は、メッセージ・チャネルと MQI チャネルの両方で使用できます。 これらの出口は、チャネル上を流れるあらゆるタイプのデータ、および両方向のフローに対して、呼び出されます。

送信出口と受信出口は、各伝送セグメントにアクセスできます。 送信出口と受信出口は、伝送セグメントの内容を変更し、その長さを変えることができます。

メッセージ・チャネル上で、MCA がメッセージを分割し、複数の伝送セグメントで送信する場合、メッセージの各部が入っている伝送セグメントごとに、送信出口が呼び出されます。受信側では、伝送セグメン

トごとに受信出口が呼び出されます。 MQI チャネル上でも、MQI 呼び出しの入力パラメーターまたは出力パラメーターが、1 つのセグメントで送信するには大きすぎる場合、同じことが行われます。

MQI チャネル上で、伝送セグメントのバイト 10 は、MQI 呼び出しを識別し、その伝送セグメントに、その呼び出しの入力パラメーターが入っているのか、出力パラメーターが入っているのかを示します。 送信出口と受信出口は、このバイトを調べると、その MQI 呼び出しに、保護が必要なアプリケーション・データが入っているかどうかを判別することができます。

必要なリソースを取得し、初期化するために、送信出口が初めて呼び出される場合、送信出口は、伝送セグメントを保持する指定量のスペースをバッファー内に予約するように、MCA に依頼することができます。その後、伝送セグメントを処理するために送信出口が呼び出されると、送信出口は、そのスペースを使用して、暗号化された鍵やデジタル署名などを追加できます。 チャネルの相手側にある対応する受信出口は、送信出口によって追加されたデータを除去し、そのデータを伝送セグメントの処理に使用できます。

送信出口と受信出口は、処理対象のデータの構造を理解する必要がなく、したがって、各伝送セグメントをバイナリー・オブジェクトとして処理できるような状況で使用するのが最適です。

送信出口と受信出口は、機密性とデータ保全性を実装するために使用できますが、セキュリティー以外の理由で使用することも可能です。

**関連タスク**

送信または受信出口プログラムでの API 呼び出しの識別

## アプリケーション・レベル・セキュリティー

アプリケーション・レベル・セキュリティー とは、アプリケーションと、そのアプリケーションが接続されているキュー・マネージャーとの間のインターフェースで起動されるセキュリティー・サービスを指します。

これらのサービスは、アプリケーションが、キュー・マネージャーに対する MQI 呼び出しを行うときに起動されます。 このサービスは、アプリケーション、キュー・マネージャー、WebSphere MQ をサポートする別の製品、またはこれらの任意の組み合わせによって、直接または間接に起動されます。 アプリケーション・レベル・セキュリティーは、60 ページの図 8 に図示されています。

アプリケーション・レベル・セキュリティーは、エンドツーエンド・セキュリティー またはメッセージ・レベル・セキュリティー とも呼ばれます。

次にアプリケーション・レベル・セキュリティー・サービスの例をいくつか挙げます。

- アプリケーションがメッセージをキューに入れるときに、メッセージ記述子には、そのアプリケーションに関連したユーザー ID が入ります。 しかし、ユーザー ID の認証に使用できるデータ (例えば、暗号化されたパスワード) はありません。 セキュリティー・サービスは、このデータを追加することができます。メッセージが最終的に受信側アプリケーションによって取り出されるときに、このサービスの別のコンポーネントが、メッセージと一緒に移動したデータを使用して、ユーザー ID を認証することができます。これは、識別と認証サービスの例です。
- メッセージがアプリケーションによってキューに入れられるときに、そのメッセージは暗号化でき、受信側アプリケーションによって取り出されるときに復号できます。これは、機密性サービスの例です。
- メッセージが受信側アプリケーションによって取り出されるときに、そのメッセージを検査することができます。 この検査により、メッセージの内容が、送信側アプリケーションによって最初にキューに入れられた時点以降に意図的に変更されたかどうかを判断します。これは、データ保全性サービスの例です。

### 計画 *Advanced Message Security*

IBM WebSphere MQ Advanced Message Security (AMS) は、別個にライセンス交付される IBM WebSphere MQ のコンポーネントです。これを使用すると、末端のアプリケーションに影響を与えずに、IBM WebSphere MQ ネットワーク経由で流れる機密データを高水準で保護できます。

機密性の高い重要な情報 (特に患者記録などの内密情報や、クレジット・カード詳細などの支払い関連情報) をやり取りする場合には、機密保護に特別の注意を払う必要があります。 企業内でやり取りされる情報の保全性を確実に維持して無許可アクセスから保護することは、常に課題であり、重要な責任です。 さらに、多くの場合、セキュリティー上の規定に従う必要があり、これに違反すると罰則が適用される恐れがあります。

IBM WebSphere MQ のセキュリティー拡張を独自に開発することができます。 ただし、そのようなソリューションは専門的な技術を必要とし、保守が複雑で多大なコストがかかる可能性があります。 IBM WebSphere MQ Advanced Message Security は、企業において実質的にあらゆる種類の商用 IT システム間で情報をやり取りする際にこのような課題に取り組むうえで役立ちます。

IBM WebSphere MQ Advanced Message Security は、IBM WebSphere MQ のセキュリティー機能を次のように拡張します。

- メッセージの暗号化またはデジタル署名を使用して、Point-to-Point メッセージング・インフラストラクチャー向けに、アプリケーション・レベルのエンドツーエンド・データ保護を提供します。
- 複雑なセキュリティー・コードを作成したり、既存のアプリケーションを変更/再コンパイルしたりしなくても、総合的なセキュリティーが提供されます。
- Public Key Infrastructure (PKI) テクノロジーを使用して、メッセージの認証、許可、機密性、およびデータ保全性のサービスを提供します。
- メインフレームおよび分散サーバーに関するセキュリティー・ポリシーの管理が可能です。
- IBM WebSphere MQ サーバーとクライアントをどちらもサポートします。
- IBM WebSphere MQ Managed File Transfer と統合して、エンドツーエンドの保護されたメッセージング・ソリューションを提供します。

詳細については、272 ページの『IBM WebSphere MQ Advanced Message Security』 を参照してください。

### 独自のアプリケーション・レベル・セキュリティーの提供

このトピック集では、独自のアプリケーション・レベル・セキュリティー・サービスを用意する方法を取り上げます。

アプリケーション・レベルのセキュリティーの実装に役立つように、IBM WebSphere MQ には、API 出口と API 交差出口という 2 つの出口が用意されています。

これらの出口によって、識別と認証、アクセス制御、機密性、データ保全性、否認防止のサービスを用意できますが、セキュリティーとは無関係の機能を用意することも可能です。

API 出口または API 交差出口が、ご使用のシステム環境でサポートされない場合、独自のアプリケーション・レベル・セキュリティーを提供する別の方法を検討する必要が生じます。 1 つの方法は、MQI をカプセル化する、上位レベルの API を開発することです。 プログラマーは、MQI の代わりにこの API を使用して、IBM WebSphere MQ アプリケーションを作成します。

上位レベルの API を使用する最も一般的な理由は、次のとおりです。

- MQI の拡張機能をプログラマーから見えないようにする。
- MQI 使用の標準を実施する。
- MQI に機能を追加する。 この追加機能は、セキュリティー・サービスにすることができます。

一部のベンダーの製品では、この手法を使用して、IBM WebSphere MQ 用のアプリケーション・レベル・セキュリティーを提供します。

この方法でセキュリティー・サービスを提供する計画の場合は、データ変換について、次の項目に注意してください。

- セキュリティー・トークン (例えば、デジタル署名) がメッセージ内のアプリケーション・データに追加された場合、データ変換を実行する任意のコードは、このトークンの存在を認識する必要があります。
- セキュリティー・トークンは、アプリケーション・データのバイナリー・イメージから得られた可能性があります。 したがって、トークンの検査はすべて、データの変換前に実行する必要があります。
- メッセージ内のアプリケーション・データが暗号化された場合、そのデータはデータの変換前に復号する必要があります。

## チャネル出口プログラム

チャネル出口プログラム は、MCA の処理シーケンス内で、指定された場所で呼び出されるプログラムです。 ユーザーとベンダーは、独自のチャネル出口プログラムを作成することができます。 いくつかのチャネル出口プログラムが、IBM によって提供されています。

チャネル出口プログラムにはいくつかのタイプがありますが、リンク・レベル・セキュリティーを提供する役割を持つのは、次の 4 つだけです。

- セキュリティー出口
- メッセージ出口
- 送信出口
- 受信出口

この 4 つのタイプのチャネル出口プログラムを図にまとめたのが、65 ページの図 9 です。以下の各トピックでは、その 4 つのタイプのチャネル出口プログラムを取り上げます。

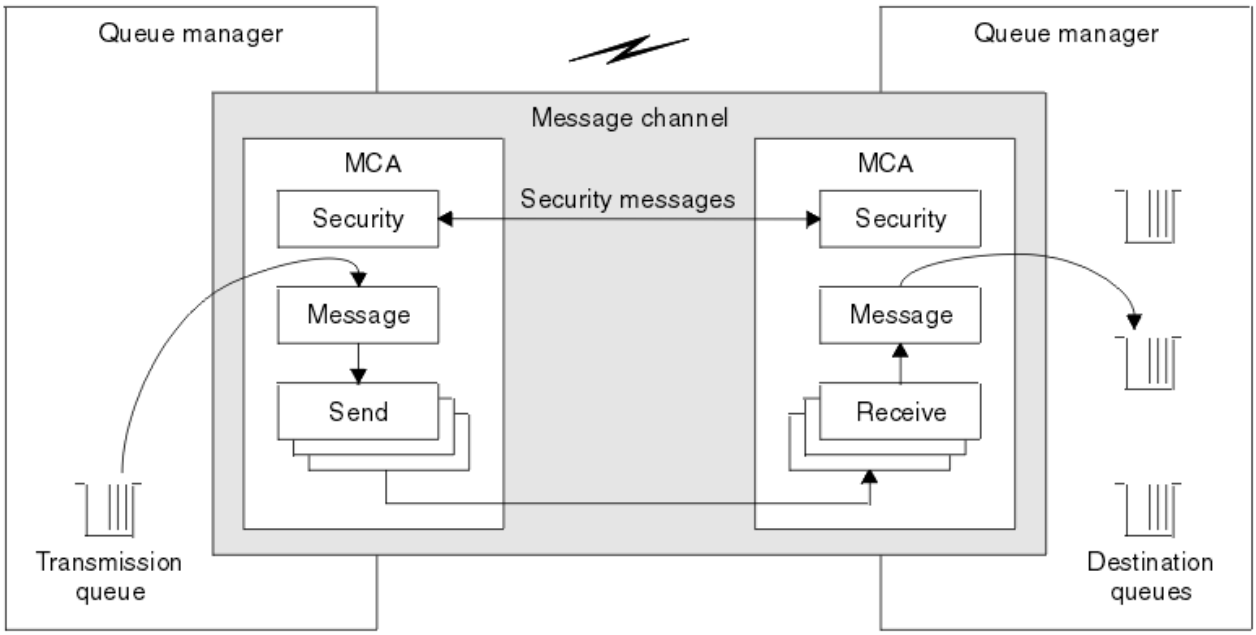



図 9. メッセージ・チャネル上のセキュリティー出口、メッセージ出口、送信出口、および受信出口

**関連概念**

メッセージング・チャネルのためのチャネル出口プログラム

## セキュリティー出口の概要

通常、セキュリティー出口は、ペアで使用します。 セキュリティー出口を呼び出すのは、メッセージ・フローの前です。目的は、 MCA がパートナーを認証できるようにすることです。

セキュリティー出口 は、通常、チャネルの両端に 1 つずつあって、ペアで機能します。 チャネルの始動時に初期のデータ・ネゴシエーションが完了した直後から、メッセージが流れ始めるまでの間に、セキュリティー出口は呼び出されます。 セキュリティー出口の主な目的は、チャネルの両端にある MCA が、相手側の MCA を認証できるようにすることです。 ただし、セキュリティー出口がその他の機能 (セキュリティーには無関係な機能であっても) を実行することを妨げるものはありません。

セキュリティー出口は、セキュリティー・メッセージ を送信することによって、互いに情報を交換することができます。 セキュリティー・メッセージのフォーマットは定義されていないため、ユーザーが決定します。 セキュリティー・メッセージの交換で起こり得る結果の 1 つは、セキュリティー出口のいずれかが、処理を続行しないことを決定することです。 その場合、チャネルはクローズされ、メッセージは流れません。 チャネルの一方の側だけにセキュリティー出口がある場合であっても、その出口は呼び出され、チャネルを続行するか、クローズするかを選択できます。

セキュリティー出口は、メッセージ・チャネルと MQI チャネルの両方で呼び出すことができます。 セキュリティー出口の名前は、チャネルの両端のチャネル定義で、パラメーターとして指定されます。

このセキュリティー出口について詳しくは、62 ページの『セキュリティー出口を使用したリンク・レベル・セキュリティー』 を参照してください。

## メッセージ出口

メッセージ出口は、メッセージ・チャネルでのみ動作し、通常はペアで機能します。 メッセージ出口は、メッセージ全体で動作し、メッセージ全体に対してさまざまな変更を加えることができます。

チャネルの送信側と受信側にあるメッセージ出口 は、通常、ペアで機能します。 チャネルの送信側にあるメッセージ出口は、MCA が伝送キューからメッセージを受け取った後で呼び出されます。 チャネルの受信側では、MCA が宛先キューにメッセージを入れる前に、メッセージ出口が呼び出されます。

メッセージ出口は、伝送キュー見出し MQXQH (組み込みメッセージ記述子が組み込まれている) と、メッセージ内のアプリケーション・データの両方にアクセスできます。 メッセージ出口は、メッセージの内容を変更し、メッセージの長さを変えることができます。 長さの変更は、メッセージの圧縮、圧縮解除、暗号化、または復号の結果です。 また、メッセージにデータを追加したり、メッセージからデータを削除した結果、長さが変わる場合もあります。

メッセージ出口は、メッセージの一部へのアクセスではなくメッセージ全体へのアクセスを必要とする任意の目的 (必ずしも、セキュリティーのためとは限らない) に使用できます。

メッセージ出口は、現在処理しているメッセージが、その宛先に向かってそれ以上進むべきではないことを決定できます。 この場合、MCA はそのメッセージを送達不能キューに入れます。 また、メッセージ出口は、チャネルを閉じることもできます。

メッセージ出口は、メッセージ・チャネル上でしか呼び出すことができず、MQI チャネル上では呼び出すことができません。 これは、MQI チャネルの目的が、MQI 呼び出しの入出力パラメーターが、IBM WebSphere MQ MQI クライアント・アプリケーションとキュー・マネージャーとの間で流れることを可能にすることであるからです。

メッセージ出口の名前は、チャネルの両端のチャネル定義で、パラメーターとして指定されます。 また、連続して実行されるメッセージ出口のリストを指定することもできます。

このメッセージ出口について詳しくは、62 ページの『メッセージ出口を使用したリンク・レベル・セキュリティー』 を参照してください。

## 送信出口と受信出口

通常、送信出口と受信出口は、ペアで使用します。 作動対象は伝送セグメントです。処理対象のデータの構造が重要な意味を持たない状況で使用するのがベストです。

チャネルの一方の側にある送信出口 と、もう一方の側にある受信出口 は、通常、ペアで機能します。 送信出口は、MCA が communications send を発行して、通信接続を介してデータを送信する直前に呼び出されます。 受信出口は、MCA が communications receive の後に制御を取り戻し、通信接続からデータを受信した直後に、呼び出されます。 MQI チャネルを通じての共有会話が使用中なら、各会話ごとに、送受信出口の異なるインスタンスが呼び出されます。

メッセージ・チャネル上の 2 つの MCA 間の IBM WebSphere MQ チャネル・プロトコル・フローには、メッセージ・データとともに、制御情報が入っています。 同様に、MQI チャネル上のフローには、MQI 呼び出しのパラメーターとともに、制御情報が入っています。 送信出口と受信出口は、すべてのタイプのデータに対して呼び出されます。

メッセージ・チャネル上では、メッセージ・データは一方向のみに流れますが、MQI チャネル上では、MQI 呼び出しの入力パラメーターが 1 つの方向に流れると、出力パラメーターは、逆の方向に流れます。 メッセージ・チャネルと MQI チャネルの両方で、制御情報は両方向に流れます。 その結果、送信出口と受信出口は、チャネルの両端で呼び出されることが可能です。

2 つの MCA 間の 1 つのフローで伝送されるデータの単位は、伝送セグメント と呼ばれます。 送信出口と受信出口は、各伝送セグメントにアクセスできます。 送信出口と受信出口は、伝送セグメントの内容を変更し、その長さを変えることができます。 ただし、送信出口で伝送セグメントの先頭の 8 バイトを変更することはできません。 その 8 バイトは、IBM WebSphere MQ チャネル・プロトコルのヘッダーの一部です。 また、送信出口が伝送セグメントの長さを増やすことができる量にも制限があります。 特に、送信出口は、チャネルの始動時に 2 つの MCA 間でネゴシエーションされた最大の長さ以上に、伝送セグメントを長くすることはできません。

メッセージ・チャネル上で、メッセージが大きすぎて、1 つの伝送セグメントで送信できない場合、送信側の MCA は、メッセージを分割して、複数の伝送セグメントとしてメッセージを送信します。 その結果、送信出口は、メッセージの一部が入っている伝送セグメントごとに呼び出され、受信側では、受信出口が伝

送セグメントごとに呼び出されます。 伝送セグメントが受信出口によって処理された後、受信側 MCA は、伝送セグメントからメッセージを再構成します。

同様に MQI チャネル上でも、MQI 呼び出しの入力パラメーターまたは出力パラメーターが大きすぎる場合、複数の伝送セグメントとして送信されます。 これは、例えば、アプリケーション・データが大きい場合に MQPUT、MQPUT1、または MQGET 呼び出しで行われることがあります。

上記の考慮事項を考慮に入れると、処理しようとするデータの構造を理解する必要がなく、したがって、各伝送セグメントをバイナリー・オブジェクトとして扱うことができるような目的に、送信出口と受信出口を使用する方が妥当であるといえます。

送信出口または受信出口でチャネルを閉じることもできます。

送信出口と受信出口の名前は、チャネルの両端のチャネル定義で、パラメーターとして指定されます。 また、連続して実行される送信出口のリストを指定することもできます。 同様に、受信出口のリストも指定することができます。

この送信出口または受信出口について詳しくは、62 ページの『送信出口と受信出口を使用したリンク・レベル・セキュリティー』 を参照してください。

## データ保全性の計画

データ保全性を保持する方法を計画します。

データ保全性は、アプリケーション・レベルまたはリンク・レベルで実装できます。

アプリケーション・レベルでは、IBM WebSphere MQ Advanced Message Security を使用してメッセージにデジタル署名し、許可されていない変更から保護します。 標準の機能が要件を満たさない場合、API 出口プログラムを使用することもできます。

リンク・レベルでは、SSL または TLS の使用を選択することもできます。 この場合、デジタル証明書の使用を計画する必要があります。 標準の機能が要件を満たさない場合、チャネル出口プログラムを使用することもできます。

**関連概念**

71 ページの『SSL を使用したチャネルの保護』
IBM WebSphere MQ の SSL サポートは、キュー・マネージャー認証情報オブジェクトや、さまざまな MQSC コマンドを使用します。 また、デジタル証明書の使用についても検討する必要があります。

22 ページの『IBM WebSphere MQ でのデータ保全性』
データ保全性サービスを使用して、メッセージが変更されたかどうかを検出できます。

63 ページの『計画 Advanced Message Security』
IBM WebSphere MQ Advanced Message Security (AMS) は、別個にライセンス交付される IBM WebSphere MQ のコンポーネントです。これを使用すると、末端のアプリケーションに影響を与えずに、IBM WebSphere MQ ネットワーク経由で流れる機密データを高水準で保護できます。

**関連資料**

API 出口参照

チャネル出口呼び出しおよびデータ構造体

## 監査の計画

どのデータを監査する必要があるか、どのように監査情報の収集と処理を行うか決めます。 システムが正しく構成されているかチェックする方法を考慮します。

アクティビティー・モニターには、いくつかの面があります。 考慮しなければならない面はしばしば監査員要件により定義され、これらの要件はしばしば、HIPAA (医療保険の積算と責任に関する法律) または SOX (サーベンス・オクスリー) などの規制基準により主導されます。 IBM WebSphere MQ は、それらの基準に準拠するのに役立つように意図されたフィーチャーを提供します。

例外だけに関心があるのか、システムのすべての振る舞いに関心があるのかを考慮します。

監査のいくつかの面は、運用のモニターとしても考慮できます。この監査に関する違いの 1 つは、リアルタイム・アラートだけを見るのではなく、しばしば履歴データを見るということです。 モニターについては、「モニターおよびパフォーマンス」セクションで説明しています。

### どのデータを監査するか

次のセクションで説明されているようにして、どのタイプのデータまたはアクティビティーを監査する必要があるか考慮します。

**IBM WebSphere MQ インターフェースを使用して IBM WebSphere MQ に行われた変更**
装備イベント、特にコマンド・イベントおよび構成イベントを発行するように IBM WebSphere MQ を構成します。

**IBM WebSphere MQ に、制御外で行われた変更**
変更によっては、IBM WebSphere MQ の動作に影響を及ぼす可能性がありますが、IBM WebSphere MQ によって直接モニターすることはできません。 このような変更の例としては、構成ファイル `mqs.ini`、`qm.ini`、および `mqclient.ini` への変更、キュー・マネージャーの作成と削除、バイナリー・ファイルのインストール (ユーザー出口プログラムなど)、およびファイル許可の変更などがあります。 これらのアクティビティーをモニターするには、オペレーティング・システムのレベルで実行するツールを使用しなければなりません。 異なるオペレーティング・システムには、異なるツールが使用可能であり適切です。 *sudo* などの関連ツールにより作成されるログもあるかもしれません。

**IBM WebSphere MQ の操作制御**
キュー・マネージャーの始動や停止などのアクティビティーを監査するには、オペレーティング・システム・ツールを使用しなければならないかもしれません。 場合によっては、IBM WebSphere MQ を装備イベントを発行するように構成できます。

**IBM WebSphere MQ 内のアプリケーション・アクティビティー**
アプリケーションのアクション (例えば、キューのオープンやメッセージの取得) を監査するには、適切なイベントを発行するように IBM WebSphere MQ を構成します。

**侵入者アラート**
セキュリティー突破の試みを監査するには、許可イベントを発行するようにシステムを構成します。 チャネル・イベントも、特に予期しないチャネル終了の場合に、アクティビティーを表示する上で役に立ちます。

### 監査データの収集、表示、保存の計画

必要な要素の多くは、IBM WebSphere MQ イベント・メッセージとして報告されます。 これらのメッセージを読み取り、形式化できるツールを選択しなければなりません。 長期保管や分析に関心がある場合、データベースなどの補助ストレージ・メカニズムにそれらを移動しなければなりません。 これらのメッセージを処理しない場合、イベント・キューに残ったままになり、キューが満杯になるかもしれません。 なんらかのイベントに基づいて自動的にアクションを取る (例えば、セキュリティー障害が発生した際にアラートを発行する) ツールを実装することに決定する場合があるかもしれません。

### システムが正しく構成されているか検証する

IBM WebSphere MQ Explorer では、テストのセットが供給されています。 これらを使用して、問題になっているオブジェクト定義をチェックします。

また、システム構成が予期したとおりであるか、定期的にチェックしてください。 何かが変更された時にコマンドと構成イベントを報告することはできますが、構成をダンプし、既知の健全な構成のコピーと比較することも役に立ちます。

## トポロジーによるセキュリティーの計画

このセクションでは、特定の状況、つまり、チャネル、キュー・マネージャー・クラスター、パブリッシュ/サブスクライブ・アプリケーション、マルチキャスト・アプリケーション、およびファイアウォール使用時におけるセキュリティーについて説明します。

詳しくは、以下のサブトピックを参照してください。

## チャネル許可

チャネルを介してメッセージを送信または受信するときは、さまざまな IBM WebSphere MQ リソースに対するアクセス権限を持つユーザー ID が必要です。

MCA の PUT 時にメッセージを受信するときには、MCA に関連付けられたユーザー ID、またはメッセージに関連付けられたユーザー ID のいずれかを使用できます。

CONNECT 時に、**CHLAUTH** チャネル認証レコードを使用して、表明されたユーザー ID を代替ユーザーにマップできます。

WebSphere MQ では、SSL または TLS サポートを使用してチャネルを保護することができます。

MCAUSER 属性が使用されていない送信側チャネルを除く、送信側チャネルおよび受信側チャネルに関連付けられたユーザー ID は、以下のリソースに対するアクセス権限を必要とします。

- 送信側チャネルに関連付けられたユーザー ID は、キュー・マネージャー、伝送キュー、送達不能キューに対するアクセス権限と、チャネル出口が必要とするその他のすべてのリソースに対するアクセス権限を必要とします。
- 受信側チャネルの MCAUSER ユーザー ID は、*+setall* 権限を必要とします。

  その理由は、受信側チャネルは、リモート送信側チャネルから受信したデータを使用して、すべてのコンテキスト・フィールドを含む完全な MQMD を作成する必要があるからです。

  したがって、キュー・マネージャーは、このアクティビティーを実行するユーザーに *+setall* 権限があることを必要とします。 この *+setall* 権限を、以下のユーザーに付与しなければなりません。

  – 受信側チャネルがメッセージを有効に書き込むすべてのキュー。
  – キュー・マネージャー・オブジェクト。 詳細については、コンテキストについての許可を参照してください。
- 発信元が COA レポート・メッセージを要求した受信側チャネルの MCAUSER ユーザー ID には、レポート・メッセージを返す伝送キューの *+passid* 権限が必要です。 この権限がない場合、AMQ8077 エラー・メッセージがログに記録されます。
- 受信側チャネルに関連付けられたユーザー ID で、ターゲット・キューを開いてキューにメッセージを書き込むことができます。

  これにはメッセージ・キューイング・インターフェース (MQI) が関係するため、WebSphere MQ オブジェクト権限マネージャー (OAM) を使用していない場合は、追加のアクセス制御検査を行わなければならない場合があります。 許可検査を、MCA に関連付けられたユーザー ID に対して行うか (このトピックに記載されている方法)、それともメッセージに関連付けられたユーザー ID に対して行うか (MQMD の UserIdentifier フィールドで指定) を指定できます。

  チャネル定義の **PUTAUT** パラメーターが適用されるチャネル・タイプの場合、これらの検査で使用されるユーザー ID は、このパラメーターで指定されます。

  – チャネルはデフォルトではキュー・マネージャーのサービス・アカウントを使用します。 このアカウントには全管理権限があり、特殊権限は不要です。

    サーバー接続チャネルの場合、デフォルトでは管理接続は CHLAUTH 規則によってブロックされるので、明示的なプロビジョニングを必要とします。

    管理者がこのアクセスを制限するステップを行っていなければ、受信側、要求側、クラスター受信側タイプのチャネルを、隣接するキュー・マネージャーによってローカル管理できます。
  – WebSphere 管理特権がないユーザー ID を使用している場合、チャネルが機能するためには、チャネルに関する dsp 権限と ctrlx 権限をそのユーザー ID に付与する必要があります。 SDR チャネル・タイプには MCAUSER 属性は使用されません。
  – メッセージに関連付けられたユーザー ID を使用する場合、ユーザー ID はリモート・システムからのものである可能性があります。

    このリモート・システムのユーザー ID は、ターゲット・システムで認識されなければなりません。 例えば、以下のコマンドを発行します。

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect +inq +setall
```

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

ここで、*Profile* はチャネルです。

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

ここで、*Profile* は送達不能キューです (設定されている場合)。

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

ここで、*Profile* は許可されたキューのリストです。

**重要 :** コマンド・キューや他の機密性の高いシステム・キューにメッセージを挿入する許可をユーザー ID に与える際には注意が必要です。

MCA に関連付けられるユーザー ID は、MCA のタイプによって異なります。 MCA には、次の 2 つのタイプがあります。

**呼び出し側 MCA**
チャネルを開始する MCA。 呼び出し側 MCA は、個々のプロセスとして、チャネル・イニシエーターのスレッドとして、あるいはプロセス・プールのスレッドとして開始できます。 使用されるユーザー ID は、親プロセス (チャネル・イニシエーター) に関連付けられたユーザー ID、または MCA を開始するプロセスに関連付けられたユーザー ID です。

**応答側 MCA**
応答側 MCA は、呼び出し側 MCA による要求の結果として開始される MCA です。 応答側 MCA は、個々のプロセスとして、リスナーのスレッドとして、あるいはプロセス・プールのスレッドとして開始できます。 このユーザー ID は、以下のいずれかのタイプにすることができます (この優先順位で設定します)。

1. APPC では、呼び出し側 MCA は、応答側 MCA に使用するユーザー ID を指定できます。 これはネットワーク・ユーザー ID によって呼び出され、個々のプロセスとして開始したチャネルにのみ適用されます。 チャネル定義の USERID パラメーターを使用してネットワーク・ユーザー ID を設定します。
2. **USERID** パラメーターが使用されない場合は、MCA が使用しなければならないユーザー ID を応答側 MCA のチャネル定義で指定できます。 チャネル定義の **MCAUSER** パラメーターを使用してユーザー ID を設定します。
3. この (2 つの) いずれの方法でもユーザー ID が設定されていない場合は、MCA を開始するプロセスのユーザー ID、または親プロセス (リスナー) のユーザー ID が使用されます。

**関連概念**
39 ページの『チャネル認証レコード』
チャネル認証レコードを使用すれば、接続システムに与えるアクセス権限をチャネル・レベルで細かく制御できるようになります。

チャネル認証レコード・プロパティー

## チャネル・イニシエーター定義の保護
チャネル・イニシエーターを操作できるのは、mqm グループのメンバーに限られます。

IBM WebSphere MQ チャネル・イニシエーターは IBM WebSphere MQ オブジェクトではないため、それらへのアクセスは OAM によって制御されません。 IBM WebSphere MQ では、ユーザーまたはアプリケーションのユーザー ID が mqm グループのメンバーでない限り、ユーザーまたはアプリケーションがそれらのオブジェクトを操作することはできません。 PCF コマンド `StartChannelInitiator` を発行するアプリケーションがある場合、PCF メッセージのメッセージ記述子で指定したユーザー ID は、ターゲット・キュー・マネージャーの mqm グループのメンバーである必要があります。

エスケープ PCF コマンドや間接モードの `runmqsc` を使用して同等の MQSC コマンドを発行するには、ユーザー ID は宛先マシンでも mqm グループのメンバーでなければなりません。

## 伝送キュー

キュー・マネージャーは、伝送キューにリモート・メッセージを自動的に書き込みます。 これには特別の権限は必要ありません。

ただし、メッセージを伝送キューに直接書き込む必要がある場合は、特別な権限が必要です。90 ページの表 10 を参照してください。

## チャネル出口

チャネル認証レコードが適切でない場合、追加されたセキュリティーのためにチャネル出口を使用することができます。 セキュリティー出口は、2 つのセキュリティー出口プログラムの間のセキュア接続を形成します。 一方のプログラムは、送信側メッセージ・チャネル・エージェント (MCA) 用で、もう一方は受信側 MCA 用です。

チャネル出口についての詳細は、64 ページの『チャネル出口プログラム』を参照してください。

## *SSL* を使用したチャネルの保護

IBM WebSphere MQ の SSL サポートは、キュー・マネージャー認証情報オブジェクトや、さまざまな MQSC コマンドを使用します。 また、デジタル証明書の使用についても検討する必要があります。

## SSL サポート用のコマンドおよび属性

Secure Sockets Layer (SSL) プロトコルには、盗聴、改ざん、偽名の使用から保護するためのチャネル・セキュリティーが用意されています。 IBM WebSphere MQ の SSL サポートにより、チャネル定義で特定のチャネルが SSL セキュリティーを使用することを指定できます。 また、使用する暗号化アルゴリズムなど、望ましいセキュリティーのタイプを詳しく指定することもできます。

以下の MQSC コマンドは SSL をサポートしています。

**ALTER AUTHINFO**
認証情報オブジェクトの属性を変更します。

**DEFINE AUTHINFO**
認証情報オブジェクトを作成します。

**DELETE AUTHINFO**
認証情報オブジェクトを削除します。

**DISPLAY AUTHINFO**
特定の認証情報オブジェクトの属性を表示します。

以下のキュー・マネージャーのパラメーターは SSL をサポートしています。

**SSLCRLNL**
SSLCRLNL 属性は、証明書取り消し場所を提供して、拡張 TLS/SSL 証明書検査を実行できるようにするために使用される、 認証情報オブジェクトの名前リストを指定します。

**SSLCRYP**
Windows、 UNIX and Linux システムでは、 SSLCryptoHardware キュー・マネージャー属性を設定します。 この属性は、システムに存在する暗号ハードウェアを構成するときに使用できる、パラメーター・ストリングの名前です。

**SSLEV**
SSL を使用しているチャネルが SSL 接続の確立に失敗した場合に SSL イベント・メッセージを報告するかどうかを決定します。

**SSLFIPS**
暗号ハードウェアではなく IBM WebSphere MQ で暗号化を実行する場合に、FIPS 認証アルゴリズムのみを使用するかどうかを指定します。 暗号ハードウェアが構成されている場合、ハードウェア製品で提供される暗号モジュールが使用されます。これらのモジュールは、特定のレベルの FIPS 認定を受けている場合があります。 これは、使用されているハードウェア製品によって異なります。

**SSLKEYR**
Windows、UNIX and Linux システムでは、キー・リポジトリーをキュー・マネージャーに関連付けます。 キー・データベースは *GSKit* キー・データベースに入れられています。 (IBM Global Security Kit

(GSKit) を使用すると、Windows、UNIX and Linux システムで SSL セキュリティーを使用できるようになります。)

**SSLRKEYC**
秘密鍵を再ネゴシエーションする前に SSL 会話内で送受信されるバイト数。 このバイト数には、MCA によって送信される制御情報が含まれます。

以下のチャネル・パラメーターは SSL をサポートしています。

**SSLCAUTH**
IBM WebSphere MQ が SSL クライアントからの証明書を必要としているかどうか、そしてそれを検査するかどうかを定義します。

**SSLCIPH**
暗号化の強力度と機能を指定します (CipherSpec)。例えば、NULL_MD5 や RC4_MD5_US。 CipherSpec は、チャネルの両端で一致していなければなりません。

**SSLPEER**
許可されたパートナーの識別名 (固有の ID) を指定します。

このセクションでは、認証情報オブジェクトをサポートする `setmqaut`、`dspmqaut`、`dmpmqaut`、`rcrmqobj`、`rcdmqimg`、および `dspmqfls` の各コマンドについて説明します。 また、 UNIX and Linux システムで証明書を管理するための iKeycmd コマンド、および UNIX、 Linux 、および Windows システムで証明書を管理するための runmqakm ツールについても説明します。 以下のセクションを参照してください。

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- 鍵と証明書の管理

SSL を使用したチャネル・セキュリティーの概要については、以下を参照してください。

- 23 ページの『IBM WebSphere MQ での SSL および TLS のサポート』

SSL に関連した MQSC コマンドの詳細については、以下を参照してください。

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

SSL に関連した PCF コマンドの詳細については、以下を参照してください。

- Change Authentication Information Object、Copy Authentication Information Object、および Create Authentication Information Object
- Delete Authentication Information Object
- Inquire Authentication Information Object

## 自己署名証明書と CA 署名証明書

アプリケーションの開発およびテストを行う間と、実動環境で使用する場合の両方で、デジタル証明書の使用法を計画することは重要です。 キュー・マネージャーとクライアント・アプリケーションの使用法に応じて、CA 署名証明書か自己署名証明書を使用できます。

**CA 署名証明書**
実動システムの場合、信頼できる認証局 (CA) から証明書を取得します。 外部 CA から証明書を取得する場合、そのサービスの料金を支払います。

**自己署名証明書**

アプリケーションの開発中には、プラットフォームに応じて自己署名証明書かローカル CA で発行された証明書を使用できます。

Windows、 UNIX、および Linux システムでは、自己署名証明書を使用できます。 説明は、122 ページの『UNIX, Linux, and Windows システムでの自己署名個人証明書の作成』を参照してください。

自己署名証明書は、以下の理由で実動環境での使用には適切ではありません。

- 自己署名証明書は、取り消すことができません。したがって、アタッカーが秘密鍵を不正に取得してしまうと、身分を偽って勝手に操作を実行する、という事態が発生しかねません。 一方、CA は、暗号の漏えいが発生した証明書を取り消して、その証明書がそれ以上使用される事態を防止できます。 したがって、実稼働環境では、CA 署名証明書を使用するほうが安全です。一方テスト・システムでは、自己署名証明書を使用するほうが便利です。
- 自己署名証明書は、期限が切れることがありません。 これはテスト環境では便利で安全ですが、実稼働環境では最終的にセキュリティー・ブリーチ (抜け穴) につながります。 自己署名証明書は取り消せないので、リスクがさらに大きくなります。
- 自己署名証明書は、個人証明書として使用したり、ルート (トラスト・アンカー) CA 証明書として使用したりします。 自己署名の個人証明書があるユーザーは、この証明書を使用して他の個人証明書に署名することもできます。 一般的にこのような署名は、CA で発行された個人証明書では行うことができず、重大な機密漏れが生じることを示しています。

## CipherSpec およびデジタル証明書

サポートされている CipherSpec のサブセットのみが、サポートされているすべてのタイプのデジタル証明書で使用可能です。 そのため、使用するデジタル証明書に適した CipherSpec を選択する必要があります。同様に、組織のセキュリティー・ポリシーで、特定の CipherSpec の使用が求められている場合は、適切なデジタル証明書を取得しなければなりません。

CipherSpec とデジタル証明書の関係について詳しくは、34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』を参照してください。

## 証明書妥当性検査ポリシー

IETF RFC 5280 標準には一連の証明書妥当性検査のルールが規定されており、偽名攻撃を予防するために準拠アプリケーション・ソフトウェアはこのルールを実装する必要があります。 証明書妥当性検査ルール一式は、証明書妥当性検査ポリシーとして知られています。 WebSphere MQ での証明書妥当性検査ポリシーの詳細については、33 ページの『IBM WebSphere MQ での証明書妥当性検査ポリシー』を参照してください。

## *SNA LU 6.2* セキュリティー・サービス

SNA LU 6.2 には、セッション・レベルの暗号化、セッション・レベルの認証、会話レベルの認証の機能が用意されています。

**注 :** このトピック集は、システム・ネットワーク体系 (SNA) に関する基本的な理解を前提としています。このセクションで参照される他の資料には、関係する概念と用語が簡単に紹介されています。 さらに包括的で技術的な SNA の紹介が必要な場合は、「*Systems Network Architecture Technical Overview*」、GC30-3073 を参照してください。

SNA LU 6.2 は、次の 3 つのセキュリティー・サービスを提供します。

- セッション・レベルの暗号化方式
- セッション・レベルの認証
- 会話レベルの認証

セッション・レベルの暗号化方式とセッション・レベルの認証の場合、SNA は *Data Encryption Standard (DES)* アルゴリズムを使用します。 DES アルゴリズムは、ブロック暗号化アルゴリズムであり、データの暗号化と復号に対称鍵を使用します。 ブロックと鍵のどちらも、長さは 8 バイトです。

### セッション・レベルの暗号化方式

セッション・レベルの暗号化方式 は、DES アルゴリズムを使用してセッション・データの暗号化と復号を行います。 したがって、SNA LU 6.2 チャネル上でリンク・レベルの機密性サービスを提供するのに使用できます。

論理装置 (LU) は、強制 (または必須) データ暗号方式、選択可能データ暗号方式、またはデータ暗号方式なしを提供できます。

強制暗号セッション では、LU は、すべてのアウトバウンド・データ要求単位を暗号化し、すべてのインバウンド・データ要求単位を復号します。

選択可能暗号セッション では、LU は、送信側のトランザクション・プログラム (TP) によって指定されたデータ要求単位だけを暗号化します。 送信側 LU は、要求見出し内に標識を設定することによって、データが暗号化されることを知らせます。 この標識を調べると、受信側 LU は、受信側 TP に渡す前に、どの要求単位を復号するかを判別できます。

SNA ネットワークでは、WebSphere MCA は、トランザクション・プログラムです。 MCA は、送信するデータに対して暗号化を要求しません。 したがって、選択可能データ暗号化方式は、選択できません。強制データ暗号化方式またはデータ暗号化方式なしが、セッション上で選択可能です。

強制データ暗号化方式をインプリメントする方法については、ご使用の SNA サブシステムの資料を参照してください。 ご使用のプラットフォーム上で使用可能な、より強い形式の暗号化 (例えば、z/OS 上での Triple DES 24 バイト暗号化) についても、同じ資料を参照してください。

セッション・レベルの暗号化方式の一般的な解説については、「*Systems Network Architecture LU 6.2 Reference: Peer Protocols*」、SC31-6808 を参照してください。

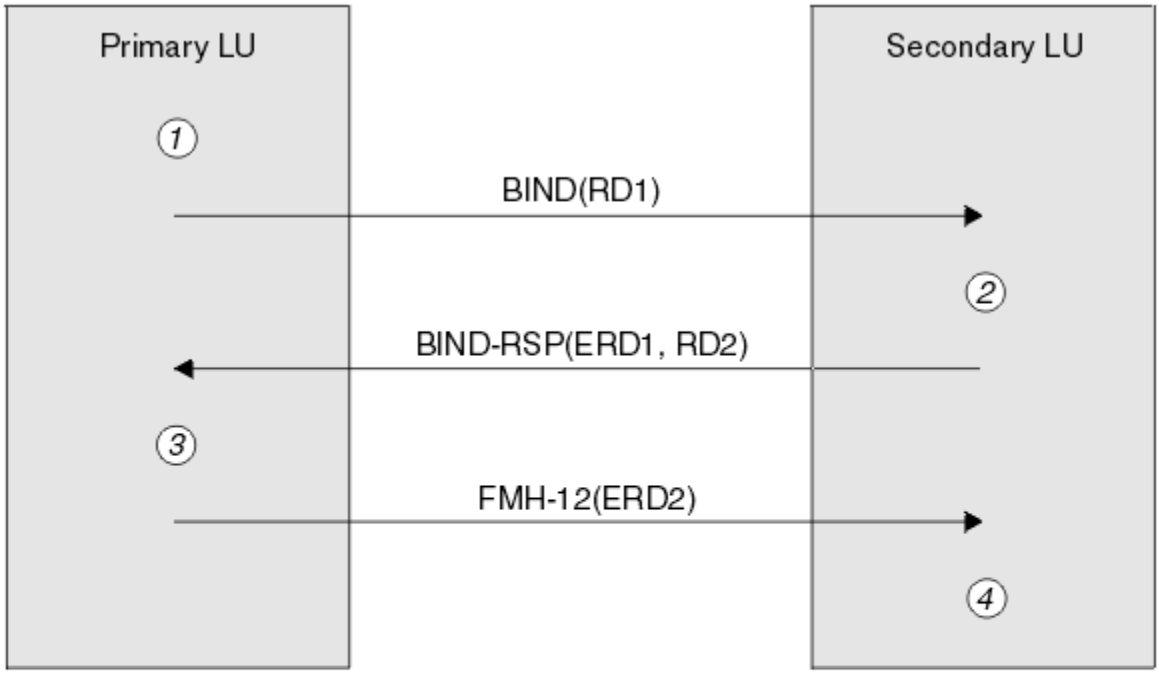
### セッション・レベルの認証

セッション・レベルの認証 は、2 つの LU がセッションをアクティブにしている間に相互に認証できるようにする、セッション・レベルのセキュリティー・プロトコルです。 これは、*LU-LU* 検証 とも呼ばれます。

LU はネットワークからシステムへの実質的な "「gateway」" であるため、特定の状況ではこのレベルの認証で十分であると考慮できます。 例えば、キュー・マネージャーが、制御されたトラステッド環境で稼働しているリモート・キュー・マネージャーとメッセージを交換する必要がある場合、LU が認証された後では、リモート・システムの残りのコンポーネントの ID を信頼することができます。

セッション・レベルの認証は、各 LU が相手側の LU のパスワードを検証することによって行われます。 このパスワードは、*LU-LU* パスワード と呼ばれます。これは、LU の各ペア間で 1 つのパスワードが設定されるからです。 LU-LU パスワードが設定される方法は、インプリメンテーションにより異なり、SNA の範囲外です。

75 ページの図 10 は、セッション・レベルの認証のフローを示しています。




Legend:

| | | |
|---|---|---|
| BIND | = | BIND request unit |
| BIND-RSP | = | BIND response unit |
| ERD | = | Encrypted random data |
| FMH-12 | = | Function Management Header 12 |
| RD | = | Random data |

図 *10.* セッション・レベルの認証のフロー

セッション・レベル認証用のプロトコルは、次のとおりです。 この手順内の番号は、75 ページの図 10 の番号に対応しています。

1. 1 次 LU は、ランダム・データ値 (RD1) を生成し、BIND 要求でそのデータ値を 2 次 LU に送信します。
2. 2 次 LU は、ランダム・データと一緒に BIND 要求を受信すると、LU-LU パスワードのコピーを鍵とする DES アルゴリズムを使用して、そのデータを暗号化します。 次に、2 次 LU は、2 つ目のランダム・データ値 (RD2) を生成し、暗号化されたデータ (ERD1) と一緒に、BIND 応答でそのデータ値を 1 次 LU に送信します。
3. 1 次 LU は、BIND 応答を受信すると、独自のバージョンの暗号化データを、最初に生成したランダム・データから計算します。 1 次 LU は、LU-LU パスワードのコピーを鍵とする DES アルゴリズムを使用して、この計算を行います。 次に、そのバージョンを、BIND 応答で受信した暗号化データと比較します。 2 つの値が同一である場合、1 次 LU は、2 次 LU が同じパスワードを持っていること、および 2 次 LU が認証されたことを認識します。 2 つの値が一致しない場合、1 次 LU はセッションを終了します。

   次に、1 次 LU は、BIND 応答で受信したランダム・データを暗号化し、暗号化されたデータ (ERD2) を、Function Management Header 12 (FMH-12) で 2 次 LU に送信します。
4. 2 次 LU は、FMH-12 を受信すると、生成したランダム・データから、独自のバージョンの暗号化データを計算します。 次に、そのバージョンを、FMH-12 で受信した暗号化データと比較します。 2 つの値が同一である場合、1 次 LU は認証されます。 2 つの値が一致しない場合、2 次 LU はセッションを終了します。

中間一致攻撃に対する保護が改善されている拡張バージョンのプロトコルでは、2 次 LU は、LU-LU パスワードのコピーを鍵として使用して、RD1、RD2、および 2 次 LU の完全修飾名から、DES メッセージ認証コード (MAC) を計算します。 2 次 LU は、ERD1 ではなく、BIND 応答で、1 次 LU に MAC を送信します。

1 次 LU は、独自のバージョンの MAC を計算し、それを BIND 応答で受信した MAC と比較することによって、2 次 LU を認証します。 次に、1 次 LU は、RD1 と RD2 から 2 つ目の MAC を計算し、ERD2 ではなく、FMH-12 で、その MAC を 2 次 LU に送信します。

2 次 LU は、独自のバージョンの 2 つ目の MAC を計算し、それを FMH-12 で受信した MAC と比較することによって、1 次 LU を認証します。

セッション・レベル認証の構成方法については、ご使用の SNA サブシステムの資料を参照してください。 セッション・レベル認証の一般的な解説については、「*Systems Network Architecture LU 6.2 Reference: Peer Protocols*」、SC31-6808 を参照してください。

### 会話レベルの認証

ローカル TP が、相手側 TP との会話を割り振ろうとすると、ローカル LU は、相手側の LU に接続要求を送信し、相手側の TP を接続するように依頼します。 ある種の状況のもとでは、接続要求にセキュリティー情報が含まれる場合があります。相手側の LU は、この情報を使用して、ローカル TP を認証することができます。 これは、会話レベルの認証、またはエンド・ユーザー検査 と呼ばれます。

以下の各トピックでは、IBM WebSphere MQ で会話レベルの認証がどのようにサポートされているかについて説明します。

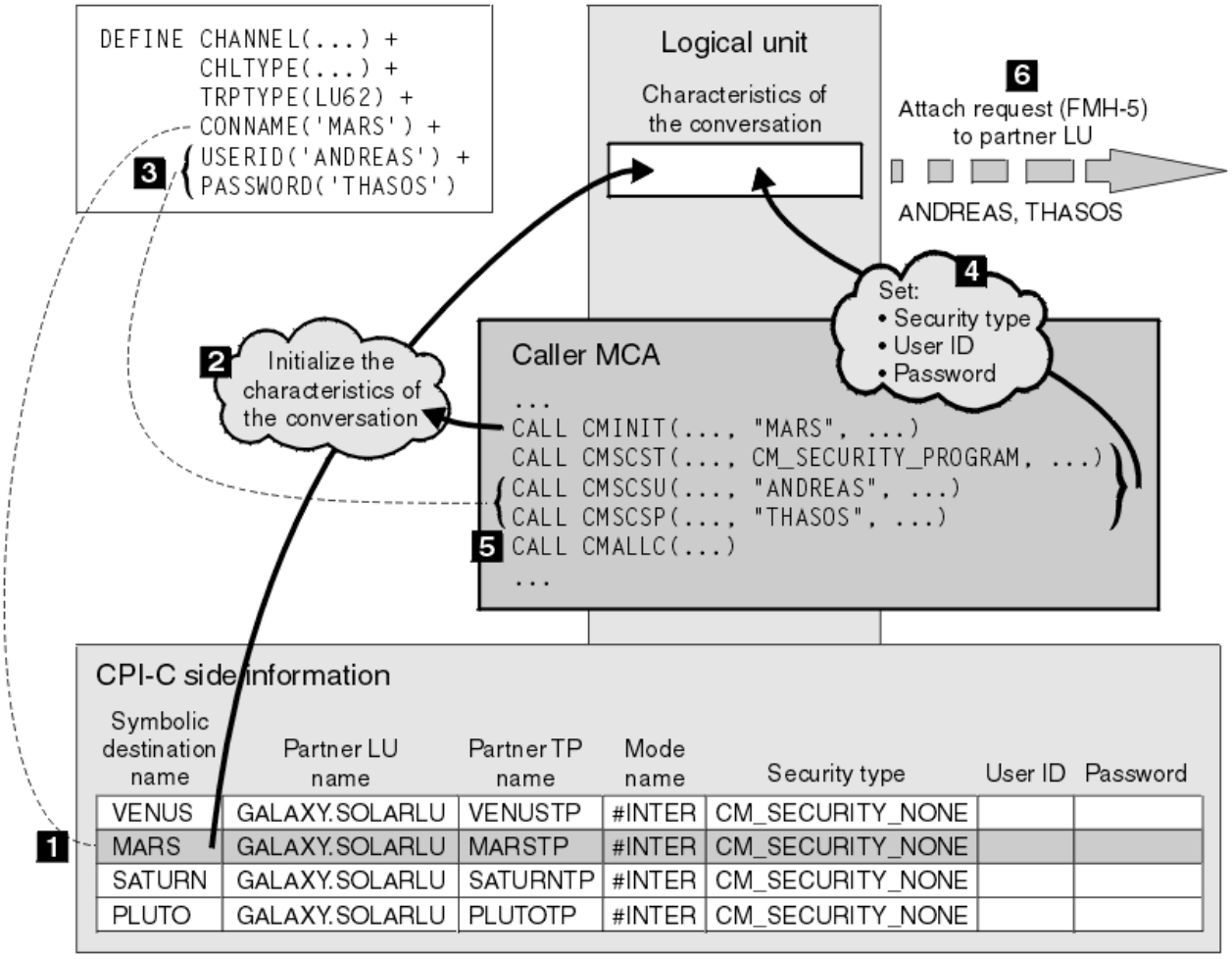
会話レベル認証の詳細については、「*Systems Network Architecture LU 6.2 Reference: Peer Protocols*」、SC31-6808 を参照してください。 z/OS に固有の情報については、「*z/OS MVS™* 計画*: APPC/MVS* 管理」、SA88-8571 を参照してください。

CPI-C の詳細については、「*Common Programming Interface Communications CPI-C Specification*」、SC31-6180 を参照してください。 APPC/MVS TP Conversation Callable Services の詳細については、「*z/OS MVS* プログラミング： *APPC/MVS* トランザクション・プログラムの書き方」、SA88-8587 を参照してください。

#### *UNIX* システム、および *Windows* システム上の *IBM WebSphere MQ* での会話レベル認証のサポート

このトピックでは、UNIX, Linux, and Windows で会話レベルの認証がどのように動作するのかに関する概要を取り上げます。

IBM WebSphere MQ for WebSphere MQ on UNIX システム、および WebSphere MQ for Windows での会話レベル認証のサポートを、 77 ページの図 11 に示します。 図の中の番号は、以下の説明内の番号と対応しています。




| Symbolic destination name | Partner LU name | Partner TP name | Mode name | Security type | User ID | Password |
|---|---|---|---|---|---|---|
| VENUS | GALAXY.SOLARLU | VENUSTP | #INTER | CM_SECURITY_NONE | | |
| MARS | GALAXY.SOLARLU | MARSTP | #INTER | CM_SECURITY_NONE | | |
| SATURN | GALAXY.SOLARLU | SATURNTP | #INTER | CM_SECURITY_NONE | | |
| PLUTO | GALAXY.SOLARLU | PLUTOTP | #INTER | CM_SECURITY_NONE | | |

図 *11.* 会話レベルの認証に対する *WebSphere MQ* のサポート

IBM i システム、UNIX システム、および Windows システム上で、MCA は、共通プログラミング・インターフェース・コミュニケーション (CPI-C) 呼び出しを使用して、SNA ネットワーク上で相手側 MCA と通信します。 チャネルの呼び出し側のチャネル定義では、CONNAME パラメーターの値は、CPI-C サイド情報項目を識別するシンボリック宛先名です (1)。 この項目は、次のものを指定します。

- 相手側 LU の名前
- 応答側 MCA である、相手側 TP の名前
- 会話に使用されるモードの名前

サイド情報項目は、次のセキュリティー情報も指定できます。

- セキュリティー・タイプ

  一般にインプリメントされるセキュリティー・タイプは、CM_SECURITY_NONE、CM_SECURITY_PROGRAM、および CM_SECURITY_SAME ですが、CPI-C 仕様では別のタイプが定義されます。
- ユーザー ID
- パスワード

呼び出し側 MCA は、CONNAME の値を呼び出しのパラメーターの 1 つとして使用して、CPI-C 呼び出し CMINIT を発行することによって、応答側 MCA との会話を割り振ります。 CMINIT 呼び出しは、ローカル LU のために、MCA が会話に使用する予定のサイド情報項目を識別します。 ローカル LU は、この項目内の値を使用して、会話の特性を初期化します (2)。

次に、呼び出し側 MCA は、チャネル定義内の USERID パラメーターと PASSWORD パラメーターの値を検査します (3)。 USERID が設定されると、呼び出し側 MCA は、次の CPI-C 呼び出しを発行します (4)。

- CMSCST。会話のセキュリティー・タイプを CM_SECURITY_PROGRAM に設定します。
- CMSCSU。会話のユーザー ID を USERID の値に設定します。
- CMSCSP。会話のパスワードを PASSWORD の値に設定します。 PASSWORD が設定されない限り、CMSCSP は呼び出されません。

これらの呼び出しによって設定されたセキュリティー・タイプ、ユーザー ID、およびパスワードは、サイド情報項目から以前に取得された値はすべて指定変更されます。

次に、呼び出し側 MCA は、CPI-C 呼び出し CMALLC を発行して、会話を割り振ります (5)。 この呼び出しに応えて、ローカル LU は、相手側 LU に接続要求 (Function Management Header 5、すなわち FMH-5) を送信します (6)。

相手側 LU がユーザー ID とパスワードを受け入れると、USERID と PASSWORD の値が接続要求に組み込まれます。 相手側 LU がユーザー ID とパスワードを受け入れない場合、これらの値は接続要求に組み込まれません。 ローカル LU は、両方の LU がバインドしてセッションを作成するときに、相手側 LU が、情報交換の一部としてユーザー ID とパスワードを受け入れるかどうかを検出します。

今後のバージョンの接続要求では、クリア・パスワードではなく、置換パスワードが LU 間を通過できます。置換パスワードは、パスワードから作成される、DES メッセージ認証コード (MAC)、または SHA-1 メッセージ・ダイジェストです。 置換パスワードが使用できるのは、両方の LU が置換パスワードをサポートする場合だけです。

相手側 LU は、ユーザー ID とパスワードが入っている着信接続要求を受信すると、識別と認証のために、ユーザー ID とパスワードを使用する場合があります。 相手側 LU は、アクセス制御リストを参照することによって、会話を割り振り、応答側 MCA を接続する権限が、ユーザー ID にあるかどうかも判別します。

さらに、応答側 MCA は、接続要求に組み込まれているユーザー ID の下で稼働する場合があります。この場合、このユーザー ID が、応答側 MCA のデフォルト・ユーザー ID になり、この MCA がキュー・マネージャーに接続しようとするときの権限検査に使用されます。 また、MCA が後でキュー・マネージャーのリソースにアクセスしようとするときにも、権限検査に使用される場合があります。

接続要求におけるユーザー ID とパスワードが識別、認証、およびアクセス制御に使用される方法は、インプリメンテーションによって決まります。 ご使用の SNA サブシステムに固有の情報については、該当する資料を参照してください。

USERID が設定されない場合、呼び出し側の MCA は、CMSCST、CMSCSU、および CMSCSP を呼び出しません。 この場合、接続要求で流れるセキュリティー情報は、サイド情報項目で何が指定されるか、および相手側 LU が何を受け入れるかのみによって決まります。

## キュー・マネージャー・クラスターのセキュリティー

キュー・マネージャー・クラスターは便利ですが、使用に際してはセキュリティーに特に注意する必要があります。

キュー・マネージャー・クラスター とは、なんらかの点で論理的に関連付けられているキュー・マネージャーのネットワークです。 クラスターのメンバーであるキュー・マネージャーは、クラスター・キュー・マネージャー と呼ばれます。

クラスター・キュー・マネージャーに属するキューを、クラスター内の他のキュー・マネージャーに知らせることができます。 このようなキューは、クラスター・キュー と呼ばれます。 クラスター内の任意のキュー・マネージャーは、次のものがなくても、クラスター・キューにメッセージを送信できます。

- 各クラスター・キューに対する明示的なリモート・キュー定義
- 各リモート・キュー・マネージャーとの間で明示的に定義される、両方向のチャネル
- アウトバウンド・チャネルごとに別々の伝送キュー

複数のキュー・マネージャーがクローンとして存在するクラスターを作成することができます。 つまり、これらのキュー・マネージャーは、クラスター・キューとして宣言されたすべてのローカル・キューを含めて、同じローカル・キューのインスタンスを持ち、同じサーバー・アプリケーションのインスタンスをサポートできます。

クラスター・キュー・マネージャーに接続されているアプリケーションが、複製された各キュー・マネージャー上にインスタンスがあるクラスター・キューに、メッセージを送信する場合、IBM WebSphere MQ

は、そのメッセージをどのキュー・マネージャーに送信するかを決定します。 複数のアプリケーションがクラスター・キューにメッセージを送信する場合、WebSphere MQ は、そのキューのインスタンスがあるキュー・マネージャーのそれぞれに、ワークロードを分散して調整します。 複製されたキュー・マネージャーをホストするシステムのいずれかに障害が起きても、WebSphere MQ は、障害が起きたシステムが再始動するまで、残りのキュー・マネージャー全体で引き続き、ワークロードを調整します。

キュー・マネージャー・クラスターを使用する場合は、次のセキュリティー項目を考慮する必要があります。

- 選択されたキュー・マネージャーだけが、ご使用のキュー・マネージャーにメッセージを送信できるようにする
- リモート・キュー・マネージャーの選択されたユーザーだけが、ご使用のキュー・マネージャー上のキューにメッセージを送信できるようにする
- ご使用のキュー・マネージャーに接続されているアプリケーションが、選択されたリモート・キューだけにメッセージを送信できるようにする

上記の考慮事項は、クラスターを使用しない場合であっても該当しますが、クラスターを使用する場合の方が、重要度が高くなります。

アプリケーションが 1 つのクラスター・キューにメッセージを送信できる場合、そのアプリケーションは、リモート・キューの定義、伝送キュー、またはチャネルを追加しなくても、ほかの任意のクラスター・キューにメッセージを送信できます。 したがって、ご使用のキュー・マネージャー上のクラスター・キューへのアクセスを制限する必要があるかどうか、およびアプリケーションがメッセージを送信する先のクラスター・キューを制限する必要があるかどうかを検討することが、さらに重要になります。

このほかにも次のセキュリティー上の考慮事項があります。この考慮事項は、キュー・マネージャー・クラスターを使用する場合だけ該当します。

- 選択されたキュー・マネージャーだけがクラスターに加われるようにする
- 不必要なキュー・マネージャーをクラスターから退去させる

これらのすべての考慮事項の詳細については、『クラスターのセキュリティーの確保』を参照してください。

**関連タスク**

250 ページの『キュー・マネージャーのメッセージ受信の防止』
出口プログラムを使用することによって、受信する権限のないメッセージをクラスター・キュー・マネージャーが受信できないようにすることができます。

## IBM WebSphere MQ Publish/Subscribe のセキュリティー

IBM WebSphere MQ Publish/Subscribe を使用する場合は、セキュリティーに関する考慮事項が増えます。

パブリッシュ/サブスクライブ・システムには、パブリッシャーとサブスクライバーという、2 つのタイプの アプリケーションがあります。 パブリッシャー は、IBM WebSphere MQ メッセージの形式で情報を提供します。 パブリッシャーは、メッセージを発行するときに、メッセージ内の情報の主題を指定するトピック を指定します。

サブスクライバー は、発行される情報のコンシューマーです。 サブスクライバーは、関心のあるトピックをサブスクライブすることによって、それらを指定します。

キュー・マネージャー は、IBM WebSphere MQ Publish/Subscribe に用意されているアプリケーションです。 ブローカーは、パブリッシャーから発行されたメッセージと、サブスクライバーからのサブスクリプション要求を受け取り、発行されたメッセージをサブスクライバーに経路指定します。 サブスクライバーがサブスクリプション要求を出したトピックについてのメッセージだけが、サブスクライバーに送られます。

詳細については、パブリッシュ/サブスクライブのセキュリティーを参照してください。

## マルチキャストのセキュリティー

この情報を利用して、IBM WebSphere MQ Multicast でなぜセキュリティー・プロセスが必要になることがあるのかに関する理解を深めてください。

IBM WebSphere MQ Multicast には、標準装備のセキュリティーはありません。 セキュリティー検査は MQOPEN 時にキュー・マネージャーで処理され、MQMD フィールド設定はクライアントによって処理されます。 IBM WebSphere MQ アプリケーションではないアプリケーション (LLM アプリケーションなど、詳しくは WebSphere MQ Low Latency Messaging とのマルチキャスト相互運用性を参照) がネットワーク内に存在する場合があります。したがって、受信側のアプリケーションがコンテキスト・フィールドの妥当性を確認できないために、独自のセキュリティー手順を実装する必要が生じることがあります。

考慮するセキュリティー・プロセスとして、次の 3 種類があります。

**アクセス制御**

IBM WebSphere MQ でのアクセス制御は、ユーザー ID に基づいて行われます。 この件について詳しくは、57 ページの『クライアントへのアクセス制御』を参照してください。

**ネットワーク・セキュリティー**

ネットワークを分離することは、偽のメッセージを防ぐための実行可能なセキュリティー・オプションです。 マルチキャスト・グループ・アドレス上のアプリケーションが、ネイティブの通信機能を使用して、有害なメッセージをパブリッシュすることがあります。これは同じマルチキャスト・グループ・アドレス上のアプリケーションからのメッセージなので、MQ メッセージと区別できません。

マルチキャスト・グループ・アドレス上のクライアントが、同じマルチキャスト・グループ・アドレス上の他のクライアント宛てのメッセージを受け取ることもあります。

マルチキャスト・ネットワークを分離すると、有効なクライアントとアプリケーションのみがアクセスできるようになります。 このセキュリティー上の予防措置により、有害なメッセージが着信したり、機密情報が流出したりしないようにできます。

マルチキャスト・グループ・ネットワーク・アドレスについては、マルチキャスト・トラフィックに適したネットワークの設定を参照してください。

**デジタル署名**

デジタル署名は、メッセージの表記を暗号化することによって作成されます。 この暗号化は、署名者の秘密鍵を使用し、通常、効率を上げるために、メッセージ自体ではなく、メッセージ・ダイジェストを対象とし行われます。 MQPUT の前にメッセージをデジタル署名することも適切なセキュリティー上の予防措置ですが、メッセージが大量になる場合は、このプロセスはパフォーマンスに悪影響を及ぼすおそれがあります。

デジタル署名は、署名されるデータによってさまざまです。 2 つの別々のメッセージが、同じエンティティーによってデジタル署名される場合、2 つの署名は異なりますが、両方の署名を同じ公開鍵、つまり、メッセージを署名したエンティティーの公開鍵で検証することができます。

このセクションで前述されているように、マルチキャスト・グループ・アドレス上のアプリケーションが、ネイティブの通信機能を使用して、MQ メッセージと区別できない有害なメッセージをパブリッシュすることがあります。 デジタル署名は発信証明を提供し、送信側だけが秘密鍵を知っているので、送信側がメッセージの発信元であるという強固な証拠になります。

この件について詳しくは、7 ページの『暗号の概念』を参照してください。

# ファイアウォールおよび Internet Pass-Thru

通常はファイアウォールを使用して、悪意のある IP アドレスからのアクセス (サービス妨害アタックの場合など) を防ぎます。 ただし、セキュリティー管理者によるファイアウォール・ルールの更新を待機する間など、IBM WebSphere MQ で IP アドレスを一時的にブロックする必要がある場合があります。

1 つ以上の IP アドレスをブロックするには、タイプ BLOCKADDR または ADDRESSMAP のチャネル認証レコードを作成します。 詳しくは、182 ページの『特定の IP アドレスのブロッキング』を参照してください。

## IBM WebSphere MQ Internet Pass-Thru のセキュリティー

Internet Pass-Thru を使用すると、ファイアウォールを通過する通信を簡略化できますが、セキュリティーに関する影響もいくつかあります。

IBM WebSphere MQ Internet Pass-thru は、SupportPac MS81 で提供されている IBM WebSphere MQ 基本製品拡張機能です。

WebSphere MQ Internet Pass-thru を使用すると、直接の TCP/IP 接続なしにインターネットを介して、2 つのキュー・マネージャーがメッセージを交換したり、WebSphere MQ クライアント・アプリケーションがキュー・マネージャーに接続したりすることができるようになります。 これは、ファイアウォールにより、2 つのシステム間の直接 TCP/IP 接続が禁止される場合に便利です。 この機能は、HTTP 内部のフローのトンネリングを行ったり、プロキシーの役目をすることによって、ファイアウォールとの双方向の WebSphere MQ チャネル・プロトコル・フローの通過を簡単、かつ管理しやすくします。 Secure Sockets Layer (SSL) を使用すると、 インターネットを介して送信されるメッセージの暗号化と復号にも使用できます。

WebSphere MQ システムが IPT と通信する場合に IPT で SSLProxyMode を使用していない場合は、次のように、WebSphere MQ が使用する CipherSpec が、IPT が使用する CipherSuite と一致することを確認してください。

- IPT が SSL または TLS サーバーの役目をし、WebSphere MQ が SSL または TLS クライアントとして接続する場合、WebSphere MQ が使用する CipherSpec は、関連した IPT 鍵リングで使用可能になっている CipherSuite と対応する必要があります。
- IPT が SSL または TLS クライアントの役目をし、WebSphere MQ SSL または TLS サーバーに接続する場合、IPT CipherSuite は、受信側の WebSphere MQ チャネル上で定義された CipherSpec と一致する必要があります。

IPT から統合 WebSphere MQ SSL および TLS サポートに移行する場合は、iKeyman を使用して IPT からデジタル証明書を転送します。

詳しくは、WebSphere MQ Internet Pass-Thru (SupportPac MS81) を参照してください。

# セキュリティーのセットアップ

このトピック集には、さまざまなオペレーティング・システムおよびクライアントの使用法に固有の情報が含まれています。

## UNIX, Linux, and Windows システムでのセキュリティーのセットアップ

UNIX, Linux, and Windows システムに固有のセキュリティーに関する考慮事項。

IBM WebSphere MQ キュー・マネージャーは、価値があると思われる情報を転送します。そのため、許可されていないユーザーがキュー・マネージャーにアクセスできなくするために、権限システムを使用する必要があります。 以下のタイプのセキュリティー制御について考えてみてください。

**IBM WebSphere MQ を管理できるユーザー**
IBM WebSphere MQ を管理するコマンドを発行できるユーザー群を定義できます。

**IBM WebSphere MQ オブジェクトをだれが使用できるか**
以下のことを実行するために MQI 呼び出しと PCF コマンドを使用できるユーザー (通常はアプリケーション) を定義できます。

- キュー・マネージャーにだれが接続できるか。
- オブジェクト (キュー、プロセス定義、名前リスト、チャネル、クライアント接続チャネル、リスナー、サービス、および認証情報オブジェクト) にアクセスできるのはどのユーザーか、また、これらのオブジェクトに対して該当ユーザーが持つアクセス権の種類は何か。
- IBM WebSphere MQ メッセージにだれがアクセスできるか。
- メッセージと関連付けられたコンテキスト情報にだれがアクセスできるか。

**チャネル・セキュリティー**
リモート・システムへメッセージを送信するのに使用するチャネルが必要なリソースにアクセスできることを確認する必要があります。

標準の操作機能を使用することにより、プログラム・ライブラリー、MQI リンク・ライブラリー、およびコマンドに対するアクセス権を付与することができます。 ただし、キューおよびその他のキュー・マネー

ジャー・データを入れるディレクトリーは、IBM WebSphere MQ 専用です。標準オペレーティング・システム・コマンドを使用して MQI リソースへの許可を与えたり、取り消したりしないでください。

## ターミナル・サービスを使用した IBM WebSphere MQ への接続

ターミナル・サービスを使用している場合は、 **Create global objects** ユーザー権限によって問題が発生する可能性があります。

Terminal Services を使用して Windows システムに接続しているときに、キュー・マネージャーの作成または開始で問題が発生する場合は、 Windows の最新バージョンのユーザー権限 **Create global objects** が原因である可能性があります。

**Create global objects** ユーザー権限は、グローバル・ネームスペースにオブジェクトを作成する権限を持つユーザーを制限します。 アプリケーションがグローバル・オブジェクトを作成するには、グローバル名前空間で実行されているか、アプリケーションを実行するユーザーに **Create global objects** ユーザー権限が適用されている必要があります。

管理者にはデフォルトで適用される **Create global objects** ユーザー権限があるため、管理者は、ユーザー権限を変更することなく、Terminal Services を使用して接続したときにキュー・マネージャーを作成して開始することができます。

端末サービスを使用するときに WebSphere MQ を管理するさまざまな方法が機能しない場合は、 **Create global objects** ユーザー権限を設定してみてください。

1. 以下のようにして、「管理ツール」パネルを開きます。

   **Windows 2003 および Windows XP**
   このパネルには、「**コントロールパネル**」 > 「**管理ツール**」を使用してアクセスします。

   **Windows Vista および Windows Server 2008**
   このパネルには、「**コントロールパネル**」 > 「**システムと保守**」 > 「**管理ツール**」を使用してアクセスします。
2. 「**ローカル セキュリティ ポリシー**」をダブルクリックします。
3. `Local Policies` を展開します。
4. `User Rights Assignment` をクリックします。
5. 新しいユーザーまたはグループを **Create global objects** ポリシーに追加します。

## Windows でのグループの作成と管理

ここでは、ワークステーションまたはメンバー・サーバー・マシンでグループを管理するプロセスを取り上げます。

ドメイン・コントローラーの場合、ユーザーおよびグループは Active Directory を使用して管理されます。Active Directory の使用について詳しくは、適切なオペレーティング・システムの説明を参照してください。

プリンシパルのグループ・メンバーシップに変更を加えても、キュー・マネージャーを再始動するか、MQSC コマンド REFRESH SECURITY (または PCF でこれに相当するコマンド) を実行するまで、その変更は認識されません。

ユーザーとグループを操作するには、「コンピュータの管理」パネルを使用します。 現在ログオンしているユーザーに対して加えられた変更は、そのユーザーが再度ログインするまで有効にならない場合があります。

**Windows 2003 および Windows XP**
このパネルには、「**コントロールパネル**」 > 「**管理ツール**」 > 「**コンピューターの管理**」を使用してアクセスします。

**Windows Vista および Windows Server 2008**
このパネルには、「**コントロールパネル**」 > 「**システムとメンテナンス**」 > 「**管理ツール**」 > 「**コンピューターの管理**」を使用してアクセスします。

**Windows 7**
「**管理ツール**」 > 「**コンピューターの管理**」を使用して、このパネルにアクセスします。

## *Windows* でのグループの作成

コントロール パネルを使用してグループを作成します。

### 手順

1. コントロール パネルを開きます。
2. 「**管理ツール**」をダブルクリックします。
   「管理ツール」パネルが開きます。
3. 「**コンピュータの管理**」をダブルクリックします。
   「コンピュータの管理」パネルが開きます。
4. 「**ローカル ユーザーとグループ**」を展開します。
5. 「**グループ**」を右クリックして、「**新しいグループ**」を選択します。
   「新しいグループ」パネルが表示されます。
6. 「グループ名」フィールドに適切な名前を入力し、「**作成**」をクリックします。
7. 「**クローズ**」をクリックします。

## *Windows* でグループにユーザーを追加する操作

コントロール パネルを使用してグループにユーザーを追加します。

### 手順

1. コントロール パネルを開きます。
2. 「**管理ツール**」をダブルクリックします。
   「管理ツール」パネルが開きます。
3. 「**コンピュータの管理**」をダブルクリックします。
   「コンピュータの管理」パネルが開きます。
4. 「コンピュータの管理」パネルから、「**ローカル ユーザーとグループ**」を展開します。
5. 「**ユーザー**」
6. グループを追加するユーザーをダブルクリックします。
   「ユーザー プロパティ」パネルが表示されます。
7. 「**所属するグループ**」タブを選択します。
8. ユーザーを追加するグループを選択します。 該当のグループが表示されない場合、以下の処理を行います。
   a) 「**追加...**」をクリックします。
      「グループの選択」パネルが表示されます。
   b) 「**Locations...(ロケーション...)**」をクリックします。
      「Locations (ロケーション)」パネルが表示されます。
   c) ユーザーを追加するグループのロケーションをリストから選択し、「**OK (了解)**」をクリックします。
   d) 表示されたフィールドにグループ名を入力します。

      または、 「**拡張 ...**」 をクリックします。 次に、 「**検索**」 をクリックして、現在選択されている場所で使用可能なグループをリストします。 ここから、 ユーザーを追加するグループを選択し、「**OK (了解)**」をクリックします。
   e) 「**OK**」をクリックします。
      「ユーザー プロパティ」パネルが表示され、追加したグループが表示されます。
   f) グループを選択します。
9. 「**OK**」をクリックします。
   「コンピュータの管理」パネルが表示されます。

### *Windows* でグループのメンバーを表示する操作

コントロール パネルを使用してグループのメンバーを表示します。

#### 手順

1. コントロール パネルを開きます。
2. 「**管理ツール**」をダブルクリックします。
   「管理ツール」パネルが開きます。
3. 「**コンピュータの管理**」をダブルクリックします。
   「コンピュータの管理」パネルが開きます。
4. 「コンピュータの管理」パネルから、「**ローカル ユーザーとグループ**」を展開します。
5. 「**グループ**」を選択します。
6. グループをダブルクリックします。 「グループ プロパティ」パネルが表示されます。
   「グループ プロパティ」パネルが表示されます。

#### タスクの結果

グループのメンバーが表示されます。

### *Windows* でグループからユーザーを削除する操作

コントロール パネルを使用してグループからユーザーを削除します。

#### 手順

1. コントロール パネルを開きます。
2. 「**管理ツール**」をダブルクリックします。
   「管理ツール」パネルが開きます。
3. 「**コンピュータの管理**」をダブルクリックします。
   「コンピュータの管理」パネルが開きます。
4. 「コンピュータの管理」パネルから、「**ローカル ユーザーとグループ**」を展開します。
5. 「**ユーザー**」を選択します。
6. グループを追加するユーザーをダブルクリックします。
   「ユーザー プロパティ」パネルが表示されます。
7. 「**所属するグループ**」タブを選択します。
8. ユーザーを除去するグループを選択し、「**削除**」をクリックします。
9. 「**OK**」をクリックします。
   「コンピュータの管理」パネルが表示されます。

#### タスクの結果

グループからユーザーが削除されました。

## HP-UX でのグループの作成と管理

HP-UX では、NIS および NIS+ を使用していない場合、System Administration Manager (SAM) を使用してグループを処理します。

### *HP-UX* でのグループの作成

System Administration Manager を使用してグループにユーザーを追加します。

#### 手順

1. System Administration Manager (SAM) で、「Accounts for Users and Groups (ユーザーおよびグループのアカウント)」をダブルクリックします。

2.「Groups (グループ)」をダブルクリックします。
3.「Actions (アクション)」プルダウンで「Add (追加)」を選択し、「Add a New Group (新規グループの追加)」パネルを表示します。
4. グループの名前を入力し、グループに追加するユーザーを選択します。
5.「Apply (適用)」をクリックしてグループを作成します。

### タスクの結果
グループが作成されました。

## *HP-UX* でグループにユーザーを追加する操作
System Administration Manager を使用してグループにユーザーを追加します。

### 手順
1. System Administration Manager (SAM) で、「Accounts for Users and Groups (ユーザーおよびグループのアカウント)」をダブルクリックします。
2.「Groups (グループ)」をダブルクリックします。
3. グループの名前を強調表示して、「Actions (アクション)」プルダウンで「Modify (変更)」を選択し、「Modify an Existing Group (既存のグループの変更)」パネルを表示します。
4. グループに追加するユーザーを選択して、「Add (追加)」をクリックします。
5. グループに別のユーザーも追加する場合には、ユーザーごとにステップ 4 を繰り返します。
6. リストへの名前の追加が完了したら、「OK (了解)」をクリックします。

### タスクの結果
グループにユーザーが追加されました。

## *HP-UX* でグループのメンバーを表示する操作
System Administration Manager を使用してグループのメンバーを表示します。

### 手順
1. System Administration Manager (SAM) で、「Accounts for Users and Groups (ユーザーおよびグループのアカウント)」をダブルクリックします。
2.「Groups (グループ)」をダブルクリックします。
3. グループの名前を強調表示して、「Actions (アクション)」プルダウンで「Modify (変更)」を選択し、「Modify an Existing Group (既存のグループの変更)」パネルを表示し、グループのユーザーのリストを表示します。

### タスクの結果
グループのメンバーが表示されます。

## *HP-UX* でグループからユーザーを削除する操作
System Administration Manager を使用してグループからユーザーを削除します。

### 手順
1. System Administration Manager (SAM) で、「Accounts for Users and Groups (ユーザーおよびグループのアカウント)」をダブルクリックします。
2.「Groups (グループ)」をダブルクリックします。
3. グループの名前を強調表示して、「Actions (アクション)」プルダウンで「Modify (変更)」を選択し、「Modify an Existing Group (既存のグループの変更)」パネルを表示します。
4. グループから除去するユーザーを選択して、「Remove (除去)」をクリックします。
5. グループから別のユーザーも除去する場合には、ユーザーごとにステップ 4 を繰り返します。

6. リストからの名前の除去が完了したら、「OK (了解)」をクリックします。

### タスクの結果
グループからユーザーが除去されました。

## AIX でのグループの作成と管理
AIX では、NIS および NIS+ を使用していない場合、SMITTY を使用してグループを処理します。

### グループの作成
SMITTY を使用してグループを作成します。

### 手順
1. SMITTY で、「Security and Users (セキュリティーおよびユーザー)」を選択して Enter キーを押します。
2. 「Groups (グループ)」を選択して Enter キーを押します。
3. 「Add a Group (グループの追加)」を選択して Enter キーを押します。
4. グループの名前と、グループに追加するユーザーの名前をコンマで区切って入力します。
5. Enter キーを押してグループを作成します。

### タスクの結果
グループが作成されました。

### グループへのユーザーの追加
SMITTY を使用してグループにユーザーを追加します。

### 手順
1. SMITTY で、「Security and Users (セキュリティーおよびユーザー)」を選択して Enter キーを押します。
2. 「Groups (グループ)」を選択して Enter キーを押します。
3. 「Change / Show Characteristics of Groups (グループの特性の変更/表示)」を選択して Enter キーを押します。
4. グループの名前を入力し、グループのメンバーのリストを表示します。
5. グループに追加するユーザーの名前をコンマで区切って追加します。
6. Enter キーを押してグループにその名前を追加します。

### グループ内のユーザーの表示
SMITTY を使用してグループのメンバーを表示します。

### 手順
1. SMITTY で、「Security and Users (セキュリティーおよびユーザー)」を選択して Enter キーを押します。
2. 「Groups (グループ)」を選択して Enter キーを押します。
3. 「Change / Show Characteristics of Groups (グループの特性の変更/表示)」を選択して Enter キーを押します。
4. グループの名前を入力し、グループのメンバーのリストを表示します。

### タスクの結果
グループのメンバーが表示されます。

### グループからのユーザーの除去
SMITTY を使用してグループからユーザーを削除します。

### 手順

1. SMITTY で、「Security and Users (セキュリティーおよびユーザー)」を選択して Enter キーを押します。
2. 「Groups (グループ)」を選択して Enter キーを押します。
3. 「Change / Show Characteristics of Groups (グループの特性の変更/表示)」を選択して Enter キーを押します。
4. グループの名前を入力し、グループのメンバーのリストを表示します。
5. グループから除去するユーザーの名前を削除します。
6. Enter キーを押してグループからその名前を除去します。

### タスクの結果
グループからユーザーが除去されました。

## Solaris でのグループの作成と管理
Solaris では、NIS および NIS + を使用していない場合、 /etc/group ファイルを使用してグループを処理します。

### *Solaris* でのグループの作成
**groupadd** コマンドを使用してグループを作成します。

### 手順

次のコマンドを入力します。 groupadd *group-name*
*group-name* は、グループの名前です。

### タスクの結果
/etc/group ファイルにグループの情報が書き込まれます。

### *Solaris* でグループにユーザーを追加する操作
**usermod** コマンドを使用してグループにユーザーを追加します。

### 手順

補足グループにメンバーを追加するには、usermod コマンドを実行して、そのユーザーが現在メンバーになっている補足グループと、そのユーザーがメンバーになる補足グループをリストします。
例えば、ユーザーがグループ groupa のメンバーであり、 groupb のメンバーにもなる場合は、コマンド usermod -G groupa,groupb *user-name* を使用します。ここで、 *user-name* はユーザー名です。

### *Solaris* でグループのメンバーを表示する操作
グループのメンバーを確認する場合は、/etc/group ファイルでそのグループの項目を調べます。

### *Solaris* でグループからユーザーを削除する操作
**usermod** コマンドを使用してグループからユーザーを削除します。

### 手順

補助グループからメンバーを除去するには、ユーザーにメンバーとして残す補助グループをリストする **usermod** コマンドを実行します。
例えば、users という 1 次グループのメンバーであるユーザーが、mqm、groupa、groupb の各グループのメンバーにもなっている場合に、そのユーザーを mqm グループから削除するには、コマンド usermod -G groupa,groupb *user-name* を使用します (*user-name* は、ユーザー名です)。

## Linux でのグループの作成と管理

Linux では、NIS または NIS+ を使用していない場合は、/etc/group ファイルを使用してグループを処理します。

### *Linux* でのグループの作成

**groupadd** コマンドを使用してグループを作成します。

#### 手順

新規グループを作成するには、コマンド groupadd -g *group-ID group-name* を入力します。

*group-ID* はグループの数値 ID、*group-name* はグループの名前です。

#### タスクの結果

/etc/group ファイルにグループの情報が書き込まれます。

### *Linux* でグループにユーザーを追加する操作

**usermod** コマンドを使用してグループにユーザーを追加します。

#### 手順

補足グループにメンバーを追加するには、usermod コマンドを実行して、そのユーザーが現在メンバーになっている補足グループと、そのユーザーがメンバーになる補足グループをリストします。
例えば、groupa というグループのメンバーであるユーザーを groupb のメンバーとして追加する場合は、コマンド usermod -G groupa,groupb *user-name* を使用します。

*user-name* は、ユーザー名です。

### *Linux* でグループのメンバーを表示する操作

**getent** コマンドを使用してグループのメンバーを表示します。

#### 手順

グループのメンバーを表示するには、次のコマンドを入力します。 getent group *group-name*

*group-name* は、グループの名前です。

### グループからのユーザーの除去

**usermod** コマンドを使用してグループからユーザーを削除します。

#### 手順

補助グループからメンバーを除去するには、ユーザーにメンバーとして残す補助グループをリストする **usermod** コマンドを実行します。
例えば、users という 1 次グループのメンバーであるユーザーが、mqm、groupa、groupb の各グループのメンバーにもなっている場合に、そのユーザーを mqm グループから削除するには、コマンド usermod -G groupa,groupb *user-name* を使用します。

*user-name* は、ユーザー名です。

## 許可が機能する仕組み

このセクションの各トピックには、各種の権限の機能とそれぞれに該当する制限を詳細に定義した権限指定表が含まれています。

これらの表は、次のような状態に適用されます。

- MQI 呼び出しを発行するアプリケーション
- MQSC コマンドをエスケープ PCF として発行する管理プログラム

- PCF コマンドを発行する管理プログラム

このセクションでは、次のものを指定する 1 組のテーブルという形で情報を提供しています。

**実行するアクション**
MQI オプション、MQSC コマンド、または PCF コマンド

**アクセス制御オブジェクト**
キュー、プロセス、キュー・マネージャー、名前リスト、認証情報、チャネル、クライアント接続チャネル、リスナー、またはサービス。

**必要な権限**
MQZAO_ 定数で表す

テーブルの中で、接頭部が MQZAO_ の定数は、特定のエンティティーに関する `setmqaut` コマンドの許可リストのキーワードに対応します。 例えば、MQZAO_BROWSE はキーワード `+browse` に対応します。 MQZAO_SET_ALL_CONTEXT はキーワード `+setall` などに対応します。 これらの定数は、プロダクトと共に提供される ヘッダー・ファイル cmqzc.h に定義されています。

## *MQI* 呼び出しについての許可

**MQCONN**、**MQOPEN**、**MQPUT1**、**MQCLOSE** では、許可検査が必要になる場合があります。 このトピックでは、それぞれの呼び出しで必要になる権限をいくつかの表にまとめています。

MQI 呼び出しおよびオプションのいくつかは、アプリケーションを実行するユーザー ID (またはアプリケーションが許可を想定できるユーザー ID) が適切な許可を与えられている場合にのみ、アプリケーションから発行できます。

許可検査を必要とする MQI 呼び出しは、**MQCONN**、**MQOPEN**, **MQPUT1**、および **MQCLOSE** の 4 つです。

MQOPEN および MQPUT1 の場合、権限検査は、名前が解決された結果の 1 つ以上の名前についてではなく、オープンされるオブジェクトの名前について行われます。 例えば、アプリケーションが別名キューをオープンする権限を与えられていても、別名が解決される基本キューをオープンする権限は与えられていない場合があります。 検査の規則は次のとおりです。キュー・マネージャー別名定義が直接オープンされない場合、キュー・マネージャー別名ではない名前を解決している間に検出された最初の定義に対して検査が実行されます。つまり、その名前はオブジェクト記述子の *ObjectName* フィールドに表示されます。 オブジェクトをオープンするためには、必ず権限が必要です。 場合によっては、キュー・マネージャー・オブジェクトの許可を通して入手される、キューに依存しない別の権限が必要です。

89 ページの表 8、90 ページの表 9、90 ページの表 10、および 91 ページの表 11 では、各呼び出しに必要な許可を要約しています。 表の適用しない は、許可検査がこの操作には該当しないことを意味します。検査しない は、許可検査が実行されないことを意味します。

**注 :** これらの表には、名前リスト、チャネル、クライアント接続チャネル、リスナー、サービス、または認証情報の各オブジェクトについての記載はありません。 これらのオブジェクトには、どの許可も適用されないためです。ただし、他のオブジェクトの場合と同じ許可が適用される MQOO_INQUIRE は例外となります。

特殊許可 MQZAO_ALL_MQI には、オブジェクト・タイプに関係した、表の中のすべての許可が含まれます。ただし、MQZAO_DELETE と MQZAO_DISPLAY は除きます。これらは、管理許可として分類されます。

メッセージ・コンテキスト・オプションのいずれかを変更するためには、呼び出しを発行するための適切な許可が必要です。 例えば、MQOO_SET_IDENTITY_CONTEXT または MQPMO_SET_IDENTITY_CONTEXT を使用するには、`+setid` アクセス権が必要です。

表 *8. MQCONN* 呼び出しに必要なセキュリティー許可

| **必要な条件** | **キュー・オブジェクト (91 ページの『1』)** | **プロセス・オブジェクト** | **キュー・マネージャー・オブジェクト** |
|---|---|---|---|
| **MQCONN** | 適用外 | 適用外 | MQZAO_CONNECT |

表 9. *MQOPEN* 呼び出しに必要なセキュリティー許可

| 必要な条件 | キュー・オブジェクト (91 ページの『1』) | プロセス・オブジェクト | キュー・マネージャー・オブジェクト |
|---|---|---|---|
| MQOO_INQUIRE | MQZAO_INQUIRE | MQZAO_INQUIRE | MQZAO_INQUIRE |
| MQOO_BROWSE | MQZAO_BROWSE | 適用外 | 検査しない |
| MQOO_INPUT_* | MQZAO_INPUT | 適用外 | 検査しない |
| MQOO_SAVE_ALL_CONTEXT (91 ページの『2』) | MQZAO_INPUT | 適用外 | 適用外 |
| MQOO_OUTPUT (通常キュー) (91 ページの『3』) | MQZAO_OUTPUT | 適用外 | 適用外 |
| MQOO_PASS_IDENTITY_CONTEXT (91 ページの『4』) | MQZAO_PASS_IDENTITY_CONTEXT | 適用外 | 検査しない |
| MQOO_PASS_ALL_CONTEXT (91 ページの『4』, 91 ページの『5』) | MQZAO_PASS_ALL_CONTEXT | 適用外 | 検査しない |
| MQOO_SET_IDENTITY_CONTEXT (91 ページの『4』, 91 ページの『5』) | MQZAO_SET_IDENTITY_CONTEXT | 適用外 | MQZAO_SET_IDENTITY_CONTEXT (91 ページの『6』) |
| MQOO_SET_ALL_CONTEXT (91 ページの『4』, 91 ページの『7』) | MQZAO_SET_ALL_CONTEXT | 適用外 | MQZAO_SET_ALL_CONTEXT (91 ページの『6』) |
| MQOO_OUTPUT (伝送キュー) (91 ページの『8』) | MQZAO_SET_ALL_CONTEXT | 適用外 | MQZAO_SET_ALL_CONTEXT (91 ページの『6』) |
| MQOO_SET | MQZAO_SET | 適用外 | 検査しない |
| MQOO_ALTERNATE_USER_AUTHORITY | (91 ページの『9』) | (91 ページの『9』) | MQZAO_ALTERNATE_USER_AUTHORITY (91 ページの『9』, 91 ページの『10』) |

表 10. *MQPUT1* 呼び出しに必要なセキュリティー許可

| 必要な条件 | キュー・オブジェクト (91 ページの『1』) | プロセス・オブジェクト | キュー・マネージャー・オブジェクト |
|---|---|---|---|
| MQPMO_PASS_IDENTITY_CONTEXT | MQZAO_PASS_IDENTITY_CONTEXT (92 ページの『11』) | 適用外 | 検査しない |
| MQPMO_PASS_ALL_CONTEXT | MQZAO_PASS_ALL_CONTEXT (92 ページの『11』) | 適用外 | 検査しない |
| MQPMO_SET_IDENTITY_CONTEXT | MQZAO_SET_IDENTITY_CONTEXT (92 ページの『11』) | 適用外 | MQZAO_SET_IDENTITY_CONTEXT (91 ページの『6』) |

| 表 *10. MQPUT1* 呼び出しに必要なセキュリティー許可 *(続き)* | | | |
|---|---|---|---|
| **必要な条件** | **キュー・オブジェクト (91 ページの『1』)** | **プロセス・オブジェクト** | **キュー・マネージャー・オブジェクト** |
| MQPMO_SET_ ALL_CONTEXT | MQZAO_SET_ ALL_CONTEXT **(92 ページの『11』)** | 適用外 | MQZAO_SET_ ALL_CONTEXT **(91 ページの『6』)** |
| (伝送キュー) **(91 ページの『8』)** | MQZAO_SET_ ALL_CONTEXT | 適用外 | MQZAO_SET_ ALL_CONTEXT **(91 ページの『6』)** |
| MQPMO_ALTERNATE_ USER_AUTHORITY | **(92 ページの『12』)** | 適用外 | MQZAO_ALTERNATE_ USER_AUTHORITY **(91 ページの『10』)** |

| 表 *11. MQCLOSE* 呼び出しに必要なセキュリティー許可 | | | |
|---|---|---|---|
| **必要な条件** | **キュー・オブジェクト (91 ページの『1』)** | **プロセス・オブジェクト** | **キュー・マネージャー・オブジェクト** |
| MQCO_DELETE | MQZAO_DELETE **(92 ページの『13』)** | 適用外 | 適用外 |
| MQCO_DELETE _PURGE | MQZAO_DELETE **(92 ページの『13』)** | 適用外 | 適用外 |

**表の注:**

1. モデル・キューをオープンする場合:
   - モデル・キューの場合、オープンするアクセスのタイプごとにモデル・キューをオープンするための権限に加えて、モデル・キューの場合は MQZAO_DISPLAY 権限が必要です。
   - 動的キューを作成する場合、MQZAO_CREATE 権限は必要ありません。
   - モデル・キューのオープンに使用したユーザー ID には、作成された動的キューに関するキュー特有のあらゆる権限が自動的に与えられます (MQZAO_ALL と同等)。
2. MQOO_INPUT_* も指定する必要があります。 これは、ローカル・キュー、モデル・キュー、または別名キューの場合に有効です。
3. この検査は、伝送キュー (注 91 ページの『8』 を参照) 以外は、すべての場合の出力において実行されます。
4. MQOO_OUTPUT も指定する必要があります。
5. このオプションは、MQOO_PASS_IDENTITY_CONTEXT も暗黙的に指定されます。
6. この権限は、キュー・マネージャー・オブジェクトと個々のキューの 両方に対して必要です。
7. MQOO_PASS_IDENTITY_CONTEXT、MQOO_PASS_ALL_CONTEXT、および MQOO_SET_IDENTITY_CONTEXT も、このオプションによって暗黙的に指定されます。
8. この検査は、*Usage* キュー属性として MQUS_TRANSMISSION を持ち、出力のために直接オープンされているローカル・キューまたはモデル・キューについて実行されます。 リモート・キューがオープンされる場合 (リモート・キュー・マネージャーとリモート・キューの名前を指定するか、リモート・キューのローカル定義の名前を指定して) は、この検査は適用されません。
9. MQOO_INQUIRE (あらゆるオブジェクト・タイプの場合)、または MQOO_BROWSE、MQOO_INPUT_*、MQOO_OUTPUT、または MQOO_SET (キューの場合) の中から、少なくとも 1 つを指定する必要があります。 検査は他の指定されたオプションの場合と同じで、提供されている代替ユーザー ID を使用し、特有の名前のあるオブジェクト権限と、MQZAO_ALTERNATE_USER_IDENTIFIER 検査の現行アプリケーション権限を調べます。
10. この許可では、任意の *AlternateUserId* を指定できます。

11. MQUS_TRANSMISSION の *Usage* キュー属性がないキューの場合は、MQZAO_OUTPUT 検査も行われます。
12. 検査は他の指定されたオプションの場合と同じで、提供されている代替ユーザー ID を使用し、特有の名前のあるキューの権限と、MQZAO_ALTERNATE_USER_IDENTIFIER 検査の現行アプリケーション権限を調べます。
13. この検査は、次の 2 つの条件が両方とも満たされた場合にのみ行われます。
    - 永続動的キューがクローズされて削除中である。
    - 使用中のオブジェクト・ハンドルを戻した MQOPEN 呼び出しが作成したキューではない。

    上記以外の場合は、検査は行われません。

## エスケープ *PCF* の中の *MQSC* コマンドに関する許可

ここでは、エスケープ PCF に含まれている各 MQSC コマンドで必要な権限をまとめます。

「適用外」は、この操作がこのオブジェクト・タイプには該当しないことを意味します。

コマンドを実行依頼するプログラムを実行させるユーザー ID には、以下の権限も必要になります。

- キュー・マネージャーに対する MQZAO_CONNECT 権限
- PCF コマンドを実行するためのキュー・マネージャー上の MQZAO_DISPLAY 権限
- エスケープ PCF コマンドのテキスト内で MQSC コマンドを発行する権限

**ALTER *object***

| **オブジェクト** | **必要な権限** |
|---|---|
| キュー | MQZAO_CHANGE |
| トピック | MQZAO_CHANGE |
| プロセス | MQZAO_CHANGE |
| キュー・マネージャー | MQZAO_CHANGE |
| 名前リスト | MQZAO_CHANGE |
| 認証情報 | MQZAO_CHANGE |
| チャネル | MQZAO_CHANGE |
| クライアント接続チャネル | MQZAO_CHANGE |
| リスナー | MQZAO_CHANGE |
| サービス | MQZAO_CHANGE |
| コミュニケーション情報 | MQZAO_CHANGE |

**CLEAR *object***

| **オブジェクト** | **必要な権限** |
|---|---|
| キュー | MQZAO_CLEAR |
| トピック | MQZAO_CLEAR |
| プロセス | 適用外 |
| キュー・マネージャー | 適用外 |
| 名前リスト | 適用外 |
| 認証情報 | 適用外 |
| チャネル | 適用外 |

| オブジェクト | 必要な権限 |
| --- | --- |
| クライアント接続チャネル | 適用外 |
| リスナー | 適用外 |
| サービス | 適用外 |
| コミュニケーション情報 | 適用外 |

**DEFINE *object* NOREPLACE (96 ページの『1』)**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_CREATE (96 ページの『2』) |
| トピック | MQZAO_CREATE (96 ページの『2』) |
| プロセス | MQZAO_CREATE (96 ページの『2』) |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_CREATE (96 ページの『2』) |
| 認証情報 | MQZAO_CREATE (96 ページの『2』) |
| チャネル | MQZAO_CREATE (96 ページの『2』) |
| クライアント接続チャネル | MQZAO_CREATE (96 ページの『2』) |
| リスナー | MQZAO_CREATE (96 ページの『2』) |
| サービス | MQZAO_CREATE (96 ページの『2』) |
| コミュニケーション情報 | MQZAO_CREATE (96 ページの『2』) |

**DEFINE *object* REPLACE (96 ページの『1』、96 ページの『3』)**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_CHANGE |
| トピック | MQZAO_CHANGE |
| プロセス | MQZAO_CHANGE |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_CHANGE |
| 認証情報 | MQZAO_CHANGE |
| チャネル | MQZAO_CHANGE |
| クライアント接続チャネル | MQZAO_CHANGE |
| リスナー | MQZAO_CHANGE |
| サービス | MQZAO_CHANGE |
| コミュニケーション情報 | MQZAO_CHANGE |

**DELETE *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_DELETE |
| トピック | MQZAO_DELETE |

| オブジェクト | 必要な権限 |
| --- | --- |
| プロセス | MQZAO_DELETE |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_DELETE |
| 認証情報 | MQZAO_DELETE |
| チャネル | MQZAO_DELETE |
| クライアント接続チャネル | MQZAO_DELETE |
| リスナー | MQZAO_DELETE |
| サービス | MQZAO_DELETE |
| コミュニケーション情報 | MQZAO_DELETE |

**DISPLAY *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_DISPLAY |
| トピック | MQZAO_DISPLAY |
| プロセス | MQZAO_DISPLAY |
| キュー・マネージャー | MQZAO_DISPLAY |
| 名前リスト | MQZAO_DISPLAY |
| 認証情報 | MQZAO_DISPLAY |
| チャネル | MQZAO_DISPLAY |
| クライアント接続チャネル | MQZAO_DISPLAY |
| リスナー | MQZAO_DISPLAY |
| サービス | MQZAO_DISPLAY |
| コミュニケーション情報 | MQZAO_DISPLAY |

**START *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | 適用外 |
| トピック | 適用外 |
| プロセス | 適用外 |
| キュー・マネージャー | 適用外 |
| 名前リスト | 適用外 |
| 認証情報 | 適用外 |
| チャネル | MQZAO_CONTROL |
| クライアント接続チャネル | 適用外 |
| リスナー | MQZAO_CONTROL |
| サービス | MQZAO_CONTROL |

| オブジェクト | 必要な権限 |
| --- | --- |
| コミュニケーション情報 | 適用外 |

**STOP *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | 適用外 |
| トピック | 適用外 |
| プロセス | 適用外 |
| キュー・マネージャー | 適用外 |
| 名前リスト | 適用外 |
| 認証情報 | 適用外 |
| チャネル | MQZAO_CONTROL |
| クライアント接続チャネル | 適用外 |
| リスナー | MQZAO_CONTROL |
| サービス | MQZAO_CONTROL |
| コミュニケーション情報 | 適用外 |

**チャネル・コマンド**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| PING CHANNEL | チャネル | MQZAO_CONTROL |
| RESET CHANNEL | チャネル | MQZAO_CONTROL_EXTENDED |
| RESOLVE CHANNEL | チャネル | MQZAO_CONTROL_EXTENDED |

**サブスクリプション・コマンド**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| ALTER SUB | トピック | MQZAO_CONTROL |
| DEFINE SUB | トピック | MQZAO_CONTROL |
| DELETE SUB | トピック | MQZAO_CONTROL |
| DISPLAY SUB | トピック | MQZAO_DISPLAY |

**Security Commands**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| SET AUTHREC | キュー・マネージャー | MQZAO_CHANGE |
| DELETE AUTHREC | キュー・マネージャー | MQZAO_CHANGE |
| DISPLAY AUTHREC | キュー・マネージャー | MQZAO_DISPLAY |
| DISPLAY AUTHSERV | キュー・マネージャー | MQZAO_DISPLAY |
| DISPLAY ENTAUTH | キュー・マネージャー | MQZAO_DISPLAY |
| SET CHLAUTH | キュー・マネージャー | MQZAO_CHANGE |

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| DISPLAY CHLAUTH | キュー・マネージャー | MQZAO_DISPLAY |
| REFRESH SECURITY | キュー・マネージャー | MQZAO_CHANGE |

**Status Displays**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| DISPLAY CHSTATUS | キュー・マネージャー | MQZAO_DISPLAY<br>チャネル・タイプが CLUSSDR の場合は、伝送キューに対する +inq 権限 (または同等の MQZAO_INQUIRE) が必要です。 |
| DISPLAY LSSTATUS | キュー・マネージャー | MQZAO_DISPLAY |
| DISPLAY PUBSUB | キュー・マネージャー | MQZAO_DISPLAY |
| DISPLAY SBSTATUS | キュー・マネージャー | MQZAO_DISPLAY |
| DISPLAY SVSTATUS | キュー・マネージャー | MQZAO_DISPLAY |
| DISPLAY TPSTATUS | キュー・マネージャー | MQZAO_DISPLAY |

**クラスター・コマンド**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| DISPLAY CLUSQMGR | キュー・マネージャー | MQZAO_DISPLAY |
| REFRESH CLUSTER | 「mqm」グループ・メンバーシップが必要 | |
| RESET CLUSTER | 「mqm」グループ・メンバーシップが必要 | |
| SUSPEND QMGR | 「mqm」グループ・メンバーシップが必要 | |
| RESUME QMGR | 「mqm」グループ・メンバーシップが必要 | |

**Other Administrative Commands**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| PING QMGR | キュー・マネージャー | MQZAO_DISPLAY |
| REFRESH QMGR | キュー・マネージャー | MQZAO_CHANGE |
| RESET QMGR | キュー・マネージャー | MQZAO_CHANGE |
| DISPLAY CONN | キュー・マネージャー | MQZAO_DISPLAY |
| STOP CONN | キュー・マネージャー | MQZAO_CHANGE |

**注 :**

1. DEFINE コマンドでは、LIKE オブジェクトが指定されている場合は LIKE オブジェクトに関する、または LIKE が省略されている場合は適切な SYSTEM.DEFAULT.xxx オブジェクトに関する、MQZAO_DISPLAY 権限も必要です。
2. MQZAO_CREATE 権限は、特定のオブジェクトまたはオブジェクト・タイプに特有のものではありません。 `setmqaut` コマンドで QMGR のオブジェクト・タイプを指定すれば、指定したキュー・マネージャーのすべてのオブジェクトについて作成権限が与えられます。
3. これは、置き換えようとするオブジェクトがすでに存在している場合に適用されます。 存在していない場合は、DEFINE *object* NOREPLACE の検査になります。

**関連情報**
クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

## *PCF* コマンドについての許可

ここでは、PCF コマンドごとに必要な許可について要約します。

「検査しない」は、権限の検査が行われないことを意味します。「適用外」は、この操作がこのオブジェクト・タイプには該当しないことを意味します。

コマンドを実行依頼するプログラムを実行させるユーザー ID には、以下の権限も必要になります。

- キュー・マネージャーに対する MQZAO_CONNECT 権限
- PCF コマンドを実行するためのキュー・マネージャー上の MQZAO_DISPLAY 権限

特殊権限 MQZAO_ALL_ADMIN には、以下のリストのとおり、特定のオブジェクトまたはオブジェクト・タイプに固有でないオブジェクト・タイプ (MQZAO_CREATE を除く) に関連するすべての権限が含まれています。

**Change *object***

| **オブジェクト** | **必要な権限** |
|---|---|
| キュー | MQZAO_CHANGE |
| トピック | MQZAO_CHANGE |
| プロセス | MQZAO_CHANGE |
| キュー・マネージャー | MQZAO_CHANGE |
| 名前リスト | MQZAO_CHANGE |
| 認証情報 | MQZAO_CHANGE |
| チャネル | MQZAO_CHANGE |
| クライアント接続チャネル | MQZAO_CHANGE |
| リスナー | MQZAO_CHANGE |
| サービス | MQZAO_CHANGE |
| 通信情報 | MQZAO_CHANGE |

**Clear *object***

| **オブジェクト** | **必要な権限** |
|---|---|
| キュー | MQZAO_CLEAR |
| トピック | MQZAO_CLEAR |
| プロセス | 適用外 |
| キュー・マネージャー | 適用外 |
| 名前リスト | 適用外 |
| 認証情報 | 適用外 |
| チャネル | 適用外 |
| クライアント接続チャネル | 適用外 |
| リスナー | 適用外 |
| サービス | 適用外 |
| コミュニケーション情報 | 適用外 |

**オブジェクト のコピー (置換なし) ( 1)**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_CREATE ( 2) |
| トピック | MQZAO_CREATE ( 2) |
| プロセス | MQZAO_CREATE ( 2) |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_CREATE ( 2) |
| 認証情報 | MQZAO_CREATE ( 2) |
| チャネル | MQZAO_CREATE ( 2) |
| クライアント接続チャネル | MQZAO_CREATE ( 2) |
| リスナー | MQZAO_CREATE ( 2) |
| サービス | MQZAO_CREATE ( 2) |
| 通信情報 | MQZAO_CREATE (102 ページの『2』) |

**オブジェクト のコピー (置換あり) ( 1, 4)**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_CHANGE |
| トピック | MQZAO_CHANGE |
| プロセス | MQZAO_CHANGE |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_CHANGE |
| 認証情報 | MQZAO_CHANGE |
| チャネル | MQZAO_CHANGE |
| クライアント接続チャネル | MQZAO_CHANGE |
| リスナー | MQZAO_CHANGE |
| サービス | MQZAO_CHANGE |
| 通信情報 | MQZAO_CHANGE |

**オブジェクト の作成 (置換なし) ( 3)**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_CREATE ( 2) |
| トピック | MQZAO_CREATE ( 2) |
| プロセス | MQZAO_CREATE ( 2) |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_CREATE ( 2) |
| 認証情報 | MQZAO_CREATE ( 2) |
| チャネル | MQZAO_CREATE ( 2) |

| オブジェクト | 必要な権限 |
| --- | --- |
| クライアント接続チャネル | MQZAO_CREATE **(2)** |
| リスナー | MQZAO_CREATE **(2)** |
| サービス | MQZAO_CREATE **(2)** |
| 通信情報 | MQZAO_CREATE **(2)** |

**オブジェクト の作成 (置換あり) (3, 4)**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_CHANGE |
| トピック | MQZAO_CHANGE |
| プロセス | MQZAO_CHANGE |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_CHANGE |
| 認証情報 | MQZAO_CHANGE |
| チャネル | MQZAO_CHANGE |
| クライアント接続チャネル | MQZAO_CHANGE |
| リスナー | MQZAO_CHANGE |
| サービス | MQZAO_CHANGE |
| 通信情報 | MQZAO_CHANGE |

**Delete *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_DELETE |
| トピック | MQZAO_DELETE |
| プロセス | MQZAO_DELETE |
| キュー・マネージャー | 適用外 |
| 名前リスト | MQZAO_DELETE |
| 認証情報 | MQZAO_DELETE |
| チャネル | MQZAO_DELETE |
| クライアント接続チャネル | MQZAO_DELETE |
| リスナー | MQZAO_DELETE |
| サービス | MQZAO_DELETE |
| 通信情報 | MQZAO_DELETE |

**Inquire *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | MQZAO_DISPLAY |
| トピック | MQZAO_DISPLAY |

| オブジェクト | 必要な権限 |
| --- | --- |
| プロセス | MQZAO_DISPLAY |
| キュー・マネージャー | MQZAO_DISPLAY |
| 名前リスト | MQZAO_DISPLAY |
| 認証情報 | MQZAO_DISPLAY |
| チャネル | MQZAO_DISPLAY |
| クライアント接続チャネル | MQZAO_DISPLAY |
| リスナー | MQZAO_DISPLAY |
| サービス | MQZAO_DISPLAY |
| 通信情報 | MQZAO_DISPLAY |

**Inquire *object* names**

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | 検査しない |
| トピック | 検査しない |
| プロセス | 検査しない |
| キュー・マネージャー | 検査しない |
| 名前リスト | 検査しない |
| 認証情報 | 検査しない |
| チャネル | 検査しない |
| クライアント接続チャネル | 検査しない |
| リスナー | 検査しない |
| サービス | 検査しない |
| コミュニケーション情報 | 検査しない |

**Start *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | 適用外 |
| トピック | 適用外 |
| プロセス | 適用外 |
| キュー・マネージャー | 適用外 |
| 名前リスト | 適用外 |
| 認証情報 | 適用外 |
| チャネル | MQZAO_CONTROL |
| クライアント接続チャネル | 適用外 |
| リスナー | MQZAO_CONTROL |
| サービス | MQZAO_CONTROL |

| オブジェクト | 必要な権限 |
| --- | --- |
| コミュニケーション情報 | 適用外 |

**Stop *object***

| オブジェクト | 必要な権限 |
| --- | --- |
| キュー | 適用外 |
| トピック | 適用外 |
| プロセス | 適用外 |
| キュー・マネージャー | 適用外 |
| 名前リスト | 適用外 |
| 認証情報 | 適用外 |
| チャネル | MQZAO_CONTROL |
| クライアント接続チャネル | 適用外 |
| リスナー | MQZAO_CONTROL |
| サービス | MQZAO_CONTROL |
| コミュニケーション情報 | 適用外 |

**チャネル・コマンド**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Ping Channel | チャネル | MQZAO_CONTROL |
| Reset Channel | チャネル | MQZAO_CONTROL_EXTENDED |
| Resolve Channel | チャネル | MQZAO_CONTROL_EXTENDED |

**サブスクリプション・コマンド**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Change Subscription | トピック | MQZAO_CONTROL |
| Create Subscription | トピック | MQZAO_CONTROL |
| Delete Subscription | トピック | MQZAO_CONTROL |
| Inquire Subscription | トピック | MQZAO_DISPLAY |

**Security Commands**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Set Authority Record | キュー・マネージャー | MQZAO_CHANGE |
| Delete Authority Record | キュー・マネージャー | MQZAO_CHANGE |
| Inquire Authority Records | キュー・マネージャー | MQZAO_DISPLAY |
| Inquire Authority Service | キュー・マネージャー | MQZAO_DISPLAY |
| Inquire Entity Authority | キュー・マネージャー | MQZAO_DISPLAY |
| Set Channel Authentication Record | キュー・マネージャー | MQZAO_CHANGE |

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Inquire Channel Authentication Records | キュー・マネージャー | MQZAO_DISPLAY |
| Refresh Security | キュー・マネージャー | MQZAO_CHANGE |

**Status Displays**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Inquire Channel Status | キュー・マネージャー | MQZAO_DISPLAY<br>チャネル・タイプが CLUSSDR の場合は、伝送キューに対する +inq 権限 (または同等の MQZAO_INQUIRE) が必要です。 |
| Inquire Channel Listener Status | キュー・マネージャー | MQZAO_DISPLAY |
| Inquire Pub/Sub Status | キュー・マネージャー | MQZAO_DISPLAY |
| Inquire Subscription Status | キュー・マネージャー | MQZAO_DISPLAY |
| Inquire Service Status | キュー・マネージャー | MQZAO_DISPLAY |
| Inquire Topic Status | キュー・マネージャー | MQZAO_DISPLAY |

**クラスター・コマンド**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Inquire Cluster Queue Manager | キュー・マネージャー | MQZAO_DISPLAY |
| Refresh Cluster | 「mqm」グループ・メンバーシップが必要 | |
| Reset Cluster | 「mqm」グループ・メンバーシップが必要 | |
| Suspend Queue Manager Cluster | 「mqm」グループ・メンバーシップが必要 | |
| Resume Queue Manager Cluster | 「mqm」グループ・メンバーシップが必要 | |

**Other Administrative Commands**

| コマンド | オブジェクト | 必要な権限 |
| --- | --- | --- |
| Ping Queue Manager | キュー・マネージャー | MQZAO_DISPLAY |
| Refresh Queue Manager | キュー・マネージャー | MQZAO_CHANGE |
| Reset Queue Manager | キュー・マネージャー | MQZAO_CHANGE |
| Reset Queue Statistics | キュー | MQZAO_DISPLAY および MQZAO_CHANGE |
| Inquire Connection | キュー・マネージャー | MQZAO_DISPLAY |
| Stop Connection | キュー・マネージャー | MQZAO_CHANGE |

**注 :**

1. Copy コマンドでは、From オブジェクトに関する MQZAO_DISPLAY 権限も必要です。
2. MQZAO_CREATE 権限は、特定のオブジェクトまたはオブジェクト・タイプに特有のものではありません。 `setmqaut` コマンドで QMGR のオブジェクト・タイプを指定すれば、指定したキュー・マネージャーのすべてのオブジェクトについて作成権限が与えられます。

3. 作成コマンドの場合は、該当する SYSTEM.DEFAULT.* オブジェクト。
4. これは、置き換えようとするオブジェクトがすでに存在している場合に適用されます。 存在しない場合は、Copy または Create (置き換えなし) と同じ検査になります。

## Windows のセキュリティーに関する特別な考慮事項

Windows では、バージョンによって一部のセキュリティー機能の動作が異なる場合があります。

IBM WebSphere MQ セキュリティーは、ユーザー許可およびグループ・メンバーシップについての情報について、オペレーティング・システム API への呼び出しに依存しています。 いくつかの機能は、Windows システムで同じように動作しません。 この一連のトピックでは、 Windows 環境で IBM WebSphere MQ を実行する場合に、これらの違いが IBM WebSphere MQ セキュリティーにどのように影響する可能性があるかについて説明します。

### *SSPI* チャネル出口プログラム

WebSphere MQ for Windows には、メッセージ・チャネルと MQI チャネルの両方で使用できるセキュリティー出口プログラムが組み込まれています。 その出口のソース・コードとオブジェクト・コードが用意されており、片方向と両方向の認証が可能です。

このセキュリティー出口は、Windows プラットフォームの統合セキュリティー機能を提供するセキュリティー・サポート・プロバイダー・インターフェース (SSPI) を使用します。

セキュリティー出口は、次の識別と認証サービスを提供します。

**単方向認証**

これは、Windows NT LAN Manager (NTLM) の認証サポートを使用します。 NTLM により、サーバーは、クライアントを認証できるようになります。 クライアントがサーバーを認証したり、あるサーバーが別のサーバーを認証したりすることは許可しません。 NTLM は、サーバーが本物であることを前提とするネットワーク環境用に設計されています。 NTLM は、 WebSphere MQ バージョン 7.0 でサポートされるすべての Windows プラットフォームでサポートされます。

このサービスは、一般に、サーバー・キュー・マネージャーが WebSphereMQ MQI クライアント・アプリケーションを認証できるようにするために、MQI チャネル上で使用されます。 クライアント・アプリケーションは、実行中のプロセスに関連したユーザー ID によって識別されます。

この認証を実行するには、チャネルのクライアント側にあるセキュリティー出口が、NTLM から認証トークンを取得し、そのトークンをセキュリティー・メッセージ内で、チャネルの相手側のセキュリティー出口に送信します。 相手側のセキュリティー出口は、そのトークンを NTLM に渡し、NTLM が、そのトークンが本物であるかどうかを検査します。 相手側のセキュリティー出口は、トークンの確実性を確信できない場合、チャネルをクローズするように MCA に指示します。

**両方向認証または相互認証**

これは、Kerberos 認証サービスを使用します。 Kerberos プロトコルは、ネットワーク環境内のサーバーが本物であることを前提としません。 サーバーは、クライアントやその他のサーバーを認証することができ、クライアントはサーバーを認証できます。 Kerberos は、 WebSphere MQ バージョン 7.0 によってサポートされるすべての Windows プラットフォームでサポートされます。

このサービスは、メッセージ・チャネルと MQI チャネルの両方で使用できます。 メッセージ・チャネル上では、2 つのキュー・マネージャーの相互認証を提供します。 MQI チャネル上では、サーバー・キュー・マネージャーと WebSphere MQ MQI クライアント・アプリケーションが、互いに認証できるようにします。 キュー・マネージャーは、ストリング `ibmMQSeries/` を接頭部として持つ名前で識別されます。 クライアント・アプリケーションは、実行中のプロセスに関連したユーザー ID によって識別されます。

この相互認証を実行するため、開始側のセキュリティー出口は Kerberos セキュリティー・サーバーから認証トークンを取得し、そのトークンをセキュリティー・メッセージ内で相手側に送信します。 相手側のセキュリティー出口は、トークンを Kerberos サーバーに渡し、本物であるかどうかを検査します。 Kerberos セキュリティー・サーバーは 2 番目のトークンを生成し、相手側はそれをセキュリティー・メッセージ内で開始側のセキュリティー出口に送信します。 次に、開始側のセキュリティー出口は、2 番目のトークンが本物であるかどうかを検査するよう Kerberos サーバーに要求します。 このやりとりの間、一方のセキュリティー出口が他方のセキュリティー出口によって送信されたトークンの確

実性を確信できない場合、そのセキュリティー出口は、チャネルをクローズするように MCA に指示します。

セキュリティー出口は、ソース形式とオブジェクト形式の両方で提供されます。 独自のチャネル出口プログラムを作成するための開始点として、ソース・コードを使用するか、あるいは提供されたオブジェクト・モジュールを使用することができます。 オブジェクト・モジュールには、2 つの入り口点があります。1 つは、NTLM 認証サポートを使用する単方向認証用であり、もう 1 つは、Kerberos 認証サービスを使用する両方向認証用です。

SSPI チャネル出口プログラムの機能の詳細や実装方法については、Windows システムでの SSPI セキュリティー出口の使用を参照してください。

## *Windows* で「*group not found*」エラーが表示される場合

この問題は、 Windows サーバーがドメイン・コントローラーにプロモートまたはドメイン・コントローラーからデモートされると、 WebSphere MQ がローカル mqm グループにアクセスできなくなるために発生する可能性があります。 この問題を解決するには、ローカル mqm グループを再作成します。

症状は、ローカル mqm グループがないことを示すエラーとして表示されます。例えば、次のように表示されます。

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

WebSphere MQ はローカルに定義された mqm グループを使用するため、サーバーとドメイン・コントローラーの間のマシンの状態を変更すると、WebSphere MQ の運用に影響する可能性があります。 サーバーがプロモートしてドメイン・コントローラーになると、スコープがローカルからドメイン・ローカルに変わります。 このマシンがサーバーにデモートすると、すべてのドメイン・ローカル・グループが除去されます。 すなわち、マシンをサーバーからドメイン・コントローラーに変更して再びサーバーに戻すと、ローカル mqm グループへのアクセスが失われてしまいます。

この問題を解決するには、標準の Windows 管理ツールを使用してローカル mqm グループを再作成します。すべてのグループ・メンバーシップ情報が消失するため、新しく作成したローカル mqm グループに、特権のある WebSphere MQ ユーザーを復元する必要があります。 マシンがドメイン・メンバーである場合、ドメイン mqm グループをローカル mqm グループに追加し、特権のあるドメイン WebSphere MQ ユーザー ID に、必要なレベルの権限を与える必要もあります。

## *Windows* 上の *IBM WebSphere MQ* およびドメイン・コントローラーで問題が発生した場合

Windows サーバーがドメイン・コントローラーにプロモートするときに、セキュリティーの設定で問題が生じる可能性があります。

Windows 2000、Windows 2003、Windows Server 2008 サーバーをドメイン・コントローラーにプロモートさせるときには、ユーザーおよびグループ許可に関連して、デフォルトのセキュリティー設定かデフォルト以外のセキュリティー設定を選択するオプションが示されます。 このオプションは、任意のユーザーが Active Directory からグループ・メンバーシップを取り出せるかどうかを制御します。 WebSphere MQ は、セキュリティー・ポリシーをインプリメントするために、グループ・メンバーシップ情報に依存しているので、WebSphere MQ の運用を行っているユーザー ID が、他のユーザーのグループ・メンバーシップを判別できることは重要です。

Windows 2000 でデフォルトのセキュリティー・オプションを使用してドメインが作成される場合、インストール・プロセス中に WebSphere MQ によって作成されたデフォルトのユーザー ID は、必要に応じて、他のユーザーのグループ・メンバーシップを取得することができます。 その後、製品は通常どおりにインストールされ、デフォルトのオブジェクトを作成し、(必要であれば) キュー・マネージャーは、ローカルおよびドメイン・ユーザーのアクセス権限を判別します。

Windows 2000 でデフォルト以外のセキュリティー・オプションを使用してドメインが作成される場合、または、Windows 2003 または Windows Server 2008 でデフォルト・セキュリティー・オプションを使用してドメインが作成される場合に、インストール中に WebSphere MQ によって作成されるユーザー ID が常に必要なグループ・メンバーシップを決定するわけではありません。 この場合、以下の点についての知識が必要です。

- Windows 2000 のデフォルト以外のセキュリティー権限、または Windows 2003 および Windows Server 2008 のデフォルト・セキュリティー権限の動作
- ドメイン mqm グループがグループ・メンバーシップを読み取れるようにする方法
- ドメイン・ユーザーの下で実行する IBM WebSphere MQ Windows サービスを構成する方法

デフォルト以外のセキュリティー権限を持つ *Windows 2000* ドメインとデフォルト・セキュリティー権限を持つ *Windows 2003* および *Windows Server 2008* ドメイン
これらのオペレーティング・システムでは、WebSphere MQ をインストールするのがローカル・ユーザーかドメイン・ユーザーかによって、インストールの動作が変わります。

**ローカル**・ユーザーが WebSphere MQ をインストールする場合、WebSphere MQ の準備ウィザードは、IBM WebSphere MQ Windows サービス用に作成したローカル・ユーザーが、インストールしているユーザーのグループ・メンバーシップ情報を取り出せることを検出します。 WebSphere MQ の準備ウィザードは、ネットワーク構成についてユーザーに質問を出し、Windows 2000 以降で稼働するドメイン・コントローラーに定義された別のユーザー・アカウントがあるかどうかを判別します。 ある場合、IBM WebSphere MQ Windows サービスは、特定の設定と権限を持つドメイン・ユーザー・アカウントの下で実行する必要があります。 WebSphere MQ の準備ウィザードは、このユーザーのアカウント詳細の入力を求めてきます。 オンライン・ヘルプには、ドメイン管理者に送信することのできる、必要なドメイン・ユーザー・アカウントの詳細が示されています。

**ドメイン**・ユーザーが WebSphere MQ をインストールする場合、WebSphere MQ の準備ウィザードは、IBM WebSphere MQ Windows サービス用に作成したローカル・ユーザーが、インストールしているユーザーのグループ・メンバーシップ情報を取り出せないことを検出します。 この場合、WebSphere MQ の準備ウィザードは必ず、使用する IBM WebSphere MQ Windows サービスのドメイン・ユーザー・アカウントのアカウント詳細を入力するようユーザーに求めてきます。

IBM WebSphere MQ Windows サービスでドメイン・ユーザー・アカウントを使用する必要がある場合、WebSphere MQ の準備ウィザードを使用してアカウントを構成し終わるまで、WebSphere MQ は正常に機能できません。 WebSphere MQ 準備ウィザードでは、適切なアカウントを使用して Windows サービスを構成してしまうまで、他の作業を続けることはできません。

Windows 2000 ドメインがデフォルト以外のセキュリティー権限で構成されている場合、WebSphere MQ を正常に稼働させるための通常の解決方法としては、前述のとおり、適切なドメイン・ユーザー・アカウントを使って WebSphere MQ を構成します。

詳細については、『WebSphere MQ のドメイン・アカウントの作成とセットアップ』を参照してください。

*Windows* 上のドメイン・ユーザーの下で実行するための *IBM WebSphere MQ* サービスの構成
IBM WebSphere MQ の準備ウィザードを使用して、ユーザー・アカウントのアカウント詳細を入力します。あるいは、「コンピューターの管理」パネルを使用して、インストール済み環境固有の IBM WebSphere MQ サービスの 「**ログオン**」 の詳細を変更することもできます。

詳しくは、 IBM WebSphere MQ Windows サービス・ユーザー・アカウントのパスワードの変更 を参照してください。

## *Windows* にセキュリティー・テンプレート・ファイルを適用する操作

テンプレートを適用すると、WebSphere MQ のファイルとディレクトリーに適用されるセキュリティー設定が影響を受ける可能性があります。 高セキュア・テンプレートを使用する場合は、WebSphere MQ をインストールする前に適用してください。

Windows では、テキスト・ベースのセキュリティー・テンプレート・ファイルがサポートされており、これを使用して、Security Configuration and Analysis MMC スナップインを持つ 1 つ以上のコンピューターに、統一されたセキュリティー設定を適用することができます。 特に、Windows には、特定レベルのセキュリティーを実現することを目的として、特定範囲のセキュリティー設定を備えたいくつかのテンプレートが備えられています。 具体的には、互換、セキュア、高セキュアのテンプレートがあります。

このいずれかのテンプレートを適用すると、WebSphere MQ のファイルとディレクトリーに適用されるセキュリティー設定が影響を受ける可能性があります。 高セキュア・テンプレートを使用する場合は、WebSphere MQ をインストールする前にマシンを構成してください。

WebSphere MQ が既にインストールされているマシンに高セキュア・テンプレートを適用すると、WebSphere MQ のファイルとディレクトリーに対して設定されているすべてのアクセス権が削除されま

す。 それらのアクセス権が削除されれば、エラー・ディレクトリーに対する *Administrator*、*mqm*、*Everyone* (該当する場合) の各グループのアクセス権を失うことになります。

### ネストされたグループ

ネストされたグループの使用には、制限があります。 これらには、ドメイン機能レベルに由来するものと、WebSphere MQ の制限に由来するものがあります。

Active Directory は、ドメイン機能レベルに応じて、ドメイン・コンテキスト内のさまざまなグループ・タイプをサポートできます。 デフォルトでは、Windows 2003 ドメインは *Windows 2000* 混在 機能レベルになります。 (Windows server 2003、Windows XP、Windows Vista、および Windows Server 2008 はすべて、Windows 2003 ドメイン・モデルに従います。) ドメイン機能レベルは、ドメイン環境内でのユーザー ID の構成時に許可される、サポートされるグループ・タイプおよびネストのレベルを決定します。 Group Scope および組み込み基準については、Active Directory の資料を参照してください。

Active Directory 要件の他に、WebSphere MQ によって使用される ID に関する制限があります。 WebSphere MQ によって使用されるネットワーク API では、ドメイン機能レベルでサポートされているすべての構成がサポートされている訳ではありません。 そのため WebSphere MQ は、ローカル・グループ内でネストされている、ドメイン・ローカル・グループにあるドメイン ID のグループ・メンバーシップを照会することはできません。 さらに、グローバルおよびユニバーサル・グループの複数ネストはサポートされていません。 ただし、直近にネストされたグローバルおよびユニバーサル・グループはサポートされてます。

### *IBM WebSphere MQ* に接続する *Windows* アプリケーションの追加権限の構成

アプリケーション・プロセスに対する SYNCHRONIZE アクセスが認められるようにするには、IBM WebSphere MQ プロセスを実行するアカウントで追加の権限が必要になる場合があります。

通常より高いセキュリティー・レベルで実行するように構成された Windows アプリケーション (ASP ページなど) が、IBM WebSphere MQ に接続 する場合、問題が発生する可能性があります。

IBM WebSphere MQ は、特定のアクションを調整するために、アプリケーション・プロセスへの SYNCHRONIZE アクセスを必要とします。 APAR IC35116 により、適切な特権が指定されるように IBM WebSphere MQ が変更されました。 ただし、要求されたアクセスが許可されるようにするために、IBM WebSphere MQ プロセスを実行するアカウントで追加権限が必要となる場合があります。

サーバー・アプリケーションが初めてキュー・マネージャーに接続しようとすると、IBM WebSphere MQ がプロセスを変更して、IBM WebSphere MQ 管理者に SYNCHRONIZE 権限を付与します。 IBM WebSphere MQ プロセスが実行されているユーザー ID に対して追加権限を構成するには、以下のステップを完了します。

1. ローカル・セキュリティー・ポリシー・ツールを開始し、「セキュリティー設定」->「ローカル・ポリシー」->「ユーザーの権利の割り当て」をクリックし、「プログラムのデバッグ」をクリックします。
2. 「プログラムのデバッグ」をダブルクリックしてから、自分の IBM WebSphere MQ ユーザー ID をリストに追加します。

システムが Windows ドメイン内にあり、有効なポリシー設定がまだ設定されていない場合、ローカル・ポリシー設定が指定されていても、「ドメイン セキュリティ ポリシー」ツールを使用して、ドメイン・レベルでも 同様にユーザー ID へ許可を与える必要があります。

## HP Integrity NonStop Server でのセキュリティーのセットアップ

HP Integrity NonStop Server システムに固有のセキュリティーに関する考慮事項。

IBM WebSphere MQ Client for HP Integrity NonStop Server は、Transport Layer Security (TLS) プロトコルと Secure Sockets Layer (SSL) プロトコルの両方をサポートして、キュー・マネージャーに接続するときに、リンク・レベルのセキュリティーを提供します。 これらのプロトコルは、OpenSSL の実装を使用することによってサポートされます。 OpenSSL には、強力な暗号操作を提供するため乱数データのソースが必要です。

### OpenSSL

IBM WebSphere MQ Client for HP Integrity NonStop Server の OpenSSL セキュリティーの概要。

OpenSSL ツールキットは、ネットワーク上のセキュア通信用の Secure Sockets Layer (SSL) および Transport Layer Security (TLS) プロトコルのオープン・ソース実装です。

このツールキットは、OpenSSL Project によって開発されています。 OpenSSL Project の詳細については、https://www.openssl.org を参照してください。 IBM WebSphere MQ Client for HP Integrity NonStop Server には、OpenSSL ライブラリーおよび **openssl** コマンドの変更されたバージョンが含まれています。 このライブラリーおよび **openssl** コマンドは、OpenSSL ツールキット 1.0.1c から移植されており、オブジェクト・コードとしてのみ提供されます。 ソース・コードは提供されません。

OpenSSL ライブラリーは、必要に応じて、IBM WebSphere MQ クライアント・アプリケーションのプログラムによって動的にロードされます。 IBM WebSphere MQ によって提供される OpenSSL ライブラリーのみが、IBM WebSphere MQ クライアント・アプリケーションの使用のためにサポートされます。

証明書の管理目的で使用できる **openssl** コマンドは、OSS ディレクトリー `opt_installation_path/opt/mqm/bin` にインストールされます。

**openssl** コマンドを使用して、さまざまな一般的なデータ形式で鍵とデジタル証明書を作成および管理できます。また、簡単な認証局 (CA) のタスクを実行できます。

OpenSSL によって処理される鍵および証明書データのデフォルトの形式は、Privacy Enhanced Mail (PEM) 形式です。 PEM 形式のデータは、Base64 エンコードの ASCII データです。 したがって、データは、E メールなどのテキスト・ベース・システムを使用して転送したり、テキスト・エディターや Web ブラウザーを使用してカット・アンド・ペーストしたりすることができます。 PEM は、テキスト・ベース暗号交換のインターネット標準であり、Internet RFC 1421、1422、1423、および 1424 で明記されています。 IBM WebSphere MQ は、拡張子が `.pem` のファイルに PEM 形式のデータが含まれていると想定します。 PEM 形式のファイルには、複数の証明書および他のエンコード・オブジェクトを入れることができ、コメントも含めることができます。

他のオペレーティング・システム上の IBM WebSphere MQ SSL サポートでは、ファイル内の鍵および証明書データを Distinguished Encoding Rule (DER) を使用してエンコードしなければならない場合があります。 DER は、ASN.1 表記をセキュア通信で使用するためのエンコード・ルールのセットです。 DER を使用してエンコードされるデータはバイナリー・データで、DER を使用してエンコードされる鍵および証明書データの形式は PKCS#12 または PFX とも呼ばれます。 一般に、このデータを含むファイルの拡張子は `.p12` または `.pfx` です。 **openssl** コマンドは、PEM 形式と PKCS#12 形式の間で変換を行うことができます。

## エントロピー・デーモン

OpenSSL には、強力な暗号操作を提供するため乱数データのソースが必要です。 乱数の生成は、通常は、オペレーティング・システムまたはシステム全体のデーモン・プロセスによって提供される機能です。 HP Integrity NonStop Server オペレーティング・システムは、オペレーティング・システム内でこの機能を提供しません。

IBM WebSphere MQ Client for HP Integrity NonStop Server によって提供される SSL および TLS サポートを使用するとき、乱数データのソースを提供するためにエントロピー・デーモンと呼ばれるプロセスが必要です。 SSL または TLS を要求するクライアント・チャネルを開始するときに、OpenSSL は、エントロピー・デーモンが実行され、`/etc/egd-pool` にある OSS ファイル・システムのソケットでサービスが提供されていることを想定します。

エントロピー・デーモンは、IBM WebSphere MQ Client for HP Integrity NonStop Server によって提供されません。 IBM WebSphere MQ Client for HP Integrity NonStop Server は、以下のエントロピー・デーモンでテストされます。

- amqjkdm0 (IBM WebSphere MQ 5.3 サーバーで提供)
- `/usr/local/bin/prngd` (バージョン 0.9.27、HP Integrity NonStop Server オープン・ソース技術ライブラリーによって提供)

# IBM WebSphere MQ MQI クライアント・セキュリティーのセットアップ

クライアント・アプリケーションがサーバー上のリソースへ無制限にアクセスしないように、IBM WebSphere MQ MQI クライアント・セキュリティーについて考慮する必要があります。

クライアント・アプリケーションの実行時には、必要以上のアクセス権を持つユーザー ID を使用してそのアプリケーションを実行しないでください。例えば、mqm グループ内のユーザーや mqm ユーザー自体も該当します。

アクセス権が過度に多いユーザーとしてアプリケーションを実行すると、アプリケーションがキュー・マネージャーの一部に対して故意または不慮にアクセスしたり変更したりするリスクが生じます。

クライアント・アプリケーションとそのキュー・マネージャー・サーバー間のセキュリティーには、認証およびアクセス管理という 2 つの局面があります。

- 認証を使用して、特定のユーザーとして実行しているクライアント・アプリケーションが本人であることを確認できます。 認証を使用すると、アタッカーがいずれかのアプリケーションの偽名を使用してキュー・マネージャーに対するアクセス権を獲得しないようにすることができます。

  SSL または TLS 内で相互認証を使用する必要があります。 詳細については、111 ページの『SSL または TLS の取り扱い』
- アクセス管理を使用して、特定のユーザーかユーザーのグループに関するアクセス権を付与したり削除したりできます。 特別に作成したユーザー (または、特定のグループ内のユーザー) を使用してクライアント・アプリケーションを実行すると、アクセス管理を使用して、そのアプリケーションが想定外のキュー・マネージャーの部分にアクセスできないようにすることができます。

  アクセス管理のセットアップ時には、チャネル認証規則とチャネル上の MCAUSER フィールドを考慮しなければなりません。 これらのフィーチャーは両方とも、アクセス管理権限の検証に使用するユーザー ID を変更できます。

  アクセス管理の詳細については、159 ページの『オブジェクトに対するアクセス権限の設定』を参照してください。

制限付き ID を使用して特定のチャネルに接続するようにクライアント・アプリケーションをセットアップしているものの、そのチャネルの MCAUSER フィールドで管理者 ID が設定されている場合には、クライアント・アプリケーションが正常に接続すると、管理者 ID を使用してアクセス管理が検査されます。 したがって、クライアント・アプリケーションはキュー・マネージャーに対する全アクセス権限を持ちます。

MCAUSER 属性の詳細については、185 ページの『MCAUSER ユーザー ID への表明済みユーザー ID のマッピング』を参照してください。

チャネル認証規則をキュー・マネージャーに対するアクセス管理方式として使用することもできます。 この場合は、受諾する接続に関する特定の規則と基準をセットアップします。

チャネル認証規則の詳細については、39 ページの『チャネル認証レコード』を参照してください。

## MQI クライアントでの実行時に FIPS 認定の CipherSpec のみを使用するように指定する

FIPS 準拠のソフトウェアを使用して鍵リポジトリーを作成し、次にチャネルで FIPS 認定の CipherSpec を使用しなければならないことを指定します。

実行時に FIPS 準拠になるようにするには、鍵リポジトリーが、-fips オプションを指定した runmqakm などの、FIPS 準拠のソフトウェアのみを使用して作成および管理されている必要があります。

以下の 3 つの方法 (優先順にリストしています) で、SSL チャネルまたは TLS チャネルで FIPS 認定の CipherSpec のみを使用しなければならないことを指定できます。

1. MQSCO 構造体の FipsRequired フィールドを MQSSL_FIPS_YES に設定する。
2. 環境変数 MQSSLFIPS を YES に設定する。
3. クライアント構成ファイルの SSLFipsRequired 属性を YES に設定する。

デフォルトでは、FIPS 認定の暗号方式は必要ありません。

これらの値の意味は、ALTER QMGR SSLFIPS で同等のパラメーター値が持つ意味と同じです (ALTER QMGR を参照)。 現在、アクティブな SSL 接続または TLS 接続がクライアント・プロセスに存在せず、FipsRequired の値が SSL MQCONNX に正しく指定されている場合、それ以降にこのプロセスと関連して行われる SSL 接続では、この値に関連付けられた CipherSpec のみが使用されます。 この条件は、これとその他の SSL 接

続または TLS 接続がすべて停止され、後続の MQCONNX により FipsRequired に対して新しい値が提供されるまで適用されます。

暗号ハードウェアが存在する場合、ハードウェア製品によって提供される暗号モジュールを使用するように、WebSphere MQ を構成することができます。 これらのモジュールは、特定のレベルの FIPS 認定を受けている場合があります。 構成可能なモジュール、およびそれらが FIPS 証明されているかどうかは、使用しているハードウェア製品によって異なります。

FIPS のみの CipherSpec が構成されている場合、MQI クライアントは、FIPS 以外の CipherSpec が指定されている接続を、MQRC_SSL_INITIALIZATION_ERROR として拒否します (可能な場合)。 WebSphere MQ では、そのような接続をすべて拒否することを保証していません。 WebSphere MQ を FIPS 準拠の構成にするかどうかの決定は、お客様の責任で行ってください。

**関連概念**

26 ページの『 UNIX、 Linux、および Windows の連邦情報処理標準 (FIPS)』
Windows、UNIX and Linux システム上の SSL または TLS チャネルで暗号化が必要な場合、WebSphere MQ は IBM Crypto for C (ICC) という名前の暗号化パッケージを使用します。 Windows、 UNIX and Linux プラットフォームでは、ICC ソフトウェアは、米国連邦情報・技術局の連邦情報処理標準 (FIPS) 暗号モジュール評価プログラム (レベル 140-2) に合格しました。

クライアント構成ファイルの SSL スタンザ

**関連資料**

FipsRequired (MQLONG)

MQSSLFIPS

## 複数の GSKit V8.0 インストールがある AIX 上での SSL または TLS クライアント・アプリケーションの実行

AIX システムに複数の GSKit V8.0 がインストールされている場合、その AIX 上で SSL または TLS クライアント・アプリケーションを実行すると、MQRC_CHANNEL_CONFIG_ERROR およびエラー AMQ6175 が発生する場合があります。

複数の GSKit V8.0 インストールがある AIX システム上でクライアント・アプリケーションを実行するときに SSL または TLS を使用すると、クライアント接続呼び出しが MQRC_CHANNEL_CONFIG_ERROR を返す場合があります。 /var/mqm/errors ログには、失敗したクライアント・アプリケーションのエラー AMQ6175 および AMQ9220 が記録されます。以下に例を示します。

```
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 ------------------------------------------------------
```

このエラーの一般的な原因の 1 つとして、LIBPATH または LD_LIBRARY_PATH 環境変数の設定により、IBM WebSphere MQ クライアントが 2 つの異なる GSKit V8.0 インストールからのライブラリーのセットを混ぜてロードする場合があります。 IBM WebSphere MQ クライアント・アプリケーションを DB2® 環境で実行すると、このエラーが発生する可能性があります。

このエラーを回避するには、ライブラリー・パスの初めに IBM WebSphere MQ ライブラリーのディレクトリーを組み込んで、IBM WebSphere MQ ライブラリーが優先されるようにします。 これは、 **-k** パラメーターを指定した **setmqenv** コマンドを使用して行うことができます。以下に例を示します。

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

**setmqenv** コマンドの使用方法について詳しくは、『setmqenv (WebSphere MQ 環境の設定)』を参照してください。

# UNIX, Linux, and Windows システムでの SSL 通信または TLS 通信のセットアップ

SSL または TLS 暗号セキュリティー・プロトコルを使用するセキュア通信では、 通信チャネルをセットアップし、認証に使用するデジタル証明書を管理する必要があります。

SSL または TLS インストール環境をセットアップするには、SSL または TLS を使用するようにチャネルを定義する必要があります。 また、デジタル証明書を作成し、管理することも必要です。 UNIX システム、Linux システム、および Windows システムでは、自己署名証明書を使用してテストを実行できます。

自己署名証明書は、取り消すことができません。したがって、アタッカーが秘密鍵を不正に取得してしまうと、身分を偽って勝手に操作を実行する、という事態が発生しかねません。 一方、CA は、暗号の漏えいが発生した証明書を取り消して、その証明書がそれ以上使用される事態を防止できます。 したがって、実稼働環境では、CA 署名証明書を使用するほうが安全です。一方テスト・システムでは、自己署名証明書を使用するほうが便利です。

証明書の作成と管理の詳細については、114 ページの『UNIX, Linux, and Windows システムでの SSL または TLS の取り扱い』を参照してください。

このトピック集では、SSL 通信のセットアップに関連したタスクをいくつか取り上げ、それらのタスクを実行するための段階的な手順を説明します。

また、プロトコルのオプション部分である SSL または TLS クライアント認証をテストすることもできます。SSL または TLS ハンドシェーク中に、SSL または TLS クライアントは常にサーバーからデジタル証明書を取得し検証します。 IBM WebSphere MQ の実装では、SSL または TLS サーバーは、常にクライアントから証明書を要求します。

UNIX、Linux、および Windows システムでは、SSL または TLS クライアントは、正しい IBM WebSphere MQ 形式のラベルが付いた証明書が存在する場合に限って証明書を送信します。

- キュー・マネージャーの場合は、ibmwebspheremq の後にキュー・マネージャーの名前を小文字に変換して追加した形式になります。 例えば、QM1 の場合は、ibmwebspheremqqm1 です。
- IBM WebSphere MQ クライアント場合は、ibmwebspheremq の後にログオン・ユーザー ID を小文字に変換して追加した形式になります (例えば、ibmwebspheremqmyuserid)。

IBM WebSphere MQ は、他の製品の証明書との混同を避けるために、ibmwebspheremq という接頭部をラベルに付けます。 証明書ラベル全体を小文字で指定してください。

SSL または TLS サーバーは、クライアント証明書が送信される場合は、常にそのクライアント証明書を検証します。 クライアントが証明書を送信しない場合に認証が失敗するのは、チャネルの SSL または TLS サ

ーバー側の定義で、SSLCAUTH パラメーターが REQUIRED に設定されている場合、または SSLPEER パラメーター値が設定されている場合に限られます。 詳しくは、206 ページの『SSL または TLS による 2 つのキュー・マネージャーの接続』を参照してください。

# SSL または TLS の取り扱い

これらのトピックでは、IBM WebSphere MQ での SSL または TLS の使用に関連した単一タスクを実行する方法について説明します。

これらのタスクの多くは、以下のセクションで説明されている高いレベルのタスクにおけるステップとして使用されます。

- 145 ページの『ユーザーの識別および認証』
- 159 ページの『オブジェクトに対するアクセス権限の設定』
- 206 ページの『メッセージの機密性』
- 227 ページの『メッセージのデータ保全性』
- 246 ページの『クラスターのセキュリティーの確保』

## HP Integrity NonStop Server

セキュリティー・サービス、コンポーネント、サポートされるプロトコル・バージョン、サポートされる CipherSpec、およびサポートされないセキュリティー機能を含む、IBM WebSphere MQ Client for HP Integrity NonStop Server OpenSSL のセキュリティー実装について説明します。

IBM WebSphere MQ SSL & TLS サポートは、クライアント・チャネルに対して以下のセキュリティー・サービスを提供します。

- サーバーの認証、およびオプションでクライアントの認証。
- チャネルを流れるデータの暗号化と暗号化解除。
- チャネルを流れるデータの保全性検査。

IBM WebSphere MQ Client for HP Integrity NonStop Server によって提供される SSL および TLS サポートには、以下のコンポーネントが含まれます。

- OpenSSL ライブラリーと **openssl** コマンド。
- IBM WebSphere MQ パスワード・スタッシュ・コマンド **amqrsslc**。

SSL または TLS のクライアント・チャネル操作に必要な以下のコンポーネントは、IBM WebSphere MQ Client for HP Integrity NonStop Server では提供されません。

- OpenSSL 暗号化のために乱数データのソースを提供するエントロピー・デーモン。

### サポートされるプロトコル・バージョン

IBM WebSphere MQ Client for HP Integrity NonStop Server では、以下のプロトコル・バージョンがサポートされます。

- SSL 3.0
- TLS 1.0
- TLS 1.2

### サポートされる CipherSpec

IBM WebSphere MQ Client for HP Integrity NonStop Server では、以下の CipherSpec のバージョンがサポートされます。

- 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 (非推奨)
- 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

## サポートされないセキュリティー機能

IBM WebSphere MQ Client for HP Integrity NonStop Server では、現在以下はサポートされません。

- PKCS#11 暗号化ハードウェアのサポート
- LDAP 証明書取り消しリストのチェック
- Online Certificate Status Protocol (OCSP) のチェック
- FIPS 140-2、NSA SUITE B 暗号スイート制御

## 証明書マネージャー

一連のファイルを使用して、デジタル証明書および証明書の取り消し情報を保管します。

IBM WebSphere MQ の SSL および TLS サポートは、一連のファイルを使用して、デジタル証明書および証明書の取り消し情報を保管します。 これらのファイルが配置されるディレクトリーは、MQCONNX 呼び出しで渡される MQSCO 構造の KeyRepository フィールドを使用してプログラムで指定することも、*MQSSLKEYR* 環境変数で指定することも、SSLKeyRepository 属性を使用する `mqclient.ini` の SSL スタンザで指定することもできます。

MQSCO 構造は MQSSLKEYR 環境変数よりも優先され、MQSSLKEYR 環境変数は ini ファイルのスタンザ値よりも優先されます。

**重要 :** HP Integrity NonStop Server プラットフォームでは、鍵リポジトリーの位置によってディレクトリー位置が指定されますが、ファイル名は指定されません。

IBM WebSphere MQ Client for HP Integrity NonStop Server は、鍵リポジトリーの位置の、以下の名前 (大/小文字を区別します) のファイルを使用します。

- 113 ページの『個人証明書ストア』
- 113 ページの『証明書トラストストア』
- 113 ページの『パスフレーズ stash ファイル』
- 114 ページの『証明書取り消しリスト・ファイル』

### 個人証明書ストア

個人証明書ストア・ファイル `cert.pem`。

このファイルには、使用するクライアント用の個人証明書および暗号化された秘密鍵が PEM 形式で保管されます。 クライアント認証を必要としない SSL または TLS チャネルを使用する場合、このファイルの存在はオプションです。 チャネルがクライアント認証を必要とするときに、SSLCAUTH(REQUIRED) がチャネル定義で指定されている場合、このファイルが存在し、証明書と暗号化された秘密鍵の両方が保管されていなければなりません。

証明書ストアの所有者への読み取りアクセスを許可するには、ファイル許可をこのファイルで設定する必要があります。

正しくフォーマット設定された `cert.pem` ファイルには、次のヘッダーとフッターがある 2 つのセクションが含まれていなければなりません。

```
-----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-----
```

暗号化された秘密鍵のパスフレーズは、パスフレーズ stash ファイル (`Stash.sth`) に保管されます。

### 証明書トラストストア

証明書トラストストア・ファイル `trust.pem`。

このファイルには、クライアントが接続するキュー・マネージャーによって使用される個人証明書を検証するために必要な証明書が PEM 形式で格納されます。 証明書トラストストアは、すべての SSL または TLS クライアント・チャネルで必須です。

ファイル許可を設定してこのファイルへの書き込みアクセスを制限する必要があります。

正しくフォーマット設定された `trust.pem` ファイルには、次のヘッダーとフッターがある 1 つ以上のセクションが含まれていなければなりません。

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

### パスフレーズ *stash* ファイル

パスフレーズ stash ファイル `Stash.sth`。

このファイルは、バイナリー形式の IBM WebSphere MQ 専用ファイルで、`cert.pem` ファイルに保持される秘密鍵にアクセスするときに使用する暗号化されたパスフレーズが含まれます。 秘密鍵そのものは、`cert.pem` 証明書ストアに保管されています。

このファイルは、 **-s** パラメーターを指定した IBM WebSphere MQ **amqrsslc** コマンド行ツールを使用して作成または変更されます。 例えば、ディレクトリー `/home/alice` に `cert.pem` ファイルが格納されている場合、以下のように実行します。

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

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

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

このファイルにファイル許可を設定して、関連付けられた個人証明書ストアの所有者への読み取りアクセスができるようにしなければなりません。

証明書取り消しリスト・ファイル
証明書取り消しリスト・ファイル (`crl.pem`)。

このファイルは、デジタル証明書をクライアントが妥当性検査するのに使用する PEM 形式の証明書取り消しリスト (CRL) を格納します。 このファイルの存在はオプションです。 このファイルが存在しない場合は、証明書の妥当性検査をしても証明書の失効検査は行われません。

ファイル許可を設定してこのファイルへの書き込みアクセスを制限する必要があります。

正しくフォーマット設定された `crl.pem` ファイルには、次のヘッダーおよびフッターがある 1 つ以上のセクションが含まれていなければなりません。

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

## UNIX, Linux, and Windows システムでの SSL または TLS の取り扱い

UNIX、 Linux 、および Windows システムでは、Secure Sockets Layer (SSL) サポートは IBM WebSphere MQ と共にインストールされます。

証明書妥当性検査ポリシーについて詳しくは、証明書の妥当性検査およびトラスト・ポリシーの設計を参照してください。

### *iKeyman*、*iKeycmd*、*runmqakm*、および *runmqckm* の使用

UNIX、Linux 、および Windows システムでは、iKeyman GUI を使用して、またはコマンド行から iKeycmd または runmqakm を使用して、鍵とデジタル証明書を管理します。

- **UNIX and Linux** システムの場合:
  - **strmqikm** コマンドを使用し、iKeyman GUI を開始する。
  - **runmqckm** コマンドを使用し、iKeycmd コマンド行インターフェースでタスクを実行する。
  - **runmqakm** コマンドを使用し、runmqakm コマンド行インターフェースでタスクを実行する。 **runmqakm** のコマンド構文は **runmqckm** の構文と同じです。

    SSL 証明書を、FIPS に準拠した方法で管理する必要がある場合には、**runmqckm** または **strmqikm** コマンドではなく、**runmqakm** コマンドを使用します。

  **runmqckm** コマンドおよび **runmqakm** コマンド用のコマンド行インターフェースの詳細については、鍵と証明書の管理を参照してください。

  PKCS #11 暗号ハードウェアに保管した証明書またはキーを使用する場合には、 iKeycmd および iKeyman が 64 ビット・プログラムであることに注意してください。 PKCS #11 サポートに必要な外部モジュールは、64 ビット・プロセスにロードされます。したがって、暗号ハードウェアの管理用に 64 ビットの PKCS #11 ライブラリーがインストールされています。 Windows および Linux x86 32 ビット・プラットフォームは唯一の例外です。これらのプラットフォームでは、 iKeyman および iKeycmd プログラムは 32 ビットです。

  以下のプラットフォームでは、JRE は前のバージョンの製品では 32 ビットでしたが、 IBM WebSphere MQ Version 7.5 では 64 ビットのみであるため、 **iKeyman** および **iKeycmd** JRE のアドレッシング・モードに適した追加の PKCS#11 ドライバーをインストールする必要がある場合があります。 それは、PKCS#11 ドライバーが使用するアドレッシング・モードが JRE と同じでなければならないためです。 以下の表に、IBM WebSphere MQ Version 7.5 JRE アドレッシング・モードを示します。

| 表 *12. IBM WebSphere MQ Version 7.5 JRE* アドレッシング・モード | |
|---|---|
| **プラットフォーム** | **JRE アドレッシング・モード** |
| Windows (32 ビットまたは 64 ビット) | 32 |
| Linux for System x 32 ビット | 32 |
| Linux for System x 64 ビット | 64 |
| Linux (System p の場合) | 64 |
| Linux (System z の場合) | 64 |
| HP-UX | 64 |
| Solaris SPARC | 64 |
| Solaris x86-64 | 64 |
| AIX | 64 |

iKeyman GUI を開始するために **strmqikm** コマンドを実行する場合は、事前に X Window システムを実行できるマシンを使用していることを確認し、次のことを行ってください。

- DISPLAY 環境変数を設定する。例えば、次のようにします。

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

- PATH 環境変数に **/usr/bin** および **/bin** が含まれていることを確認する。 これは、 **runmqckm** コマンドおよび **runmqakm** コマンドにも必要です。 以下に例を示します。

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

- **Windows** システムの場合:
  - **strmqikm** コマンドを使用し、iKeyman GUI を開始する。
  - **runmqckm** コマンドを使用し、iKeycmd コマンド行インターフェースでタスクを実行する。

    SSL 証明書を、FIPS に準拠した方法で管理する必要がある場合には、**runmqckm** または **strmqikm** コマンドではなく、**runmqakm** コマンドを使用します。

UNIX、 Linux 、または Windows システムで SSL トレースを要求するには、 strmqtrc を参照してください。

**関連資料**
runmqckm コマンドおよび runmqakm コマンド

## *UNIX, Linux, and Windows* システムでの鍵リポジトリーのセットアップ

鍵リポジトリーは、iKeyman ユーザー・インターフェースを使用して、あるいは **iKeycmd** または **runmqakm** コマンドを使用してセットアップできます。

### このタスクについて

SSL または TLS 接続では、接続の両端に鍵リポジトリー が必要です。 各 IBM WebSphere MQ キュー・マネージャーおよび IBM WebSphere MQ MQI client には、鍵リポジトリーへのアクセス権が必要です。 詳細については、23 ページの『SSL または TLS 鍵リポジトリー』 を参照してください。

UNIX, Linux, and Windows システムでは、デジタル証明書が鍵データベース・ファイルに保管されます。このファイルは、**iKeyman** ユーザー・インターフェースを使用するか、あるいは **iKeycmd** コマンドまたは **runmqakm** コマンドを使用して管理します。 これらのデジタル証明書には、ラベルがあります。 特定のラベルは、個人の証明書をキュー・マネージャーまたは IBM WebSphere MQ MQI client に関連付けます。SSL および TLS は、認証のためにその証明書を使用します。 UNIX, Linux, and Windows システムでは、IBM WebSphere MQ は、他の製品の証明書との混同を避けるために、ibmwebspheremq という接頭部をラベル

に付けます。 プレフィックスの後には、キュー・マネージャー名または IBM WebSphere MQ MQI client のユーザー・ログオン ID を小文字に変換したものが続きます。 証明書ラベル全体を小文字で指定してください。

鍵データベース・ファイルの名前は、パスと語幹名から構成されます。

- UNIX and Linux システムでは、キュー・マネージャーのデフォルトのパス (キュー・マネージャーの作成時に設定される) は `/var/mqm/qmgrs/<queue_manager_name>/ssl` です。

  Windows システムでは、デフォルトのパスは *MQ_INSTALLATION_PATH*`\Qmgrs\`*queue_manager_name*`\ssl` です (*MQ_INSTALLATION_PATH* は、IBM WebSphere MQ のインストール先のディレクトリーです)。 例えば、`C:\program files\IBM\WebSphere MQ\Qmgrs\QM1\ssl` などです。

  デフォルトの語幹名は key です。 オプショナルで、独自のパスと語幹名を選択できますが、拡張子は `.kdb` でなければなりません。

  独自のパスまたはファイル名を選択する場合は、そのファイルに対するアクセス権を設定して、そのファイルへのアクセスを厳密に制御してください。

- WebSphere MQ クライアントの場合、デフォルトのパスも語幹名もありません。 このファイルへのアクセスを厳密に制御してください。 拡張子は `.kdb` である必要があります。

ファイル・レベルのロックをサポートしないファイル・システム (Linux システム上の NFS バージョン 2 など) で鍵リポジトリーを作成しないでください。

鍵データベース・ファイル名の検査と指定については、120 ページの『UNIX、Linux または Windows システムでキュー・マネージャーの鍵リポジトリーの位置を変更する操作』を参照してください。 鍵データベース・ファイルの名前は、鍵データベース・ファイルの作成前または後に指定できます。

**iKeyman** または **iKeycmd** コマンドを実行する際のユーザー ID には、鍵データベース・ファイルが作成または更新されるディレクトリーに対する書き込み権限が必要です。 デフォルト `ssl` ディレクトリーを使用するキュー・マネージャーの場合、**iKeyman** または **iKeycmd** を実行するユーザー ID は、mqm グループのメンバーでなければなりません。 IBM WebSphere MQ MQI client の場合、クライアント稼働時に使用されたユーザー ID とは異なるユーザー ID から **iKeyman** または **iKeycmd** を実行するときには、ファイルのアクセス権を変更して、IBM WebSphere MQ MQI client が実行時に鍵データベース・ファイルにアクセスできるようにする必要があります。 詳しくは、118 ページの『Windows 上の鍵データベース・ファイルへのアクセスおよび保護』または 118 ページの『UNIX and Linux システム上の鍵データベース・ファイルへのアクセスおよび保護』を参照してください。

**iKeyman** または **iKeycmd** バージョン 7.0 では、新規鍵データベースには、定義済み認証局 (CA)証明書セットが自動的に取り込まれます。 **iKeyman** または **iKeycmd** バージョン 8.0 では、鍵データベースに自動的にデータが取り込まれることはありません。 必要な CA 証明書のみを鍵データベース・ファイルに組み込むため、初期セットアップの安全性が増しています。

**注 :** GSKit バージョン 8.0 のこの動作変更によって、CA 証明書が自動的にリポジトリーに追加されることがなくなったため、好みの CA 証明書を手動で追加する必要があります。 この動作変更によって、使用する CA 証明書をより詳細に制御できるようになりました。 119 ページの『GSKit バージョン 8.0 を使用して UNIX, Linux, and Windows システムで空の鍵リポジトリーにデフォルト CA 証明書を追加する』を参照してください。

鍵データベースを作成するには、コマンド行を使用するか、 **strmqikm** (iKeyman) ユーザー・インターフェースを使用します。

**注 :** TLS 証明書を FIPS に準拠した方法で管理する必要がある場合は、**runmqakm** コマンドを使用します。 **strmqikm** ユーザー・インターフェースは、FIPS 準拠のオプションを提供していません。

## 手順

コマンド・ラインを使用して鍵データベースを作成します。

1. 以下のいずれかのコマンドを実行します。

- UNIX, Linux, and Windows システムの場合:

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

- runmqakm を使用する場合:

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

ここで、

**-db *filename***
CMS 鍵データベースの完全修飾ファイル名を指定します。ファイル拡張子は .kdb である必要があります。

**-pw *password***
CMS 鍵データベースのパスワードを指定します。

**-type *cms***
データベースのタイプを指定します。 (IBM WebSphere MQ の場合、これは cms でなければなりません。)

**-stash**
鍵データベース・パスワードをファイルに保存します。

**-fips**
BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードでは、ICC コンポーネントは FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**runmqakm** コマンドは失敗します。

**-強い**
入力されたパスワードがパスワード強度の最小要件を満たしているかどうかを確認します。 パスワードの最小要件は、次のとおりです。

- パスワードは、最小 14 文字の長さでなければならない。
- パスワードには、少なくとも 1 つの小文字、1 つの大文字、および 1 つの数字または特殊文字が含まれている必要がある。 特殊文字には、アスタリスク (*)、ドル記号 ($)、番号記号 (#)、およびパーセント記号 (%) が含まれます。スペースは特殊文字として分類されます。
- パスワード中の同じ文字はそれぞれ最大 3 回までしか使用できない。
- パスワード内に連続して出現する同じ文字は最大 2 文字。
- すべての文字が、ASCII の標準印刷可能文字セットの 0x20 から 0x7E までの範囲内の文字である。

あるいは、 **strmqikm** (iKeyman) ユーザー・インターフェースを使用して、鍵データベースを作成します。

2. UNIX and Linux システムの場合は、root ユーザーとしてログインする。 Windows システムの場合は、管理者または MQM グループのメンバーとしてログインする。
3. **strmqikm** コマンドを実行して、iKeyman ユーザー・インターフェースを開始する。
4. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**New (新規)**」をクリックする。
「新規」ウィンドウが開きます。
5. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
6. 「**File Name (ファイル名)**」フィールドに、ファイル名を入力する。
このフィールドには、既に key.kdb というテキストが入っています。 語幹名が key である場合は、このフィールドを変更しないでください。 別の語幹名を指定した場合は、key をご使用の語幹名で置き換えてください。 ただし、.kdb 拡張子は変更してはなりません。
7. 「**Location (ロケーション)**」フィールドに、パスを入力します。
以下に例を示します。

- キュー・マネージャーの場合: `/var/mqm/qmgrs/QM1/ssl` ( UNIX and Linux システムの場合) または `C:\Program Files\IBM\WebSphere MQ\qmgrs\QM1\ssl` ( Windows システムの場合)。

  このパスは、キュー・マネージャーの **SSLKeyRepository** 属性の値と一致している必要があります。
- IBM WebSphere MQ クライアントの場合: `/var/mqm/ssl` (UNIX and Linux システム) または `C:\mqm\ssl` (Windows システム)。

8. 「**Open (オープン)**」をクリックする。

   「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
9. 「**Password (パスワード)**」フィールドにパスワードを入力し、「**Confirm Password (パスワードの確認)**」 フィールドにそのパスワードをもう一度入力する。
10. 「**Stash the password to a file (ファイルにパスワードを隠す)**」チェック・ボックスを選択する。

    **注 :** パスワードを隠さない場合は、鍵データベース・ファイルへのアクセスに必要なパスワードを取得できないので、SSL または TLS チャネルを開始しようとしても失敗します。
11. 「**OK**」をクリックします。

    「個人証明書」ウィンドウが開きます。
12. 118 ページの『Windows 上の鍵データベース・ファイルへのアクセスおよび保護』 または 118 ページの『UNIX and Linux システム上の鍵データベース・ファイルへのアクセスおよび保護』 で説明されているようにアクセス許可を設定します。

*Windows* 上の鍵データベース・ファイルへのアクセスおよび保護
鍵データベース・ファイルには、適切なアクセス権がないことがあります。 これらのファイルへの適切なアクセス権を設定する必要があります。

*key*.kdb、*key*.sth、*key*.crl、 および *key*.rdb (*key* は、鍵データベースの語幹名) の各ファイルへのアクセス制御を設定し、限定されたユーザーのセットに権限を付与します。

アクセス権の付与の際には、次の事柄を考慮します。

**全権限**
BUILTIN¥Administrators、NT AUTHORITY¥SYSTEM、およびデータベース・ファイルを作成したユーザー。

**読み取り権限**
キュー・マネージャーの場合、ローカル mqm グループのみ。 これは、MCA が mqm グループ内のユーザー ID 下で稼働していると想定しています。

クライアントの場合、クライアント・プロセスが実行されているユーザー ID。

*UNIX and Linux* システム上の鍵データベース・ファイルへのアクセスおよび保護
鍵データベース・ファイルには、適切なアクセス権がないことがあります。 これらのファイルへの適切なアクセス権を設定する必要があります。

キュー・マネージャーの場合、鍵データベース・ファイルに対して許可を設定します。そうすれば、キュー・マネージャーおよびチャネルのプロセスが必要な時にそれらのファイルを読み取れると同時に、他のユーザーがそれらを読み取ったり変更したりするのを禁止できます。 通常、mqm ユーザーは読み取り権限を必要とします。 mqm ユーザーとしてログインして鍵データベース・ファイルを作成したユーザーの場合、権限は十分であると考えられます。しかし、mqm ユーザーではなく、mqm グループの別のユーザーであった場合には、mqm グループの他のユーザーに読み取り権限を付与する必要があるかもしれません。

同様に、クライアントの場合も、鍵データベース・ファイルに対して許可を設定します。そうすれば、クライアント・アプリケーション・プロセスが必要な時にそれらのファイルを読み取れると同時に、他のユーザーがそれらを読み取ったり変更したりするのを禁止できます。 通常、クライアント・プロセスを実行するユーザーには、読み取り権限が必要です。 そのユーザーとしてログインして鍵データベース・ファイルを作成したユーザーの場合、権限は十分であると考えられます。しかし、クライアント・プロセスのユーザーではなく、そのグループの別のユーザーであった場合には、グループの他のユーザーに読み取り権限を付与する必要があるかもしれません。

*key*.kdb、*key*.sth、*key*.crl、および *key*.rdb (*key* は鍵データベースの語幹名) の各ファイルに対して、読み取りおよび書き込み権限 (ファイル所有者の場合)、および読み取り権限 (mqm またはクライアント・ユーザー・グループの場合) を設定します。(-rw-r-----)

*GSKit* バージョン *8.0* を使用して *UNIX, Linux, and Windows* システムで空の鍵リポジトリーにデフォルト *CA* 証明書を追加する
以下の手順に従って、1 つ以上のデフォルト CA 証明書を GSKit バージョン 8 を使用して空の鍵リポジトリーに追加します。

GSKit バージョン 7.0 では、新しい鍵リポジトリーを作成する場合の動作は、通常使用される認証局のデフォルト CA 証明書セットに自動的に追加することでした。 GSKit バージョン 8 では、この動作が変更され、CA 証明書はリポジトリーに自動的に追加されなくなりました。 ユーザーは、CA 証明書を手動で鍵リポジトリーに追加しなければならなくなりました。

## iKeyman の使用

CA 証明書の追加先マシンで、次の手順を実行してください。

1. **strmqikm** コマンドを使用して iKeyman GUI を開始します ( UNIX、 Linux および Windows システムの場合)。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書を追加する先の鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Key database content (鍵データベースの内容)**」フィールドで、「**Signer Certificates (署名者証明書)**」を選択する。
9. 「**挿入**」をクリックする。 「CA の証明書の追加」ウィンドウが開きます。
10. リポジトリーに追加できる CA 証明書が、階層ツリー構造に表示されます。 有効な CA 証明書の完全なリストを表示するには、信頼する CA 証明書を持つ組織の最上位エントリーを選択します。
11. 信頼する CA 証明書をリストから選択し、「**OK**」をクリックします。 証明書が鍵リポジトリーに追加されます。

## コマンド行の使用

以下のコマンドを使用してリストし、次に iKeycmd を使用して CA 証明書を追加します。

- 次のコマンドを発行して、デフォルトの CA 証明書とそれらを発行している組織をリストします。

```
runmqckm -cert -listsigners
```

- 次のコマンドを発行して、*label* フィールドで指定した組織の CA 証明書をすべて追加します。

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

ここで、

-db *filename*　　鍵データベースの完全修飾パス名です。

| | |
|---|---|
| -pw *password* | 鍵データベースのパスワードです。 |
| -label *label* | 証明書に付加するラベルです。 |

**注 :** CA 証明書を鍵リポジトリーに追加すると、WebSphere MQ は、その CA 証明書により署名されたすべての個人証明書を信頼します。 どの認証局を信頼するかをよく検討し、クライアントおよびマネージャーの認証に必要な CA 証明書セットのみを追加してください。 デフォルトの CA 証明書セット全体を追加することは、それがセキュリティー・ポリシーにおける明確な要件でない限り、お勧めしません。

## *UNIX, Linux, and Windows* システムでキュー・マネージャーの鍵リポジトリーの位置を取得する操作

キュー・マネージャーの鍵データベース・ファイルの位置を取得する手順を取り上げます。

### 手順

1. 次のどちらかの MQSC コマンドを使用して、キュー・マネージャーの属性を表示する。

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

   IBM WebSphere MQ エクスプローラーまたは PCF コマンドを使用してキュー・マネージャーの属性を表示することもできます。
2. コマンドの出力を調べて、鍵データベース・ファイルのパスと語幹名を見つける。
   例:
   a. UNIX and Linux システム: `/var/mqm/qmgrs/QM1/ssl/key` (`/var/mqm/qmgrs/QM1/ssl` はパス、key は語幹名)
   b. Windows の場合: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`。ここで、*MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` はパス、 key は語幹名です。*MQ_INSTALLATION_PATH* WebSphere MQ がインストールされている上位ディレクトリーを表します。

## *UNIX*、*Linux* または *Windows* システムでキュー・マネージャーの鍵リポジトリーの位置を変更する操作

キュー・マネージャーの鍵データベース・ファイルの位置を変更する方法はいくつかありますが、そのうちの 1 つは、MQSC コマンド ALTER QMGR です。

MQSC コマンド ALTER QMGR を使用してキュー・マネージャーの鍵リポジトリー属性を設定することにより、キュー・マネージャーの鍵データベース・ファイルの位置を変更できます。 UNIX and Linux システムの場合:

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

この鍵データベース・ファイルの完全修飾ファイル名は、`/var/mqm/qmgrs/QM1/ssl/MyKey.kdb` になります。

Windows の場合:

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

この鍵データベース・ファイルの完全修飾ファイル名は、`C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\Mykey.kdb` になります。

**重要 :** 拡張子 .kdb は、キュー・マネージャーが自動的に付加するので、SSLKEYR キーワードのファイル名には含めないでください。

WebSphere MQ エクスプローラーまたは PCF コマンドを使用してキュー・マネージャーの属性を変更することもできます。

キュー・マネージャーの鍵データベース・ファイルの場所を変更する場合、証明書は、旧の場所から転送されません。 現在アクセスしている鍵データベース・ファイルが新しい鍵データベース・ファイルである場合は、 134 ページの『個人証明書を鍵リポジトリーにインポートする操作 (UNIX, Linux, and Windows システム)』 で説明されているように、必要な CA 証明書および個人証明書をそのファイルに取り込む必要があります。

## *UNIX, Linux, and Windows システムでの IBM WebSphere MQ MQI クライアントの鍵リポジトリーの位置*

鍵リポジトリーの位置は、MQSSLKEYR 変数から取得できます。MQCONNX 呼び出しで指定することも可能です。

MQSSLKEYR 環境変数を調べて、 IBM WebSphere MQ MQI クライアントの鍵データベース・ファイルの場所を取得します。 以下に例を示します。

```
echo $MQSSLKEYR
```

また、鍵データベース・ファイル名は MQCONNX 呼び出しでも設定できるので、ご使用のアプリケーションも調べてください (121 ページの『UNIX, Linux, and Windows システム上の IBM WebSphere MQ MQI クライアントの鍵リポジトリーの場所の指定』を参照)。 MQCONNX 呼び出しで設定された値は、MQSSLKEYR の値を指定変更します。

## *UNIX, Linux, and Windows システム上の IBM WebSphere MQ MQI クライアントの鍵リポジトリーの場所の指定*

IBM WebSphere MQ MQI クライアントにはデフォルトのキー・リポジトリーはありません。 その位置を指定する方法は、2 つあります。 その他のシステムへの無許可のコピーを防ぐために、鍵データベース・ファイルには、所定のユーザーまたは管理者しかアクセスできないようにしてください。

IBM WebSphere MQ MQI クライアントの鍵データベース・ファイルの場所は、以下の 2 つの方法のいずれかで指定できます。

- MQSSLKEYR 環境変数を設定する。 UNIX and Linux システムの場合:

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

  鍵データベース・ファイルには、次の完全修飾ファイル名があります。

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

  Windows の場合:

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

  鍵データベース・ファイルには、次の完全修飾ファイル名があります。

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

  **注 :** 拡張子 .kdb は、ファイル名の必須部分ですが、環境変数の値の一部としては組み込まれていません。
- アプリケーションが MQCONNX 呼び出しを行うときに、MQSCO 構造の *KeyRepository* フィールドに、鍵データベース・ファイルのパスと語幹名を指定する。 MQCONNX における MQSCO 構造の使用について詳しくは、MQSCO の概要を参照してください。

## *UNIX、Linux または Windows システムで証明書または証明書ストアの変更が有効になる時点*

証明書ストアに含まれている証明書や、証明書ストアの位置を変更した場合に、その変更が有効になる時点は、チャネルのタイプとチャネルの実行方法によって異なります。

鍵データベース・ファイルに含まれている証明書と鍵リポジトリーの属性の変更が有効になるのは、以下の時点です。

- 新規アウトバウンド単一チャネル・プロセスが SSL チャネルとして最初に実行されたとき。
- 新規インバウンド TCP/IP 単一チャネル・プロセスが SSL チャネルの開始要求を最初に受信したとき。
- MQSC コマンド REFRESH SECURITY TYPE(SSL) が発行され、WebSphere MQ SSL 環境が最新表示されたとき。
- クライアント・アプリケーション・プロセスにおいて、プロセスの最後の SSL 接続が閉じられるとき。次の SSL 接続で、証明書の変更が受け入れられます。
- プロセス・プール・プロセス (amqrmppa) のスレッドとして実行されるチャネルの場合は、プロセス・プール・プロセスが開始または再開始され、SSL チャネルを最初に実行したとき。 プロセス・プール・プロセスが既に SSL チャネルを実行している場合に 変更内容をただちに有効にするには、MQSC コマンド REFRESH SECURITY TYPE(SSL) を実行してください。
- チャネル・イニシエーターのスレッドとして実行されるチャネルの場合は、チャネル・イニシエーターが開始または再開始され、SSL チャネルを最初に実行したとき。 チャネル・イニシエーター・プロセスが既に SSL チャネルを実行している場合に 変更内容をただちに有効にするには、MQSC コマンド REFRESH SECURITY TYPE(SSL) を実行してください。
- TCP/IP リスナーのスレッドとして実行されるチャネルの場合は、リスナーが開始または再開始され、SSL チャネル開始要求を最初に受信したとき。 リスナーが既に SSL チャネルを実行している場合に 変更内容をただちに有効にするには、MQSC コマンド REFRESH SECURITY TYPE(SSL) を実行してください。

IBM WebSphere MQ エクスプローラーまたは PCF コマンドを使用して、WebSphere MQ SSL 環境をリフレッシュすることもできます。

## *UNIX, Linux, and Windows* システムでの自己署名個人証明書の作成

iKeyman、iKeycmd、runmqakm のいずれかを使用して、自己署名証明書を作成できます。

**注 :** IBM WebSphere MQ は、SHA-3 アルゴリズムと SHA-5 アルゴリズムをサポートしません。 デジタル署名アルゴリズム名 SHA384WithRSA および SHA512WithRSA は SHA-2 ファミリーのメンバーであるため、これらのアルゴリズムは使用可能です。

デジタル署名アルゴリズム名 SHA3WithRSA および SHA5WithRSA は、それぞれ SHA384WithRSA および SHA512WithRSA の簡略形であるため、これらは推奨されません。

自己署名証明書を使用するのが望ましい場合がある理由の詳細については、207 ページの『2 つのキュー・マネージャーの相互認証への自己署名証明書の使用』を参照してください。

すべてのデジタル証明書がすべての CipherSpec と共に使用できるわけではありません。 必ず、使用する必要のある CipherSpec と互換性のある証明書を作成してください。 WebSphere MQ は、3 タイプの CipherSpec をサポートしています。 詳細については、34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』トピックの 35 ページの『楕円曲線と RSA CipherSpec の相互運用性』を参照してください。 タイプ 1 の CipherSpec (名前が ECDHE_ECDSA_ で始まる) を使用するには、**runmqakm** コマンドを使用して証明書を作成し、Elliptic Curve ECDSA 署名アルゴリズム・パラメーター (**-sig_alg** EC_ecdsa_with_SHA384 など) を指定する必要があります。

## iKeyman の使用

iKeyman は、FIPS 準拠のオプションを提供していません。 SSL または TLS 証明書を FIPS に準拠した方法で管理する必要がある場合は、**runmqakm** コマンドを使用します。

キュー・マネージャーまたは WebSphere MQ MQI クライアント用の自己署名証明書を取得するには、次の手順を使用してください。

1. **strmqikm** コマンドを使用して、iKeyman GUI を開始する。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが表示されます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。

4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書を保管する鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが表示されます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Create (作成)**」メニューから、「**New Self-Signed Certificate (新規自己署名証明書の作成)**」をクリックする。 「Create New Self-Signed Certificate (新規自己署名証明書の作成)」ウィンドウが表示されます。
9. 「**Key Label (鍵ラベル)**」フィールドに、次のどちらかを入力する。
   - キュー・マネージャーの場合は、ibmwebspheremq の後に続けて、小文字に変換されたキュー・マネージャーの名前。 例えば、QM1 の場合は、ibmwebspheremqqm1 です。
   - WebSphere MQ クライアントの場合、ibmwebspheremq の後に続けて、小文字に変換されたログオン・ユーザー ID。例えば、ibmwebspheremqmyuserid です。
10. **Distinguished name** の任意のフィールド、または任意の **Subject alternative name** フィールドの値を入力または選択します。
11. 残りのフィールドでは、デフォルト値を受け入れるか、別の値を入力または選択します。 識別名について詳しくは、 11 ページの『識別名』を参照してください。
12. 「**OK**」をクリックします。 「**Personal Certificates (個人用証明書)**」リストに、作成した自己署名の個人用証明書のラベルが表示されます。

## コマンド行の使用

iKeycmd または runmqakm を使用して自己署名個人証明書を作成するには、次のコマンドを使用します。

- UNIX、 Linux および Windows システムで iKeycmd を使用する場合:

```
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* の代わりに、-san_dsname *DNS_names* 、-san_emailaddr *email_addresses* 、または -san_ipaddr *IP_addresses* を使用できます。

- runmqakm を使用する場合:

```
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 鍵データベースの完全修飾ファイル名です。 |
| -pw *password* | CMS 鍵データベースのパスワードです。 |
| -label *label* | 証明書に付加されている鍵ラベルです。 |
| -dn *distinguished_name* | X.500 識別名を二重引用符で囲んだものです。 少なくとも 1 つの属性が必要です。 複数の OU または DC 属性を指定することも可能です。 |
| -size *key_size* | 鍵のサイズです。 iKeycmd の場合、値は 512 または 1024 にします。 runmqakm の場合、値は 512、1024、2048、または 4096 にすることができます。 |

| | |
|---|---|
| -x509version *version* | 作成する X.509 証明書のバージョン。 値は 1、2、または 3 にすることができます。 デフォルトは 3 です。 |
| -expire *days* | 証明書の有効期限 (日数)。 証明書の場合のデフォルトは 365 日です。 |
| -fips | コマンドが FIPS モードで実行されるように指定します。 このモードは、BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードの場合、ICC コンポーネントは、FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**runmqakm** コマンドは失敗します。 |
| -sig_alg | runmqakm の場合に、自己署名証明書の作成時に使用されるハッシュ・アルゴリズム。 このハッシュ・アルゴリズムは、新たに作成された自己署名証明書に関連付けられた署名を作成するために使用されます。 値は、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、または EC_ecdsa_with_SHA512 のいずれかです。 デフォルト値は SHA1WithRSA です。 |
| -sig_alg | iKeycmd の場合に、項目の鍵ペアの作成に使用される非対称署名アルゴリズム。 値は、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、または SHAWithRSA のいずれかです。 デフォルト値は SHA1WithRSA です。 |
| -san_dnsname *DNS_names* | 作成される項目の DNS 名のコンマ区切りまたはスペース区切りリスト。 |
| -san_emailaddr *email_addresses* | 作成される項目の E メール・アドレスのコンマ区切りまたはスペース区切りリスト。 |
| -san_ipaddr *IP_addresses* | 作成される項目の IP アドレスのコンマ区切りまたはスペース区切りリスト。 |

## distributed *UNIX, Linux, and Windows システムでの個人証明書の要求*

個人証明書は、**strmqikm** (iKeyman) GUI を使用して、あるいは **runmqckm** または **runmqakm** コマンドを使用してコマンド行から要求できます。 SSL または TLS 証明書を FIPS に準拠した方法で管理する必要がある場合は、**runmqakm** コマンドを使用します。

### このタスクについて

iKeyman GUI かコマンド行で個人証明書を要求できます。注意点を以下にまとめます。

- WebSphere MQ は、SHA-3 アルゴリズムと SHA-5 アルゴリズムをサポートしません。 デジタル署名アルゴリズム名 SHA384WithRSA および SHA512WithRSA は SHA-2 ファミリーのメンバーであるため、これらのアルゴリズムは使用可能です。
- デジタル署名アルゴリズム名 SHA3WithRSA および SHA5WithRSA は、それぞれ SHA384WithRSA および SHA512WithRSA の簡略形であるため、これらは推奨されません。

- すべてのデジタル証明書がすべての CipherSpec と共に使用できるわけではありません。 必ず、使用する必要のある CipherSpec と互換性のある証明書を要求してください。 WebSphere MQ は、3 タイプの CipherSpec をサポートしています。 詳細については、34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』トピックの 35 ページの『楕円曲線と RSA CipherSpec の相互運用性』を参照してください。
- タイプ 1 の CipherSpecs (名前が ECDHE_ECDSA_で始まる) を使用するには、**runmqakm** コマンドを使用して証明書を要求し、Elliptic Curve ECDSA 署名アルゴリズム・パラメーター ( **-sig_alg** EC_ecdsa_with_SHA384 など) を指定する必要があります。
- FIPS 準拠オプションを使用できるのは runmqakm コマンドだけです。
- 暗号ハードウェアを使用している場合は、141 ページの『PKCS #11 ハードウェア用の個人用証明書の要求』を参照してください。

*iKeyman* ユーザー・インターフェースの使用

## このタスクについて

iKeyman は、FIPS 準拠のオプションを提供していません。 SSL または TLS 証明書を FIPS に準拠した方法で管理する必要がある場合は、**runmqakm** コマンドを使用します。

## 手順

iKeyman ユーザー・インターフェースを使用して個人証明書を申請するには、以下の手順に従います。

1. **strmqikm** コマンドを使用して、 iKeyman ユーザー・インターフェースを開始します。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。
   「**Open (オープン)**」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 要求が生成される鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。
   「**Password Prompt (パスワード・プロンプト)**」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。
   鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Create (作成)**」メニューから、「**New Certificate Request (新規認証要求)**」をクリックする。「**Create New Key and Certificate Request (新規鍵および認証要求の作成)**」ウィンドウが開きます。
9. 「**鍵ラベル**」フィールドに、以下のラベルを入力します。
   - キュー・マネージャーの場合は、ibmwebspheremq の後に続けて、小文字に変更されたキュー・マネージャーの名前。 例えば、 QM1 というキュー・マネージャーの場合は、 ibmwebspheremqqm1 と入力します。
   - IBM WebSphere MQ MQI client の場合は、 ibmwebspheremq の後にログオン・ユーザー ID をすべて小文字で入力します。例えば、 ibmwebspheremqmyuserid のように入力します。
10. 「**識別名**」フィールドのいずれかのフィールド、またはいずれかの「**サブジェクト代替名**」フィールドで、値を入力または選択する。 残りのフィールドについては、デフォルト値を受け入れるか、新しい値を入力または選択します。
    識別名について詳しくは、11 ページの『識別名』を参照してください。
11. 「**Enter the name of a file in which to store the certificate request (認証要求を保管するファイル名の入力)**」フィールドで、デフォルトの certreq.arm を受け入れるか、絶対パスと一緒に新しい値を入力する。
12. 「**OK**」をクリックします。

確認ウィンドウが表示されます。

13. 「**OK**」をクリックします。

    「**Personal Certificate Requests (個人用証明書の要求)**」 リストに、作成した新しい個人用証明書要求のラベルが表示されます。 証明書要求は、ステップ 125 ページの『11』 で選択したファイルに保管されます。

14. そのファイルを認証局 (CA) に送信するか、CA の Web サイト上の要求フォームにそのファイルをコピーして、新しい個人用証明書を要求する。

コマンド行の使用

## 手順

**runmqckm** または **runmqakm** コマンドのいずれかを使用して個人証明書を要求するには、次のコマンドを使用します。

- **runmqckm** を使用する場合:

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

  -dn *distinguished_name* の代わりに、-san_dsname *DNS_names* 、-san_emailaddr *email_addresses* 、または -san_ipaddr *IP_addresses* を使用できます。

- runmqakm を使用する場合:

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

ここで、

**-db *filename***
CMS 鍵データベースの完全修飾ファイル名を指定します。

**-pw *password***
CMS 鍵データベースのパスワードを指定します。

**-label *label***
証明書に付加する鍵ラベルを指定します。

**-dn *distinguished_name***
二重引用符で囲んだ X.500 識別名を指定します。 少なくとも 1 つの属性が必要です。 複数の OU および DC 属性を指定できます。

**-size *key_size***
鍵のサイズを指定します。 **runmqckm** を使用している場合、値は 512 または 1024 にします。 **runmqakm** を使用している場合、値は 512、1024、または 2048 にします。

**-file *filename***
認証要求のファイル名を指定します。

**-fips**
コマンドが FIPS モードで実行されるように指定します。 このモードは、BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードでは、ICC コンポーネントは FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**runmqakm** コマンドは失敗します。

**-sig_alg**
**runmqckm** の場合、項目の鍵ペアの作成に使用される非対称署名アルゴリズムを指定します。 値は、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、または SHAWithRSA のいずれかです。 デフォルト値は SHA1WithRSA です。

**-sig_alg**
**runmqakm** の場合、認証要求の作成中に使用されるハッシュ・アルゴリズムを指定します。 このハッシュ・アルゴリズムは、新たに作成された証明書要求に関連付けられた署名を作成するために使用されます。 値は、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、または EC_ecdsa_with_SHA512 のいずれかです。 デフォルト値は SHA1WithRSA です。

**-san_dnsname *DNS_names***
作成される項目の DNS 名のコンマ区切りまたはスペース区切りリストを指定します。

**-san_emailaddr *email_addresses***
作成される項目の E メール・アドレスのコンマ区切りまたはスペース区切りリストを指定します。

**-san_ipaddr *IP_addresses***
作成される項目の IP アドレスのコンマ区切りまたはスペース区切りリストを指定します。

### *UNIX, Linux, and Windows システムでの既存の個人証明書の更新*
iKeyman ユーザー・インターフェースを使用するか、 **iKeycmd** コマンドまたは **runmqakm** コマンドを使用して、個人証明書を更新できます。

## 始める前に

より大きい鍵サイズを個人証明書に使用する必要がある場合、下記の更新ステップではうまくいきません。再作成される認証要求は、既存の鍵から生成されるからです。

124 ページの『UNIX, Linux, and Windows システムでの個人証明書の要求』に記載されているステップに従い、必要な鍵サイズを使用して新しい認証要求を作成してください。 このプロセスは、既存の鍵を置き換えるためのものです。

## このタスクについて

個人証明書には有効期限日があり、その期限を過ぎると証明書は使用できなくなります。 このタスクでは、既存の個人証明書を有効期限が切れる前に更新する方法について説明します。

*iKeyman* ユーザー・インターフェースの使用

## このタスクについて
iKeyman は、FIPS 準拠のオプションを提供していません。 SSL または TLS 証明書を FIPS に準拠した方法で管理する必要がある場合は、**runmqakm** コマンドを使用します。

## 手順

iKeyman ユーザー・インターフェースを使用して個人証明書を申請するには、以下の手順に従います。

1. UNIX, Linux, and Windows システムで **strmqikm** コマンドを使用して、iKeyman ユーザー・インターフェースを開始します。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。

「**Open (オープン)**」ウィンドウが開きます。

3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 要求が生成される鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。
   「**Password Prompt (パスワード・プロンプト)**」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。
   鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. ドロップダウン選択メニューから「**個人証明書**」を選択し、更新する証明書をリストから選択する。
9. 「**要求の再作成 ...**」 をクリックします。 」ボタンをクリックするとここに表示されます。
   ファイル名とファイルの場所の情報を入力するためのウィンドウが開きます。
10. 「**ファイル名**」フィールドで、デフォルトの certreq.arm を受け入れるか、新規の値を絶対ファイル・パスを含めて入力する。
11. 「**OK**」をクリックします。 証明書要求は、ステップ 128 ページの『9』 で選択したファイルに保管されます。
12. そのファイルを認証局 (CA) に送信するか、CA の Web サイト上の要求フォームにそのファイルをコピーして、新しい個人用証明書を要求する。

コマンド行の使用

## 手順

**iKeycmd** または **runmqakm** コマンドのいずれかを使用して個人証明書を要求するには、次のコマンドを使用します。

- UNIX, Linux, and Windows システムで **iKeycmd** を使用する場合:

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

- runmqakm を使用する場合:

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

ここで、

**-db *filename***
CMS 鍵データベースの完全修飾ファイル名を指定します。

**-pw *password***
CMS 鍵データベースのパスワードを指定します。

**-target *filename***
認証要求のファイル名を指定します。

## 次のタスク

認証局から署名付き個人証明書を受け取ったら、129 ページの『UNIX、Linux、および Windows システムで個人証明書を鍵リポジトリーに受信する操作』に記載されているステップを使用して、鍵データベースに追加することができます。

### *UNIX*、*Linux*、および *Windows* システムで個人証明書を鍵リポジトリーに受信する操作

この手順を使用して、鍵データベース・ファイルに個人証明書を受信します。 鍵リポジトリーは、証明書要求を作成したリポジトリーと同じでなければなりません。

CA から新しい個人用証明書が送信された後、新しい証明書要求の生成に使用した鍵データベース・ファイルにその証明書を追加します。 CA が E メール・メッセージの一部として証明書を送信する場合、その証明書を別のファイルにコピーしてください。

### iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

インポートされる証明書ファイルに現在のユーザー用の書き込み権限があることを確認し、 キュー・マネージャーまたは WebSphere MQ MQI クライアントが個人用証明書を受信して鍵データベース・ファイルに入れるようにするため、次の手順を実行します。

1. **strmqikm** コマンドを使用して、iKeyman GUI を開始する (Windows UNIX and Linux の場合)。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書を追加する先の鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックし、「**OK (了解)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。 「**Personal Certificates (個人用証明書)**」ビューを選択する。
8. 「**Receive (受信)**」をクリックする。 「Receive Certificate from a File (ファイルからの証明書の受信)」ウィンドウが開きます。
9. 新しい個人用証明書の証明書ファイルの名前と位置を入力するか、「**Browse (ブラウズ)**」をクリックして名前と位置を選択する。
10. 「**OK**」をクリックします。 既に鍵データベースに個人用証明書がある場合は、追加する鍵をデータベース内のデフォルト鍵として設定するかどうかを確認するウィンドウが開きます。
11. 「**はい**」または「**いいえ**」をクリックする。 「Enter a Label (ラベルの入力)」ウィンドウが開きます。
12. 「**OK**」をクリックします。 「**Personal Certificates (個人用証明書)**」フィールドに、追加した新しい個人用証明書のラベルが表示されます。

### コマンド行の使用

iKeycmd を使用して鍵データベース・ファイルに個人証明書を追加するには、次のコマンドを使用します。

- UNIX、 Linux および Windows では、以下のコマンドを発行します。

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

ここで、

| | |
|---|---|
| -file *filename* | 個人用証明書を含むファイルの完全修飾ファイル名です。 |
| -db *filename* | CMS 鍵データベースの完全修飾ファイル名です。 |
| -pw *password* | CMS 鍵データベースのパスワードです。 |

| | |
|---|---|
| -format *ascii* | 証明書の形式です。 値は、Base64 エンコードの ASCII の場合は ascii、バイナリー DER データの場合は binary とします。 デフォルトは ascii です。 |

暗号ハードウェアを使用している場合は、144 ページの『PKCS #11 ハードウェアへの個人用証明書のインポート』を参照してください。

## 鍵リポジトリーからの *CA* 証明書の取り出し

CA 証明書を取り出すには、次の手順に従います。

### iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

取り出したい CA 証明書が入っているマシン上で、次の手順を実行してください。

1. **strmqikm** コマンドを使用して、iKeyman GUI を開始する。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 抽出元になる鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Key database content (鍵データベースの内容)**」フィールドで、「**Signer Certificates (署名者証明書)**」を選択し、取り出す証明書を選択する。
9. 「**Extract (取り出し)**」をクリックする。 「Extract a Certificate to a File (ファイルへの証明書の取り出し)」ウィンドウが開きます。
10. 証明書の「**Data type (データ・タイプ)**」を選択する。例えば、拡張子が .arm のファイルの場合は「**Base64-encoded ASCII data (Base64 エンコードの ASCII データ)**」を選択します。
11. 証明書ファイルの名前と、証明書を保管したい位置を入力するか、「**Browse (ブラウズ)**」 をクリックして名前と位置を選択する。
12. 「**OK**」をクリックします。 指定したファイルに証明書が書き込まれます。

### コマンド行の使用

iKeycmd を使用して CA 証明書を取り出すには、次のコマンドを使用します。

- UNIX、 Linux および Windows の場合:

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

ここで、

| | |
|---|---|
| -db *filename* | CMS 鍵データベースの完全修飾パス名です。 |
| -pw *password* | CMS 鍵データベースのパスワードです。 |
| -label *label* | 証明書に付加するラベルです。 |

| | |
|---|---|
| -target *filename* | 宛先ファイルの名前です。 |
| -format *ascii* | 証明書の形式です。 値は、Base64 エンコードの ASCII の場合は ascii、バイナリー DER データの場合は binary とします。 デフォルトは ascii です。 |

## *UNIX*、*Linux*、および *Windows* システムで鍵リポジトリーから自己署名証明書の公開部分を抽出する

自己署名証明書の公開部分を抽出する手順を取り上げます。

### iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

抽出する自己署名証明書の公開部分があるマシンで以下の手順を実行します。

1. **strmqikm** コマンドを使用して iKeyman GUI を開始します ( UNIX、 Linux および Windows の場合)。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書の抽出元になる鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Key database content (鍵データベースの内容)**」フィールドで、「**Personal Certificates (個人用証明書)**」を選択し、証明書を選択する。
9. 「**Extract Certificate (証明書の取り出し)**」をクリックします。 「Extract a Certificate to a File (ファイルへの証明書の取り出し)」ウィンドウが開きます。
10. 証明書の「**Data type (データ・タイプ)**」を選択する。例えば、拡張子が .arm のファイルの場合は「**Base64-encoded ASCII data (Base64 エンコードの ASCII データ)**」を選択します。
11. 証明書ファイルの名前と、証明書を保管したい位置を入力するか、「**Browse (ブラウズ)**」 をクリックして名前と位置を選択する。
12. 「**OK**」をクリックします。 指定したファイルに証明書が書き込まれます。 証明書を取り出す (エクスポートではなく) 場合は、証明書の公用部分だけが含まれるので、パスワードは必要ありません。

### コマンド行の使用

iKeycmd または runmqakm を使用して自己署名証明書の公開部分を抽出するには、以下のコマンドを実行します。

- UNIX、 Linux および Windows の場合:

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

- runmqakm を使用する場合:

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

ここで、

| | |
|---|---|
| -db *filename* | CMS 鍵データベースの完全修飾パス名です。 |
| -pw *password* | CMS 鍵データベースのパスワードです。 |
| -label *label* | 証明書に付加するラベルです。 |
| -target *filename* | 宛先ファイルの名前です。 |
| -format *ascii* | 証明書の形式です。 値は、Base64 エンコードの ASCII の場合は ascii、バイナリー DER データの場合は binary とします。 デフォルトは ascii です。 |

## *UNIX, Linux, and Windows* システムで *CA* 証明書 (または自己署名証明書の公開部分) を鍵リポジトリーに追加する操作

CA 証明書または自己署名証明書の公開部分を鍵リポジトリーに追加する手順を取り上げます。

追加する証明書が証明書チェーン内にある場合は、チェーン内でそれよりも上にある証明書もすべて追加する必要があります。 証明書は、root と、チェーン内でその直下にある CA 証明書から始めて、完全な降順で追加する必要があります。

ここで CA 証明書について取り上げている手順は、自己署名証明書の公開部分にも当てはまります。

**注 :** 追加する証明書が証明書チェーン内にある場合は、チェーン内でそれより上にある証明書もすべて追加する必要があります。 証明書が ASCII (UTF-8) またはバイナリー (DER) でエンコードされていることを確認する必要があります。これは、IBM Global Secure Toolkit (GSKit) はその他のタイプのエンコードによる証明書をサポートしないためです。 証明書を追加する順序は厳密に決まっており、最初は root で、次にチェーン内のその直下にある CA 証明書というように、降順で行う必要があります。

## iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

CA 証明書の追加先マシンで、次の手順を実行してください。

1. **strmqikm** コマンドを使用して iKeyman GUI を開始します ( UNIX、 Linux および Windows システムの場合)。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書を追加する先の鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Key database content (鍵データベースの内容)**」フィールドで、「**Signer Certificates (署名者証明書)**」を選択する。
9. 「**Add (追加)**」をクリックする。 「Add CA's Certificate from a File (ファイルからの CA の証明書の追加)」ウィンドウが開きます。
10. 証明書が保管されている証明書ファイルの名前と位置を入力するか、「**Browse (ブラウズ)**」 をクリックして名前と位置を選択する。
11. 「**OK**」をクリックします。 「Enter a Label (ラベルの入力)」ウィンドウが開きます。
12. 「Enter a Label (ラベルの入力)」ウィンドウに、証明書の名前を入力する。

13.「**OK**」をクリックします。 証明書が鍵データベースに追加されます。

## コマンド行の使用

iKeycmd を使用して CA 証明書を追加するには、次のコマンドを使用します。

- UNIX、 Linux および Windows では、以下のコマンドを発行します。

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

ここで、

| | |
|---|---|
| -db *filename* | CMS 鍵データベースの完全修飾パス名です。 |
| -pw *password* | CMS 鍵データベースのパスワードです。 |
| -label *label* | 証明書に付加するラベルです。 |
| -file *filename* | 証明書を含むファイルの名前です。 |
| -format *ascii* | 証明書の形式です。 値は、Base64 エンコードの ASCII の場合は ascii、バイナリー DER データの場合は binary とします。 デフォルトは ascii です。 |

## 鍵リポジトリーからの個人用証明書のエクスポート

個人証明書をエクスポートする手順を取り上げます。

## iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

エクスポートする個人用証明書が入っているマシンで、次の手順を実行してください。

1. **strmqikm** コマンドを使用して、iKeyman GUI を開始する (Windows UNIX and Linux の場合)。
2.「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3.「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4.「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書のエクスポート元になる鍵データベース・ファイル (例えば、key.kdb) を選択する。
6.「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8.「**Key database content (鍵データベースの内容)**」フィールドで、「**Personal Certificates (個人用証明書)**」を選択し、エクスポートしたい証明書を選択する。
9.「**Export/Import (エクスポート/インポート)**」をクリックする。 「Export/Import key (鍵のエクスポート/インポート)」ウィンドウが開きます。
10.「**Export Key (鍵のエクスポート)**」を選択する。
11. エクスポートする証明書の「**Key file type (鍵ファイル・タイプ)**」 を選択する (例: **PKCS12**)。
12. ファイル名および証明書のエクスポート先を入力するか、「**Browse (ブラウズ)**」をクリックして名前および位置を選択する。
13.「**OK**」をクリックします。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。 証明書をエクスポートする (取り出しではなく) 場合は、証明書の公用部分と私用部分の両方が含まれ

ます。 そのため、エクスポート・ファイルはパスワードによって保護されています。 証明書を抽出する場合は、証明書の公用部分だけが含まれるので、パスワードは必要ありません。

14. 「**Password (パスワード)**」フィールドにパスワードを入力し、「**Confirm Password (パスワードの確認)**」 フィールドにそのパスワードをもう一度入力する。
15. 「**OK**」をクリックします。 指定したファイルに証明書がエクスポートされます。

### コマンド行の使用

iKeycmd を使用して個人証明書をエクスポートするには、次のコマンドを使用します。

- UNIX、 Linux および Windows の場合:

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

ここで、

| | |
|---|---|
| `-db` *filename* | CMS 鍵データベースの完全修飾パス名です。 |
| `-pw` *password* | CMS 鍵データベースのパスワードです。 |
| `-label` *label* | 証明書に付加するラベルです。 |
| `-type` *cms* | データベースのタイプです。 |
| `-target` *filename* | 宛先ファイルの完全修飾パス名です。 |
| `-target_pw` *password* | 証明書を暗号化するためのパスワードです。 |
| `-target_type` *pkcs12* | 証明書のタイプです。 |

## 個人証明書を鍵リポジトリーにインポートする操作 *(UNIX, Linux, and Windows システム)*

個人証明書をインポートする手順を取り上げます。

PKCS #12 形式の個人用証明書を鍵データベース・ファイルにインポートする場合は、まず CA 証明書発行の有効なフル・チェーンを鍵データベース・ファイルに追加する必要があります (132 ページの『UNIX, Linux, and Windows システムで CA 証明書 (または自己署名証明書の公開部分) を鍵リポジトリーに追加する操作』を参照してください)。

PKCS #12 ファイルは一時的なものであり、使用後は削除する必要があります。

### iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

個人用証明書のインポート先マシンで、次の手順を実行してください。

1. **strmqikm** コマンドを使用して、iKeyman GUI を開始する。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが表示されます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書を追加する先の鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが表示されます。

7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. 「**Key database content (鍵データベースの内容)**」フィールドで、「**Personal Certificates (個人用証明書)**」を選択する。
9. 「個人証明書」ビューに証明書が存在する場合は、以下の手順に従います。
   a. 「**Export/Import (エクスポート/インポート)**」をクリックする。 「Export/Import key (鍵のエクスポート/インポート)」 ウィンドウが表示されます。
   b. 「**Import Key (鍵のインポート)**」を選択する。
10. 「個人証明書」ビューに証明書が存在しない場合は、「**インポート**」をクリックします。
11. インポートする証明書の「**Key file type (鍵ファイル・タイプ)**」を選択する (例: PKCS12)。
12. 証明書が保管されている証明書ファイルの名前と位置を入力するか、「**Browse (ブラウズ)**」 をクリックして名前と位置を選択する。
13. 「**OK**」をクリックします。 「Password Prompt (パスワード・プロンプト)」ウィンドウが表示されます。
14. 「**Password (パスワード)**」フィールドに、証明書のエクスポート時に使用したパスワードを入力する。
15. 「**OK**」をクリックします。 「Change Labels (ラベルの変更)」ウィンドウが表示されます。 例えば同じラベルを持つ証明書がターゲット鍵データベースに既に存在する場合は、このウィンドウでインポートする証明書のラベルを変更できます。 証明書ラベルを変更しても、証明書チェーンの妥当性検査には影響しません。 これは、証明書を特定のキュー・マネージャーまたはクライアント (例えば、`ibmwebspheremqqm1` ) に関連付けるために、個人証明書ラベルを WebSphere MQ が必要とするラベルに変更するために使用できます。
16. 「**Select a label to change (変更するラベルの選択)**」リストから必要なラベルを選択する。 ラベルは「**Enter a new label (新規ラベルの入力)**」入力フィールドにコピーされます。 ラベル・テキストを新規ラベルのものに置き換え、「**Apply (適用)**」をクリックします。
17. 「**Enter a new label (新規ラベルの入力)**」入力フィールドのテキストが「**Select a label to change (変更するラベルの選択):**」フィールドにコピーされて最初に選択されたラベルが置換され、それによって対応する証明書のラベルが変更される。
18. 変更する必要があるラベルをすべて変更したら、「**OK (了解)**」をクリックする。 「Change Labels (ラベルの変更)」ウィンドウが閉じ、最初の「IBM Key Management (IBM 鍵管理)」ウィンドウが再表示され、「**Personal Certificates (個人用証明書)**」 フィールドおよび「**Signer Certificates (署名者証明書)**」フィールドに正しいラベルの証明書が示されます。
19. 証明書がターゲット鍵データベースにインポートされる。

## コマンド行の使用

iKeycmd を使用して個人証明書をインポートするには、次のコマンドを使用します。

- UNIX、 Linux および Windows の場合:

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

ここで、

| | |
|---|---|
| `-file filename` | PKCS #12 証明書を含むファイルの完全修飾ファイル名です。 |
| `-pw password` | PKCS #12 証明書のパスワードです。 |
| `-type pkcs12` | ファイルのタイプです。 |
| `-target filename` | 宛先 CMS 鍵データベースの名前です。 |
| `-target_pw password` | CMS 鍵データベースのパスワードです。 |
| `-target_type cms` | `-target` で指定したデータベースのタイプです。 |

| | |
|---|---|
| -label *label* | ソース鍵データベースからインポートする証明書のラベルです。 |
| -new_label *label* | ターゲット・データベースで証明書に割り当てるラベルです。 -new_label オプションを 省略すると、デフォルトで -label オプションと同じものが使用されます。 |

iKeycmd には、証明書ラベルを直接変更するコマンドが提供されていません。 証明書ラベルを変更するには、次の手順に従ってください。

1. **-cert -export** コマンドを使用して、証明書を PKCS #12 ファイルにエクスポートします。 -label オプションに既存の証明書ラベルを指定します。
2. **-cert -delete** コマンドを使用して、証明書の既存コピーを元の鍵データベースから除去する。
3. **-cert -import** コマンドを使用して、PKCS #12 ファイルから証明書をインポートします。 -label オプションに古いラベルを、-new_label オプションに必要な新規ラベルを指定します。 必要なラベルが指定された証明書が、鍵データベースにインポートされて戻されます。

### *Microsoft .pfx* ファイルからのインポート

iKeyman を使用して Microsoft の .pfx ファイルからインポートする手順を取り上げます。 .pfx ファイルをインポートするのに runmqakm を使用することはできません。

.pfx ファイルには、同じ鍵に関連付けられた証明書が 2 つ含まれている場合があります。 1 つは個人用証明書またはサイト証明書です (公開鍵と秘密鍵の両方を含みます)。 もう 1 つは CA (署名者) 証明書です (公開鍵のみを含みます)。 これらの証明書は同じ CMS 鍵データベース・ファイル内で共存できないので、いずれか 1 つだけをインポートできます。 また、「分かりやすい名前」またはラベルは、署名者証明書にのみ付加されます。

個人用証明書は、システムによって生成される UUID (Unique User Identifier) で識別されます。 この節では、pfx ファイルから個人用証明書をインポートし、以前 CA (署名者) 証明書に割り当てられていた分かりやすい名前でラベルを付ける方法を説明します。 発行する CA (署名者) 証明書は、既にターゲット鍵データベースに追加されている必要があります。 PKCS#12 ファイルは一時的なものであり、使用後は削除する必要があります。

以下のステップを実行して、ソース pfx 鍵データベースから個人用証明書をインポートします。

1. **strmqikm** コマンドを使用して iKeyman GUI を開始します ( Linux、 UNIX または Windows の場合)。「IBM Key Management (IBM 鍵管理)」ウィンドウが表示されます。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが表示されます。
3. **PKCS12** の鍵データベース・タイプを選択する。
4. **このステップを実行する前に、pfx データベースのバックアップを取ることが推奨されている**。 インポートする pfx 鍵データベースを選択する。「**Open (オープン)**」をクリックする。「Password Prompt (パスワード・プロンプト)」ウィンドウが表示されます。
5. 鍵データベース・パスワードを入力し、「**OK (了解)**」をクリックする。 「IBM Key Management (IBM 鍵管理)」ウィンドウが表示されます。 タイトル・バーに、選択した pfx 鍵データベース・ファイルの名前が表示され、ファイルが開かれて準備できていることが示されます。
6. リストから「**Signer Certificates (署名者証明書)**」を選択する。 必要な証明書の「分かりやすい名前」が、ラベルとして「署名者証明書」パネルに表示されます。
7. ラベル項目を選択し、「**Delete (削除)**」をクリックして署名者証明書を削除する。 「Confirm (確認)」ウィンドウが表示されます。
8. 「**Yes (はい)**」をクリックする。 選択したラベルが「Signer Certificates (署名者証明書)」パネルに表示されなくなります。
9. すべての署名者証明書に関して、ステップ 6、7、および 8 を繰り返す。
10. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが表示されます。
11. pfx ファイルのインポート先となるターゲット鍵 CMS データベースを選択する。 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが表示されます。

12. 鍵データベース・パスワードを入力し、「**OK (了解)**」をクリックする。 「IBM Key Management (IBM 鍵管理)」ウィンドウが表示されます。 タイトル・バーに、選択した鍵データベース・ファイルの名前が表示され、ファイルが開かれて準備が整っていることが示されます。
13. リストから「**Personal Certificates (個人用証明書)**」を選択する。
14. 「個人証明書」ビューに証明書が存在する場合は、以下の手順に従います。
    a. 「**Export/Import key (鍵のエクスポート/インポート)**」をクリックする。 「Export/Import key (鍵のエクスポート/インポート)」 ウィンドウが表示されます。
    b. 「Choose Action Type (操作の選択)」から「**Import (インポート)**」を選択する。
15. 「個人証明書」ビューに証明書が存在しない場合は、「**インポート**」をクリックします。
16. PKCS12 ファイルを選択します。
17. ステップ 4 で使用した pfx ファイルの名前を入力します。 「**OK**」をクリックします。 「Password Prompt (パスワード・プロンプト)」ウィンドウが表示されます。
18. 署名者証明書を削除したときに指定したパスワードを指定する。 「**OK**」をクリックします。
19. 「Change Labels (ラベルの変更)」ウィンドウが表示される (インポートできるのは単一の証明書だけなので)。 証明書のラベルは、xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx の形式の UUID である必要があります。
20. ラベルを変更するには、「**Select a label to change (変更するラベルの選択):**」パネル から UUID を選択する。 ラベルは「**Enter a new label (新規ラベルの入力):**」フィールドに複製されます。 ラベル・テキストをステップ 7 で削除した分かりやすい名前に置き換え、「**Apply (適用)**」をクリックします。 分かりやすい名前の形式は、 ibmwebspheremq の後にキュー・マネージャー名または WebSphere MQ MQI クライアント・ユーザー・ログオン ID (小文字) を続ける必要があります。
21. 「**OK**」をクリックします。 「Change Labels (ラベルの変更)」ウィンドウが閉じ、最初の「IBM Key Management (IBM 鍵管理)」ウィンドウが再表示され、「Personal Certificates (個人用証明書)」パネルおよび「Signer Certificates (署名者証明書)」パネルに正しいラベルの個人用証明書が示されます。
22. pfx 個人用証明書が (ターゲット) データベースにインポートされます。

iKeycmd を使用して証明書ラベルを変更することはできません。

### *PKCS #7 ファイルからのインポート*

iKeyman および iKeycmd ツールは、PKCS #7 (.p7b) ファイルをサポートしていません。 PKCS #7 ファイルから証明書をインポートするには runmqckm ツールを使用します。

PKCS #7 ファイルから CA 証明書を追加するには、次のコマンドを使用します。

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

| | |
|---|---|
| -db *filename* | CMS 鍵データベースの完全修飾ファイル名です。 |
| -pw *password* | 鍵データベースのパスワードです。 |
| -type *cms* | 鍵データベースのタイプです。 |
| -file *filename* | PKCS #7 ファイルの名前です。 |
| -label *label* | ターゲット・データベースで証明書に割り当てるラベルです。 最初の証明書は指定されたラベルを使用します。 他の証明書があれば、それらの証明書にはすべてサブジェクト名のラベルが付きます。 |

PKCS #7 ファイルから個人証明書をインポートするには、次のコマンドを使用します。

```
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 証明書を含むファイルの完全修飾ファイル名です。 |
| -pw *password* | PKCS #7 証明書のパスワードです。 |

| | |
|---|---|
| -type *pkcs7* | ファイルのタイプです。 |
| -target *filename* | 宛先鍵データベースの名前です。 |
| -target_pw *password* | 宛先鍵データベースのパスワードです。 |
| -target_type *cms* | -target で指定したデータベースのタイプです。 |
| -label *label* | インポートする証明書のラベルです。 |
| -new_label *label* | ターゲット・データベースで証明書に割り当てるラベルです。<br>-new_label オプションを 省略すると、デフォルトで -label オプションと同じものが使用されます。 |

## *UNIX, Linux, and Windows* システムで鍵リポジトリーから証明書を削除する操作

個人証明書または CA 証明書を削除する手順を取り上げます。

### iKeyman の使用

SSL 証明書を FIPS に準拠した方法で管理する必要がある場合は、runmqakm コマンドを使用します。 iKeyman は、FIPS 準拠のオプションを提供していません。

1. **strmqikm** コマンドを使用して iKeyman GUI を開始します ( UNIX、 Linux および Windows システムの場合)。
2. 「**Key Database File (鍵データベース・ファイル)**」メニューから、「**Open (オープン)**」をクリックする。 「Open (オープン)」ウィンドウが開きます。
3. 「**Key database type (鍵データベース・タイプ)**」をクリックし、「**CMS**」 (Certificate Management System) を選択する。
4. 「**Browse (ブラウズ)**」をクリックして、鍵データベース・ファイルが入っているディレクトリーまでナビゲートする。
5. 証明書の削除元になる鍵データベース・ファイル (例えば、key.kdb) を選択する。
6. 「**Open (オープン)**」をクリックする。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
7. 鍵データベースの作成時に設定したパスワードを入力し、「**OK (了解)**」をクリックする。 鍵データベース・ファイルの名前が 「**File Name (ファイル名)**」フィールドに表示されます。
8. ドロップダウン・リストから、「**Personal Certificates (個人用証明書)**」または「**Signer Certificates (署名者証明書)**」を選択します。
9. 削除する証明書を選択する。
10. 証明書のコピーがまだないときに、その証明書を保管したい場合は、「**Export/Import (エクスポート/インポート)**」をクリックしてエクスポートする (133 ページの『鍵リポジトリーからの個人用証明書のエクスポート』を参照)。
11. 証明書を選択して、「**Delete (削除)**」をクリックする。 「Confirm (確認)」ウィンドウが開きます。
12. 「**Yes (はい)**」をクリックする。 「**Personal Certificates (個人用証明書)**」フィールドに、削除した証明書のラベルが表示されなくなります。

### コマンド行の使用

iKeycmd または runmqakm を使用して証明書を削除するには、以下のコマンドを実行します。

- UNIX、 Linux および Windows の場合:

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

ここで、

| | |
|---|---|
| -db *filename* | CMS 鍵データベースの完全修飾ファイル名です。 |

| | |
|---|---|
| -pw *password* | CMS 鍵データベースのパスワードです。 |
| -label *label* | 個人用証明書に付加するラベルです。 |
| -fips | コマンドが FIPS モードで実行されるように指定します。 このモードは、BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードの場合、ICC コンポーネントは、FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**runmqakm** コマンドは失敗します。 |

## 鍵リポジトリー保護のための強力なパスワードの生成

**runmqakm** コマンドを使用して、鍵リポジトリー保護のための強力なパスワードを生成できます。

**runmqakm** コマンドに次のパラメーターを指定して使用することにより、強力なパスワードを生成することができます。

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

それ以降、生成されたパスワードを証明書管理コマンドの **-pw** パラメーターに使用するときは、必ずパスワードを二重引用符で囲みます。 UNIX and Linux システムでは、パスワード・ストリングに以下の文字が含まれている場合、それらをエスケープするためにバックスラッシュ文字を使用することも必要です。

```
!   \   "    '
```

**runmqckm**、**runmqakm** または iKeyman GUI からのプロンプトに対する応答にパスワードを入力する場合は、パスワードを引用符で囲んだり、エスケープしたりする必要はありません。 それは、この場合にデータ入力がオペレーティング・システム・シェルの影響を受けないためです。

## 暗号ハードウェア *(UNIX, Linux, and Windows システム)*

キュー・マネージャーまたはクライアントの暗号ハードウェアを構成する方法はいくつかあります。

次の方式のどちらかを使用すると、UNIX、Linux、または Windows システム上でキュー・マネージャー用に暗号ハードウェアを構成できます。

- ALTER QMGR で説明しているように、SSLCRYP パラメーターを指定した ALTER QMGR MQSC コマンドを使用する。
- IBM WebSphere MQ エクスプローラーを使用して、UNIX、Linux、または Windows システム上で暗号ハードウェアを構成する。 詳細については、オンライン・ヘルプを参照してください。

以下のいずれかの方法を使用して、 UNIX、 Linux 、または Windows システム上の WebSphere MQ クライアント用に暗号ハードウェアを構成できます。

- MQSSLCRYP 環境変数を設定する。 ALTER QMGR で説明しているように、MQSSLCRYP に指定可能な値は、SSLCRYP パラメーターの場合と同じです。 SSLCRYP パラメーターの GSK_PCS11 バージョンを使用する場合は、PKCS #11 トークン・ラベル全体を小文字で指定する必要があります。
- MQCONNX 呼び出しで、SSL 構成オプション構造である MQSCO の **CryptoHardware** フィールドを設定する。 詳しくは、MQSCO の概要を参照してください。

PKCS #11 インターフェースを使用する暗号ハードウェアを上記のいずれかの方法で構成した場合は、チャネルで使用する個人用証明書を、構成した暗号トークンの鍵データベース・ファイルに保管する必要があります。 これについては、139 ページの『PKCS #11 ハードウェアでの証明書の管理』で説明されています。

### *PKCS #11* ハードウェアでの証明書の管理

PKCS #11 インターフェースをサポートする暗号ハードウェアにおける デジタル証明書を管理できます。

## このタスクについて

認証局 (CA) 証明書を保管する予定はないものの、証明書をすべて暗号化ハードウェアに保管する場合であっても、鍵データベースを作成して IBM WebSphere MQ 環境を用意する必要があります。 鍵データベース

は、キュー・マネージャーがその SSLKEYR フィールドを参照するため、またはクライアント・アプリケーションが MQSSLKEYR 環境変数を参照するために必要です。 この鍵データベースは、認証要求を作成するときにも必要です。

鍵データベースを作成するには、コマンド行を使用するか、 **strmqikm** (iKeyman) ユーザー・インターフェースを使用します。

## 手順

コマンド・ラインを使用して鍵データベースを作成します。

1. 以下のいずれかのコマンドを実行します。
   - UNIX, Linux, and Windows システムの場合:

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

   - runmqakm を使用する場合:

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

   ここで、

   **-db *filename***
   CMS 鍵データベースの完全修飾ファイル名を指定します。ファイル拡張子は .kdb である必要があります。

   **-pw *password***
   CMS 鍵データベースのパスワードを指定します。

   **-type *cms***
   データベースのタイプを指定します。 (IBM WebSphere MQ の場合、これは cms でなければなりません。)

   **-stash**
   鍵データベース・パスワードをファイルに保存します。

   **-fips**
   BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードでは、ICC コンポーネントは FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**runmqakm** コマンドは失敗します。

   **-強い**
   入力されたパスワードがパスワード強度の最小要件を満たしているかどうかを確認します。 パスワードの最小要件は、次のとおりです。

   - パスワードは、最小 14 文字の長さでなければならない。
   - パスワードには、少なくとも 1 つの小文字、1 つの大文字、および 1 つの数字または特殊文字が含まれている必要がある。 特殊文字には、アスタリスク (*)、ドル記号 ($)、番号記号 (#)、およびパーセント記号 (%) が含まれます。スペースは特殊文字として分類されます。
   - パスワード中の同じ文字はそれぞれ最大 3 回までしか使用できない。
   - パスワード内に連続して出現する同じ文字は最大 2 文字。
   - すべての文字が、ASCII の標準印刷可能文字セットの 0x20 から 0x7E までの範囲内の文字である。

あるいは、 **strmqikm** (iKeyman) ユーザー・インターフェースを使用して、鍵データベースを作成します。

2. UNIX and Linux システムの場合は、root ユーザーとしてログインする。 Windows システムの場合は、管理者または MQM グループのメンバーとしてログインする。
3. **strmqikm** コマンドを実行して、iKeyman ユーザー・インターフェースを開始する。
4. 「**鍵データベース・ファイル**」 > 「**オープン**」をクリックします。
5. 「**鍵データベース・タイプ**」をクリックして、**PKCS11Direct** を選択します。

6. 「**File Name (ファイル名)**」フィールドに、暗号ハードウェアを管理するためのモジュールの名前を入力する (例: PKCS11_API.so)。

   PKCS #11 暗号ハードウェアに保管した証明書またはキーを使用する場合には、 iKeycmd および iKeyman が 64 ビット・プログラムであることに注意してください。 PKCS #11 サポートに必要な外部モジュールは、64 ビット・プロセスにロードされます。したがって、暗号ハードウェアの管理用に 64 ビットの PKCS #11 ライブラリーがインストールされています。 Windows および Linux x86 32 ビット・プラットフォームは唯一の例外です。これらのプラットフォームでは、 iKeyman および iKeycmd プログラムは 32 ビットです。
7. 「**Location (ロケーション)**」フィールドに、パスを入力します。

   - UNIX and Linux システムでは、例えば /usr/lib/pksc11 のように入力します。
   - Windows システムでは、cryptoki などのライブラリー名を入力できます。

   「**OK**」をクリックします。 「Open Cryptographic Token (暗号トークンのオープン)」ウィンドウが開きます。
8. 「**Cryptographic Token Password (暗号トークン・パスワード)**」フィールドに、暗号ハードウェアの構成時に設定したパスワードを入力する。
9. 個人用証明書の受信またはインポートに 必要な署名者証明書を保持するための容量が暗号ハードウェアにある場合は、2 次鍵データベースのチェック・ボックスを両方ともクリアし、ステップ 141 ページの『13』 から続行する。

   2 次 CMS 鍵データベースに署名者証明書を保持させる必要がある場合は、「**Open existing secondary key database file (既存の 2 次鍵データベース・ファイルをオープン)**」または「**Create new secondary key database file (新規 2 次鍵データベース・ファイルを作成)**」 のいずれかを選択する。
10. 「**File Name (ファイル名)**」フィールドに、ファイル名を入力する。 このフィールドには、既に key.kdb というテキストが入っています。 語幹名が key である場合は、このフィールドを変更しないでください。 別の語幹名を指定した場合は、key をご使用の語幹名で置き換えてください。 .kdb 接尾部は変更してはいけません。
11. 「**Location (ロケーション)**」フィールドに、パスを入力する。例えば、次のようにします。

    - キュー・マネージャーの場合、/var/mqm/qmgrs/QM1/ssl
    - IBM WebSphere MQ MQI クライアントの場合: /var/mqm/ssl

    「**OK**」をクリックします。 「Password Prompt (パスワード・プロンプト)」ウィンドウが開きます。
12. パスワードを入力してください。

    ステップ 141 ページの『9』 で「**既存の 2 次鍵データベース・ファイルをオープン**」を選択した場合は、「**パスワード**」フィールドにパスワードを入力します。

    ステップ 141 ページの『9』 で 「**新規 2 次鍵データベース・ファイルの作成**」 を選択した場合は、以下のサブステップを実行します。

    a) 「**Password (パスワード)**」フィールドにパスワードを入力し、「**Confirm Password (パスワードの確認)**」 フィールドにそのパスワードをもう一度入力する。

    b) 「**Stash the password to a file (ファイルにパスワードを隠す)**」を選択する。 パスワードを隠さない場合は、鍵データベース・ファイルへのアクセスに必要なパスワードを取得できないので、SSL チャネルを開始しようとしても失敗します。

    c) 「**OK**」をクリックします。 パスワードが key.sth ファイル内にあることを確認するウィンドウが開きます (別の語幹名を指定した場合を除く)。
13. 「**OK**」をクリックします。 「Key database content (鍵データベースの内容)」フレームが表示されます。

*PKCS #11* ハードウェア用の個人用証明書の要求

キュー・マネージャーまたは IBM WebSphere MQ MQI クライアントから暗号化ハードウェア用の個人証明書を要求するには、以下の手順を使用します。

*iKeyman* ユーザー・インターフェースの使用

## このタスクについて

**注 :** WebSphere MQ は、SHA-3 アルゴリズムと SHA-5 アルゴリズムをサポートしません。 デジタル署名アルゴリズム名 SHA384WithRSA および SHA512WithRSA は SHA-2 ファミリーのメンバーであるため、これらのアルゴリズムは使用可能です。

デジタル署名アルゴリズム名 SHA3WithRSA および SHA5WithRSA は、それぞれ SHA384WithRSA および SHA512WithRSA の簡略形であるため、これらは推奨されません。

## 手順

iKeyman ユーザー・インターフェースから個人証明書を要求するには、以下の手順に従います。

1. 暗号ハードウェアを使用する手順を実行します。 139 ページの『PKCS #11 ハードウェアでの証明書の管理』を参照してください。
2. 「**Create (作成)**」メニューから、「**New Certificate Request (新規認証要求)**」をクリックする。
   「Create New Key and Certificate Request (新規鍵および認証要求の作成)」ウィンドウが開きます。
3. 「**鍵ラベル**」フィールドに、以下のラベルを入力します。
   - キュー・マネージャーの場合は、`ibmwebspheremq` の後に続けて、小文字に変更されたキュー・マネージャーの名前。 例えば、 `QM1` というキュー・マネージャーの場合は、 `ibmwebspheremqqm1` と入力します。
   - IBM WebSphere MQ MQI client の場合は、`ibmwebspheremq` の後にログオン・ユーザー ID をすべて小文字で入力します (例: `ibmwebspheremqmyuserid` )。
4. 「**共通名**」および「**組織**」に値を入力し、「**国**」を選択します。 残りのオプション・フィールドでは、デフォルト値を受け入れるか、別の値を入力または選択します。
   「**Organizational Unit (部門)**」フィールドに 指定できる名前は 1 つだけです。 これらのフィールドについて詳しくは、11 ページの『識別名』を参照してください。
5. 「**Enter the name of a file in which to store the certificate request (認証要求を保管するファイル名の入力)**」フィールドで、デフォルトの `certreq.arm` を受け入れるか、絶対パスと一緒に新しい値を入力する。
6. 「**OK**」をクリックします。
   確認ウィンドウが開きます。
7. 「**OK**」をクリックします。
   「**Personal Certificate Requests (個人用証明書の要求)**」 リストに、作成した新しい個人用証明書要求のラベルが表示されます。 証明書要求は、ステップ 142 ページの『5』 で選択したファイルに保管されます。
8. そのファイルを認証局 (CA) に送信するか、CA の Web サイト上の要求フォームにそのファイルをコピーして、新しい個人用証明書を要求する。

コマンド行の使用

## 手順

**runmqckm** または **runmqakm** コマンドのいずれかを使用して個人証明書を要求するには、次のコマンドを使用します。

- **runmqckm** を使用する場合:

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

-dn *distinguished_name* の代わりに、-san_dsname *DNS_names* 、-san_emailaddr *email_addresses* 、または -san_ipaddr *IP_addresses* を使用できます。

- runmqakm を使用する場合:

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

ここで、

**-db *filename***
CMS 鍵データベースの完全修飾ファイル名を指定します。

**-pw *password***
CMS 鍵データベースのパスワードを指定します。

**-label *label***
証明書に付加する鍵ラベルを指定します。

**-dn *distinguished_name***
二重引用符で囲んだ X.500 識別名を指定します。 少なくとも 1 つの属性が必要です。 複数の OU および DC 属性を指定できます。

**-size *key_size***
鍵のサイズを指定します。 **runmqckm** を使用している場合、値は 512 または 1024 にします。 **runmqakm** を使用している場合、値は 512、1024、または 2048 にします。

**-file *filename***
認証要求のファイル名を指定します。

**-fips**
コマンドが FIPS モードで実行されるように指定します。 このモードは、BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードでは、ICC コンポーネントは FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**runmqakm** コマンドは失敗します。

**-sig_alg**
**runmqckm** の場合、項目の鍵ペアの作成に使用される非対称署名アルゴリズムを指定します。 値は、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、または SHAWithRSA のいずれかです。 デフォルト値は SHA1WithRSA です。

**-sig_alg**
**runmqakm** の場合、認証要求の作成中に使用されるハッシュ・アルゴリズムを指定します。 このハッシュ・アルゴリズムは、新たに作成された証明書要求に関連付けられた署名を作成するために使用されます。 値は、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、または EC_ecdsa_with_SHA512 のいずれかです。 デフォルト値は SHA1WithRSA です。

**-san_dnsname *DNS_names***
作成される項目の DNS 名のコンマ区切りまたはスペース区切りリストを指定します。

**-san_emailaddr *email_addresses***
作成される項目の E メール・アドレスのコンマ区切りまたはスペース区切りリストを指定します。

**-san_ipaddr *IP_addresses***
作成される項目の IP アドレスのコンマ区切りまたはスペース区切りリストを指定します。

*PKCS #11* ハードウェアへの個人用証明書のインポート
キュー・マネージャーまたは IBM WebSphere MQ MQI クライアントが個人証明書を暗号ハードウェアにインポートするための手順を取り上げます。

*iKeyman* の使用

## 手順

iKeyman ユーザー・インターフェースから個人証明書を要求するには、以下の手順に従います。

1. 暗号ハードウェアを使用する手順を実行します。 139 ページの『PKCS #11 ハードウェアでの証明書の管理』を参照してください。
2. 「**Receive (受信)**」をクリックする。 「Receive Certificate from a File (ファイルからの証明書の受信)」ウィンドウが開きます。
3. 新しい個人証明書の 「**データ・タイプ**」 を選択します。例えば、拡張子が `.arm` のファイルの場合は `Base64-encoded ASCII data` を選択します。
4. 新しい個人用証明書の証明書ファイルの名前と位置を入力するか、「**Browse (ブラウズ)**」をクリックして名前と位置を選択する。
5. 「**OK**」をクリックします。 既に鍵データベースに個人用証明書がある場合は、追加する鍵をデータベース内のデフォルト鍵として設定するかどうかを確認するウィンドウが開きます。
6. 「**はい**」または「**いいえ**」をクリックする。 「Enter a Label (ラベルの入力)」ウィンドウが開きます。
7. ラベルを入力します。
   例えば、個人証明書を要求したときに使用したラベルと同じラベルを使用できます。 このラベルは、正しい IBM WebSphere MQ 形式でなければならないことに注意してください。
   - キュー・マネージャーの場合は、`ibmwebspheremq` の後に続けてキュー・マネージャーの名前を小文字で入力します。 例えば、 QM1 という名前のキュー・マネージャーの場合、ラベルは `ibmwebspheremqqm1` になります。
   - IBM WebSphere MQ MQI クライアントの場合は、`ibmwebspheremq` の後に続けてログオン・ユーザー ID を小文字で入力します。 例えば、ユーザー ID `MyUserID` の場合、ラベルは `ibmwebspheremqmyuserid` になります。
8. 「**OK**」をクリックします。 「**個人用証明書**」リストに、追加した新しい個人用証明書のラベルが表示されます。 このラベルは、ユーザーが提供したラベルの前に暗号トークン・ラベルが追加された構成になっています。

コマンド行の使用

## 手順

コマンド・ラインから個人証明書を要求するには、以下の手順に従います。

1. ご使用の環境に合わせて構成されたコマンド・ウィンドウを開きます。
2. ご使用のオペレーティング・システムおよび構成に該当する以下のコマンドを入力します。
   - Windows、UNIX and Linux システムでは、以下のいずれかのコマンドを使用します。

```
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
```

   ここで、

   **-file *filename***
   個人用証明書を含むファイルの完全修飾ファイル名を指定します。

**-crypto *path***
ハードウェアに用意されている PKCS #11 ライブラリーの完全修飾パスを指定します。

**-tokenlabel *hardware_token***
インストール中に暗号化ハードウェアのストレージ部分に付けられたラベルを指定します。

**-pw *hardware_password***
ハードウェアへアクセスするためのパスワードを指定します。

**-format *cert_format***
証明書の形式を指定します。 値は、Base64 エンコードの ASCII の場合は `ascii`、バイナリー DER データの場合は `binary` とします。 デフォルトは CSIF です。

**-fips**
コマンドが FIPS モードで実行されるように指定します。このモードは、BSafe 暗号ライブラリーを使用不可にします。 ICC コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化される必要があります。 FIPS モードでは、ICC コンポーネントは FIPS 140-2 で検証されたアルゴリズムを使用します。 ICC コンポーネントが FIPS モードで初期化されない場合、**`runmqakm`** コマンドは失敗します。

# ユーザーの識別および認証

MQCSP 構造を使用して、またはユーザー出口プログラムのいくつかのタイプで、ユーザーを識別および認証することができます。

## MQCSP 構造の使用

MQCSP 接続セキュリティー・パラメーター構造体は MQCONNX 呼び出しで指定します。この構造体にはユーザー ID とパスワードが含まれます。 必要に応じて、セキュリティー出口で MQCSP を変更することができます。

**注 :** オブジェクト権限マネージャー (OAM) はパスワードを使用しません。 ただし、OAM は、ユーザー ID に対し、簡単な認証と言えなくもない限られた処理は行います。 上記のパラメーターをアプリケーションで使用すると、これらのチェック作業によって、ユーザーが他のユーザーの ID を採用するのを防ぐことができます。

## セキュリティー出口による識別と認証の実装

セキュリティー出口の主な目的は、チャネルの両端にある MCA が、相手側の MCA を認証できるようにすることです。 メッセージ・チャネルの両端、および MQI チャネルのサーバー側にある MCA は、通常、接続しているキュー・マネージャーの代理をします。 MQI チャネルのクライアント側にある MCA は、通常、WebSphere MQ クライアント・アプリケーションのユーザーの代理をします。 この状況での相互認証は、実際には、2 つのキュー・マネージャー間で、またはキュー・マネージャーと、WebSphere MQ MQI クライアント・アプリケーションのユーザーとの間で行われます。

用意されているセキュリティー出口 (SSPI チャネル出口) は、認証トークンの交換によって相互認証を実装する方法を示した実例です。認証トークンは、信頼できる認証サーバー (Kerberos など) によって生成され、その後検査されることになります。 詳細については、103 ページの『SSPI チャネル出口プログラム』を参照してください。

また、相互認証は、公開鍵インフラストラクチャー (PKI) テクノロジーを使用することによってもインプリメントすることができます。 各セキュリティー出口は、ランダム・データを生成し、そのセキュリティー出口が代理をするキュー・マネージャーまたはユーザーの秘密鍵を使用して署名し、その署名されたデータをセキュリティー・メッセージ内で相手側に送信します。 相手側のセキュリティー出口は、そのキュー・マネージャーまたはユーザーの公開鍵を使用してデジタル署名を検査することによって、認証を実行します。 複数のアルゴリズムが使用可能である場合、デジタル署名を交換する前に、双方のセキュリティー出口が、メッセージ・ダイジェストの生成用のアルゴリズムを合意する必要があります。

セキュリティー出口が、署名されたデータを相手側に送信する場合、そのセキュリティー出口が代理をするキュー・マネージャーまたはユーザーを識別する手段も送信する必要があります。 これは、識別名、またはデジタル証明書にすることができます。 デジタル証明書が送信される場合、相手側のセキュリティー

出口は、証明書チェーンをルート CA 証明書までたどることによって、その証明書を検証することができます。 これによって、デジタル署名の検査に使用される公開鍵の所有権が保証されます。

相手側のセキュリティー出口がデジタル証明書を検証できるのは、証明書チェーン内の残りの証明書が入っている鍵リポジトリーへのアクセス権がある場合だけです。 キュー・マネージャーまたはユーザー用のデジタル証明書が送信されない場合、相手側のセキュリティー出口がアクセス権を持つ鍵リポジトリーで、デジタル証明書が入手可能でなければなりません。 相手側のセキュリティー出口は、署名者の公開鍵が見つからない場合、デジタル署名を検査することはできません。

Secure Sockets Layer (SSL) と Transport Layer Security (TLS) では、ここで取り上げたような PKI 手法が採用されています。 SSL および TLS がどのように認証を実行するかの詳細については、14 ページの『Secure Sockets Layer (SSL) および Transport Layer Security (TLS) の概念』を参照してください。

信頼される認証サーバーまたは PKI サポートが使用できない場合、別の手法を使用できます。 セキュリティー出口でインプリメントできる一般的な手法では、対称鍵アルゴリズムを使用します。

セキュリティー出口の 1 つで、出口 A は乱数を生成し、それをセキュリティー・メッセージの中でパートナー・セキュリティー出口である出口 B に送信します。 出口 B は、2 つのセキュリティー出口にしか認識されない鍵のコピーを使用して、番号を暗号化します。 出口 B は、暗号化された乱数を、出口 B が生成した 2 つ目の乱数とともに、セキュリティー・メッセージ内で出口 A に送信します。 出口 A は、最初の乱数が正しく暗号化されたかどうかを確認し、鍵のコピーを使用して 2 つ目の乱数を暗号化し、その暗号化された乱数を、セキュリティー・メッセージ内で出口 B に送信します。 次に、出口 B は、2 つ目の乱数が正しく暗号化されたかどうかを検証します。 このやりとりの間、どちらかのセキュリティー出口が、相手側の確実性を確信できない場合、そのセキュリティー出口は、チャネルをクローズするように MCA に指示できます。

この手法の利点は、このやりとりの間に通信接続を介して鍵もパスワードも送信されないことです。 不利な点は、共有鍵を安全な方法で配布する方法の問題に対する解決法を提供しないことです。 この問題の 1 つの解決法が、226 ページの『ユーザー出口プログラムでの機密性の実装』で説明されています。 SNA では、2 つの LU がバインドしてセッションを形成するときに、ほぼ同じ手法が、この 2 つの LU の相互認証に使用されます。 この手法は、74 ページの『セッション・レベルの認証』で説明されています。

ここでは、相互認証の手法を取り上げましたが、そのすべては、片方向認証に合わせて調整できます。

## メッセージ出口による識別と認証の実装

アプリケーションがメッセージをキューに入れるときに、メッセージ記述子内の *UserIdentifier* フィールドには、そのアプリケーションに関連したユーザー ID が入ります。 しかし、ユーザー ID の認証に使用できるデータはありません。 このデータは、チャネルの送信側のメッセージ出口によって追加し、そのチャネルの受信側のメッセージ出口によって検査することができます。 認証データは、例えば、暗号化されたパスワード、またはデジタル署名にすることができます。

このサービスは、アプリケーション・レベルでインプリメントされる場合の方が効果的です。 基本的な要件は、メッセージを受信するアプリケーションのユーザーが、メッセージを送信したアプリケーションのユーザーを識別し、認証できることです。 したがって、当然、このサービスをアプリケーション・レベルでインプリメントすることを検討すべきです。 詳しくは、149 ページの『API 出口と API 交差出口による識別マッピング』を参照してください。

## API 出口と API 交差出口による識別と認証の実装

個々のメッセージのレベルでは、識別と認証は、2 つのユーザー、つまりメッセージの送信側と受信側に関与するサービスです。 基本的な要件は、メッセージを受信するアプリケーションのユーザーが、メッセージを送信したアプリケーションのユーザーを識別し、認証できることです。 この要件は、両方向の認証ではなく、単方向の認証用であることに注意してください。

このサービスのインプリメントの方法に応じて、ユーザーとそのアプリケーションは、このサービスとのインターフェースを取るか、このサービスと相互作用する必要がある場合があります。 さらに、サービスの使用時期と使用方法は、ユーザーとそのアプリケーションが置かれる場所により、またアプリケーション自体の性質により異なります。 したがって、このサービスを、リンク・レベルではなく、アプリケーション・レベルでインプリメントすることを検討する方が妥当です。

このサービスをリンク・レベルでインプリメントすることを検討する場合は、次のような問題の解決が必要になる場合があります。

- メッセージ・チャネル上で、このサービスを必要とするメッセージに対してのみ、このサービスを適用するには、どうするか
- ユーザーとそのアプリケーションが、このサービスとのインターフェースを取るか、相互作用することができるようにする (それが要件である場合) には、どうするか
- メッセージがあて先までの途中にある複数のメッセージ・チャネルを介して送信される、マルチホップ状況では、このサービスのコンポーネントをどこで起動するか

ここでは、識別と認証のサービスをアプリケーション・レベルで実装する例をいくつか取り上げます。 ここで使用する *API* 出口 という語は、API 出口または API 交差出口のいずれかを指します。

- アプリケーションがキューにメッセージを書き込むときに、API 出口は、Kerberos などの信頼される認証サーバーから認証トークンを取得できます。 API 出口は、このトークンをメッセージ内のアプリケーション・データに追加することができます。 メッセージが受信側アプリケーションによって取り出されるときに、2 つ目の API 出口は、そのトークンを検査して送信側を認証するように、認証サーバーに依頼することができます。
- アプリケーションがキューにメッセージを入れるときに、API 出口は、メッセージ内のアプリケーション・データに次の項目を付加することができます。
  - 送信側のデジタル証明書
  - 送信側のデジタル署名

  メッセージ・ダイジェストの生成に複数のアルゴリズムが使用可能である場合、API 出口には、使用しているアルゴリズムの名前を組み込むことができます。

  メッセージが受信側アプリケーションによって取り出されるときに、2 つ目の API 出口は、次の検査を実行できます。

  - API 出口は、証明書チェーンをルート CA 証明書までたどることによって、デジタル証明書を検証できます。 これを行うために、API 出口には、証明書チェーン内の残りの証明書が入っている鍵リポジトリーへのアクセス権が必要です。 この検査により、識別名によって指定される送信側が、その証明書に入っているる公開鍵の本物の所有者であることが保証されます。
  - API 出口は、証明書に入っている公開鍵を使用して、デジタル署名を検査することができます。 この検査は、送信側を認証します。

  デジタル証明書全体を送信する代わりに、送信側の識別名を送信することができます。 この場合、2 番目の API 出口が送信側の公開鍵を検出できるように、鍵リポジトリーには、送信側の証明書が入っている必要があります。 もう 1 つの可能性は、証明書チェーン内のすべての証明書を送信することです。
- アプリケーションがメッセージをキューに入れるときに、メッセージ記述子内の *UserIdentifier* フィールドには、そのアプリケーションに関連したユーザー ID が入ります。 このユーザー ID は、送信側を識別するのに使用できます。 認証を使用可能にするために、API 出口は、暗号化されたパスワードなどのデータを、メッセージ内のアプリケーション・データに付加することができます。 メッセージが受信側アプリケーションによって取り出されるときに、2 つ目の API 出口は、メッセージと一緒に移動したデータを使用して、ユーザー ID を認証できます。

  この手法は、制御されたトラステッド環境で発信されるメッセージ、および信頼される認証サーバーまたは PKI サポートが使用できない状況で発信されるメッセージには、十分であると考えられます。

## 特権ユーザー

特権ユーザーは、WebSphere MQ の全管理権限を付与されたユーザーです。

以下の表にリストされているユーザーに加えて、キューに対する +crt 権限を持つ任意のグループのメンバーは間接的な管理者です。 同様に、キュー・マネージャーに対する +set 権限、およびコマンド・キューに対する +put 権限を持つすべてのユーザーも管理者です。

これらの特権を通常のユーザーおよびアプリケーションに付与しないでください。

表 13. プラットフォーム別の特権ユーザー.

特権ユーザーの表。 Windows の場合、特権ユーザーは、SYSTEM、mqm グループのすべてのメンバー、および管理者 (Administrators) グループのすべてのメンバーです。 UNIX and Linux システムの場合、特権ユーザーは、mqm グループのすべてのメンバーです。 IBM i の場合、特権ユーザーは、プロファイル (ユーザー) qmqm および qmqmadm、qmqmadm グループのすべてのメンバー、および *ALLOBJ を設定して定義されているすべてのユーザーです。

| プラットフォーム | 特権ユーザー |
| --- | --- |
| Windows システム | • SYSTEM<br>• mqm グループのメンバー<br>• 管理者 (Administrators) グループのメンバー |
| UNIX and Linux システム | • mqm グループのメンバー |

## MQCSP 構造を使用したユーザーの識別および認証

MQCSP 接続セキュリティー・パラメーター構造体は MQCONNX 呼び出しで指定することができます。

MQCSP 接続セキュリティー・パラメーター構造体にはユーザー ID とパスワードが含まれており、それらは、許可サービスでユーザーの識別および認証に使用することができます。

IBM WebSphere MQ と共に提供される許可サービス・コンポーネントは、 オブジェクト権限マネージャー (OAM) と呼ばれます。 OAM は、MQCSP に含まれている ID に基づいてユーザーを許可しますが、パスワードは確認しません。 チェーニングされた出口を OAM と組み合わせて使用するか、または、OAM を別の許可サービスと置き換えることにより、許可サービスにパスワードの確認を実装することはできます。

MQCSP はセキュリティー出口で変更することができます。

## セキュリティー出口による識別と認証の実装

セキュリティー出口を使用して、片方向認証または相互認証を実装できます。

セキュリティー出口の主な目的は、チャネルの両端にある MCA が、相手側の MCA を認証できるようにすることです。 メッセージ・チャネルの両端、および MQI チャネルのサーバー側にある MCA は、通常、接続しているキュー・マネージャーの代理をします。 MQI チャネルのクライアント側にある MCA は、通常、WebSphere MQ MQI クライアント・アプリケーションのユーザーの代理をします。 この状況での相互認証は、実際には、2 つのキュー・マネージャー間で、またはキュー・マネージャーと、WebSphere MQ MQI クライアント・アプリケーションのユーザーとの間で行われます。

用意されているセキュリティー出口 (SSPI チャネル出口) は、認証トークンの交換によって相互認証を実装する方法を示した実例です。認証トークンは、信頼できる認証サーバー (Kerberos など) によって生成され、その後検査されることになります。 詳細については、103 ページの『SSPI チャネル出口プログラム』を参照してください。

また、相互認証は、公開鍵インフラストラクチャー (PKI) テクノロジーを使用することによってもインプリメントすることができます。 各セキュリティー出口は、ランダム・データを生成し、そのセキュリティー出口が代理をするキュー・マネージャーまたはユーザーの秘密鍵を使用して署名し、その署名されたデータをセキュリティー・メッセージ内で相手側に送信します。 相手側のセキュリティー出口は、そのキュー・マネージャーまたはユーザーの公開鍵を使用してデジタル署名を検査することによって、認証を実行します。 複数のアルゴリズムが使用可能である場合、デジタル署名を交換する前に、双方のセキュリティー出口が、メッセージ・ダイジェストの生成用のアルゴリズムを合意する必要があります。

セキュリティー出口が、署名されたデータを相手側に送信する場合、そのセキュリティー出口が代理をするキュー・マネージャーまたはユーザーを識別する手段も送信する必要があります。 これは、識別名、またはデジタル証明書にすることができます。 デジタル証明書が送信される場合、相手側のセキュリティー出口は、証明書チェーンをルート CA 証明書までたどることによって、その証明書を検証することができます。 これによって、デジタル署名の検査に使用される公開鍵の所有権が保証されます。

相手側のセキュリティー出口がデジタル証明書を検証できるのは、証明書チェーン内の残りの証明書が入っている鍵リポジトリーへのアクセス権がある場合だけです。 キュー・マネージャーまたはユーザー用のデジタル証明書が送信されない場合、相手側のセキュリティー出口がアクセス権を持つ鍵リポジトリーで、デジタル証明書が入手可能でなければなりません。 相手側のセキュリティー出口は、署名者の公開鍵が見つからない場合、デジタル署名を検査することはできません。

Secure Sockets Layer (SSL) と Transport Layer Security (TLS) では、ここで取り上げたような PKI 手法が採用されています。 Secure Sockets Layer がどのように認証を実行するかの詳細については、14 ページの『Secure Sockets Layer (SSL) および Transport Layer Security (TLS) の概念』を参照してください。

信頼される認証サーバーまたは PKI サポートが使用できない場合、別の手法を使用できます。 セキュリティー出口でインプリメントできる一般的な手法では、対称鍵アルゴリズムを使用します。

セキュリティー出口の 1 つで、出口 A は乱数を生成し、それをセキュリティー・メッセージの中でパートナー・セキュリティー出口である出口 B に送信します。 出口 B は、2 つのセキュリティー出口にしか認識されない鍵のコピーを使用して、番号を暗号化します。 出口 B は、暗号化された乱数を、出口 B が生成した 2 つ目の乱数とともに、セキュリティー・メッセージ内で出口 A に送信します。 出口 A は、最初の乱数が正しく暗号化されたかどうかを確認し、鍵のコピーを使用して 2 つ目の乱数を暗号化し、その暗号化された乱数を、セキュリティー・メッセージ内で出口 B に送信します。 次に、出口 B は、2 つ目の乱数が正しく暗号化されたかどうかを検証します。 このやりとりの間、どちらかのセキュリティー出口が、相手側の確実性を確信できない場合、そのセキュリティー出口は、チャネルをクローズするように MCA に指示できます。

この手法の利点は、このやりとりの間に通信接続を介して鍵もパスワードも送信されないことです。 不利な点は、共有鍵を安全な方法で配布する方法の問題に対する解決法を提供しないことです。 この問題の 1 つの解決法が、226 ページの『ユーザー出口プログラムでの機密性の実装』で説明されています。 SNA では、2 つの LU がバインドしてセッションを形成するときに、ほぼ同じ手法が、この 2 つの LU の相互認証に使用されます。 この手法は、74 ページの『セッション・レベルの認証』で説明されています。

ここでは、相互認証の手法を取り上げましたが、そのすべては、片方向認証に合わせて調整できます。

## メッセージ出口による識別マッピング

認証を実装するのは、アプリケーション・レベルのほうが望ましいですが、メッセージ出口を使用して、ユーザー ID を認証するための情報を処理することも可能です。

アプリケーションがメッセージをキューに入れるときに、メッセージ記述子内の *UserIdentifier* フィールドには、そのアプリケーションに関連したユーザー ID が入ります。 しかし、ユーザー ID の認証に使用できるデータはありません。 このデータは、チャネルの送信側のメッセージ出口によって追加し、そのチャネルの受信側のメッセージ出口によって検査することができます。 認証データは、例えば、暗号化されたパスワード、またはデジタル署名にすることができます。

このサービスは、アプリケーション・レベルでインプリメントされる場合の方が効果的です。 基本的な要件は、メッセージを受信するアプリケーションのユーザーが、メッセージを送信したアプリケーションのユーザーを識別し、認証できることです。 したがって、当然、このサービスをアプリケーション・レベルでインプリメントすることを検討すべきです。 詳細については、149 ページの『API 出口と API 交差出口による識別マッピング』を参照してください。

## API 出口と API 交差出口による識別マッピング

メッセージを受信するアプリケーションでは、そのメッセージを送信したアプリケーションのユーザーを識別して認証する機能が必要です。 通常は、そのサービスをアプリケーション・レベルで実装するのがベストです。 API 出口を使用すれば、いくつかの方法でそのサービスを実装できます。

個々のメッセージのレベルでは、識別と認証は、2 つのユーザー、つまりメッセージの送信側と受信側に関与するサービスです。 基本的な要件は、メッセージを受信するアプリケーションのユーザーが、メッセージを送信したアプリケーションのユーザーを識別し、認証できることです。 この要件は、両方向の認証ではなく、単方向の認証用であることに注意してください。

このサービスのインプリメントの方法に応じて、ユーザーとそのアプリケーションは、このサービスとのインターフェースを取るか、このサービスと相互作用する必要がある場合があります。 さらに、サービスの使用時期と使用方法は、ユーザーとそのアプリケーションが置かれる場所により、またアプリケーショ

ン自体の性質により異なります。 したがって、このサービスを、リンク・レベルではなく、アプリケーション・レベルでインプリメントすることを検討する方が妥当です。

このサービスをリンク・レベルでインプリメントすることを検討する場合は、次のような問題の解決が必要になる場合があります。

- メッセージ・チャネル上で、このサービスを必要とするメッセージに対してのみ、このサービスを適用するには、どうするか
- ユーザーとそのアプリケーションが、このサービスとのインターフェースを取るか、相互作用することができるようにする (それが要件である場合) には、どうするか
- メッセージがあて先までの途中にある複数のメッセージ・チャネルを介して送信される、マルチホップ状況では、このサービスのコンポーネントをどこで起動するか

ここでは、識別と認証のサービスをアプリケーション・レベルで実装する例をいくつか取り上げます。 ここで使用する *API* 出口 という語は、API 出口または API 交差出口のいずれかを指します。

- アプリケーションがキューにメッセージを書き込むときに、API 出口は、Kerberos などの信頼される認証サーバーから認証トークンを取得できます。 API 出口は、このトークンをメッセージ内のアプリケーション・データに追加することができます。 メッセージが受信側アプリケーションによって取り出されるときに、2 つ目の API 出口は、そのトークンを検査して送信側を認証するように、認証サーバーに依頼することができます。
- アプリケーションがキューにメッセージを入れるときに、API 出口は、メッセージ内のアプリケーション・データに次の項目を付加することができます。
  - 送信側のデジタル証明書
  - 送信側のデジタル署名

  メッセージ・ダイジェストの生成に複数のアルゴリズムが使用可能である場合、API 出口には、使用しているアルゴリズムの名前を組み込むことができます。

  メッセージが受信側アプリケーションによって取り出されるときに、2 つ目の API 出口は、次の検査を実行できます。

  - API 出口は、証明書チェーンをルート CA 証明書までたどることによって、デジタル証明書を検証できます。 これを行うために、API 出口には、証明書チェーン内の残りの証明書が入っている鍵リポジトリーへのアクセス権が必要です。 この検査により、識別名によって指定される送信側が、その証明書に入っているる公開鍵の本物の所有者であることが保証されます。
  - API 出口は、証明書に入っている公開鍵を使用して、デジタル署名を検査することができます。 この検査は、送信側を認証します。

  デジタル証明書全体を送信する代わりに、送信側の識別名を送信することができます。 この場合、2 番目の API 出口が送信側の公開鍵を検出できるように、鍵リポジトリーには、送信側の証明書が入っている必要があります。 もう 1 つの可能性は、証明書チェーン内のすべての証明書を送信することです。
- アプリケーションがメッセージをキューに入れるときに、メッセージ記述子内の *UserIdentifier* フィールドには、そのアプリケーションに関連したユーザー ID が入ります。 このユーザー ID は、送信側を識別するのに使用できます。 認証を使用可能にするために、API 出口は、暗号化されたパスワードなどのデータを、メッセージ内のアプリケーション・データに付加することができます。 メッセージが受信側アプリケーションによって取り出されるときに、2 つ目の API 出口は、メッセージと一緒に移動したデータを使用して、ユーザー ID を認証できます。

  この手法は、制御されたトラステッド環境で発信されるメッセージ、および信頼される認証サーバーまたは PKI サポートが使用できない状況で発信されるメッセージには、十分であると考えられます。

## 取り消された証明書の取り扱い

デジタル証明書は、認証局によって取り消されることがあります。 プラットフォームに応じて OCSP を使用するかまたは LDAP サーバーで CRL を使用することにより、証明書の失効状況を確認することができます。

SSL ハンドシェーク時に、通信するパートナーは、デジタル証明書を使用して互いに認証します。 認証には、受信された証明書が引き続き信頼できるかどうかの検査が組み込まれる場合があります。 認証局 (CA) は、次の理由を含めて、さまざまな理由で証明書を取り消します。

- 所有者が別の組織に移動した
- 秘密鍵が秘密でなくなった

CA は、証明書取り消しリスト (CRL) で、取り消された個人用証明書を公開します。 取り消された CA 証明書は、権限取り消しリスト (ARL) で公開されます。

UNIX、 Linux 、および Windows システムでは、 WebSphere MQ SSL サポートは、OCSP (Online Certificate Status Protocol) または LDAP (Lightweight Directory Access Protocol) サーバー上の CRL と ARL を使用して、失効した証明書を検査します。 OCSP が推奨される方法です。 IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS は、クライアント・チャネル定義テーブル・ファイル内の OCSP 情報を使用できません。 ただし、OCSP を構成することはできます (Online Certificate Protocol の使用セクションを参照)。

z/Os および IBM i WebSphere MQ SSL サポートは、LDAP サーバー上の CRL および ARL のみを使用して、取り消された証明書を検査します。

証明書について詳しくは、

権限については、 9 ページの『デジタル証明書』 を参照してください。

## 失効した証明書および OCSP

IBM WebSphere MQ は、どの Online Certificate Status Protocol (OCSP) 応答側を使用するのかを決定し、受信した応答を処理します。 OCSP 応答側をアクセス可能にするための手順を実行しなければならない場合があります。

**注 :** この情報は、 Windows、 UNIX and Linux システム上の WebSphere MQ にのみ適用されます。

WebSphere MQ は、OCSP を使用してデジタル証明書の失効状況を検査するときに、以下の 2 つのメソッドを使用してどの OCSP 応答側と連絡を取るのかを決定することができます。

- 検査対象の証明書内の AuthorityInfoAccess (AIA) 証明書拡張を使用する。
- 認証情報オブジェクトで指定されたか、またはクライアント・アプリケーションによって指定された URL を使用する。

認証情報オブジェクトに指定された URL、またはクライアント・アプリケーションによって指定された URL は、AIA 証明書拡張内の URL に優先します。

OCSP 応答側の URL との間にファイアウォールが存在する場合、ファイアウォールを再構成して、OCSP 応答側にアクセスできるようにするか、または OCSP プロキシー・サーバーをセットアップしてください。 SSL スタンザで SSLHTTPProxyName 変数を使用して、プロキシー・サーバーの名前を指定します。 クライアント・システム上では、環境変数 MQSSLPROXY を使用することによっても、プロキシー・サーバー名を指定できます。 詳細については、関連情報を参照してください。

テスト環境で実行しているなどの理由で、TLS または SSL 証明書が失効してもかまわない場合には、SSL スタンザの OCSPCheckExtensions を NO に設定できます。 この変数を設定すると、AIA 証明書拡張が無視されます。 この解決方法は、実稼働環境では、ほとんどの場合に不適切です。実稼働環境では、失効した証明書を提示するユーザーからのアクセスは許可できないからです。

OCSP 応答側にアクセスするために呼び出しを行うと、次の 3 つのいずれかの結果になります。

**良好**
証明書は有効です。

**失効**
証明書は取り消されています。

**不明**
この結果になるのは、次の 3 つのうちのいずれかが原因です。

- IBM WebSphere MQ が OCSP 応答側にアクセスできない。

- OCSP 応答側が応答を送信したが、WebSphere MQ が応答のデジタル署名を検証できない。
- OCSP 応答側が、その証明書に関する取り消しデータを保持していないことを示す応答を送信した。

「不明」という OCSP 結果を受信した場合の IBM WebSphere MQ の動作は、OCSPAuthentication 属性の設定値によって決まります。 キュー・マネージャーの場合、この属性は UNIX and Linux システムの qm.ini ファイルの SSL スタンザ、または Windows レジストリーに保持されます。 その値は、IBM WebSphere MQ エクスプローラーで設定できます。 クライアントでは、これはクライアント構成ファイルの SSL スタンザに保持されます。

OCSPAuthentication が REQUIRED (デフォルト値) に設定されている場合に「不明」という結果を受信すると、WebSphere MQ は接続を拒否し、タイプ AMQ9716 のエラー・メッセージを発行します。 キュー・マネージャーの SSL イベント・メッセージが有効な場合、ReasonQualifier が MQRQ_SSL_HANDSHAKE_ERROR に設定された、タイプ MQRC_CHANNEL_SSL_ERROR の SSL イベント・メッセージが生成されます。

OCSPAuthentication が OPTIONAL に設定されている場合に「不明」という結果を受信すると、WebSphere MQ はその SSL チャネルの開始を許可し、警告や SSL イベント・メッセージは生成されません。

OCSPAuthentication が WARN に設定されている場合に「不明」という結果を受信すると、SSL チャネルは開始されますが、IBM WebSphere MQ はタイプ AMQ9717 の警告メッセージをエラー・ログに出力します。 キュー・マネージャーの SSL イベント・メッセージが有効になっている場合、タイプが MQRC_CHANNEL_SSL_WARNING で ReasonQualifier が MQRQ_SSL_UNKNOWN_REVOCATION に設定された SSL イベント・メッセージが生成されます。

## OCSP 応答のデジタル署名

OCSP 応答側は、3 つの方法のいずれかでその応答に署名します。 応答側からは、使用する方法が通知されます。

- OCSP 応答に、検査中の証明書を発行した同一の CA 証明書を使用してデジタル署名を付加できます。 この場合、追加の証明書をセットアップする必要はありません。SSL 接続を確立するために既に実行したステップで OCSP 応答を検証できます。
- OCSP 応答に、検査中の証明書を発行した同一の認証局 (CA) によって署名された別の証明書を使用して、デジタル署名を付加できます。 この場合、OCSP 応答と一緒に署名証明書が送信されます。 OCSP 応答側から流れてくる証明書では、拡張キー使用法という拡張機能が id-kp-OCSPSigning に設定されていなければなりません。その設定があれば、その証明書は応答の証明書として信頼できるということになります。 OCSP 応答は、署名された証明書と一緒に送信される (さらに、SSL 接続のために既に信頼されている CA によって証明書が署名されている) ため、追加の証明書のセットアップは必要ありません。
- OCSP 応答に、検査中の証明書に直接関係のない別の証明書を使用して、デジタル署名を付加できます。この場合、OCSP 応答は OCSP 応答側自体によって発行された証明書によって署名されます。 OCSP 応答側の証明書コピーを、OCSP 検査を実行しているクライアントまたはキュー・マネージャーの鍵データベースに追加する必要があります。132 ページの『UNIX, Linux, and Windows システムで CA 証明書 (または自己署名証明書の公開部分) を鍵リポジトリーに追加する操作』を参照してください。 CA 証明書が追加される場合、デフォルトで、このコンテキストで必要な設定であるトラステッド・ルートとして追加されます。 この証明書が追加されない場合、WebSphere MQ は OCSP 応答のデジタル署名を検証できず、OCSP 検査の結果が「不明」になります。この際、OCSPAuthentication の値に応じて IBM WebSphere MQ がチャネルを閉じる可能性があります。

## Java および JMS クライアント・アプリケーションでの Online Certificate Status Protocol (OCSP)

Java API の制約により、WebSphere MQ が SSL および TLS セキュア・ソケットの Online Certificate Status Protocol (OCSP) 証明書失効検査を使用できるのは、OCSP が Java 仮想マシン (JVM) プロセス全体に対して有効になっている場合のみです。 JVM のすべてのセキュア・ソケットに対して OCSP を有効にするには、以下の 2 つの方法があります。

- 表 1 に示す OCSP 構成設定を含めるように JRE java.security ファイルを編集し、アプリケーションを再起動します。

- 適用されるすべての Java セキュリティー・マネージャー・ポリシーに従って、java.security.Security.setProperty() API を使用します。

少なくとも、ocsp.enable 値と ocsp.responderURL 値のいずれかを指定する必要があります。

| プロパティー名 | 説明 |
| --- | --- |
| ocsp.enable | このプロパティーの値は `true` または `false` です。 `true` の場合、証明書失効検査の実行時に OCSP 検査が有効になります。`false` の場合、または設定しない場合は OCSP 検査が無効になります。 |
| ocsp.responderURL | このプロパティーの値は、OCSP 応答側の場所を示す URL です。 例えば、`ocsp.responderURL=http://ocsp.example.net:80` です。デフォルトでは、OCSP 応答側の場所は、検証される証明書により暗黙的に決定されます。 このプロパティーは、Authority Information Access 拡張 (RFC 3280 で定義) が証明書にない場合、またはその指定変更が必要な場合に使用されます。 |
| ocsp.responderCertSubjectName | このプロパティーの値は、OCSP 応答側の証明書のサブジェクト名です。 例えば、`ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"` です。 デフォルトでは、OCSP 応答側の証明書は、検証される証明書の発行者の証明書です。 このプロパティーは、デフォルトが適用されない場合に OCSP 応答側の証明書を特定します。 その値はストリングでの識別名 (RFC 2253 で定義) で、証明書パスの検証時に提供される一連の証明書の中から 1 つの証明書を特定します。 サブジェクト名のみでは証明書を一意的に特定できない場合は、代わりに ocsp.responderCertIssuerName プロパティーおよび ocsp.responderCertSerialNumber プロパティーの両方を使用する必要があります。 このプロパティーが設定されると、ocsp.responderCertIssuerName プロパティーおよび ocsp.responderCertSerialNumber プロパティーは無視されます。 |
| ocsp.responderCertIssuerName | このプロパティーの値は、OCSP 応答側の証明書の発行者名です。 例えば、`ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"` です。 デフォルトでは、OCSP 応答側の証明書は、検証される証明書の発行者の証明書です。 このプロパティーは、デフォルトが適用されない場合に OCSP 応答側の証明書を特定します。 その値はストリングでの識別名 (RFC 2253 で定義) で、証明書パスの検証時に提供される一連の証明書の中から 1 つの証明書を特定します。 このプロパティーを設定する場合は、ocsp.responderCertSerialNumber プロパティーも設定する必要があります。 ocsp.responderCertSubjectName プロパティーが設定されている場合、このプロパティーは無視されます。 |
| ocsp.responderCertSerialNumber | このプロパティーの値は、OCSP 応答側の証明書のシリアル番号です。例えば、`ocsp.responderCertSerialNumber=2A:FF:00` です。デフォルトでは、OCSP 応答側の証明書は、検証される証明書の発行者の証明書です。 このプロパティーは、デフォルトが適用されない場合に OCSP 応答側の証明書を特定します。 この値は 16 進数字のストリング (分離文字のコロンまたはスペースが存在することがあります) であり、証明書パスの検証時に提供される一連の証明書の中から、1 つの証明書を特定します。 このプロパティーを設定する場合は、ocsp.responderCertIssuerName プロパティーも設定する必要があります。 ocsp.responderCertSubjectName プロパティーが設定されている場合、このプロパティーは無視されます。 |

この方法で OCSP を有効にする前に、以下に示す、いくつかの点を考慮してください。

- OCSP 構成を設定すると、JVM プロセス内のすべてのセキュア・ソケットに影響します。 場合によっては、SSL セキュア・ソケットまたは TLS セキュア・ソケットを使用する他のアプリケーション・コードと

JVM が共用されるときに好ましくない副次作用が発生することがあります。 選択した OCSP 構成が、同じ JVM 内で実行されているすべてのアプリケーションに適したものであることを確認してください。

- JRE にメンテナンスを適用すると、java.security ファイルが上書きされる場合があります。 Java インテリム・フィックスおよび製品メンテナンスを適用する際には、java.security ファイルを上書きしないよう注意してください。 場合によっては、メンテナンスの適用後に java.security の変更を再適用する必要があります。 このため、代わりに java.security.Security.setProperty() API を使用して OCSP 構成を設定することも検討してください。
- OCSP 検査を有効にすることが意味を持つのは、失効検査も有効になっている場合のみです。 失効検査は `PKIXParameters.setRevocationEnabled()` メソッドにより有効にします。
- AMS Java インターセプター (ネイティブ・インターセプターでの OCSP 検査の有効化を参照) を使用している場合は、鍵ストア構成ファイルの AMS OCSP 構成と競合する java.security OCSP 構成は使用しないよう注意してください。

## 証明書取り消しリストおよび権限取り消しリストの取り扱い

WebSphere MQ による CRL および ARL のサポートは、プラットフォームによって異なります。

各プラットフォームでの CRL および ARL サポートは、次のとおりです。

- z/OS では、System SSL は、Tivoli® の公開鍵インフラストラクチャー製品によって LDAP サーバーに保管される CRL および ARL をサポートします。
- その他のプラットフォームでは、CRL および ARL サポートは、PKIX X.509 V2 CRL プロファイルの推奨事項に従います。

WebSphere MQ は、直近の 12 時間のあいだにアクセスされた CRL と ARL のキャッシュを管理します。

キュー・マネージャーまたは WebSphere MQ MQI クライアントは証明書を受け取ると、CRL を調べてその証明書が有効であることを確認します。 WebSphere MQ はまず、キャッシュ内を調べます。 CRL がキャッシュ内にない場合、WebSphere MQ は *SSLCRLNamelist* 属性によって指定される認証情報オブジェクトの名前リスト内に現れる順に、WebSphere MQ が使用可能な CRL を検出するまで LDAP CRL サーバーのロケーションを問い合わせます。 名前リストが指定されていない場合、またはブランク値が指定されている場合、CRL は検査されません。

LDAP について詳しくは、WebSphere MQ for Windows での Lightweight Directory Access Protocol (LDAP) サービスの使用を参照してください。

### *LDAP* サーバーのセットアップ

CA の識別名の階層に合わせて LDAP Directory Information Tree 構造を構成します。 そのためには、LDAP Data Interchange Format ファイルを使用します。

証明書と CRL を発行する CA の識別名に対応する階層を使用するように、LDAP Directory Information Tree (DIT) 構造を構成します。 LDAP Data Interchange Format (LDIF) を使用するファイルを使用して、DIT 構造をセットアップできます。 また、LDIF ファイルを使用してディレクトリーを更新することもできます。

LDIF ファイルは、LDAP ディレクトリー内のオブジェクトを定義するのに必要な情報が入っている、ASCII テキスト・ファイルです。 LDIF ファイルには、1 つ以上の項目が入っています。この項目はそれぞれ、識別名、1 つ以上のオブジェクト・クラス定義、およびオプショナルでの複数の属性定義から構成されます。

`certificateRevocationList;binary` 属性には、取り消されたユーザー証明書のリストが、バイナリー形式で含まれています。 `authorityRevocationList;binary` 属性には、取り消された CA 証明書のバイナリー・リストが入っています。 WebSphere MQ SSL とともに使用する場合、これらの属性のバイナリー・データ は DER (Definite Encoding Rules) 形式に準拠している必要があります。 LDIF ファイルの詳細については、ご使用の LDAP サーバーに付属の資料を参照してください。

155 ページの図 12 は、 IBM 内のテスト組織によってセットアップされた、識別名 "CN=CA1, OU = テスト、O=IBM、C=GB"を持つ架空の認証局である CA1 によって発行された CRL および ARL をロードするために、LDAP サーバーへの入力として作成できるサンプル LDIF ファイルを示しています。

```
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)
```

図 *12.* 認証局のサンプル *LDIF* ファイル。 これは、インプリメンテーションによって異なる可能性があります。

155 ページの図 13 は、155 ページの図 12 に示されているサンプル LDIF ファイルをロードする際に、LDAP サーバーが作成する DIT 構造を示しています。また、IBM 社内の PKI 組織によってセットアップされた仮想認証局である CA2 用の同種ファイルも示しています。

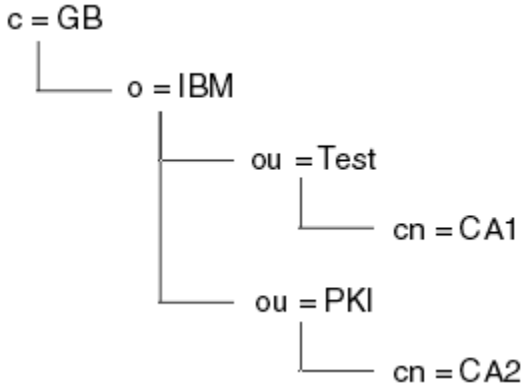



図 *13. LDAP Directory Information Tree* 構造例

WebSphere MQ は、CRL と ARL の両方を検査します。

**注 :** ご使用の LDAP サーバーのアクセス制御リストにより、許可ユーザーが、CRL と ARL を保持する項目の読み取り、検索、および比較を行うことができることを確認してください。 WebSphere MQ は、AUTHINFO オブジェクトの LDAPUSER プロパティーと LDAPPWD プロパティーを使用して LDAP サーバーにアクセスします。

*LDAP* サーバーの構成と更新

LDAP サーバーの構成または更新の手順を取り上げます。

1. 認証局 (複数の場合あり) から、DER 形式の CRL および ARL を取得する。
2. テキスト・エディター、または LDAP サーバーに付属のツールを使用して、CA の識別名、および必要なオブジェクト・クラス定義が入っている、1 つ以上の LDIF ファイルを作成する。 DER 形式のデータは、`certificateRevocationList;binary` 属性 (CRL の場合)、`authorityRevocationList;binary` 属性 (ARL の場合)、またはその両方の値として、LDIF ファイルにコピーしてください。
3. LDAP サーバーを開始する。
4. ステップ 155 ページの『2』 で作成した LDIF ファイル (複数の場合あり) から、項目を追加する。

LDAP CRL サーバーの構成後、セットアップが正しく行われたかどうかを検査します。 まず、チャネルで取り消されていない証明書を使用し、チャネルが正しく開始されることを確認します。 次に、取り消された証明書を使用し、チャネルが開始されないことを確認します。

更新された CRL を認証局から頻繁に取得してください。 LDAP サーバーで 12 時間ごとに、更新を取得することを検討してください。

## キュー・マネージャーを使用した *CRL* および *ARL* へのアクセス

キュー・マネージャーには、1 つ以上の認証情報オブジェクトを関連付けます。認証情報オブジェクトには、LDAP CRL サーバーのアドレスを格納します。

この節で示す証明書取り消しリスト (CRL) に関する情報は、権限取り消しリスト (ARL) にも当てはまります。

それぞれが LDAP CRL サーバーのアドレスを保持する認証情報オブジェクトを、キュー・マネージャーに提供することによって、CRL へのアクセス方法をキュー・マネージャーに指示します。 この認証情報オブジェクトは、*SSLCRLNamelist* キュー・マネージャー属性で指定された 名前リストに保持されます。

次の例では、MQSC を使用してパラメーターを指定します。

1. AUTHTYPE パラメーターを CRLLDAP に設定した、DEFINE AUTHINFO MQSC コマンドを使用して、認証情報オブジェクトを定義する。

   AUTHTYPE パラメーターで CRLLDAP 値を指定すると、LDAP サーバーの CRL に対するアクセスが行われるようになります。 作成したタイプ CRLLDAP の各認証情報オブジェクトは、LDAP サーバーのアドレスを保持します。 複数の認証情報オブジェクトがある場合、それらのオブジェクトが指す LDAP サーバーには、同一の情報が入っていなければなりません。 これにより、1 つ以上の LDAP サーバーに障害が起きても、サービスの継続性が確保されます。

   どのプラットフォームでも、ユーザー ID とパスワードは、暗号化されないで LDAP サーバーに送信されます。
2. DEFINE NAMELIST MQSC コマンドを使用して、認証情報オブジェクトの名前用の名前リストを定義する。
3. ALTER QMGR MQSC コマンドを使用して、キュー・マネージャーにその名前リストを提供する。 以下に例を示します。

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   ここで、`sslcrlnlname` は、認証情報オブジェクトの名前リストです。

   このコマンドは、*SSLCRLNamelist* と呼ばれる、キュー・マネージャーの属性を設定します。 この属性のキュー・マネージャーの初期値はブランクです。

1 つ以上の LDAP サーバーが失敗した場合のサービスの継続性を確保するために、代替 LDAP サーバーとの最高 10 個の接続を、名前リストに追加することができます。 LDAP サーバーには同一の情報が入っていなければなりません。

### *IBM WebSphere MQ Explorer* を使用した *CRL* および *ARL* へのアクセス

IBM WebSphere MQ Explorer を使用して、CRL へのアクセス方法をキュー・マネージャーに指示することができます。

この節で示す証明書取り消しリスト (CRL) に関する情報は、権限取り消しリスト (ARL) にも当てはまります。

CRL との LDAP 接続をセットアップする手順は、次のとおりです。

1. キュー・マネージャーを開始したことを確認する。
2. 「**認証情報**」フォルダーを右クリックして、「**新規**」**->**「**認証情報**」**の順にクリックする**。 開いたプロパティー・シートで、次の手順を実行してください。
   a. 最初のページの「**認証情報の作成**」で、CRL(LDAP) オブジェクトの名前を入力する。
   b. 「**プロパティーの変更**」の「**一般**」ページで、接続タイプを選択する。 オプションで、説明を入力できます。
   c. 「**プロパティーの変更**」の「**CRL(LDAP)**」ページを選択する。
   d. LDAP サーバーの名前を、ネットワーク名か IP アドレスのどちらかとして入力する。

e. サーバーがログインの詳細を要求する場合、ユーザー ID、および必要に応じてパスワードを入力する。

f.「**OK**」をクリックします。

3.「**名前リスト**」 フォルダーを右クリックし、「**新規**」**->**「**名前リスト**」 をクリックします。 開いたプロパティー・シートで、次の手順を実行してください。

a. 名前リストの名前を入力する。

b. CRL(LDAP) オブジェクトの名前 (ステップ 156 ページの『2.a』 から) を、リストに追加する。

c.「**OK**」をクリックします。

4. キュー・マネージャーを右クリックし、「**Properties (プロパティー)**」を選択し、「**SSL**」ページを選択する。

a.「**Check certificates received by this queue manager against Certification Revocation Lists (このキュー・マネージャーが受け取った証明書と、証明書取り消しリストを照合する)**」チェック・ボックスを選択する。

b. 名前リストの名前 (ステップ 157 ページの『3.a』 から) を、「**CRL Namelist (CRL 名前リスト)**」フィールドに入力する。

### *IBM WebSphere MQ MQI* クライアントを使用した *CRL* および *ARL* へのアクセス

IBM WebSphere MQ MQI クライアントによる検査のために CRL を保持する LDAP サーバーを指定するには、3 つのオプションがあります。

この節で示す証明書取り消しリスト (CRL) に関する情報は、権限取り消しリスト (ARL) にも当てはまります。

LDAP サーバーを指定する 3 つの方法は、以下のとおりです。

- チャネル定義テーブルを使用する
- MQCONNX 呼び出しで SSL 構成オプション構造体 MQSCO を使用する
- Active Directory を使用する (Active Directory サポートを備えた Windows システム上)

詳細については、関連情報を参照してください。

1 つ以上の LDAP サーバーが失敗した場合のサービスの継続性を確保するために、代替 LDAP サーバーとの最高 10 個の接続を含めることができます。 LDAP サーバーには同一の情報が入っていなければなりません。

Linux ( zSeries プラットフォーム) 上で実行されている WebSphere MQ MQI クライアント・チャネルから LDAP CRL にアクセスすることはできません。

#### *OCSP* レスポンダーの位置および *CRL* を保持する *LDAP* サーバーの位置

IBM WebSphere MQ MQI クライアント・システムでは、OCSP レスポンダーの場所、および証明書取り消しリスト (CRL) を保持する Lightweight Directory Access Protocol (LDAP) サーバーの場所を指定できます。

これらの位置は、優先順位の高い順に示された以下の 3 つの方法で指定できます。

## WebSphere MQ MQI クライアント・アプリケーションが MQCONNX 呼び出しを発行する場合

OCSP レスポンダーまたは **MQCONNX** 呼び出しで CRL を保持する LDAP サーバーを指定できます。

**MQCONNX** 呼び出しでは、接続オプション構造体 MQCNO が、SSL 構成オプション構造体 MQSCO を参照できます。 次に、MQSCO 構造体が、1 つ以上の認証情報レコード構造体 MQAIR を参照します。 各 MQAIR 構造体には、WebSphere MQ MQI クライアントが、OCSP レスポンダーまたは CRL を保持する LDAP サーバーにアクセスするために必要な情報がすべて含まれています。 例えば、MQAIR 構造体内のフィールドの 1 つは、レスポンダーに接続可能な URL です。 MQAIR 構造体について詳しくは、MQAIR - 認証情報レコードを参照してください。

## クライアント・チャネル定義テーブル (CCDT) を使用した OCSP レスポンダーまたは LDAP サーバーへのアクセス

WebSphere MQ MQI クライアントが、OCSP レスポンダーまたは CRL を保持する LDAP サーバーにアクセスできるように、1 つ以上の認証情報オブジェクトの属性をクライアント・チャネル定義テーブルに組み込みます。

サーバー・キュー・マネージャーでは、1 つ以上の認証情報オブジェクトを定義できます。 認証オブジェクトの属性には、(OCSP がサポートされているプラットフォーム上の) OCSP レスポンダー、または CRL を保持する LDAP サーバーにアクセスするために必要な情報がすべて含まれています。 属性の 1 つで OCSP レスポンダーの URL を指定し、別の属性では LDAP サーバーが稼働しているシステムのホスト・アドレスまたは IP アドレスを指定します。

AUTHTYPE(OCSP) の認証情報オブジェクトは IBM i または z/OS キュー・マネージャーでの使用には適用されませんが、クライアントでの使用のためにクライアント・チャネル定義テーブル (CCDT) にコピーされるように、これらのプラットフォーム上で指定することはできます。

WebSphere MQ MQI クライアントが、OCSP レスポンダーまたは CRL を保持する LDAP サーバーにアクセスできるように、1 つ以上の認証情報オブジェクトの属性をクライアント・チャネル定義テーブルに組み込むことができます。 以下のいずれかの方法で、それらの属性を組み込むことができます。

**サーバー・プラットフォームの場合 AIX、 HP-UX、 Linux、 Solaris、および Windows**

1 つ以上の認証情報オブジェクトの名前を含む名前リストを定義します。 それから、キュー・マネージャー属性 **SSLCRLNameList** を名前リストの名前に設定します。

CRL を使用する場合は、さらに高可用性が得られるように複数の LDAP サーバーを構成できます。 大切なのは、各 LDAP サーバーが同じ CRL を保持することです。 ある LDAP サーバーが必要なときに使用できない場合、WebSphere MQ MQI クライアントは別の LDAP サーバーにアクセスします。

ここでは、名前リストで示された認証情報オブジェクトの属性を、まとめて証明書取り消し場所 と呼びます。 キュー・マネージャー属性 **SSLCRLNameList** を名前リストの名前に設定すると、キュー・マネージャーに関連するクライアント・チャネル定義テーブルに証明書取り消し場所がコピーされます。 クライアント・システムから共有ファイルとして CCDT にアクセスできる場合、または CCDT がクライアント・システムにコピーされている場合は、そのシステム上の WebSphere MQ MQI クライアントは CCDT の証明書取り消し場所を使用して、OCSP レスポンダーまたは CRL を保持する LDAP サーバーにアクセスすることができます。

キュー・マネージャーの証明書取り消し場所が後で変更された場合、その変更内容は、キュー・マネージャーに関連する CCDT に反映されます。 キュー・マネージャー属性 **SSLCRLNameList** をブランクに設定すると、証明書取り消し場所が CCDT から削除されます。 これらの変更は、クライアント・システム上のテーブルのコピーには反映されません。

MQI チャネルのクライアント側とサーバー側の証明書失効ロケーションが異なっている必要があり、サーバー・キュー・マネージャーが証明書失効ロケーションの作成に使用されるサーバー・キュー・マネージャーである場合は、以下のようにして行うことができます。

1. サーバー・キュー・マネージャーで、クライアント・システムで使用する証明書取り消し場所を作成します。
2. 証明書取り消し場所を含む CCDT をクライアント・システムにコピーします。
3. サーバー・キュー・マネージャーで、MQI チャネルのサーバー側での必要に応じて証明書取り消し場所を変更します。

## Windows での Active Directory の使用

Windows システムで **setmqcrl** 制御コマンドを使用して、Active Directory で現在の CRL 情報を公開することができます。

コマンド **setmqcrl** では OCSP 情報は公開されません。

このコマンドおよびその構文については、setmqcrl を参照してください。

### *IBM WebSphere MQ classes for Java* および *IBM WebSphere MQ classes for JMS* を使用した *CRL* および *ARL* へのアクセス

IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS は、他のプラットフォームとは異なる方法で CRL にアクセスします。

IBM WebSphere MQ classes for Java で CRL と ARL を処理する方法については、 証明書失効リストの使用を参照してください。

IBM WebSphere MQ classes for JMS で CRL と ARL を操作する方法については、 SSLCERTSTORES オブジェクト・プロパティーを参照してください。

## 認証情報オブジェクトの取り扱い

認証情報オブジェクトは、MQSC または PCF コマンド、あるいは IBM WebSphere MQ Explorer を使用して取り扱うことができます。

以下の MQSC コマンドは、認証情報オブジェクトに対して機能します。

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

これらのコマンドの詳細な説明については、「スクリプト (MQSC) コマンド」を参照してください。

以下のプログラマブル・コマンド・フォーマット (PCF)・コマンドは、認証情報オブジェクトに対して機能します。

- Create Authentication Information
- Copy Authentication Information
- Change Authentication Information
- Delete Authentication Information
- Inquire Authentication Information
- Inquire Authentication Information Names

これらのコマンドの詳細については、「プログラマブル・コマンド・フォーマットの定義」を参照してください。

WebSphere MQ エクスプローラーが使用可能なプラットフォームでは、WebSphere MQ エクスプローラーを使用することもできます。

# オブジェクトに対するアクセス権限の設定

このセクションには、オブジェクト権限マネージャーおよびチャネル出口プログラムを使用してオブジェクトへのアクセスを制御する方法について情報が記載されています。

UNIX, Linux, and Windows システムの場合。 オブジェクト権限マネージャー (OAM) を使用して、オブジェクトへのアクセスを制御します。 この一連のトピックには、OAM に対するコマンド・インターフェースの使用法についての情報が含まれます。 これには、セキュリティーをシステムに適用するために実行するタスクを判別するのに使用できるチェックリストや、IBM WebSphere MQ を管理したり、IBM WebSphere MQ オブジェクトを処理したりする権限をユーザーに付与する場合の考慮事項も含まれます。 提供されるセキュリティー・メカニズムが必要を満たさない場合は、独自のチャネル出口プログラムを開発することができます。

## UNIX、Linux および Windows システムでの OAM によるオブジェクトへのアクセス制御

オブジェクト権限マネージャー (OAM) には、WebSphere MQ オブジェクトに対する権限を与えたり取り消したりするためのコマンド・インターフェースが用意されています。

197 ページの『UNIX, Linux, and Windows システム上の IBM WebSphere MQ を管理する権限』で説明されているように、これらのコマンドの使用を適切に許可されていなければなりません。 WebSphere MQ の管理を許可されたユーザー ID には、キュー・マネージャーに対するスーパーユーザー 権限があります。それで、MQI 要求またはコマンドを発行するためのこれ以上の許可を付与する必要はないことになります。

## UNIX, Linux, and Windows システムでの IBM WebSphere MQ オブジェクトへのアクセス権限の付与

**setmqaut** 制御コマンド、または **MQCMD_SET_AUTH_REC** PCF コマンドを使用して、 IBM WebSphere MQ オブジェクトへのアクセス権限をユーザーおよびユーザー・グループに付与します。

**setmqaut** 制御コマンドとその構文の詳細な定義については、『setmqaut』を参照してください。 **MQCMD_SET_AUTH_REC** PCF コマンドとその構文の詳細な定義については、『権限レコードの設定』を参照してください。

このコマンドを使用するために、キュー・マネージャーを実行する必要があります。 あるプリンシパルのアクセス権を変更した場合、OAM はその変更をすぐに反映します。

オブジェクトへのアクセス権をユーザーに付与するには、以下のことを指定する必要があります。

- 処理する予定のオブジェクトを所有するキュー・マネージャーの名前。キュー・マネージャーの名前を指定しないと、デフォルト・キュー・マネージャーが使用されます。
- オブジェクトの名前とタイプ (オブジェクトを固有に識別するため)。 名前は プロファイル として指定します。これは、オブジェクトの明示的な名前か汎用名のいずれかになり、ワイルドカード文字を含められます。 汎用プロファイルの詳細や、その中でのワイルドカード文字の使用方法は、161 ページの『UNIX, Linux, and Windows システムでの OAM 汎用プロファイルの使用』を参照してください。
- 権限を適用する 1 つ以上のプリンシパルおよびグループ名。

  ユーザー ID にスペースが含まれている場合は、このコマンドを使用するときにユーザー ID を引用符で囲みます。 Windows システムでは、ユーザー ID をドメイン・ネームで修飾できます。 実際のユーザー ID にアットマーク (@) 記号が含まれている場合は、その記号がユーザー ID とドメイン・ネームの間の区切り文字ではなくユーザー ID の一部であることを示すために @@ に置き換えてください。
- 許可のリスト。 リストの各項目では、そのオブジェクトに付与する (またはオブジェクトから取り消す) 予定のアクセス権のタイプを指定します。 リスト内の各許可はキーワードとして指定され、接頭部にプラス記号 (+) または負符号 (-)が付きます。正符号を使用して指定された許可を追加し、負符号 (-) を使用して許可を除去します。 「+」または「-」符号とキーワードの間にはスペースを入れません。

  単一のコマンドで、許可をいくつでも指定できます。 例えば、ユーザーやグループがキューにメッセージを書き込むこととそれらをブラウズすることを許可するが、メッセージを入手するアクセス権を取り消す許可のリストは、次のとおりです。

```
    +browse -get +put
```

## setmqaut コマンドの使用例

以下の例では、setmqaut コマンドを使用してあるオブジェクトを使用する許可を付与し取り消す方法が示されています。

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

この例のそれぞれの指定の意味は次のとおりです。

- saturn.queue.manager は、キュー・マネージャー名です。
- queue は、オブジェクト・タイプです。
- RED.LOCAL.QUEUE は、オブジェクト名です。
- groupa は、変更する必要のある許可を持つグループの ID です。
- +browse -get +put は指定したキューに関する許可リストです。

- +browse は、キュー上のメッセージをブラウズ (ブラウズ・オプション付き **MQGET** を発行) する許可を追加します。
- -get は、キューからメッセージを読み取る (**MQGET**) 許可を取り消します。
- +put は、キューにメッセージを書き込む (**MQPUT**) 許可を追加します。

次のコマンドはキュー MyQueue に関する書き込み権限をプリンシパル fvuser と、グループ groupa および groupb から取り消します。 UNIX and Linux システムでは、このコマンドは、fvuser と同じ 1 次グループのすべてのプリンシパルの書き込み権限を取り消します。

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

## 別の許可サービスでのコマンドの使用

OAM の代わりに独自の許可サービスを使用している場合、**setmqaut** コマンドにこのサービスの名前を指定し、コマンドをこのサービスに送信することができます。 同時に実行される複数のインストール可能なコンポーネントがある場合は、このパラメーターを指定する必要があります。指定しない場合、許可サービスのために、最初のインストール可能なコンポーネントが更新されます。 デフォルトでは、これはシステムに提供された OAM です。

# UNIX, Linux, and Windows システムでの OAM 汎用プロファイルの使用

OAM 汎用プロファイルを使用すると、作成時に個々のオブジェクトに対して別々の **setmqaut** コマンドを発行するのではなく、一度に多数のオブジェクトに対してユーザーが持つ権限を設定することができます。

**setmqaut** コマンドの中で汎用プロファイルを使用すると、そのプロファイルに適した、すべてのオブジェクトに汎用権限を設定できるようになります。

このトピック集では、汎用プロファイルの使用方法をさらに詳しく説明します。

## OAM プロファイルでのワイルドカード文字の使用

プロファイルが総称である理由は、プロファイル名において特殊文字 (ワイルドカード文字) が使用できるためです。 例えば、疑問符 (?) というワイルドカード文字は、名前に含まれる任意の 1 文字に一致します。そのため、ABC.?EF を指定すると、そのプロファイルに付与する許可は、ABC.DEF、ABC.CEF、ABC.BEF などの名前を持つすべてのオブジェクトに適用されます。

使用できるワイルドカード文字は次のとおりです。

**?**

任意の 1 文字の代わりに疑問符 (?) を使用します。 例えば、AB.?D は、AB.CD、AB.ED、AB.FD の各オブジェクトに適用されます。

*****

アスタリスク (*) は、次のように使用します。

- プロファイル名に含まれる修飾子 に使用して、オブジェクト名に含まれる任意の修飾子 1 つに一致します。 修飾子は、ピリオドで区切られた、オブジェクト名の部分です。 例えば、ABC.DEF.GHI では、修飾子は ABC、DEF、および GHI です。

  例えば、 ABC.*.JKL はオブジェクト ABC.DEF.JKL、および ABC.GHI.JKL に適用されます。 (ただし、このコンテキストで * を使用する場合は、常に 1 つの修飾子を指すので、ABC.JKL には適用され**ません**。)
- プロファイル名に含まれる修飾子の文字 1 つは、オブジェクト名に含まれる 0 個以上の文字に一致します。

  例えば、 ABC.DE*.JKL はオブジェクト ABC.DE.JKL、 ABC.DEF.JKL、および ABC.DEGH.JKL に適用されます。

******

二重アスタリスク (**) は、次のようにして、プロファイル名の中で、**1 回のみ**使用します。

- プロファイル名全体をすべてのオブジェクト名と一致させます。 例えば、プロセスを識別するために -t prcs を使用する場合、プロファイル名として ** を使用し、すべてのプロセスの許可を変更します。
- プロファイル名の先頭、中ほど、最後の修飾子のいずれかが、オブジェクト名に含まれる 0 個以上の文字に一致します。 例えば、 **.ABC は、最終修飾子 ABC を持つすべてのオブジェクトを識別します。

**注 :** UNIX and Linux システムでワイルドカード文字を使用しているとき、プロファイル名を単一引用符で囲む**必要があります**。

## プロファイルの優先順位

汎用プロファイルの使用を理解する上で重要な点は、作成するオブジェクトに適用する権限を決定するときにプロファイルに与えられる優先順位です。 例えば、次のコマンドを発行するとします。

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

1 番目は、プロファイル AB. *; と一致する名前を持つプリンシパルのすべてのキューに対する書き込み権限を付与します。 2 番目のコマンドは、プロファイル AB.C*。

AB.CD と呼ばれるキューを作成するとします。 ワイルドカード突き合わせのルールによると、いずれかの setmqaut がそのキューに適用されます。 その場合、書き込み権限と読み取り権限のどちらが付与されるのでしょうか。

答えを見つけるために、複数のプロファイルを特定のオブジェクトに適用できるときには、必ず**最も特定されたプロファイルだけを適用する**というルールを適用します。 この規則を適用する方法として、プロファイル名は左から右に比較します。 違いを見つけた箇所では、必ず非総称文字が総称文字よりも限定的ということになります。 このため、上記の例では、キュー AB.CD は**書き込み**権限を持つことになります (AB.C* は、AB.* よりも限定的)。

汎用文字を比較する場合、特定 の順序は以下のようになります。

1. ?
2. *
3. **

## プロファイル設定のダンプ

**dmpmqaut** 制御コマンドとその構文の詳細な定義については、dmpmqaut を参照してください。 **MQCMD_INQUIRE_AUTH_RECS** PCF コマンドとその構文の詳細な定義については、Inquire Authority Records を参照してください。

以下には、**dmpmqaut** 制御コマンドを使用して汎用プロファイルの権限レコードをダンプする例を示します。

1. 次の例では、プリンシパル user1 に対するキュー a.b.c と一致するプロファイルのすべての権限レコードがダンプされます。

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   結果のダンプは、次のようになります。

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   **注 :** UNIX and Linux ユーザーは **dmpmqaut** コマンドで -p オプションを使用できますが、権限を定義するときには -g groupname を代わりに使用する必要があります。

2. 次の例では、キュー a.b.c と一致するプロファイルのすべての権限レコードがダンプされます。

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

結果のダンプは、次のようになります。

```
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. この例では、プロファイル a.b. * のすべての権限レコードをダンプします。 タイプ・キュー。

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

結果のダンプは、次のようになります。

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 次の例では、キュー・マネージャー qmX に対する権限レコードすべてがダンプされます。

```
dmpmqaut -m qmX
```

結果のダンプは、次のようになります。

```
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. 次の例では、キュー・マネージャー qmX に対するプロファイル名とオブジェクト・タイプがすべてダンプされます。

```
dmpmqaut -m qmX -l
```

結果のダンプは、次のようになります。

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

**注 :** WebSphere MQ for Windows の場合に限り、表示されるすべてのプリンシパルに次のようなドメイン情報が付帯します。

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

## *OAM* プロファイルでのワイルドカード文字の使用

オブジェクト権限マネージャー (OAM) プロファイル名でワイルドカード文字を使用することによって、そのプロファイルを複数のオブジェクトに適用できます。

プロファイルが総称である理由は、プロファイル名において特殊文字 (ワイルドカード文字) が使用できるためです。 例えば、疑問符 (?) というワイルドカード文字は、名前に含まれる任意の 1 文字に一致します。そのため、ABC.?EF を指定すると、そのプロファイルに付与する許可は、ABC.DEF、ABC.CEF、ABC.BEF などの名前を持つすべてのオブジェクトに適用されます。

使用できるワイルドカード文字は次のとおりです。

**?**

任意の 1 文字の代わりに疑問符 (?) を使用します。 例えば、AB.?D は、AB.CD、AB.ED、AB.FD の各オブジェクトに適用されます。

*****

アスタリスク (*) は、次のように使用します。

- プロファイル名に含まれる修飾子 に使用して、オブジェクト名に含まれる任意の修飾子 1 つに一致します。 修飾子は、ピリオドで区切られた、オブジェクト名の部分です。 例えば、ABC.DEF.GHI では、修飾子は ABC、DEF、および GHI です。

  例えば、ABC.*.JKL は、ABC.DEF.JKL、ABC.GHI.JKL の各オブジェクトに適用されます。 (ただし、このコンテキストで * を使用する場合は、常に 1 つの修飾子を指すので、ABC.JKL には適用され**ません**。)
- プロファイル名に含まれる修飾子の文字 1 つは、オブジェクト名に含まれる 0 個以上の文字に一致します。

  例えば、ABC.DE*.JKL は、ABC.DE.JKL、ABC.DEF.JKL、ABC.DEGH.JKL の各オブジェクトに適用されます。

******

二重アスタリスク (**) は、次のようにして、プロファイル名の中で、**1 回のみ**使用します。

- プロファイル名全体をすべてのオブジェクト名と一致させます。 例えば、プロセスを識別するために -t prcs を使用する場合、プロファイル名として ** を使用し、すべてのプロセスの許可を変更します。
- プロファイル名の先頭、中ほど、最後の修飾子のいずれかが、オブジェクト名に含まれる 0 個以上の文字に一致します。 例えば、 **.ABC は、最終修飾子 ABC を持つすべてのオブジェクトを識別します。

**注 :** UNIX and Linux システムでワイルドカード文字を使用しているとき、プロファイル名を単一引用符で囲む**必要があります**。

## プロファイルの優先順位

1 つのオブジェクトに適用される汎用プロファイルが複数存在する場合もあります。 そのような場合は、最も具体的なルールが適用されます。

汎用プロファイルの使用を理解する上で重要な点は、作成するオブジェクトに適用する権限を決定するときにプロファイルに与えられる優先順位です。 例えば、次のコマンドを発行するとします。

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

1 番目は、プロファイル AB. *; と一致する名前を持つプリンシパルのすべてのキューに対する書き込み権限を付与します。 2 番目のコマンドは、プロファイル AB.C*。

AB.CD と呼ばれるキューを作成するとします。 ワイルドカード突き合わせのルールによると、いずれかの setmqaut がそのキューに適用されます。 その場合、書き込み権限と読み取り権限のどちらが付与されるのでしょうか。

答えを見つけるために、複数のプロファイルを特定のオブジェクトに適用できるときには、必ず**最も特定されたプロファイルだけを適用する**というルールを適用します。 この規則を適用する方法として、プロファイル名は左から右に比較します。 違いを見つけた箇所では、必ず非総称文字が総称文字よりも限定的ということになります。 このため、上記の例では、キュー AB.CD は**書き込み**権限を持つことになります (AB.C* は、AB.* よりも限定的)。

汎用文字を比較する場合、特定 の順序は以下のようになります。

1. ?
2. *
3. **

## プロファイル設定のダンプ

**dmpmqaut** 制御コマンドまたは **MQCMD_INQUIRE_AUTH_RECS** PCF コマンドを使用して、指定のプロファイルに関連した現在の権限をダンプします。

**dmpmqaut** 制御コマンドとその構文の詳細な定義については、dmpmqaut を参照してください。 **MQCMD_INQUIRE_AUTH_RECS** PCF コマンドとその構文の詳細な定義については、Inquire Authority Records を参照してください。

以下には、**dmpmqaut** 制御コマンドを使用して汎用プロファイルの権限レコードをダンプする例を示します。

1. 次の例では、プリンシパル user1 に対するキュー a.b.c と一致するプロファイルのすべての権限レコードがダンプされます。

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   結果のダンプは、次の例のようになります。

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   **注 :** UNIX and Linux ユーザーは -p オプションを使用できません。代わりに -g groupname を使用する必要があります。
2. 次の例では、キュー a.b.c と一致するプロファイルのすべての権限レコードがダンプされます。

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   結果のダンプは、次の例のようになります。

   ```
   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. この例では、プロファイル a.b. * のすべての権限レコードをダンプします。 タイプ・キュー。

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

結果のダンプは、次の例のようになります。

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. 次の例では、キュー・マネージャー qmX に対する権限レコードすべてがダンプされます。

```
dmpmqaut -m qmX
```

結果のダンプは、次の例のようになります。

```
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. 次の例では、キュー・マネージャー qmX に対するプロファイル名とオブジェクト・タイプがすべてダンプされます。

```
dmpmqaut -m qmX -l
```

結果のダンプは、次の例のようになります。

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

**注 :** WebSphere MQ for Windows の場合に限り、表示されるすべてのプリンシパルに次のようなドメイン情報が付帯します。

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

## アクセス設定の表示

**dspmqaut** 制御コマンドまたは **MQCMD_INQUIRE_ENTITY_AUTH** PCF コマンドは、特定のオブジェクトに関して特定のプリンシパルまたはグループが持つ許可を表示する場合に使用します。

このコマンドを使用するために、キュー・マネージャーを実行する必要があります。 プリンシパルのアクセス権を変更すると、この変更は OAM によって即時に反映されます。 一度に 1 つのグループまたはプリンシパルの許可のみが表示されます。 **dmpmqaut** 制御コマンドとその構文の詳細な定義については、『dmpmqaut』を参照してください。**MQCMD_INQUIRE_ENTITY_AUTH** PCF コマンドとその構文の詳細な定義については、『エンティティー権限の照会』を参照してください。

以下の例は、**dspmqaut** 制御コマンドを使用して、キュー・マネージャー QueueMan1 上にある Annuities という名前のプロセス定義に対してグループ GpAdmin が持つ許可を表示する方法を示しています。

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

## IBM WebSphere MQ オブジェクトへのアクセス権の変更および取り消し

あるオブジェクトに対してユーザーまたはグループが持つアクセスのレベルを変更するには、**setmqaut** コマンドを使用します。 許可を持つグループのメンバーである特定ユーザーのアクセス権を取り消すには、そのグループからそのユーザーを除去します。

グループからユーザーを削除するプロセスについては、以下で説明されています。

- 82 ページの『Windows でのグループの作成と管理』
- 84 ページの『HP-UX でのグループの作成と管理』
- 86 ページの『AIX でのグループの作成と管理』
- 87 ページの『Solaris でのグループの作成と管理』
- 88 ページの『Linux でのグループの作成と管理』

.

IBM WebSphere MQ オブジェクトを作成するユーザー ID は、そのオブジェクトに対する完全な制御権限を付与されます。 ローカル mqm グループ (または Windows システムでは Administrators グループ) からこのユーザー ID を除去しても、これらの権限は取り消されません。 あるオブジェクトを作成したユーザー ID について、そのオブジェクトへのアクセス権を取り消すには、そのユーザー ID を mqm または Administrators グループから除去してから、**setmqaut** 制御コマンドまたは **MQCMD_DELETE_AUTH_REC** PCF コマンドを使用します。 setmqaut 制御コマンドとその構文の完全な定義については、setmqaut を参照してください。また、**MQCMD_INQUIRE_ENTITY_AUTH** PCF コマンドとその構文の完全な定義については、 Inquire Entity Authority を参照してください。

Windows では、ユーザー・プロファイルを削除する前に、特定の Windows ユーザー・アカウントに対応する OAM 項目を削除してください。 ユーザー・アカウントの削除後に OAM 項目を削除することはできません。

## UNIX, Linux, and Windows システムでのセキュリティー・アクセス検査の抑止

すべてのセキュリティー検査をオフにするために、OAM を無効にできます。 テスト環境では、その設定が適している場合もあります。 OAM を無効にするか削除した後で、既存のキュー・マネージャーに OAM を追加することはできません。

(例えば、テスト環境で) セキュリティー検査を実行しないことを決定する場合、以下の 2 つのいずれかの方法で OAM を使用不可にすることができます。

- キュー・マネージャーを作成する前に、オペレーティング・システムの環境変数 MQSNOAUT を次のように設定します (このようにする場合、後で OAM を追加することはできません)。

  MQSNOAUT 変数の設定による 影響 について詳しくは、 環境変数 を参照してください。
- キュー・マネージャー構成ファイルを編集して、サービスを削除します。 (これを行った場合、後から OAM を追加することはできません)。

OAM が無効になっている状態で setmqaut または dspmqaut を使用する場合の注意点を以下にまとめます。

- OAM は、指定されたプリンシパルまたはグループを妥当性検査しません。これは、コマンドが無効な値を 受け入れる ことができることを意味します。
- OAM は、セキュリティー検査を実行しません。つまり、すべてのプリンシパルとグループに、該当するすべてのオブジェクト操作を実行する権限があると見なされます。

**警告 :** OAM が除去されると、それを既存のキュー・マネージャーに戻すことはできません。 これは、OAM がオブジェクト作成時に配置されている必要があるためです。 WebSphere MQ OAM を除去後に再び使用するには、キュー・マネージャーを再作成する必要があります。

**関連概念**

インストール可能サービス

# リソースへの必要なアクセス権限の付与

このトピックを使用して、ご使用の WebSphere MQ システムにセキュリティーを適用するために、どのタスクを実行すべきかを判別してください。

## このタスクについて

このタスクでは、ご使用の WebSphere MQ インストール済み環境の要素に適切なレベルのセキュリティーを適用するために、どのアクションが必要かを判別します。 参照先のそれぞれのタスクには、すべてのプラットフォーム用のステップバイステップの指示が記載されています。

## 手順

1. キュー・マネージャーへのアクセスを、特定のユーザーに限定する必要がありますか?

   a) いいえ: アクションは必要ありません。

   b) はい: 次の質問に進んでください。
2. アクセスを許可されるユーザーには、キュー・マネージャー・リソースのサブセットに対する部分的な管理アクセス権が必要ですか?

   a) いいえ: 次の質問に進んでください。

   b) はい: 169 ページの『キュー・マネージャー・リソースのサブセットに対する部分的な管理アクセス権の付与』を参照してください。
3. アクセスを許可されるユーザーには、キュー・マネージャー・リソースのサブセットに対する全管理アクセス権が必要ですか?

   a) いいえ: 次の質問に進んでください。

   b) はい: 174 ページの『キュー・マネージャー・リソースのサブセットに対する全管理アクセス権の付与』を参照してください。
4. アクセスを許可されるユーザーには、すべてのキュー・マネージャー・リソースに対する読み取り専用アクセス権が必要ですか?

   a) いいえ: 次の質問に進んでください。

   b) はい: 178 ページの『キュー・マネージャー上のすべてのリソースへの読み取り専用アクセス権の付与』を参照してください。

5. アクセスを許可されるユーザーには、すべてのキュー・マネージャー・リソースに対する全管理アクセス権が必要ですか?
   a) いいえ: 次の質問に進んでください。
   b) はい: 179 ページの『キュー・マネージャー上のすべてのリソースへの全管理アクセス権の付与』を参照してください。
6. ユーザー・アプリケーションがキュー・マネージャーに接続する必要がありますか?
   a) いいえ: 180 ページの『キュー・マネージャーへの接続の除去』の説明に従い、接続を無効にしてください。
   b) はい: 181 ページの『ユーザー・アプリケーションがキュー・マネージャーに接続できるようにする』を参照してください。

## キュー・マネージャー・リソースのサブセットに対する部分的な管理アクセス権の付与

特定のユーザーに、全部ではなく一部のキュー・マネージャー・リソースに対する部分的な管理アクセス権を付与する必要があります。 以下の表を使用して、行う必要のあるアクションを判別してください。

表 *14.* キュー・マネージャー・リソースのサブセットに対する部分的な管理アクセス権の付与

| **ユーザーが管理する必要のあるオブジェクトのタイプ** | **行うアクション** |
|---|---|
| キュー | 169 ページの『いくつかのキューへの限定された管理アクセス権の付与』の説明に従い、必要なキューへの部分的な管理アクセス権を付与する |
| トピック | 170 ページの『いくつかのトピックへの限定された管理アクセス権の付与』の説明に従い、必要なトピックへの部分的な管理アクセス権を付与する |
| チャネル | 171 ページの『いくつかのチャネルへの限定された管理アクセス権の付与』の説明に従い、必要なチャネルへの部分的な管理アクセス権を付与する |
| キュー・マネージャー | 171 ページの『キュー・マネージャーへの限定された管理アクセス権の付与』の説明に従い、キュー・マネージャーへの部分的な管理アクセス権を付与する |
| Processes | 172 ページの『いくつかのプロセスへの限定された管理アクセス権の付与』の説明に従い、必要なプロセスへの部分的な管理アクセス権を付与する |
| 名前リスト | 172 ページの『いくつかの名前リストへの限定された管理アクセス権の付与』の説明に従い、必要な名前リストへの部分的な管理アクセス権を付与する |
| サービス | 173 ページの『いくつかのサービスへの限定された管理アクセス権の付与』の説明に従い、必要なサービスへの部分的な管理アクセス権を付与する |

### いくつかのキューへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのキューへの部分的な管理アクセス権を付与します。

### このタスクについて

いくつかのアクションのためいくつかのキューへの限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
```

- 変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

  **ReqdAction**
  グループに実行許可を与えるアクション。

  – UNIX、Linux、および Windows システムでは、次の権限の任意の組み合わせ: +chg、+clr、+dlt、+dsp。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

  **注 :** キューに対して +crt 権限を付与すると、当該ユーザーまたはグループが間接的に管理者として設定されます。 一部のキューに対する限定された管理アクセス権を付与する際に、+crt 権限は使用しないでください。

  **QType**

  DISPLAY コマンドの場合、値 QUEUE、QLOCAL、QALIAS、QMODEL、QREMOTE、または QCLUSTER のうちの 1 つ。

  *ReqdAction* の他の値の場合は、値 QLOCAL、QALIAS、QMODEL、または QREMOTE のうちの 1 つ。

# いくつかのトピックへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのトピックへの部分的な管理アクセス権を付与します。

## このタスクについて

いくつかのアクションのためいくつかのトピックへの限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- 変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

  **ReqdAction**
  グループに実行許可を与えるアクション。

  – UNIX、 Linux および Windows システムでは、次の権限の任意の組み合わせ: + chg、+ clr、+ crt、+ dlt、+ dsp。 + ctrl。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

## いくつかのチャネルへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのチャネルへの部分的な管理アクセス権を付与します。

## このタスクについて

いくつかのアクションのためいくつかのチャネルへの限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- 変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

  **ReqdAction**
  グループに実行許可を与えるアクション。

  – UNIX、 Linux および Windows システムでは、次の権限の任意の組み合わせ: + chg、+ clr、+ crt、+ dlt、+ dsp。 + ctrl、+ ctrlx。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

## キュー・マネージャーへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャーへの部分的な管理アクセス権を付与します。

## このタスクについて

キュー・マネージャーに対していくつかのアクションを実行するために限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction)
MQMNAME('QMgrName')
```

## タスクの結果

キュー・マネージャーに対してどの MQSC コマンドをユーザーが実行できるかを決定するには、各 MQSC コマンドについて次のコマンドを実行します。

```
RDEFINE MQCMDS QMgrName.ReqdAction.QMGR UACC(NONE)
PERMIT QMgrName.ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

ユーザーが DISPLAY QMGR コマンドを使用することを許可するには、次のコマンドを実行します。

```
RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

**ReqdAction**
グループに実行許可を与えるアクション。

- UNIX、Linux、および Windows システムでは、次の権限の任意の組み合わせ: +chg、+clr、+crt、+dlt、+dsp。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

  +set は MQI 権限であり、通常は管理権限とは見なされませんが、キュー・マネージャーに対する +set の付与は、間接的に完全な管理権限を付与することになる可能性があります。 通常のユーザーおよびアプリケーションに +set を付与しないでください。

## いくつかのプロセスへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのプロセスへの部分的な管理アクセス権を付与します。

### このタスクについて

いくつかのアクションのためいくつかのプロセスへの限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
  ```

- 変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

  **ReqdAction**
  グループに実行許可を与えるアクション。

  – UNIX、Linux、および Windows システムでは、次の権限の任意の組み合わせ: +chg、+clr、+crt、+dlt、+dsp。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

## いくつかの名前リストへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかの名前リストへの部分的な管理アクセス権を付与します。

### このタスクについて

いくつかのアクションのためいくつかの名前リストへの限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- 変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

  **ReqdAction**
  グループに実行許可を与えるアクション。

  – UNIX、Linux、および Windows システムでは、次の権限の任意の組み合わせ: +chg、+clr、+crt、+dlt、+ctrl、+ctrlx、+dsp。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

# いくつかのサービスへの限定された管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのサービスへの部分的な管理アクセス権を付与します。

## このタスクについて

いくつかのアクションのためいくつかのサービスへの限定された管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

**注 :** z/OS ではサービス・オブジェクトが存在しません。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction)
MQMNAME('QMgrName')
```

## タスクの結果

これらのコマンドは、指定されたサービスへのアクセス権を付与します。 サービスに対してどの MQSC コマンドをユーザーが実行できるかを決定するには、各 MQSC コマンドについて次のコマンドを実行します。

```
RDEFINE MQCMDS QMgrName.ReqdAction.SERVICE UACC(NONE)
PERMIT QMgrName.ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

ユーザーが DISPLAY SERVICE コマンドを使用することを許可するには、次のコマンドを実行します。

```
RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

**ReqdAction**
グループに実行許可を与えるアクション。

- UNIX、Linux、および Windows システムでは、次の権限の任意の組み合わせ: +chg、+clr、+crt、+dlt、+ctrl、+ctrlx、+dsp。 権限 +alladm は +chg +clr +dlt +dsp と等価です。

## キュー・マネージャー・リソースのサブセットに対する全管理アクセス権の付与

特定のユーザーに、全部ではなく一部のキュー・マネージャー・リソースに対する全管理アクセス権を付与する必要があります。 以下の表を使用して、行う必要のあるアクションを判別してください。

表 *15.* キュー・マネージャー・リソースのサブセットに対する全管理アクセス権の付与

| **ユーザーが管理する必要のあるオブジェクトのタイプ** | **行うアクション** |
|---|---|
| キュー | 174 ページの『いくつかのキューへの全管理アクセス権の付与』の説明に従い、必要なキューへの全管理アクセス権を付与する |
| トピック | 175 ページの『いくつかのトピックへの全管理アクセス権の付与』の説明に従い、必要なトピックへの全管理アクセス権を付与する |
| チャネル | 175 ページの『いくつかのチャネルへの全管理アクセス権の付与』の説明に従い、必要なチャネルへの全管理アクセス権を付与する |
| キュー・マネージャー | 176 ページの『キュー・マネージャーへの全管理アクセス権の付与』の説明に従い、キュー・マネージャーへの全管理アクセス権を付与する |
| Processes | 177 ページの『いくつかのプロセスへの全管理アクセス権の付与』の説明に従い、必要なプロセスへの全管理アクセス権を付与する |
| 名前リスト | 177 ページの『いくつかの名前リストへの全管理アクセス権の付与』の説明に従い、必要な名前リストへの全管理アクセス権を付与する |
| サービス | 178 ページの『いくつかのサービスへの全管理アクセス権の付与』の説明に従い、必要なサービスへの全管理アクセス権を付与する |

## いくつかのキューへの全管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのキューへの全管理アクセス権を付与します。

### このタスクについて

いくつかのキューへの全管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.QUEUE.ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## いくつかのトピックへの全管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのトピックへの全管理アクセス権を付与します。

## このタスクについて

いくつかのアクションのためいくつかのトピックへの全管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM)
MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.TOPIC.ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## いくつかのチャネルへの全管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのチャネルへの全管理アクセス権を付与します。

## このタスクについて

いくつかのチャネルへの全管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.CHANNEL.ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## キュー・マネージャーへの全管理アクセス権の付与

業務上必要とする各ユーザー・グループに、キュー・マネージャーに対する完全な管理アクセス権を付与します。

### このタスクについて

キュー・マネージャーに対する完全な管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## いくつかのプロセスへの全管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのプロセスへの全管理アクセス権を付与します。

## このタスクについて

いくつかのプロセスに対する完全な管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.CHANNEL.ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

  変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

## いくつかの名前リストへの全管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかの名前リストへの全管理アクセス権を付与します。

## このタスクについて

いくつかの名前リストへの全管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM)
MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.NAMELIST.ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

  変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## いくつかのサービスへの全管理アクセス権の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャー上のいくつかのサービスへの全管理アクセス権を付与します。

### このタスクについて

いくつかのサービスへの全管理アクセス権を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.SERVICE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SERVICE.ObjectProfile CLASS(MQADMIN) ID(GroupName ) ACCESS(ALTER)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## キュー・マネージャー上のすべてのリソースへの読み取り専用アクセス権の付与

業務上それを必要とする各ユーザーまたはユーザー・グループに、キュー・マネージャー上のすべてのリソースへの読み取り専用アクセス権を付与します。

### このタスクについて

「役割に基づく権限の追加」ウィザード、またはご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- ウィザードの使用:
  a) WebSphere MQ エクスプローラーのナビゲーター・ペインで、キュー・マネージャーを右クリックし、「**オブジェクト権限**」 > 「**役割に基づく権限の追加**」をクリックします。
  「役割に基づく権限の追加」ウィザードが開きます。
- UNIX および Windows システムの場合、次のコマンドを実行します。

```
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
```

SYSTEM.ADMIN.COMMAND.QUEUE および SYSTEM.MQEXPLORER.REPLY.MODEL に対する特定の権限が必要なのは、MQ エクスプローラーを使用する場合に限られます。

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*Q) USER('GroupName') AUT(*ADMDSP *BROWSE) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*TOPIC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CHL) USER('GroupName') AUT(*ADMDSP *INQ) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CLTCN) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*AUTHINFO) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*LSR) USER('GroupName') AUT(*ADMDSP)MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*NMLIST) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*SVC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ('object-name') OBJTYPE(*MQM)  USER('GroupName') AUT(*ADMDSP *CONNECT *INQ)
MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
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)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**GroupName**
アクセス権を付与されるグループの名前。

## キュー・マネージャー上のすべてのリソースへの全管理アクセス権の付与

業務上それを必要とする各ユーザーまたはユーザー・グループに、キュー・マネージャー上のすべてのリソースへの全管理アクセス権を付与します。

### このタスクについて

「役割に基づく権限の追加」ウィザード、またはご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- ウィザードの使用:

a) WebSphere MQ エクスプローラーのナビゲーター・ペインで、キュー・マネージャーを右クリックし、「**オブジェクト権限**」 > 「**役割に基づく権限の追加**」をクリックします。

「役割に基づく権限の追加」ウィザードが開きます。

- UNIX and Linux システムの場合は、以下のコマンドを実行します。

```
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 システムの場合、UNIX and Linux システムの場合と同じコマンドを発行しますが、@class の代わりにプロファイル名 @CLASS を使用します。
- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER('GroupName') AUT(*ALLADM) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**GroupName**
アクセス権を付与されるグループの名前。

## キュー・マネージャーへの接続の除去

ユーザー・アプリケーションがキュー・マネージャーに接続しないようにするには、そのアプリケーションのキュー・マネージャーへの接続権限を除去します。

### このタスクについて

使用するオペレーティング・システムに適切なコマンドを使用して、キュー・マネージャーに接続するための権限をすべてのユーザーから取り消します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i の場合は、以下のコマンドを発行します。

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- z/OS の場合は、以下のコマンドを実行します。

```
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 コマンドは発行しないでください。

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**GroupName**
アクセスを拒否されるグループの名前。

## ユーザー・アプリケーションがキュー・マネージャーに接続できるようにする

ユーザー・アプリケーションからキュー・マネージャーへの接続を許可する必要がある場合を考慮します。このトピックの表を使用して、行うべきアクションを判別します。

最初に、クライアント・アプリケーションがキュー・マネージャーに接続するかどうかを決定します。

キュー・マネージャーに接続するアプリケーションがいずれもクライアント・アプリケーションではない場合は、リモート・アクセスを使用不可にします (188 ページの『キュー・マネージャーへのリモート・アクセスを使用不可にする』を参照)。

キュー・マネージャーに接続するアプリケーションの 1 つ以上がクライアント・アプリケーションである場合は、リモート接続を保護します (181 ページの『キュー・マネージャーへのリモート接続の保護』を参照)。

いずれの場合も、188 ページの『接続セキュリティーのセットアップ』の説明どおりに接続のセキュリティーをセットアップします。

キュー・マネージャーに接続している各ユーザーの、リソースへのアクセスを制御する必要がある場合には、以下の表を参照してください。 最初の欄の記述が真である場合に、2 番目の欄にリストされているアクションを行います。

| 記述 | 行うアクション |
|---|---|
| キューを利用するアプリケーションがある | 189 ページの『キューへのユーザー・アクセスの制御』を参照してください。 |
| トピックを利用するアプリケーションがある | 194 ページの『トピックへのユーザー・アクセスの制御』を参照してください。 |
| キュー・マネージャー・オブジェクトに対して照会を実行するアプリケーションがある | 195 ページの『キュー・マネージャーで照会を行うための権限の付与』を参照してください。 |
| プロセス・オブジェクトを使用するアプリケーションがある | 196 ページの『プロセスにアクセスするための権限の付与』を参照してください。 |
| 名前リストを利用するアプリケーションがある | 196 ページの『名前リストにアクセスするための権限の付与』を参照してください。 |

## キュー・マネージャーへのリモート接続の保護

キュー・マネージャーへのリモート接続は、SSL か TLS、セキュリティー出口、チャネル認証レコード、またはこれらの方式の組み合わせを使用して保護できます。

### このタスクについて

クライアント・ワークステーション上でクライアント接続チャネルを使用し、サーバー上でサーバー接続チャネルを使用して、クライアントをキュー・マネージャーに接続します。 以下のいずれかの方法で、この種の接続を保護します。

## 手順

1. SSL または TLS とチャネル認証レコードの併用:
   a) SSLPEERMAP チャネル認証レコードを使用し、すべての識別名 (DN) を USERSRC(NOACCESS) にマップして、DN でチャネルがオープンされないようにします。
   b) SSLPEERMAP チャネル認証レコードを使用し、特定の DN または DN の集合を USERSRC(CHANNEL) にマップして、それらの DN でチャネルをオープンできるようにします。
2. SSL または TLS とセキュリティー出口の併用:
   a) サーバー接続チャネル上の MCAUSER を、何の特権も持たないユーザー ID に設定します。
   b) 渡される MQCD 構造体内の SSLPeerNamePtr および SSLPeerNameLength フィールドで受け取る SSL DN の値に応じて MCAUSER 値を割り当てるよう、セキュリティー出口を作成します。
3. SSL または TLS と固定チャネル定義値の併用:
   a) サーバー接続チャネル上の SSLPEER を、特定の値、または狭い範囲の値に設定します。
   b) サーバー接続チャネル上の MCAUSER を、チャネルの実行時に使用するユーザー ID に設定します。
4. SSL または TLS を使用しないチャネルでのチャネル認証レコードの使用:
   a) ADDRESS(*) および USERSRC(NOACCESS) を指定したアドレス・マッピング・チャネル認証レコードを使用して、IP アドレスでチャネルがオープンされないようにします。
   b) USERSRC(CHANNEL) を指定した特定の IP アドレスに関するアドレス・マッピング・チャネル認証レコードを使用して、これらのアドレスでチャネルをオープンできるようにします。
5. セキュリティー出口の使用:
   a) 例えば発信元の IP アドレスなど、選択したプロパティーに基づいて接続権限を与えるよう、セキュリティー出口を作成します。
6. 特定の環境での必要に応じて、チャネル認証レコードとセキュリティー出口を併用することも、3 つの方式をすべて使用することもできます。

特定の *IP* アドレスのブロッキング
チャネル認証レコードを使用して、特定のチャネルが IP アドレスからのインバウンド接続を受け入れないように、またはキュー・マネージャー全体が IP アドレスからのアクセスを受け入れないようにすることができます。

## 始める前に

次のコマンドを実行して、チャネル認証レコードを使用可能にします。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## このタスクについて

特定のチャネルがインバウンド接続を受け入れないようにして、正しいチャネル名を使用している場合にのみ接続を受け入れるようにするために、1 つのタイプのルールを使用して IP アドレスをブロックすることができます。 ある IP アドレスからキュー・マネージャー全体にアクセスできないようにするには、通常はファイアウォールを使用してそのアドレスを永久にブロックします。 しかし、別のタイプのルールを使用して、ファイアウォールが更新されるのを待っている間などに、いくつかのアドレスを一時的にブロックすることができます。

## 手順

- IP アドレスが特定のチャネルを使用できないようにするには、MQSC コマンド **SET CHLAUTH** または PCF コマンド **Set Channel Authentication Record** を使用してチャネル認証レコードを設定します。

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

このコマンドは、以下の 3 つの部分で構成されます。

**SET CHLAUTH (*generic-channel-name*)**
コマンドのこの部分を使用して、キュー・マネージャー全体、単一のチャネル、またはチャネル範囲のいずれを対象にして接続をブロックするかを制御します。 ここに指定する内容によって、対象となる領域が決まります。

以下に例を示します。

- SET CHLAUTH('*') - キュー・マネージャー上のすべてのチャネル、つまりキュー・マネージャー全体をブロックします。
- SET CHLAUTH('SYSTEM.*') - SYSTEM で始まるチャネルをすべてブロックします。
- SET CHLAUTH('SYSTEM.DEF.SVRCONN') - チャネル SYSTEM.DEF.SVRCONN をブロックします。

**CHLAUTH 規則のタイプ**
コマンドのこの部分を使用して、コマンドのタイプを指定し、単一のアドレスを渡すか、それともアドレスのリストを渡すかを決定します。

以下に例を示します。

- TYPE(ADDRESSMAP) - 単一のアドレスまたはワイルドカード・アドレスを渡す場合には ADDRESSMAP を使用します。 例えば、ADDRESS('192.168.*') は 192.168 で始まる IP アドレスからのすべての接続をブロックします。

  パターンを使用した IP アドレスのフィルタリングについて詳しくは、汎用 IP アドレスを参照してください。
- TYPE(BLOCKADDR) - ブロックするアドレスのリストを渡す場合には BLOCKADDR を使用します。

**その他のパラメーター**
これらのパラメーターは、コマンドの 2 番目の部分で使用した規則のタイプに依存します。

- TYPE(ADDRESSMAP) の場合、ADDRESS を使用します。
- TYPE(BLOCKADDR) の場合、ADDRLIST を使用します。

**関連資料**
SET CHLAUTH

キュー・マネージャーが実行していない場合に特定の *IP* アドレスを一時的にブロックする
キュー・マネージャーが実行中でないために MQSC コマンドを発行できない場合、特定の IP アドレスまたは IP アドレスの範囲をブロックしなければならないことがあります。 blockaddr.ini ファイルを変更することによって、例外的なベースで IP アドレスを一時的にブロックすることができます。

## このタスクについて

blockaddr.ini ファイルには、キュー・マネージャーによって使用される BLOCKADDR 定義のコピーが含まれています。 リスナーがキュー・マネージャーより前に開始された場合、リスナーはこのファイルを読み取ります。 このような状況では、リスナーは、blockaddr.ini ファイルに手動で追加した値を使用します。

ただし、キュー・マネージャーが開始されると、BLOCKADDR 定義のセットが blockaddr.ini ファイルに書き込まれることに注意してください。これは手動による編集が行われた可能性がある場合は上書きします。 同様に、**SET CHLAUTH** コマンドを使用して BLOCKADDR 定義を追加または削除するたびに、blockaddr.ini ファイルが更新されます。 したがって、BLOCKADDR 定義を永続的に変更できるのは、キュー・マネージャーの実行中に **SET CHLAUTH** コマンドを使用して変更した場合のみです。

## 手順

1. blockaddr.ini ファイルをテキスト・エディターで開きます。
   このファイルは、キュー・マネージャーのデータ・ディレクトリーに配置されています。
2. IP アドレスを単純なキーワードと値の対として追加します。ここで、キーワードは Addr です。
   パターンを使用した IP アドレスのフィルタリングについては、汎用 IP アドレスを参照してください。

以下に例を示します。

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

**関連タスク**

182 ページの『特定の IP アドレスのブロッキング』
チャネル認証レコードを使用して、特定のチャネルが IP アドレスからのインバウンド接続を受け入れないように、またはキュー・マネージャー全体が IP アドレスからのアクセスを受け入れないようにすることができます。

**関連資料**

SET CHLAUTH

特定のユーザー *ID* のブロッキング

チャネルが終了する原因となるユーザー ID (表明されている場合) を指定して、特定のユーザーがチャネルを使用できないようにすることができます。 これを行うには、チャネル認証レコードを設定します。

## 始める前に

以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 手順

MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

TYPE(BLOCKUSER) で提供されるユーザー・リストは SVRCONN チャネルのみに適用され、キュー・マネージャー同士のチャネルには適用されません。

*userID1* および *userID2* はそれぞれ、チャネルを使用できないようにするユーザーの ID です。 特殊値 *MQADMIN を指定して特権管理ユーザーを参照することもできます。 特権ユーザーについて詳しくは、147 ページの『特権ユーザー』を参照してください。 *MQADMIN の詳細については、SET CHLAUTH を参照してください。

**関連資料**

SET CHLAUTH

*MCAUSER* ユーザー *ID* へのリモート・キュー・マネージャーのマッピング

チャネル認証レコードを使用して、チャネルの接続元であるキュー・マネージャーに従って、チャネルの MCAUSER 属性を設定することができます。

## 始める前に

以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## このタスクについて

オプションで、特定の IP アドレスにのみ規則を適用することができます。

この技法は、サーバー接続チャネルには適用されないことに注意してください。 以下に示されているコマンドでサーバー接続チャネルの名前を指定しても、効果はありません。

## 手順

- MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user)
```

  *generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

  *generic-partner-qmgr-name* は、キュー・マネージャーの名前、あるいはキュー・マネージャー名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

  *user* は、指定されたキュー・マネージャーからのすべての接続に使用するユーザー ID です。

- このコマンドを特定の IP アドレスに対してのみ実行するには、**ADDRESS** パラメーターを以下のように組み込みます。

```
SET CHLAUTH('generic-channel-name') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user) ADDRESS(
generic-ip-address)
```

  *generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

  *generic-ip-address* は、単一アドレス、あるいはアドレスと一致するパターン (ワイルドカードを示すアスタリスク (*) 記号、または範囲を指定するハイフン (-) を含む) のいずれかです。 汎用 IP アドレスについて詳しくは、 汎用 IP アドレス を参照してください。

**関連資料**

SET CHLAUTH

*MCAUSER* ユーザー *ID* への表明済みユーザー *ID* のマッピング

チャネル認証レコードを使用して、クライアントから受け取った元のユーザー ID に従って、サーバー接続チャネルの MCAUSER 属性を変更することができます。

## 始める前に

以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## このタスクについて

この技法は、サーバー接続チャネルにのみ適用されることに注意してください。 これは、他のチャネル・タイプでは効果がありません。

## 手順

MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

*client-user-name* は、クライアントによって表明されたユーザー ID です。

*user* は、クライアントのユーザー名の代わりに使用されるユーザー ID です。

**関連資料**

SET CHLAUTH

*MCAUSER* ユーザー *ID* への *SSL* または *TLS* 識別名のマッピング
チャネル認証レコードを使用して、受け取った識別名 (DN) に従って、チャネルの MCAUSER 属性を設定することができます。

## 始める前に
以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 手順
MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP) SSLPEER(generic-ssl-peer-name
) USERSRC(MAP) MCAUSER(user)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。
*generic-ssl-peer-name* は、標準 IBM WebSphere MQ ルールに従った SSLPEER 値のストリングです。 SSLPEER 値についての WebSphere MQ の規則を参照してください。
*user* は、指定された DN を使用するすべての接続に使用するユーザー ID です。

**関連資料**
SET CHLAUTH

リモート・キュー・マネージャーからのアクセスのブロック化
チャネル認証レコードを使用して、リモート・キュー・マネージャーがチャネルを始動できないようにすることができます。

## 始める前に
以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## このタスクについて
この技法は、サーバー接続チャネルには適用されないことに注意してください。 以下に示されているコマンドでサーバー接続チャネルの名前を指定しても、効果はありません。

## 手順
MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE(QMGRMAP) QMNAME('generic-partner-qmgr-name')
USERSRC(NOACCESS)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。
*generic-partner-qmgr-name* は、キュー・マネージャーの名前、あるいはキュー・マネージャー名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

**関連資料**
SET CHLAUTH

クライアントが表明したユーザー *ID* のアクセスのブロック化
チャネル認証レコードを使用して、クライアントが表明したユーザー ID がチャネルを使用できないようにすることができます。

## 始める前に
以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## このタスクについて
この技法は、サーバー接続チャネルにのみ適用されることに注意してください。 これは、他のチャネル・タイプでは効果がありません。

## 手順
MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE(USERMAP) CLNTUSER('client-user-name') USERSRC(NOACCESS)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。
*client-user-name* は、クライアントによって表明されたユーザー ID です。

**関連資料**
SET CHLAUTH

*SSL* 識別名のアクセスのブロック化
チャネル認証レコードを使用して、SSL 識別名がチャネルを始動できないようにすることができます。

## 始める前に
以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

## 手順
MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP) SSLPEER('generic-ssl-peer-name')
USERSRC(NOACCESS)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。
*generic-ssl-peer-name* は、標準 IBM WebSphere MQ ルールに従った SSLPEER 値のストリングです。 SSLPEER 値についての WebSphere MQ の規則を参照してください。

**関連資料**
SET CHLAUTH

*MCAUSER* ユーザー *ID* への *IP* アドレスのマッピング
チャネル認証レコードを使用して、接続の受信元である IP アドレスに従って、チャネルの MCAUSER 属性を設定することができます。

## 始める前に
以下のようにして、チャネル認証レコードが有効になっていることを確認します。

```
ALTER QMGR CHLAUTH(ENABLED)
```

### 手順

MQSC コマンド **SET CHLAUTH**、または PCF コマンド **Set Channel Authentication Record** を使用して、チャネル認証レコードを設定します。 例えば、以下の MQSC コマンドを発行できます。

```
SET CHLAUTH('generic-channel-name') TYPE(ADDRESSMAP) ADDRESS('generic-ip-address') USERSRC(MAP)
MCAUSER(user)
```

*generic-channel-name* は、アクセスを制御するチャネルの名前、あるいはチャネル名と一致するパターン (アスタリスク (*) 記号をワイルドカードとして含む) のいずれかです。

*user* は、指定された DN を使用するすべての接続に使用するユーザー ID です。

*generic-ip-address* は、接続の作成元となるアドレス、あるいはアドレスと一致するパターン (ワイルドカードを示すアスタリスク (*)、または範囲を指定するハイフン (-) を含む) のいずれかです。

**関連資料**

SET CHLAUTH

## キュー・マネージャーへのリモート・アクセスを使用不可にする

クライアント・アプリケーションがキュー・マネージャーに接続しないようにするには、そのキュー・マネージャーへのリモート・アクセスを使用不可にします。

### このタスクについて

以下のいずれかの方法で、クライアント・アプリケーションがキュー・マネージャーに接続できないようにします。

### 手順

- MQSC コマンド **DELETE CHANNEL** を使用して、すべてのサーバー接続チャネルを削除します。
- MQSC コマンド **ALTER CHANNEL** を使用して、チャネルのメッセージ・チャネル・エージェントのユーザー ID (MCAUSER) を、アクセス権を持たないユーザー ID に設定します。

## 接続セキュリティーのセットアップ

キュー・マネージャーに接続する業務上の必要がある各ユーザーまたはユーザー・グループに、そうする権限を付与します。

### このタスクについて

接続セキュリティーをセットアップするには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```

- IBM i の場合は、以下のコマンドを発行します。

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM)  USER('GroupName') AUT(*CONNECT)
  ```

- z/OS の場合は、以下のコマンドを実行します。

  ```
  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)
```

これらのコマンドは、バッチ、CICS、IMS、および チャネル・イニシエーター (CHIN) 用の接続権限を付与します。 特定のタイプの接続を使用しない場合は、それに対応するコマンドを省略してください。

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## キューへのユーザー・アクセスの制御

キューへのアプリケーション・アクセスを制御する必要がある場合を考慮します。 このトピックを使用して、行うべきアクションを判別します。

最初の欄の各記述が真である場合に、2 番目の欄に示されているアクションを行います。

| 記述 | アクション |
|---|---|
| アプリケーションがキューからメッセージを取得する | 189 ページの『キューからメッセージを取得する権限の付与』を参照してください。 |
| アプリケーションがコンテキストを設定する | 190 ページの『コンテキストを設定するための権限の付与』を参照してください。 |
| アプリケーションがコンテキストを渡す | 191 ページの『コンテキストを渡すための権限の付与』を参照してください。 |
| アプリケーションがクラスター・キューにメッセージを書き込む | 247 ページの『リモート・クラスター・キューへのメッセージ書き込み権限の付与』を参照してください。 |
| アプリケーションがローカル・キューにメッセージを書き込む | 191 ページの『ローカル・キューにメッセージを書き込むための権限の付与』を参照してください。 |
| アプリケーションがモデル・キューにメッセージを書き込む | 192 ページの『モデル・キューにメッセージを書き込むための権限の付与』を参照してください。 |
| アプリケーションがリモート・キューにメッセージを書き込む | 193 ページの『リモート・クラスター・キューにメッセージを書き込むための権限の付与』を参照してください。 |

### キューからメッセージを取得する権限の付与

業務上それを必要とする各ユーザー・グループに、1 つのキューまたはキューの集合からメッセージを取得する権限を付与します。

## このタスクについて

いくつかのキューからメッセージを取得する権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

コンテキストを設定するための権限の付与
業務上それを必要とする各ユーザー・グループに、書き込み中のメッセージにコンテキストを設定する権限を付与します。

## このタスクについて

いくつかのキューでコンテキストを設定する権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、 Linux 、および Windows システムの場合は、以下のいずれかのコマンドを発行します。
  - ID コンテキストのみを設定する場合:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
```

  - すべてのコンテキストを設定する場合:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
```

- IBM i の場合、次のコマンドのいずれか 1 つを実行します。
  - ID コンテキストのみを設定する場合:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('QMgrName')
```

  - すべてのコンテキストを設定する場合:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL)
MQMNAME('QMgrName')
```

- z/OS の場合、次のコマンド・セットのいずれか 1 つを実行します。
  - ID コンテキストのみを設定する場合:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  - すべてのコンテキストを設定する場合:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

コンテキストを渡すための権限の付与
業務上それを必要とする各ユーザー・グループに、取得したメッセージからのコンテキストを、書き込み中のメッセージに渡す権限を付与します。

## このタスクについて
いくつかのキューでコンテキストを渡す権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順
- UNIX、 Linux 、および Windows システムの場合は、以下のいずれかのコマンドを発行します。
  - ID コンテキストのみを渡す場合:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
    ```

  - すべてのコンテキストを渡す場合:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
    ```

- IBM i の場合、次のコマンドのいずれか 1 つを実行します。
  - ID コンテキストのみを渡す場合:

    ```
    GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID)
    MQMNAME('QMgrName')
    ```

  - すべてのコンテキストを渡す場合:

    ```
    GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL)
    MQMNAME('QMgrName')
    ```

- z/OS の場合は、次のコマンドを実行して、ID コンテキストまたはすべてのコンテキストを渡します。

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

  **ObjectProfile**
  権限を変更するオブジェクトまたは総称プロファイルの名前。

  **GroupName**
  アクセス権を付与されるグループの名前。

ローカル・キューにメッセージを書き込むための権限の付与
業務上それを必要とする各ユーザー・グループに、1 つのローカル・キューまたはローカル・キューの集合にメッセージを書き込む権限を付与します。

## このタスクについて

いくつかのローカル・キューにメッセージを書き込む権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

モデル・キューにメッセージを書き込むための権限の付与

業務上それを必要とする各ユーザー・グループに、1 つのモデル・キューまたはモデル・キューの集合にメッセージを書き込む権限を付与します。

## このタスクについて

モデル・キューは、動的キューを作成するために使用されます。 したがって、モデル・キューおよび動的キューの両方に対する権限を付与する必要があります。 これらの権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、 Linux 、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ModelQueueName') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME('QMgrName')
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
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)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ModelQueueName**
動的キューの基となるモデル・キューの名前。

**ObjectProfile**
権限を変更する動的キューまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

リモート・クラスター・キューにメッセージを書き込むための権限の付与
業務上それを必要とする各ユーザー・グループに、1 つのリモート・クラスター・キューまたはリモート・クラスター・キューの集合にメッセージを書き込む権限を付与します。

## このタスクについて

リモート・クラスター・キューにメッセージを書き込むには、リモート・キューのローカル定義、または完全修飾されたリモート・キューのいずれかにそれを書き込むことができます。 リモート・キューのローカル定義を使用する場合には、ローカル・オブジェクトに書き込むための権限が必要です。 191 ページの『ローカル・キューにメッセージを書き込むための権限の付与』を参照してください。 完全に修飾されたリモート・キューを使用する場合には、リモート・キューに書き込むための権限が必要です。 ご使用のオペレーティング・システムに対応するコマンドを使用して、この権限を付与します。

デフォルトの動作では、SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するアクセス制御を実行します。 この動作は、複数の伝送キューを使用している場合でも適用されることに注意してください。

このトピックで説明する特定の動作は、 「セキュリティー・スタンザ」 トピックで説明されているように qm.ini ファイル内の **ClusterQueueAccessControl** 属性を *RQMName* に構成し、キュー・マネージャーを再始動した場合にのみ適用されます。

UNIX、 Linux、および Windows システムでは、SET AUTHREC コマンドを使用することもできます。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -t rqmname -n
ObjectProfile -g GroupName +put
```

  リモート・クラスター・キューに関してのみ、*rqmname* オブジェクトを使用できることに注意してください。
- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
QMgrName')
```

  リモート・クラスター・キューに関してのみ、RMTMQMNAME オブジェクトを使用できることに注意してください。
- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQQUEUE  QMgrNameObjectProfile UACC(NONE)
PERMIT QMgrNameObjectProfile CLASS(MQADMIN)
ID(GroupName) ACCESS(UPDATE)
```

  リモート・クラスター・キューに関してのみ、リモート・キュー・マネージャー (またはキュー共有グループ) の名前を使用できることに注意してください。

  変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するリモート・キュー・マネージャーまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## トピックへのユーザー・アクセスの制御

トピックへのアプリケーションのアクセスを制御する必要があります。 このトピックを使用して、行うべきアクションを判別します。

最初の欄の各記述が真である場合に、2 番目の欄に示されているアクションを行います。

表 16. トピックへのユーザー・アクセスの制御

| 記述 | アクション |
|---|---|
| アプリケーションがトピックにメッセージをパブリッシュする | 194 ページの『トピックにメッセージをパブリッシュするための権限の付与』を参照してください。 |
| アプリケーションがトピックをサブスクライブする | 194 ページの『トピックをサブスクライブするための権限の付与』を参照してください。 |

### トピックにメッセージをパブリッシュするための権限の付与

業務上それを必要とする各ユーザー・グループに、1 つのトピックまたはトピックの集合にメッセージをパブリッシュする権限を付与します。

#### このタスクについて

いくつかのトピックにメッセージをパブリッシュする権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

#### 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

### トピックをサブスクライブするための権限の付与

業務上それを必要とする各ユーザー・グループに、1 つのトピックまたはトピックの集合をサブスクライブする権限を付与します。

## このタスクについて

いくつかのトピックをサブスクライブする権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

# キュー・マネージャーで照会を行うための権限の付与

業務上それを必要とする各ユーザー・グループに、キュー・マネージャーで照会を行うための権限を付与します。

## このタスクについて

キュー・マネージャーで照会を行う権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName ObjectProfile CLASS(MQCMDS) ID(GroupName ) ACCESS(READ)
```

これらのコマンドは、指定されたキュー・マネージャーへのアクセス権を付与します。 ユーザーが MQINQ コマンドを使用することを許可するには、次のコマンドを実行します。

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## プロセスにアクセスするための権限の付与

業務上それを必要とする各ユーザー・グループに、1 つのプロセスまたはプロセスの集合にアクセスする権限を付与します。

## このタスクについて

いくつかのプロセスにアクセスする権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME('QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## 名前リストにアクセスするための権限の付与

業務上それを必要とする各ユーザー・グループに、1 つの名前リストまたは名前リストの集合にアクセスする権限を付与します。

## このタスクについて

いくつかの名前リストにアクセスする権限を付与するには、ご使用のオペレーティング・システムに適切なコマンドを使用します。

## 手順

- UNIX、Linux、および Windows システムの場合は、以下のコマンドを発行します。

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
 +all
```

- IBM i の場合は、以下のコマンドを発行します。

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- z/OS の場合は、以下のコマンドを実行します。

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile
 CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

変数名の意味は次のとおりです。

**QMgrName**
キュー・マネージャーの名前。 z/OS では、この値はキュー共有グループの名前でもある可能性があります。

**ObjectProfile**
権限を変更するオブジェクトまたは総称プロファイルの名前。

**GroupName**
アクセス権を付与されるグループの名前。

## UNIX, Linux, and Windows システム上の IBM WebSphere MQ を管理する権限

IBM WebSphere MQ 管理者は、すべての IBM WebSphere MQ コマンドを使用して、他のユーザーに権限を付与することができます。 管理者がリモート・キュー・マネージャーに対してコマンドを実行する場合は、そのリモート・キュー・マネージャーで必要な権限を持っていなければなりません。 Windows システムでは、検討しなければならない考慮事項がさらにあります。

IBM WebSphere MQ 管理者には、すべての WebSphere MQ コマンド (他ユーザーに WebSphere MQ 権限を付与するコマンドを含む) を使用する権限があります。

IBM WebSphere MQ 管理者になるには、*mqm* グループという特別なグループのメンバー (Windows システムでは Administrators グループのメンバー) にならなければなりません。 mqm グループは、WebSphere MQ がインストールされると自動的に作成されます。他のユーザーが管理を実行できるようにするには、それらのユーザーをグループに追加します。 このグループのメンバー全員が、すべてのリソースに対するアクセス権を持っています。 このアクセス権は、mqm グループからユーザーを除去して REFRESH SECURITY コマンドを発行することによってのみ取り消せます。 管理者は、WebSphere MQ を管理する制御コマンドを使用できます。 これらの制御コマンドの 1 つは、**setmqaut** です。このコマンドは、WebSphere MQ リソースにアクセスまたは制御できるようにする権限を、他のユーザーに付与するのに使用されます。 権限レコードを管理するための PCF コマンドは、キュー・マネージャーで dsp および chg 権限が付与されている非管理者が使用できます。 PCF コマンドを使った権限の管理については、プログラマブル・コマンド・フォーマットを参照してください。

管理者は、**runmqsc** 制御コマンドを使用して、IBM WebSphere MQ Script (MQSC) コマンドを発行することができます。 MQSC コマンドをリモート・キュー・マネージャーに送信するために **runmqsc** が間接モードで使用される場合、各 MQSC コマンドは、Escape PCF コマンド内にカプセル化されます。 管理者は、MQSC コマンドがリモート・キュー・マネージャーによって処理されるのに必要な権限を持っていなければなりません。 WebSphere MQ エクスプローラーでは、PCF コマンドによって管理タスクを実行します。管理者には、WebSphere MQ エクスプローラーを使用してローカル・システム上のキュー・マネージャーを管理するための追加の権限は必要ありません。 IBM WebSphere MQ エクスプローラーが別のシステム上のキュー・マネージャーの管理に使用される場合、管理者には、PCF コマンドがリモート・キュー・マネージャーによって処理されるのに必要な権限が必要です。

PCF コマンドと MQSC コマンドの処理時の権限検査の詳細については、以下のトピックを参照してください。

- キュー・マネージャー、キュー、プロセス、名前リスト、認証情報オブジェクトに対して実行する PCF コマンドについては、『WebSphere MQ オブジェクトを処理する権限』を参照してください。 Escape PCF コマンド内にカプセル化される、同等の MQSC コマンドについては、このセクションを参照してください。

- チャネル、チャネル・イニシエーター、リスナー、クラスターに対して実行する PCF コマンドについては、『チャネル・セキュリティー』を参照してください。
- 権限レコードに対して実行する PCF コマンドについては、PCF コマンドの権限検査を参照してください。

さらに、 Windows システムでは、SYSTEM アカウントに WebSphere MQ リソースへの全アクセス権限があります。

UNIX and Linux プラットフォームでは、本製品でのみ使用される、mqm という特殊なユーザー ID も作成されます。 これは、特権のないユーザーは使用できません。 すべての WebSphere MQ オブジェクトはユーザー ID mqm によって所有されています。

Windows システムでは、Administrators グループのメンバーは、SYSTEM アカウントと同様に、任意のキュー・マネージャーを管理することもできます。 さらに、ドメイン内でアクティブな特権ユーザー ID すべてを含むドメイン・コントローラーでドメイン mqm グループを作成し、それをローカル mqm グループに追加することもできます。 コマンド (例えば、**crtmqm**) のなかには、IBM WebSphere MQ オブジェクト上で権限を操作するため、(以下のセクションに説明されているように) それらのオブジェクトを処理する権限が必要なものがあります。 mqm グループのメンバーには、すべてのオブジェクトを処理する権限がありますが、Windows システムでは、同じ名前のローカル・ユーザーとドメイン認証ユーザーが存在する場合、権限が拒否される場合があります。 これについては、201 ページの『プリンシパルおよびグループ』で説明されています。

ユーザー・アカウント制御 (UAC) 機能が組み込まれたバージョンの Windows は、特定のオペレーティング・システム機能に対してユーザーが実行できるアクションを制限します (そのユーザーが Administrators グループのメンバーであっても)。 ユーザー ID が Administrators グループに含まれているものの、mqm グループではない場合、昇格されたコマンド・プロンプトを使用して、**crtmqm** などの WebSphere MQ admin コマンドを発行しなければなりません。そのようにしない場合、以下のエラーが生成されます。"AMQ7077: 要求された操作の実行は許可されていません" 昇格されたコマンド・プロンプトを開くには、スタート・メニュー項目を右クリックするか、またはコマンド・プロンプトのアイコンを右クリックして、「管理者として実行」を選択します。

以下を実行するときには、mqm グループのメンバーである必要はありません。

- PCF コマンドを発行するアプリケーション・プログラムからコマンドを発行するか、またはエスケープ PCF コマンド内で MQSC コマンドを発行します。ただし、PCF コマンドがチャネル・イニシエーターを操作しない場合です。 (これらのコマンドについては、70 ページの『チャネル・イニシエーター定義の保護』を参照してください。)
- アプリケーション・プログラムから MQI 呼び出しを発行します (ただし、MQCONNX 呼び出しでファースト・パス・バインドを使用しない場合)。
- crtmqcvx コマンドを使用して、データ・タイプ構造のデータ変換を実行するコード断片を作成する。
- dspmq コマンドは、キュー・マネージャーを表示する場合に使用します。
- dspmqtrc コマンドは、WebSphere MQ の定様式トレース出力を 表示する場合に使用します。

グループおよびユーザー ID のいずれにも、12 文字までという制限が当てはまります。

UNIX and Linux プラットフォームは通常、ユーザー ID の長さを 12 文字までと制限しています。 AIX バージョン 5.3 ではこの制限が引き上げられましたが、WebSphere MQ では、すべての UNIX and Linux プラットフォームで引き続き 12 文字の制限が順守されます。 12 文字を超えるユーザー ID を使用すると、WebSphere MQ はその ID を UNKNOWN という値に置き換えます。 「UNKNOWN」という値でユーザー ID を定義しないでください。

## mqm グループの管理

mqm グループのユーザーには、WebSphere MQ に対する完全な管理特権が付与されます。 このため、アプリケーションおよび通常のユーザーを mqm グループに登録することはできません。 mqm グループには、WebSphere MQ 管理者のアカウントのみを登録してください。

これらのタスクについては、以下で説明されています。

- Windows でのグループの作成と管理
- HP-UX でのグループの作成と管理

- AIX でのグループの作成と管理
- Solaris でのグループの作成と管理
- Linux

ドメイン・コントローラーが Windows 2000 または Windows 2003 上で稼働している場合、ドメイン管理者は、使用する WebSphere MQ のために特別なアカウントを設定しなければならない場合があります。『WebSphere MQ アカウントの構成』を参照してください。

## UNIX, Linux, and Windows システム上の IBM WebSphere MQ オブジェクトを処理する権限

すべてのオブジェクトは、IBM WebSphere MQ によって保護されているので、それらのオブジェクトにアクセスするための適切な権限を各プリンシパルに与える必要があります。 プリンシパルとオブジェクトがそれぞれ異なれば、必要なアクセス権も異なります。

キュー・マネージャー、キュー、プロセス定義、名前リスト、チャネル、クライアント接続チャネル、リスナー、サービス、認証情報オブジェクトには、すべて MQI 呼び出しまたは PCF コマンドを使用するアプリケーションからアクセスします。 これらのリソースはすべて WebSphere MQ によって保護されているので、それにアクセスするための許可をアプリケーションに付与する必要があります。 要求を出すエンティティーは、ユーザー、MQI 呼び出しを発行するアプリケーション・プログラム、または PCF コマンドを発行する管理プログラムの場合があります。 要求側の ID のことをプリンシパル といいます。

同じオブジェクトに対して、プリンシパルのグループごとに異なるタイプのアクセス権限を与えることができます。 例えば、特定のキューに対して、あるグループには書き込み操作と読み取り操作の両方を許可し、別のグループにはブラウズ (ブラウズ・オプションによる MQGET) のみを許可することができます。 同様に、いくつかのグループにはあるキューに対する書き込み権限と読み取り権限がありますが、そのキューの属性を変更したり削除したりすることは許可されていません。

一部の操作は特に重要度が高いので、その実行は特権ユーザーに限る必要があります。 以下に例を示します。

- 伝送キューまたはコマンド・キュー SYSTEM.ADMIN.COMMAND.QUEUE などの 特殊キューへのアクセス
- 完全な MQI コンテキスト・オプションを使用するプログラムの実行
- アプリケーション・キューの作成と削除

オブジェクトに対する全アクセス権限は、そのオブジェクトを作成したユーザー ID と、mqm グループのすべてのメンバー (および Windows システムでは、ローカルの Administrators グループのメンバー) に対して自動的に付与されます。

**関連概念**

197 ページの『UNIX, Linux, and Windows システム上の IBM WebSphere MQ を管理する権限』
IBM WebSphere MQ 管理者は、すべての IBM WebSphere MQ コマンドを使用して、他のユーザーに権限を付与することができます。 管理者がリモート・キュー・マネージャーに対してコマンドを実行する場合は、そのリモート・キュー・マネージャーで必要な権限を持っていなければなりません。 Windows システムでは、検討しなければならない考慮事項がさらにあります。

## UNIX, Linux, and Windows システムでセキュリティー検査が行われるタイミング

通常は、キュー・マネージャーに接続するとき、オブジェクトを開いたり閉じたりするとき、メッセージを書き込んだり取り込んだりするときに、セキュリティー検査が行われます。

通常のアプリケーションで行われるセキュリティー検査は、以下のとおりです。

**キュー・マネージャーへの接続 (MQCONN または MQCONNX 呼び出し)**
アプリケーションが特定のキュー・マネージャーに関連付けられるのはこれが最初です。 キュー・マネージャーは、運用環境に問い合わせて、そのアプリケーションに関連付けられたユーザー ID を突き止めます。 続いて WebSphere MQ は、そのユーザー ID がキュー・マネージャーへ接続することを許可されていることを検査し、そのユーザー ID を将来の検査のために保存します。

ユーザーは WebSphere MQ にサインオンする必要はありません。WebSphere MQ では、ユーザーが基礎となるオペレーティング・システムにサインオンしていて、認証されているものと想定しています。

**オブジェクトのオープン (MQOPEN または MQPUT1 呼び出し)**
WebSphere MQ オブジェクトは、そのオブジェクトをオープンし、それに対してコマンドを発行することによってアクセスされます。 実際にオブジェクトがアクセスされるときではなく、オブジェクトがオープンされるときに、すべてのリソース検査が実行されます。 つまり、**MQOPEN** 要求で、必要なアクセスのタイプ (例えば、単にオブジェクトを参照するだけなのか、キューに対してメッセージを書き込むような更新を実行するのかなど) を指定しなければならないということです。

WebSphere MQ は、**MQOPEN** 要求で指定されたリソースを検査します。 別名またはリモート・キュー・オブジェクトの場合、使用される許可はオブジェクト自体に関するものであり、別名キューまたはリモート・キューが解決されるキューの許可ではありません。 そのため、ユーザーはアクセスするための許可を必要としません。 キューを作成する権限は、特権ユーザーに限定してください。 限定しないと、一部のユーザーが単に別名を作成して 通常のアクセス管理を逃れる事態になりかねません。 キュー名とキュー・マネージャー名の両方でリモート・キューが明示的に参照される場合、そのリモート・キュー・マネージャーと関連付けられた伝送キューが検査されます。

動的キューに対する権限は、それが派生したモデル・キューに対する権限に基づきます (ただし、必ずしも同じではありません)。 詳細については、注 91 ページの『1』 を参照してください。

アクセス検査のためにキュー・マネージャーによって使用されるユーザー ID は、そのキュー・マネージャーに接続されたアプリケーション運用環境から入手したユーザー ID です。 適切な権限を持つアプリケーションは、代替ユーザー ID を指定して **MQOPEN** 呼び出しを発行することができます。その後、代替ユーザー ID に対してアクセス制御検査が行われます。 この場合、アプリケーションに関連付けられたユーザー ID は変更されず、アクセス制御検査のために使用されるにすぎません。

**メッセージの書き込みと読み取り (MQPUT または MQGET 呼び出し)**
アクセス制御検査は実行されません。

**オブジェクトのクローズ (MQCLOSE)**
**MQCLOSE** の結果として動的キューが削除される場合を除き、アクセス制御検査は実行されません。 この場合、ユーザー ID がキューの削除を許可されていることについての検査は行われます。

**トピックに対するサブスクライブ (MQSUB)**
アプリケーションがトピックをサブスクライブする際、アプリケーションは、実行する必要がある操作のタイプを指定します。 新しいサブスクリプションを作成するか、既存のサブスクリプションを変更するか、既存のサブスクリプションを変更なしで再開するかのいずれかになります。 それぞれのタイプの操作についてキュー・マネージャーは、その操作を実行するための権限が、アプリケーションに関連付けられたユーザー ID に付与されていることを確認します。

アプリケーションがトピックをサブスクライブすると、トピック・ツリーのうちアプリケーションがサブスクライブした位置か、それより上で検出されたトピック・オブジェクトに対して権限検査が実行されます。 権限検査には、複数のトピック・オブジェクトに対する検査が関係する場合があります。

キュー・マネージャーが権限検査に使用するユーザー ID は、アプリケーションがキュー・マネージャーに接続されるときに、オペレーティング・システムから取得されるユーザー ID です。

キュー・マネージャーは、サブスクライバーのキューに対して権限検査を 実行しますが、管理対象キューに対しては実行しません。

## UNIX, Linux, and Windows システム上の IBM WebSphere MQ によってアクセス制御が実装される方法

IBM WebSphere MQ では、オブジェクト権限マネージャーから、基盤オペレーティング・システムに用意されているセキュリティー・サービスを利用します。 IBM WebSphere MQ には、アクセス制御リストの作成と保守のためのコマンドが用意されています。

Authorization Service Interface というアクセス制御インターフェースは、WebSphere MQ の一部です。 WebSphere MQ には、オブジェクト権限マネージャー *(OAM)* として知られている、アクセス制御マネージャー (Authorization Service Interface に準拠した) が実装されています。 これは (167 ページの『UNIX, Linux, and Windows システムでのセキュリティー・アクセス検査の抑止』で説明されているように) 特別の指定をしない限り自動的にインストールされ、作成されるキュー・マネージャーごとに使用可能になりま

す。 OAM は、Authorization Service Interface に準拠した任意のユーザー作成コンポーネント、またはベンダー作成コンポーネントで置き換えることができます。

OAM は、オペレーティング・システムのユーザー ID とグループ ID を使用し、基礎となるオペレーティング・システムのセキュリティー機能を利用します。 ユーザーは、正しい権限を持っている場合にのみ、WebSphere MQ オブジェクトにアクセスできます。 159 ページの『UNIX、Linux および Windows システムでの OAM によるオブジェクトへのアクセス制御』には、この権限を付与したり取り消したりする方法が説明されています。

OAM は、制御するリソースごとに、アクセス制御リスト (ACL) を保守します。 許可データは、SYSTEM.AUTH.DATA.QUEUE というローカル・キューに保管されます。 このキューへのアクセスは mqm グループのユーザーに制限されます。Windows の場合は、さらに Administrators グループのユーザー、および SYSTEM ID でログインしたユーザーもアクセスできます。 このキューへのユーザー・アクセス権は変更できません。

WebSphere MQ には、アクセス制御リストの作成と保守のためのコマンドが用意されています。 これらのコマンドの詳細については、159 ページの『UNIX、Linux および Windows システムでの OAM によるオブジェクトへのアクセス制御』を参照してください。

WebSphere MQ は、OAM にプリンシパル、リソース名、およびアクセス・タイプから成る要求を渡します。 OAM は、保守する ACL に基づいてアクセスを付与したり拒否したりします。 WebSphere MQ は、OAM の決定に従います。 OAM が決定できない場合は、WebSphere MQ はアクセスを許可しません。

## UNIX, Linux, and Windows システムでのユーザー ID の識別

オブジェクト権限マネージャーは、リソースへのアクセスを要求しているプリンシパルを確認します。 プリンシパルとして使用されるユーザー ID は、コンテキストによって異なります。

オブジェクト権限マネージャー (OAM) は、特定のリソースへのアクセスを要求しているユーザーを確認できなければなりません。 IBM WebSphere MQ では、この ID を指すときにプリンシパル という用語を使用します。 プリンシパルは、アプリケーションが最初にキュー・マネージャーに接続するときに確立されます。これは、接続アプリケーションに関連付けられたユーザー ID に基づき、キュー・マネージャー側で決定されます。 (アプリケーションがキュー・マネージャーに接続しないで XA 呼び出しを発行する場合、キュー・マネージャーによる権限検査には、xa_open 呼び出しを発行するアプリケーションに関連付けられたユーザー ID が使用されます。)

UNIX and Linux システムでは、権限付与ルーチンによって、実 (ログイン) ユーザー ID かアプリケーションに関連する有効ユーザー ID のどちらかが検査されます。 検査の対象になるユーザー ID は、バインド・タイプによって異なる場合もあります。詳細については、『インストール可能サービス』を参照してください。

IBM WebSphere MQ は、システムから受け取ったユーザー ID を、ユーザーを識別するものとして、各メッセージのメッセージ・ヘッダー (MQMD 構造) に伝搬します。 この ID は、メッセージ・コンテキスト情報の一部で、203 ページの『UNIX、Linux および Windows システムでのコンテキスト権限』で説明されています。 アプリケーションがコンテキスト情報の変更を許可されていない限り、そのアプリケーションでこの情報を変更することはできません。

### プリンシパルおよびグループ

プリンシパルは、グループに属します。 特定のリソースに対するアクセス権を、個人ではなくグループに付与することにより、必要とされる管理作業の量を減らすことができます。 UNIX and Linux システムでは、すべてのアクセス制御リスト (ACL) はグループに基づいていますが、 Windows システムでは、ACLS はユーザー ID とグループに基づいています。

例えば、特定のアプリケーションの実行を希望するユーザーからなるグループを定義できます。 他のユーザーの場合、そのユーザー ID を該当するグループに追加することで、必要とするすべてのリソースに対するアクセス権を付与できます。 この処理については、以下の箇所で説明されています。

- Windows でのグループの作成と管理
- HP-UX でのグループの作成と管理
- AIX でのグループの作成と管理
- Solaris でのグループの作成と管理

- Linux

プリンシパルは、複数のグループ (プリンシパルのグループ・セット) に属することができます。 グループ・セット内の各グループに付与された権限をすべて集めた権限を持ちます。 これらの権限はキャッシュに入れられるため、プリンシパルのグループ・メンバーシップに変更を加えても、MQSC コマンド REFRESH SECURITY (または PCF でこれに相当するコマンド) を発行しない限り、 キュー・マネージャーが再始動するまで認識されません。

**UNIX and Linux システム**

ACL はすべて、グループに基づきます。 特定のリソースに対するアクセス権がユーザーに与えられた場合、そのユーザー ID の 1 次グループが ACL に組み込まれます。 個別のユーザー ID は組み込まれず、そのグループのすべてのメンバーに権限が与えられます。 このため、同じグループ内の別のプリンシパルの権限を変更することにより、特定のプリンシパルの権限をうっかり変更してしまうことがないよう注意が必要です。 すべてのユーザーは、名目上はデフォルトのユーザー・グループ *nobody* に割り当てられ、このグループには権限は与えられていません。 この *nobody* グループの権限を変更することにより、特定の権限を除き、ユーザーに WebSphere MQ リソースへのアクセス権を付与できます。

ユーザー ID を値 "UNKNOWN" で定義しないでください。 "UNKNOWN" は、ユーザー ID が長すぎる場合に使用される値であり、不特定のユーザー ID が UNKNOWN のアクセス権限を使用してしまう結果になります。

ユーザー ID には 12 文字まで、またグループ名にも 12 文字までを含めることができます。

**Windows システム**

ACL は、ユーザー ID とグループの両方に基づきます。 検査は UNIX システムの場合と同じですが、ACL に個々のユーザー ID も示されることがある点が異なります。 同じユーザー ID を使って別々のドメインに別々のユーザーを持つことができます。 WebSphere MQ では、ユーザー ID をドメイン・ネームで修飾して、これらのユーザーに異なるレベルのアクセス権限を付与することができます。

グループ名には、次の形式で指定されたドメイン・ネームをオプションで含めることができます。

```
GroupName@domain
domain\GroupName
```

以下の 2 つのケースに限り、OAM によってグローバル・グループが検査されます。

1. キュー・マネージャーのセキュリティー・スタンザに `GroupModel=GlobalGroups` という設定が含まれる。セキュリティーを参照してください。
2. キュー・マネージャーが代替セキュリティー・アクセス・グループを使用している。**crtmqm** を参照してください。

ユーザー ID には 20 文字まで、ドメイン・ネームには 15 文字まで、グループ名には 64 文字まで含められます。

OAM は、まずローカル・セキュリティー・データベースを検査し、次に 1 次ドメインのデータベース、そして最後に信頼されたドメインのデータベースを検査します。 検査のために、最初に検出されるユーザー ID が OAM によって使用されます。 それぞれのユーザー ID は、特定のコンピューター上で別のグループ・メンバーシップを持つ可能性があります。

制御コマンド (例えば、`crtmqm`) の中には、オブジェクト権限マネージャー (OAM) を使用して、WebSphere MQ オブジェクト上で権限を変更するものがあります。 OAM は上記の段落に示された順序でセキュリティー・データベースを検索して、特定のユーザー ID の権限を判別します。 このため、OAM によって判別された権限が、ローカル mqm グループのメンバーとして特定のユーザー ID に与えられるはずの権限をオーバーライドすることがあります。 例えば、グローバル・グループを通じてローカル mqm グループのメンバーになっている、ドメイン・コントローラーによって認証されるユーザー ID から `crtmqm` コマンドを発行する場合、ローカル mqm グループに含まれない同じ名前のローカル・ユーザーがシステムに存在するならば、そのコマンドは失敗します。

## *Windows* セキュリティー *ID (SID)*

WebSphere MQ on Windows は、使用可能な場合は SID を使用します。 許可要求で Windows SID が指定されていなければ、WebSphere MQ は、ユーザー名だけに基づいてユーザーを識別しますが、その場合は、間違った権限が与えられる可能性があります。

Windows システムでは、ユーザー ID を補足するためにセキュリティー ID (SID) が使用されます。 SID には、ユーザーが定義される Windows セキュリティー・アカウント・マネージャー (SAM) データベース上の完全なユーザー・アカウント詳細を識別する情報が入っています。 メッセージが WebSphere MQ for Windows に作成される場合、WebSphere MQ はメッセージ記述子に SID を保管します。 WebSphere MQ on Windows が許可検査を実行する場合、SID を使用して SAM データベースから全情報を照会します。 (この照会が正常に終了するためには、ユーザーの定義を格納する SAM データベースにアクセスできることが必要です。)

デフォルトでは、許可要求に Windows SID が指定されていない場合は、WebSphere MQ は、そのユーザー名だけに基づいてユーザーを識別します。 このときに、セキュリティー・データベースを以下の順序で検索します。

1. ローカル・セキュリティー・データベース
2. 1 次ドメインのセキュリティー・データベース
3. 信頼されたドメインのセキュリティー・データベース

ユーザー名が固有でない場合、正しくない WebSphere MQ 権限が付与される可能性があります。 この問題を避けるには、各許可要求に SID を組み入れます。 この SID は、ユーザーの資格情報を確立するために WebSphere MQ によって使用されます。

すべての許可要求に SID を含めることを指定するには、`regedit` を使用します。 SecurityPolicy を NTSIDsRequired に設定します。

## UNIX、Linux および Windows システムでの代替ユーザー権限

1 つのユーザー ID で WebSphere MQ オブジェクトへのアクセス時に、別のユーザーの権限を使用できると指定することができます。 このことを代替ユーザー権限 といい、どのような WebSphere MQ オブジェクトに対しても使用できます。

代替ユーザー権限は、サーバーがプログラムから要求を受け取り、その要求に対して必要な権限をプログラムが確実に持つようにしたい場合に重要です。 サーバーは、要求に必要な権限があっても、要求したアクションに関する権限がプログラムにあるかどうかを確認する必要があります。

例えば、ユーザー ID PAYSERV のもとで実行中のサーバー・プログラムが、キューから要求メッセージを取り出したとします。この要求メッセージは、ユーザー ID USER1 によってキューに置かれたものです。 サーバー・プログラムは、要求メッセージを読み取ると、要求を処理し、要求メッセージで指定されている応答先キューに応答を書き戻します。 サーバーは、サーバーのユーザー ID (PAYSERV) を使用して 応答先キューのオープンを許可する代わりに、別のユーザー ID (この場合は USER1) を指定することができます。この例では、PAYSERV が応答先キューをオープンするときに 代替ユーザー ID として USER1 を指定できるかどうかを制御するために、代替ユーザー権限を使用することができます。

代替ユーザー ID は、オブジェクト記述子の **AlternateUserId** フィールドに 指定します。

## UNIX、Linux および Windows システムでのコンテキスト権限

コンテキストは、特定のメッセージに適用される情報であって、メッセージの一部であるメッセージ記述子 MQMD に含まれています。 アプリケーションは、MQOPEN 呼び出しまたは MQPUT 呼び出しのいずれかを出すときにコンテキスト・データを指定することができます。

コンテキスト情報は、以下の 2 つのセクションから構成されます。

**ID セクション**
メッセージの発信者。 これは、UserIdentifier、AccountingToken、および ApplIdentityData フィールドで構成されます。

**起点セクション**
メッセージの発信元およびキューに書き込まれた日時。 これは、PutApplType、PutApplName、PutDate、PutTime、および ApplOriginData フィールドで構成されます。

アプリケーションは、MQOPEN 呼び出しまたは MQPUT 呼び出しのいずれかを出すときにコンテキスト・データを指定することができます。 このデータは、アプリケーションによって生成されたり、別のメッセージから渡されたり、デフォルトでキュー・マネージャーによって生成されたりします。 例えば、コンテキ

スト・データはサーバー・プログラムによって、要求側の ID の検査、メッセージの発信元が許可ユーザー ID のもとで実行中のアプリケーションであるかどうかのテストに使用されることがあります。

サーバー・プログラムは、UserIdentifier を使用して、代替ユーザーのユーザー ID を判別することができます。 コンテキスト許可は、ユーザーが MQOPEN 呼び出しまたは MQPUT1 呼び出しにコンテキスト・オプションを使用できるかどうかを制御するのに使用できます。

コンテキスト・オプションについては、コンテキスト情報の制御 を、コンテキストに関連するメッセージ記述子フィールドの説明については MQMD の概要 を参照してください。

# セキュリティー出口によるアクセス制御の実装

MCAUserIdentifier またはオブジェクト権限マネージャーを使用して、セキュリティー出口でアクセス制御を実装できます。

## MCAUserIdentifier

カレントであるチャネルのどのインスタンスにも、チャネル定義構造 MQCD が関連付けられています。 MQCD 内のフィールドの初期値は、WebSphere MQ 管理者によって作成されるチャネル定義によって決まります。 特に、フィールドの 1 つである *MCAUserIdentifier* の初期値は、DEFINE CHANNEL コマンドの MCAUSER パラメーターの値、またはチャネル定義が別の方法で作成されている場合は、MCAUSER と等価の値によって決まります。 *MCAUserIdentifier* には、MCA ユーザー ID の最初の 12 バイトが入ります。 MCA ユーザー ID がブランクでない場合、これは、メッセージ・チャネル・エージェントが MQ リソースへのアクセス許可を受けるときに使用するユーザー ID を示します。 Windows プラットフォームでは、この MCAUSER は 12 文字未満にしてください。

MQCD 構造は、チャネル出口プログラムが MCA によって呼び出されるときに、チャネル出口プログラムに渡されます。 セキュリティー出口が MCA によって呼び出されると、そのセキュリティー出口は、*MCAUserIdentifier* の値を変更して、チャネル定義で指定された任意の値を置き換えることができます。

IBM i、UNIX、Linux および Windows システムでは、*MCAUserIdentifier* の値がブランクでない限り、MCA がキュー・マネージャーへの接続後にキュー・マネージャーのリソースにアクセスしようとするときに、キュー・マネージャーは権限検査のユーザー ID として *MCAUserIdentifier* の値を使用します。 *MCAUserIdentifier* の値がブランクである場合、キュー・マネージャーは、代わりに MCA のデフォルト・ユーザー ID を使用します。 このことは RCVR、RQSTR、CLUSRCVR および SVRCONN チャネルに当てはまります。 送信側 MCA の場合は、*MCAUserIdentifier* の値がブランクでない場合でも、権限検査には常にデフォルトのユーザー ID を使用します。

z/OS 上では、キュー・マネージャーは、*MCAUserIdentifier* の値がブランクでない場合、その値を権限検査に使用することができます。 受信側 MCA とサーバー接続 MCA の場合、キュー・マネージャーが権限検査に *MCAUserIdentifier* の値を使用するかどうかは、次に挙げるものによって決まります。

- チャネル定義内の PUTAUT パラメーターの値
- 検査に使用される RACF プロファイル
- RESLEVEL プロファイルに対する、チャネル・イニシエーター・アドレス・スペース・ユーザー ID のアクセス・レベル

送信側 MCA の場合、次のものによって決まります。

- 送信側 MCA が呼び出し側であるか、応答側であるか
- RESLEVEL プロファイルに対する、チャネル・イニシエーター・アドレス・スペース・ユーザー ID のアクセス・レベル

セキュリティー出口が *MCAUserIdentifier* に保管するユーザー ID を取得する方法には、さまざまな方法があります。 例えば、次のとおりです。

- MQI チャネルのクライアント側にセキュリティー出口がない場合、WebSphere MQ クライアント・アプリケーションに関連したユーザー ID は、クライアント・アプリケーションが MQCONN 呼び出しを発行すると、クライアント接続 MCA からサーバー接続 MCA に流れます。サーバー接続 MCA は、チャネル定義構造 MQCD の *RemoteUserIdentifier* フィールドにこのユーザー ID を保管します。 *MCAUserIdentifier* の値がこの時点でブランクである場合、MCA は *MCAUserIdentifier* に同じユーザー ID を保管します。

MCA が *MCAUserIdentifier* にユーザー ID を格納しない場合は、後からセキュリティー出口で *MCAUserIdentifier* を *RemoteUserIdentifier* の値に設定することによって、ユーザー ID を格納できます。

クライアント・システムから流れるユーザー ID が、新しいセキュリティー・ドメインに入り、サーバー・システム上で無効である場合、セキュリティー出口は、このユーザー ID を有効なユーザー ID で置き換え、置き換えられたユーザー ID を *MCAUserIdentifier* に保管することができます。

- ユーザー ID は、相手側のセキュリティー出口によってセキュリティー・メッセージ内で送信することができます。

  メッセージ・チャネル上で、送信側 MCA によって呼び出されるセキュリティー出口は、送信側 MCA が稼働するときに使用しているユーザー ID を送信することができます。 その後、受信側 MCA によって呼び出されるセキュリティー出口は、このユーザー ID を *MCAUserIdentifier* に保管することができます。 同様に、MQI チャネル上では、チャネルのクライアント側にあるセキュリティー出口は、WebSphere MQ MQI クライアント・アプリケーションに関連したユーザー ID を送信することができます。 次に、チャネルのサーバー側にあるセキュリティー出口は、このユーザー ID を *MCAUserIdentifier* に保管することができます。 上記の例のように、ユーザー ID が、ターゲット・システム上で無効である場合、セキュリティー出口は、このユーザー ID を有効なユーザー ID で置き換え、置き換えられたユーザー ID を *MCAUserIdentifier* に保管することができます。

  識別と認証サービスの一部としてデジタル証明書が受信される場合、セキュリティー出口は、証明書内の識別名を、ターゲット・システム上で有効なユーザー ID にマップすることができます。 次に、そのユーザー ID を *MCAUserIdentifier* に保管することができます。
- SSL がチャネルで使用されている場合は、相手側の識別名 (DN) が MQCD 内の SSLPeerNamePtr フィールドの出口に渡され、その証明書の発行者の DN が MQCXP の SSLRemCertIssNamePtr フィールド内の出口に渡されます。

*MCAUserIdentifier* フィールド、チャネル定義構造 MQCD、チャネル出口パラメーター構造 MQCXP の詳細については、チャネル出口呼び出しおよびデータ構造体を参照してください。 クライアント・システムから MQI チャネルに流れるユーザー ID の詳細については、『アクセス制御』を参照してください。

**注 :** WebSphere MQ v7.1 のリリースより前に構成されたセキュリティー出口アプリケーションは、更新が必要になる場合があります。 詳しくは、チャネル・セキュリティー出口プログラムを参照してください。

### WebSphere MQ オブジェクト権限マネージャーのユーザー認証

WebSphere MQ MQI クライアント接続で、セキュリティー出口を使用して オブジェクト権限マネージャー (OAM) のユーザー認証で使用される MQCSP 構造を変更または作成することができます。 『メッセージング・チャネルのためのチャネル出口プログラム』を参照してください。

## メッセージ出口によるアクセス制御の実装

メッセージ出口を使用して、1 つのユーザー ID を別のユーザー ID に置き換えなければならない場合もあります。

メッセージをサーバー・アプリケーションに送信するクライアント・アプリケーションについて考えてみましょう。 サーバー・アプリケーションは、メッセージ記述子内の *UserIdentifier* フィールドからユーザー ID を取り出すことができます。また、代替ユーザー権限を持つ場合は、クライアントに代わって WebSphere MQ リソースにアクセスするときに、このユーザー ID を権限検査に使用するように、キュー・マネージャーに依頼することができます。

チャネル定義で PUTAUT パラメーターが CTX (または、z/OS 上では ALTMCA) に設定されている場合、MCA が宛先キューを開くときに、各着信メッセージの *UserIdentifier* フィールド内のユーザー ID が、権限検査に使用されます。

ある種の状況のもとでは、レポート・メッセージが生成されると、そのメッセージは、レポートの原因であるメッセージの *UserIdentifier* フィールド内のユーザー ID の権限を使用して書き込まれます。 特に、送達後確認 (COD) レポートと有効期限レポートは、常にこの権限を使用して書き込まれます。

こうした状況があるので、メッセージが新しいセキュリティー・ドメインに入るときに、*UserIdentifier* フィールドで、ユーザー ID を別のユーザー ID に置き換える必要がある場合があります。 この置き換えは、チャネルの受信側のメッセージ出口によって行うことができます。 あるいは、着信メッセージの

*UserIdentifier* フィールド内のユーザー ID が、新しいセキュリティー・ドメインで定義されるようにすることもできます。

着信メッセージに、そのメッセージを送信したアプリケーションのユーザー用のデジタル証明書が入っている場合、メッセージ出口は、その証明書を検証し、証明書内の識別名を、受信システム上で有効なユーザー ID にマップすることができます。 その後、メッセージ出口は、メッセージ記述子内の *UserIdentifier* フィールドをこのユーザー ID に設定することができます。

メッセージ出口が、着信メッセージ内の *UserIdentifier* フィールドの値を変更する必要がある場合、メッセージ出口が、メッセージの送信側を同時に認証することが妥当である場合があります。 詳細については、149 ページの『メッセージ出口による識別マッピング』を参照してください。

## API 出口と API 交差出口によるアクセス制御の実装

API 出口または API 交差出口を使用すれば、WebSphere MQ に組み込まれているアクセス制御機能を補足する機能を提供できます。 特に、その出口では、メッセージ・レベルでアクセス制御機能を用意できます。つまり、その出口によって、アプリケーションが一定の基準を満たすメッセージだけをキューに書き込んだり、キューから取得したりするように設定できるということです。

次の例を検討してください。

- メッセージには、注文についての情報が入っているとします。 アプリケーションがキューにメッセージを書き込もうとするときに、API 出口または API 交差出口によって、その注文の合計値が一定の限界値未満であるかどうかを検査できます。
- メッセージが、リモート・キュー・マネージャーから宛先キューに着信するとします。 アプリケーションがキューからメッセージを取得しようとするときに、API 出口または API 交差出口によって、そのメッセージの送信側がそのキューにメッセージを送信する権限を持っているかどうかを検査できます。

# メッセージの機密性

機密性を維持するには、メッセージを暗号化します。 WebSphere MQ では、ユーザーのニーズに応じた、メッセージのさまざまな暗号化方法が用意されています。

どの CipherSpec を選択するかによって適用される機密性のレベルが決まります。

Point-to-Point メッセージング・インフラストラクチャー向けに、アプリケーション・レベルのエンドツーエンド・データ保護が必要な場合、WebSphere MQ Advanced Message Security を使用して、メッセージを暗号化するか、あるいは独自の API 出口または API 交差出口を作成することができます。

チャネルを介したメッセージのトランスポート中に、メッセージのみを暗号化する必要がある場合、キュー・マネージャーに対する適切なセキュリティーがあるので、SSL または TLS を使用するか、独自のセキュリティー出口やメッセージ出口を作成するか、あるいは出口プログラムを送受信することができます。

WebSphere MQ Advanced Message Security の詳細については、63 ページの『計画 Advanced Message Security』を参照してください。WebSphere MQ での SSL および TLS の使用については、23 ページの『IBM WebSphere MQ での SSL および TLS のサポート』で説明されています。 メッセージ暗号化での出口プログラムの使用法については、226 ページの『ユーザー出口プログラムでの機密性の実装』で説明されています。

## SSL または TLS による 2 つのキュー・マネージャーの接続

SSL または TLS 暗号セキュリティー・プロトコルを使用するセキュア通信では、 通信チャネルをセットアップし、認証に使用するデジタル証明書を管理する必要があります。

SSL または TLS インストール環境をセットアップするには、SSL または TLS を使用するようにチャネルを定義する必要があります。 また、デジタル証明書を取得し、管理することも必要です。 テスト・システムでは、自己署名証明書か、ローカル認証局 (CA) で発行された証明書を使用できます。 実動システムでは、自己署名証明書を使用しないでください。 細については、 ..//zs14140_.dita を参照してください。

証明書の作成と管理の詳細については、114 ページの『UNIX, Linux, and Windows システムでの SSL または TLS の取り扱い』を参照してください。

このトピック集では、SSL 通信のセットアップに関連したタスクを取り上げ、それらのタスクを実行するための段階的な手順を説明します。

また、プロトコルのオプション部分である SSL または TLS クライアント認証をテストすることもできます。SSL または TLS ハンドシェーク中に、SSL または TLS クライアントは常にサーバーからデジタル証明書を取得し検証します。 WebSphere MQ の実装では、SSL または TLS サーバーは、常にクライアントから証明書を要求します。

**注 :**

1. この状況では、SSL クライアントはハンドシェークを開始する接続を参照します。
2. 詳細については、用語集を参照してください。

UNIX、Linux、および Windows システムでは、SSL または TLS クライアントは、正しい WebSphere MQ 形式 (`ibmwebspheremq` の後に続けて、小文字にされたキュー・マネージャー名) のラベルが付いた証明書が存在する場合に限って証明書を送信します。 例えば、`QM1` の場合は、`ibmwebspheremqqm1` です。

WebSphere MQ は、他の製品の証明書との混同を避けるために、`ibmwebspheremq` という接頭部をラベルに付けます。 証明書ラベル全体を小文字で指定してください。

SSL または TLS サーバーは、クライアント証明書が送信される場合は、常にそのクライアント証明書を検証します。 クライアントが証明書を送信しない場合に認証が失敗するのは、チャネルの SSL または TLS サーバー側の定義で、SSLCAUTH パラメーターが REQUIRED に設定されている場合、または SSLPEER パラメーター値が設定されている場合に限られます。 匿名によるキュー・マネージャーへの接続 (つまり、SSL クライアントまたは TLS クライアントから証明書を送信しない場合) について詳しくは、211 ページの『片方向認証による 2 つのキュー・マネージャーの接続』を参照してください。

## 2 つのキュー・マネージャーの相互認証への自己署名証明書の使用

自己署名 SSL または TLS 証明書を使用して 2 つのキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

### このタスクについて

シナリオ

- 安全に通信する必要のあるキュー・マネージャーが 2 つ (QM1 と QM2) あります。 QM1 と QM2 の間で、相互認証を実行する必要があります。
- そこで、自己署名証明書を使用して安全な通信をテストすることにしました。

構成の結果は次のようになります。

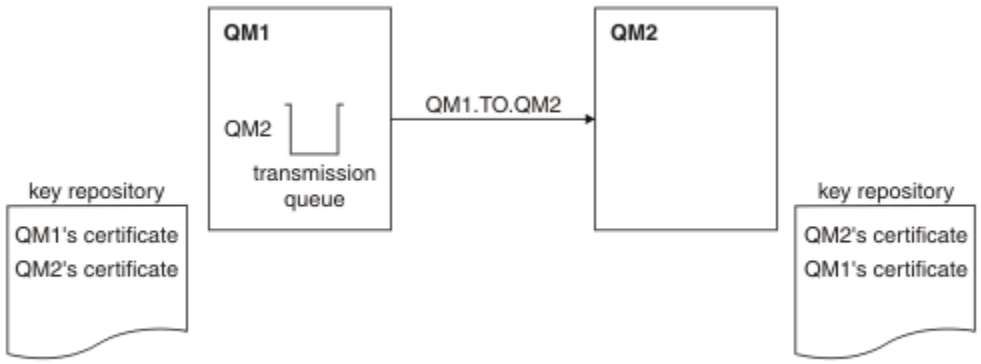



図 *14.* このタスクで実装する構成

207 ページの図 14 にあるとおり、QM1 の鍵リポジトリーには QM1 の証明書と QM2 の公開証明書を格納します。 QM2 の鍵リポジトリーには、QM2 の証明書と QM1 の公開証明書を格納します。

## 手順

1. オペレーティング・システムに従って、各キュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。
2. キュー・マネージャーごとに自己署名証明書を作成します。
   - UNIX、Linux、および Windows システムの場合。
3. 各証明書のコピーを取り出します。
   - UNIX、Linux、および Windows システムの場合。
4. FTP などのユーティリティーを使用して QM1 証明書の公開部分を QM2 システムに転送し、またその逆を実行します。
5. キュー・マネージャーごとにパートナーの証明書を鍵リポジトリーに追加します。
   - UNIX、Linux、および Windows システムの場合。
6. QM1 で以下の例のようなコマンドを実行して、送信側チャネルとそれに関連する伝送キューを定義します。

   ```
   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)
   ```

   この例では、CipherSpec RC4_MD5 を使用します。 チャネルの両端の CipherSpec は同じでなければなりません。
7. QM2 で以下の例のようなコマンドを実行して、受信側チャネルを定義します。

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   このチャネルの名前は、ステップ 6 で定義した送信側チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。
8. チャネルを開始します。

## タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、207 ページの図 14 です。

## 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

キュー・マネージャー QM2 から以下のコマンドを入力します。

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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( )
```

いずれの場合も、SSLPEER の値は、ステップ 2 で作成したパートナー証明書の DN の値と一致している必要があります。 この証明書は自己署名証明書なので、発行者の名前はピアの名前と一致しています。

SSLPEER はオプションです。 これを指定する場合は、パートナー証明書 (ステップ 2 で作成された) の DN が許可されるように値を設定する必要があります。 SSLPEER の使用について詳しくは、「WebSphere MQ rules for SSLPEER values」を参照してください。

## 2 つのキュー・マネージャーの相互認証への CA 署名証明書の使用

CA 署名 SSL または TLS 証明書を使用して 2 つのキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

### このタスクについて

シナリオ

- 安全に通信する必要のあるキュー・マネージャーが 2 つ (QMA と QMB) あります。 QMA と QMB の間で、相互認証を実行する必要があります。
- 将来的にはこのネットワークを実稼働環境で使用することを計画しているので、最初から CA 署名証明書を使用することにしました。

構成の結果は次のようになります。

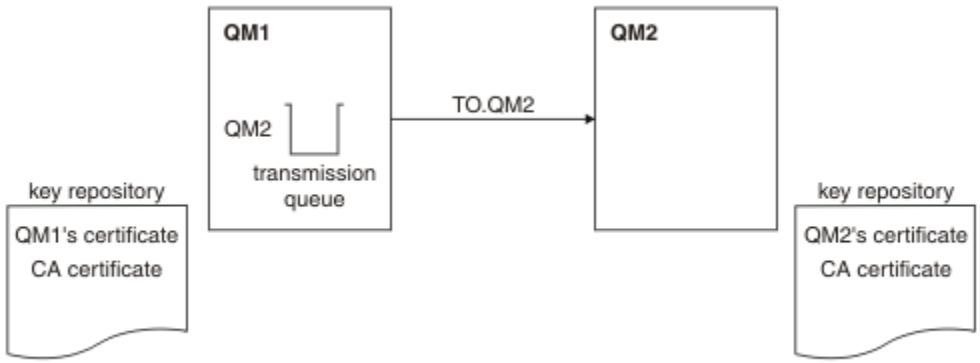



図 *15.* このタスクで実装する構成

209 ページの図 15 にあるとおり、QMA の鍵リポジトリーには、QMA の証明書と CA 証明書を格納します。 QMB の鍵リポジトリーには、QMB の証明書と CA 証明書を格納します。 この例では、QMA の証明書と QMB の証明書の両方が同じ CA から発行されました。 QMA の証明書と QMB の証明書が異なる CA から発行された場合は、QMA と QMB の鍵リポジトリーに両方の CA 証明書が含まれている必要があります。

### 手順

1. オペレーティング・システムに従って、各キュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。

2. キュー・マネージャーごとに CA 署名証明書を要求します。
   2 つのキュー・マネージャーで異なる CA を使用することができます。
   - UNIX、Linux、および Windows システムの場合。
3. キュー・マネージャーごとに認証局証明書を鍵リポジトリーに追加します。
   これらのキュー・マネージャーがそれぞれ異なる認証局を使用する場合は、各認証局の CA 証明書を両方の鍵リポジトリーに追加する必要があります。
   - UNIX、Linux、および Windows システムの場合。
4. キュー・マネージャーごとに CA 署名証明書を鍵リポジトリーに追加します。
   - UNIX、Linux、および Windows システムの場合。
5. QMA で以下の例のようなコマンドを発行して、送信側チャネルとそれに関連する伝送キューを定義します。

```
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)
```

   この例では、CipherSpec RC4_MD5 を使用します。 チャネルの両端の CipherSpec は同じでなければなりません。
6. QMB で以下の例のようなコマンドを発行して、受信側チャネルを定義します。

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
```

   このチャネルの名前は、ステップ 6 で定義した送信側チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。
7. チャネルを開始します。

## タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、209 ページの図 15 です。

## 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QMA から以下のコマンドを入力します。

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

キュー・マネージャー QMB から以下のコマンドを入力します。

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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( )
```

いずれの場合も、SSLPEER の値は、ステップ 2 で作成したパートナー証明書の識別名 (DN) の値と一致している必要があります。 発行者名は、ステップ 4 で追加された個人証明書に署名した CA 証明書の識別名と一致します。

## 片方向認証による 2 つのキュー・マネージャーの接続

相互認証を使用するシステムを変更して、キュー・マネージャーが片方向認証で別のキュー・マネージャーに接続できるようにするには (つまり、SSL クライアントまたは TLS クライアントが証明書を送信しない場合)、以下のサンプル手順に従ってください。

### このタスクについて

シナリオ

- 209 ページの『2 つのキュー・マネージャーの相互認証への CA 署名証明書の使用』 で、2 つのキュー・マネージャー (QM1 と QM2) がセットアップされました。
- 片方向認証を使用して QM2 に接続されるように QM1 を変更する必要があるとします。

構成の結果は次のようになります。

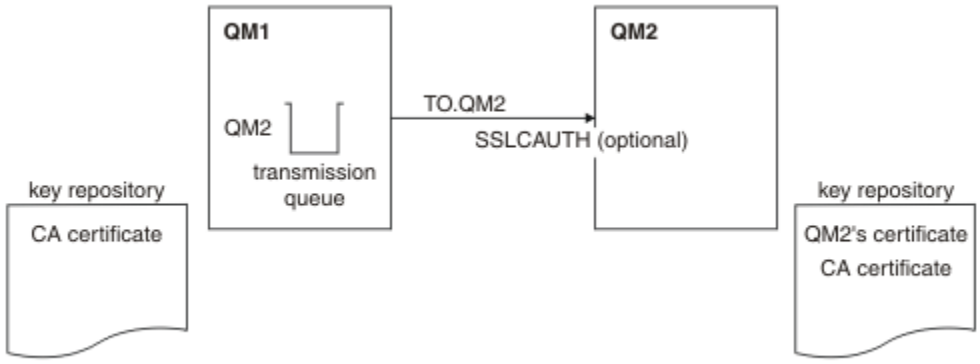



図 *16.* 片方向認証を許可するキュー・マネージャー

### 手順

1. オペレーティング・システムに従って、QM1 の個人証明書を鍵リポジトリーから削除します。
   - UNIX、Linux、および Windows システムの場合。 この証明書には、以下のようなラベルが付いています。
     - `ibmwebspheremq` の後に、小文字に変換されたキュー・マネージャーの名前が続きます。 例えば、`QM1` の場合は、`ibmwebspheremqqm1` です。
2. オプション: QM1 で、SSL または TLS チャネルが以前に実行されている場合は、の説明に従って、SSL または TLS 環境をリフレッシュします。
3. 受信側で匿名接続を許可します。

### タスクの結果

変更した鍵リポジトリーとチャネルを図にまとめたのが、211 ページの図 16 です。

### 次のタスク

送信側チャネルの稼働中に REFRESH SECURITY TYPE(SSL) コマンドを実行した場合は (手順 2)、チャネルが自動的に再始動されます。 送信側チャネルが稼働していなかった場合は、送信側チャネルを始動します。

チャネルのサーバー側で、チャネル状況表示にピア名パラメーター値が表示される場合は、クライアント証明書が流れたことを意味します。

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

QM1 キュー・マネージャーから、次のコマンドを入力します。

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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 キュー・マネージャーから、次のコマンドを入力します。

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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( )
```

QM2 で SSLPEER フィールドが空になっています。これは、QM1 が証明書を送信しなかったことを示しています。 QM1 では、SSLPEER の値が QM2 の個人証明書の DN の値と一致しています。

## キュー・マネージャーへのクライアントのセキュア接続

SSL または TLS 暗号セキュリティー・プロトコルを使用するセキュア通信では、 通信チャネルをセットアップし、認証に使用するデジタル証明書を管理する必要があります。

SSL または TLS インストール環境をセットアップするには、SSL または TLS を使用するようにチャネルを定義する必要があります。 また、デジタル証明書を取得し、管理することも必要です。 テスト・システムでは、自己署名証明書か、ローカル認証局 (CA) で発行された証明書を使用できます。 実動システムでは、自己署名証明書を使用しないでください。 細については、 ..//zs14140_.dita を参照してください。

証明書の作成と管理の詳細については、114 ページの『UNIX, Linux, and Windows システムでの SSL または TLS の取り扱い』を参照してください。

このトピック集では、SSL 通信のセットアップに関連したタスクを取り上げ、それらのタスクを実行するための段階的な手順を説明します。

また、プロトコルのオプション部分である SSL または TLS クライアント認証をテストすることもできます。SSL または TLS ハンドシェーク中に、SSL または TLS クライアントは常にサーバーからデジタル証明書を取得し検証します。 WebSphere MQ の実装では、SSL または TLS サーバーは、常にクライアントから証明書を要求します。

UNIX, Linux, and Windows システムでは、SSL または TLS クライアントは、正しい WebSphere MQ 形式 (`ibmwebspheremq` の後に続けて、小文字にされたログオン・ユーザー ID。例えば、`ibmwebspheremqmyuserid`) のラベルが付いた証明書が存在する場合に限って証明書を送信します。

WebSphere MQ は、他の製品の証明書との混同を避けるために、`ibmwebspheremq` という接頭部をラベルに付けます。 証明書ラベル全体を小文字で指定してください。

SSL または TLS サーバーは、クライアント証明書が送信される場合は、常にそのクライアント証明書を検証します。 クライアントが証明書を送信しない場合に認証が失敗するのは、チャネルの SSL または TLS サーバー側の定義で、SSLCAUTH パラメーターが REQUIRED に設定されている場合、または SSLPEER パラメーター値が設定されている場合に限られます。 匿名でのキュー・マネージャーの接続について詳しくは、216 ページの『キュー・マネージャーへのクライアントの匿名接続』を参照してください。

## クライアントとキュー・マネージャーの相互認証への自己署名証明書の使用

自己署名 SSL または TLS 証明書を使用してクライアントとキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

### このタスクについて

シナリオ

- クライアント C1、およびキュー・マネージャー QM1 が存在し、これらは安全に通信する必要があります。 C1 と QM1 の間で、相互認証を実行する必要があります。
- そこで、自己署名証明書を使用して安全な通信をテストすることにしました。

IBM i での DCM は自己署名証明書をサポートしていないため、このタスクは IBM i システムでは適用されません。

構成の結果は次のようになります。

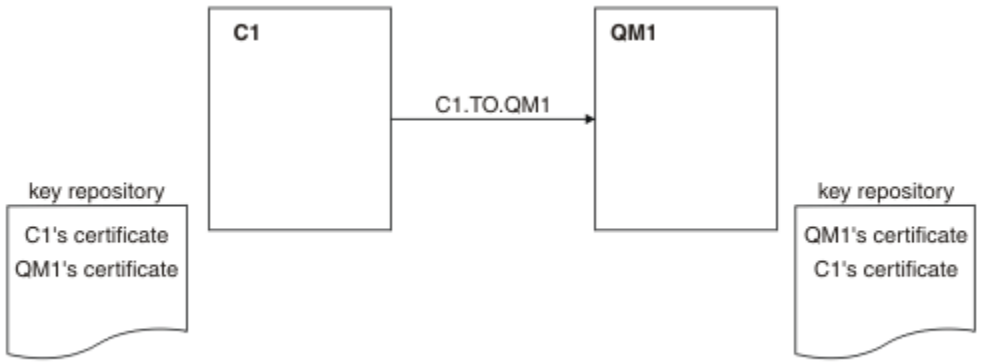



図 *17.* このタスクで実装する構成

213 ページの図 17 にあるとおり、QM1 の鍵リポジトリーには QM1 の証明書と C1 の公開証明書を格納します。 C1 の鍵リポジトリーには、C1 の証明書と QM1 の公開証明書を格納します。

### 手順

1. オペレーティング・システムに従って、クライアントとキュー・マネージャーで鍵リポジトリーを準備します。

- UNIX、Linux、および Windows システムの場合。

2. クライアントおよびキュー・マネージャー用の自己署名証明書を作成します。
   - UNIX、Linux、および Windows システムの場合。
3. 各証明書のコピーを取り出します。
   - UNIX、Linux、および Windows システムの場合。
4. FTP などのユーティリティーを使用して C1 証明書の公開部分を QM1 システムに転送し、またその逆を実行しますを参照)。
5. パートナーの証明書を、クライアントとキュー・マネージャーの鍵リポジトリーに追加します。
   - UNIX、Linux、および Windows システムの場合。
6. キュー・マネージャーでコマンド REFRESH SECURITY TYPE(SSL) を実行します。
7. 以下のいずれかの方法で、クライアント接続チャネルを定義します。
   - C1 で MQSCO 構造を持つ MQCONNX 呼び出しを使用します (WebSphere MQ MQI クライアントでのクライアント接続チャネルの作成を参照)。
   - サーバーでのサーバー接続定義およびクライアント接続定義の作成 の説明に従って、クライアント・チャネル定義テーブルを使用する。
8. QM1 で以下の例のようなコマンドを実行して、サーバー接続チャネルを定義します。

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   このチャネルの名前は、ステップ 6 で定義したクライアント接続チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。

### タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、213 ページの図 17 です。

### 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

オプションで、チャネル定義の SSLPEER フィルター属性を設定できます。 チャネル定義 SSLPEER が設定されている場合、その値は、ステップ 2 で作成したパートナー証明書のサブジェクト DN と一致している必要があります。 接続が成功すると、DISPLAY CHSTATUS 出力の SSLPEER フィールドに、リモート・クライアント証明書の対象 DN が表示されます。

## クライアントとキュー・マネージャーの相互認証への CA 署名証明書の使用

CA 署名 SSL または TLS 証明書を使用してクライアントとキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

## このタスクについて

シナリオ

- クライアント C1、およびキュー・マネージャー QM1 が存在し、これらは安全に通信する必要があります。 C1 と QM1 の間で、相互認証を実行する必要があります。
- 将来的にはこのネットワークを実稼働環境で使用することを計画しているので、最初から CA 署名証明書を使用することにしました。

構成の結果は次のようになります。

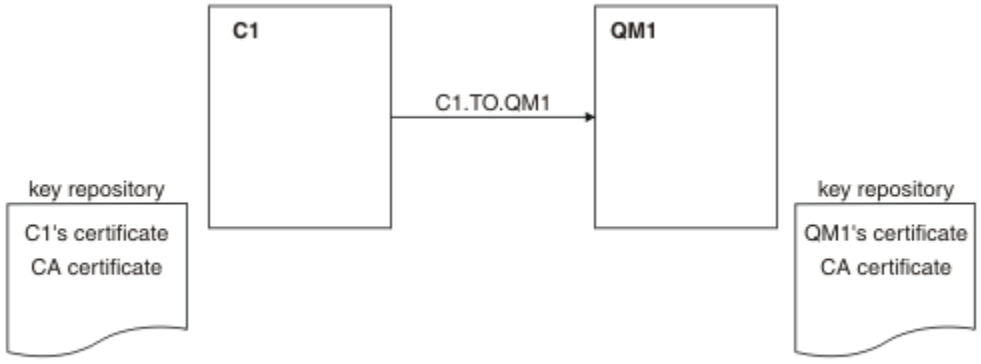



図 *18.* このタスクで実装する構成

215 ページの図 18 にあるとおり、C1 の鍵リポジトリーには、C1 の証明書と CA 証明書を格納します。 QM1 の鍵リポジトリーには、QM1 の証明書と CA 証明書を格納します。 この例では、C1 の証明書と QM1 の証明書の両方が同じ CA から発行されました。 C1 の証明書と QM1 の証明書が異なる CA から発行された場合は、C1 と QM1 の鍵リポジトリーに両方の CA 証明書が含まれている必要があります。

## 手順

1. オペレーティング・システムに従って、クライアントとキュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。
2. クライアントおよびキュー・マネージャー用の CA 署名証明書を要求します。
   クライアントとキュー・マネージャーで異なる CA を使用することがあります。
   - UNIX、Linux、および Windows システムの場合。
3. クライアントとキュー・マネージャーの鍵リポジトリーに認証局証明書を追加します。
   クライアントとキュー・マネージャーがそれぞれ異なる認証局を使用する場合は、各認証局の CA 証明書を両方の鍵リポジトリーに追加する必要があります。
   - UNIX、Linux、および Windows システムの場合。
4. クライアントとキュー・マネージャーの鍵リポジトリーに CA 署名証明書を追加します。
   - UNIX、Linux、および Windows システムの場合。
5. 以下のいずれかの方法で、クライアント接続チャネルを定義します。
   - C1 で MQSCO 構造を持つ MQCONNX 呼び出しを使用します (WebSphere MQ MQI クライアントでのクライアント接続チャネルの作成を参照)。
   - サーバーでのサーバー接続定義およびクライアント接続定義の作成 の説明に従って、クライアント・チャネル定義テーブルを使用する。
6. QM1 で以下の例のようなコマンドを実行して、サーバー接続チャネルを定義します。

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

このチャネルの名前は、ステップ 6 で定義したクライアント接続チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。

### タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、215 ページの図 18 です。

### 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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 出力の SSLPEER フィールドには、ステップ 2 で作成されたリモート・クライアント証明書の識別名が表示されます。 発行者名は、ステップ 4 で追加された個人証明書に署名した CA 証明書の識別名と一致します。

## キュー・マネージャーへのクライアントの匿名接続

相互認証のシステムを変更して、キュー・マネージャーが別のキュー・マネージャーに匿名で接続できるようにするための手順の例を取り上げます。

### このタスクについて

シナリオ

- 214 ページの『クライアントとキュー・マネージャーの相互認証への CA 署名証明書の使用』 で、キュー・マネージャーとクライアント (QM1 と C1) がセットアップされました。
- C1 が QM1 に匿名接続されるように変更する必要があるとします。

構成の結果は次のようになります。

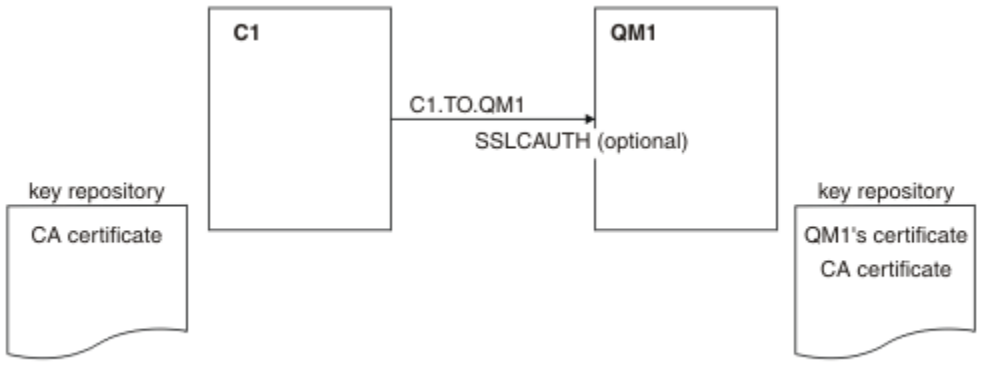



図 *19.* 匿名接続を可能にするクライアントとキュー・マネージャー

### 手順

1. オペレーティング・システムに従って、個人証明書を C1 の鍵リポジトリーから削除します。
   - UNIX、Linux、および Windows システムの場合。 この証明書には、以下のようなラベルが付いています。
     - ibmwebspheremq の後に、小文字に変換されたログオン・ユーザー ID が続きます (例: ibmwebspheremqmyuserid)。
2. クライアント・アプリケーションを再始動します。 あるいは、クライアント・アプリケーションを閉じ、すべての SSL または TLS 接続を再開します。
3. 次のコマンドを発行して、キュー・マネージャーでの匿名接続を許可します。

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

### タスクの結果

変更した鍵リポジトリーとチャネルを図にまとめたのが、216 ページの図 19 です。

### 次のタスク

チャネルのサーバー側で、チャネル状況表示にピア名パラメーター値が表示される場合は、クライアント証明書が流れたことを意味します。

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

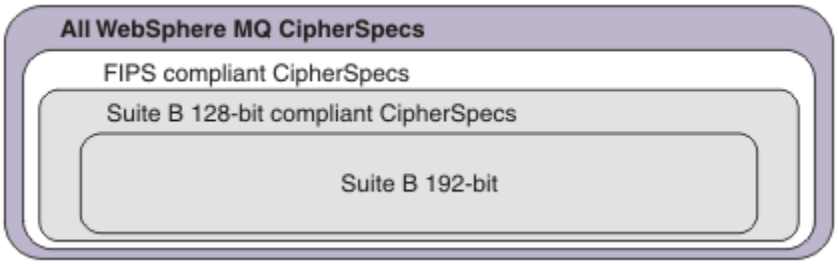
SSLCERTI フィールドと SSLPEER フィールドは空です。 これは、C1 が証明書を送信しなかったことを示しています。

## CipherSpec の指定

**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのいずれかで **SSLCIPH** パラメーターを使用して、 CipherSpec を指定します。

IBM WebSphere MQ とともに使用できる CipherSpec の一部は FIPS 準拠です。 それ以外 (NULL_MD5 など) は準拠していません。 同様に FIPS 準拠の CipherSpec の一部は Suite B 準拠でもありますが、それ以外は準拠していません。 Suite B 準拠の CipherSpec はすべて、FIPS 準拠でもあります。 Suite B 準拠の CipherSpec はすべて、128 ビット (ECDHE_ECDSA_AES_128_GCM_SHA256 など) と 192 ビット (ECDHE_ECDSA_AES_256_GCM_SHA384 など) の 2 つのグループに分けられます。

次の図は、これらのサブセットの関係を表しています。




次の表に、IBM WebSphere MQ SSL および TLS サポートとともに使用できる Cipher 仕様をリストします。個人用証明書を要求するときに、公開鍵と秘密鍵のペアの鍵サイズを指定します。 SSL ハンドシェーク時に使用される鍵のサイズは、表の注記のとおり、CipherSpec によって決定されている場合を除き、証明書に保管されているサイズです。

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|
| NULL_MD5 [a] | SSL 3.0 | MD5 | なし | 0 | いいえ | いいえ | いいえ |
| NULL_SHA [a] | SSL 3.0 | SHA-1 | なし | 0 | いいえ | いいえ | いいえ |
| RC4_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC4 | 40 | いいえ | いいえ | いいえ |
| RC4_MD5_US [a] | SSL 3.0 | MD5 | RC4 | 128 | いいえ | いいえ | いいえ |
| RC4_SHA_US [a] | SSL 3.0 | SHA-1 | RC4 | 128 | いいえ | いいえ | いいえ |
| RC2_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC2 | 40 | いいえ | いいえ | いいえ |
| DES_SHA_EXPORT [2] [a] | SSL 3.0 | SHA-1 | DES | 56 | いいえ | いいえ | いいえ |
| RC4_56_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | RC4 | 56 | いいえ | いいえ | いいえ |
| DES_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | DES | 56 | いいえ | いいえ | いいえ |
| TLS_RSA_WITH_AES_128_CBC_SHA [a] | TLS 1.0 | SHA-1 | AES | 128 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_256_CBC_SHA [4] [a] | TLS 1.0 | SHA-1 | AES | 256 | はい | いいえ | いいえ |

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|
| TLS_RSA_WITH_DES_CBC_SHA [a] | TLS 1.0 | SHA-1 | DES | 56 | いいえ [5] | いいえ | いいえ |
| FIPS_WITH_DES_CBC_SHA [B] | SSL 3.0 | SHA-1 | DES | 56 | いいえ [6] | いいえ | いいえ |
| TLS_RSA_WITH_AES_128_GCM_SHA256 [B] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_256_GCM_SHA384 [B] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_128_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 128 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_256_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 256 | はい | いいえ | いいえ |
| ECDHE_ECDSA_RC4_128_SHA256 [B] | TLS 1.2 | SHA-1 | RC4 | 128 | いいえ | いいえ | いいえ |
| ECDHE_RSA_RC4_128_SHA256 [B] | TLS 1.2 | SHA_1 | RC4 | 128 | いいえ | いいえ | いいえ |
| ECDHE_ECDSA_AES_128_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 128 | はい | いいえ | いいえ |
| ECDHE_ECDSA_AES_256_CBC_SHA384 [B] | TLS 1.2 | SHA-384 | AES | 256 | はい | いいえ | いいえ |
| ECDHE_RSA_AES_128_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 128 | はい | いいえ | いいえ |
| ECDHE_RSA_AES_256_CBC_SHA384 [B] | TLS 1.2 | SHA-384 | AES | 256 | はい | いいえ | いいえ |
| ECDHE_ECDSA_AES_128_GCM_SHA256 [B] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | はい | はい | いいえ |
| ECDHE_ECDSA_AES_256_GCM_SHA384 [B] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | はい | いいえ | はい |
| ECDHE_RSA_AES_128_GCM_SHA256 [B] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | はい | いいえ | いいえ |
| ECDHE_RSA_AES_256_GCM_SHA384 [B] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | はい | いいえ | いいえ |

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|
| TLS_RSA_WITH_NULL_SHA256 [B] | TLS 1.2 | SHA-256 | なし | 0 | いいえ | いいえ | いいえ |
| ECDHE_RSA_NULL_SHA256 [B] | TLS 1.2 | SHA-1 | なし | 0 | いいえ | いいえ | いいえ |
| ECDHE_ECDSA_NULL_SHA256 [B] | TLS 1.2 | SHA-1 | なし | 0 | いいえ | いいえ | いいえ |
| TLS_RSA_WITH_NULL_NULL [B] | TLS 1.2 | なし | なし | 0 | いいえ | いいえ | いいえ |
| TLS_RSA_WITH_RC4_128_SHA256 [B] | TLS 1.2 | SHA-1 | RC4 | 128 | いいえ | いいえ | いいえ |

**注 :**

1. FIPS 認定プラットフォーム上の FIPS 認定 CipherSpec であるかどうかを示しています。 FIPS の説明については、連邦情報処理標準 (FIPS) を参照してください。
2. ハンドシェークの最大鍵サイズは 512 ビットです。 SSL ハンドシェーク時に交換されるどちらかの証明書の鍵サイズが 512 ビットより大きい場合は、ハンドシェーク時に使用するための 512 ビットの一時鍵が生成されます。
3. ハンドシェークの鍵サイズは 1024 ビットです。
4. WebSphere MQ エクスプローラーが使用する JRE に対して適切な無制限のポリシー・ファイルが適用されていない場合には、この CipherSpec を使用して、WebSphere MQ エクスプローラーからキュー・マネージャーへの安全な接続を確立することはできません。
5. この CipherSpec は、2007 年 5 月 19 日より前は FIPS 140-2 で認証されていました。
6. この CipherSpec は、2007 年 5 月 19 日より前は FIPS 140-2 で認証されていました。 FIPS_WITH_DES_CBC_SHA という名前は歴史的なものであり、この CipherSpec がかつて FIPS 準拠であった という事実を反映しています (ただし、現在は準拠していません)。 この CipherSpec は非推奨となりました。使用することはお勧めしません。
7. この CipherSpec を使用して最大 32 GB までデータを転送できますが、それを超えるとエラー AMQ9288 を出して接続が終了します。 このエラーを回避するために、Triple-DES を使用しないか、またはこの CipherSpec を使用する際に秘密鍵リセットを有効にします。

**プラットフォームのサポート :**

- a サポート対象のすべてのプラットフォームで使用可能
- b UNIX, Linux, and Windows プラットフォームのみで使用可能

**関連概念**

34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』
このトピックでは、IBM WebSphere MQ における CipherSpec とデジタル証明書の関係を概説することにより、個々のセキュリティー・ポリシーに適した CipherSpec とデジタル証明書を選択する方法を説明します。

**関連資料**

DEFINE CHANNEL

## 推奨されない CipherSpec

必要に応じて WebSphere MQ で使用できる、推奨されない CipherSpec のリスト。

推奨されない CipherSpec を有効にする方法の詳細については、38 ページの『IBM WebSphere MQ でサポートされる CipherSpec 値』を参照してください。

次の表に、WebSphere MQ TLS サポートとともに使用できる、推奨されない CipherSpec をリストします。

| プラットフォーム・サポート [223 ページの『1』](#) | CipherSpec 名 | 使用されるプロトコル | データ整合性 | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [223 ページの『2』](#) | Suite B | 非推奨時の更新 |
|---|---|---|---|---|---|---|---|---|
| すべて | DES_SHA_EXPORT[223 ページの『3』](#) | SSL 3.0 | SHA-1 | DES | 56 | No | No | 7.5.0.6 |
| Windows UNIX Linux | DES_SHA_EXPORT1024[223 ページの『4』](#) | SSL 3.0 | SHA-1 | DES | 56 | No | No | 7.5.0.6 |
| Windows UNIX Linux | FIPS_WITH_DES_CBC_SHA | SSL 3.0 | SHA-1 | DES | 56 | いいえ [223 ページの『6』](#) | No | 7.5.0.6 |
| Windows UNIX Linux | FIPS_WITH_3DES_EDE_CBC_SHA | SSL 3.0 | SHA-1 | 3DES | 168 | いいえ [223 ページの『7』](#) | No | 7.5.0.8 |
| すべて | NULL_MD5 | SSL 3.0 | MD5 | なし | 0 | No | No | 7.5.0.6 |
| すべて | NULL_SHA | SSL 3.0 | SHA-1 | なし | 0 | No | No | 7.5.0.6 |
| すべて | RC2_MD5_EXPORT[223 ページの『3』](#) | SSL 3.0 | MD5 | RC2 | 40 | No | No | 7.5.0.7 |
| すべて | RC4_MD5_EXPORT[223 ページの『3』](#) | SSL 3.0 | MD5 | RC4 | 40 | No | No | 7.5.0.7 |
| すべて | RC4_MD5_US | SSL 3.0 | MD5 | RC4 | 128 | No | No | 7.5.0.7 |
| すべて | RC4_SHA_US | SSL 3.0 | SHA-1 | RC4 | 128 | No | No | 7.5.0.7 |
| Windows UNIX Linux | RC4_56_SHA_EXPORT1024[223 ページの『4』](#) | SSL 3.0 | SHA-1 | RC4 | 56 | No | No | 7.5.0.7 |
| すべて | TRIPLE_DES_SHA_US | SSL 3.0 | SHA-1 | 3DES | 168 | No | No | 7.5.0.8 |
| すべて | TLS_RSA_WITH_DES_CBC_SHA | TLS 1.0 | SHA-1 | DES | 56 | いいえ [223 ページの『5』](#) | No | 7.5.0.6 |

| プラットフォーム・サポート 223 ページの『1』 | CipherSpec 名 | 使用されるプロトコル | データ整合性 | 暗号化アルゴリズム | 暗号化ビット数 | FIPS 223 ページの『2』 | Suite B | 非推奨時の更新 |
|---|---|---|---|---|---|---|---|---|
| Windows UNIX Linux | ECDHE_ECDSA_NULL_SHA256 | TLS 1.2 | SHA-1 | なし | 0 | No | No | 7.5.0.6 |
| Windows UNIX Linux | ECDHE_ECDSA_RC4_128_SHA256 | TLS 1.2 | SHA-1 | RC4 | 128 | No | No | 7.5.0.7 |
| Windows UNIX Linux | ECDHE_RSA_NULL_SHA256 | TLS 1.2 | SHA-1 | なし | 0 | No | No | 7.5.0.6 |
| Windows UNIX Linux | ECDHE_RSA_RC4_128_SHA256 | TLS 1.2 | SHA-1 | RC4 | 128 | No | No | 7.5.0.7 |
| Windows UNIX Linux | TLS_RSA_WITH_NULL_NULL | TLS 1.2 | なし | なし | 0 | No | No | 7.5.0.6 |
| すべて | TLS_RSA_WITH_NULL_SHA256 | TLS 1.2 | SHA-256 | なし | 0 | No | No | 7.5.0.6 |
| Windows UNIX Linux | TLS_RSA_WITH_RC4_128_SHA256 | TLS 1.2 | SHA-1 | RC4 | 128 | No | No | 7.5.0.7 |
| すべて | TLS_RSA_WITH_3DES_EDE_CBC_SHA 223 ページの『8』 | TLS 1.0 | SHA-1 | 3DES | 168 | Yes | No | 7.5.0.8 |
| Windows UNIX Linux | ECDHE_ECDSA_3DES_EDE_CBC_SHA256 223 ページの『8』 | TLS 1.2 | SHA-1 | 3DES | 168 | Yes | No | 7.5.0.8 |
| Windows UNIX Linux | ECDHE_RSA_3DES_EDE_CBC_SHA256 223 ページの『8』 | TLS 1.2 | SHA-1 | 3DES | 168 | Yes | No | 7.5.0.8 |

| プラットフォーム・サポート 223 ページの『1』 | CipherSpec 名 | 使用されるプロトコル | データ整合性 | 暗号化アルゴリズム | 暗号化ビット数 | FIPS 223 ページの『2』 | Suite B | 非推奨時の更新 |
|---|---|---|---|---|---|---|---|---|

**注 :**

1. 具体的なプラットフォームが記載されていない場合は、CipherSpec はすべてのプラットフォームで使用可能です。
2. FIPS 認定プラットフォーム上の FIPS 認定 CipherSpec であるかどうかを示しています。 FIPS の説明については、連邦情報処理標準 (FIPS) を参照してください。
3. ハンドシェークの最大鍵サイズは 512 ビットです。 SSL ハンドシェーク時に交換される証明書のどちらかに、512 ビットより大きい鍵サイズがある場合、ハンドシェーク時に使用するために、一時的な 512 ビット鍵が生成されます。
4. ハンドシェークの鍵サイズは 1024 ビットです。
5. この CipherSpec は、2007 年 5 月 19 日より前は FIPS 140-2 で認証されていました。
6. この CipherSpec は、2007 年 5 月 19 日より前は FIPS 140-2 で認証されていました。 FIPS_WITH_DES_CBC_SHA という名前は歴史的な事情によるものであり、この CipherSpec がかつては FIPS 準拠であった (ただし現在は準拠していません) という事実を反映するものです。 この CipherSpec は非推奨となりました。使用することはお勧めしません。
7. FIPS_WITH_3DES_EDE_CBC_SHA という名前は歴史的な事情によるものであり、この CipherSpec がかつては FIPS 準拠であった (ただし現在は準拠していません) という事実を反映するものです。 この CipherSpec の使用は推奨されません。
8. この CipherSpec を使用して最大 32 GB までデータを転送できますが、それを超えるとエラー AMQ9288 を出して接続が終了します。 このエラーを回避するために、Triple-DES を使用しないか、またはこの CipherSpec を使用する際に秘密鍵リセットを有効にします。

## IBM WebSphere MQ Explorer を使用した CipherSpecs に関する情報の取得

IBM WebSphere MQ Explorer を使用して、CipherSpec の説明を表示できます。

217 ページの『CipherSpec の指定』 の CipherSpec についての情報を取得するには、次の手順を実行してください。

1. **IBM WebSphere MQ エクスプローラー**を開き、**Queue Managers** フォルダーを展開する。
2. キュー・マネージャーを開始したことを確認する。
3. 処理したいキュー・マネージャーを選択して 「**チャネル**」をクリックします。
4. 処理するチャネルを右クリックし、「**プロパティー**」を選択する。
5. 「**SSL**」プロパティー・ページを選択する。
6. 処理したい CipherSpec を、リストの中から選択する。 リストの下のウィンドウに、説明が表示されます。

## CipherSpec の代替指定方法

オペレーティング・システムの SSL サポートを使用できるプラットフォームの場合は、システムで新しい CipherSpec に対応できる可能性があります。 新しい CipherSpec を指定するには、SSLCIPH パラメーターを使用しますが、指定する値は、ご使用のプラットフォームによって異なります。

**注 :** このセクションは、UNIX、Linux、または Windows システムには適用されません。これは、CipherSpec が WebSphere MQ 製品に付属しているので、新しい CipherSpec が出荷後に使用可能にならないからです。

オペレーティング・システムが SSL サポートを提供するプラットフォームの場合、ご使用のシステムが、217 ページの『CipherSpec の指定』 に含まれていない新しい CipherSpec をサポートする場合があります。

新しい CipherSpec を指定するには、SSLCIPH パラメーターを使用しますが、指定する値は、ご使用のプラットフォームによって異なります。 いずれの場合も、CipherSpec の指定は、システムが実行している SSL のバージョンによってサポートされ、かつ有効である SSL CipherSpec と対応している必要があります。

**IBM i**
16 進値を表す 2 文字のストリング。

許可される値について詳しくは、該当する製品資料を参照してください ( IBM i 製品資料で *cipher_spec* を検索してください)。

次のように CHGMQMCHL コマンドまたは CRTMQMCHL コマンドを使用すると、値を指定できます。

```
CRTMQMCHL CHLNAME('channel name') SSLCIPH('hexadecimal value')
```

また、ALTER QMGR MQSC コマンドを使用して SSLCIPH パラメーターを設定することもできます。

**z/OS**
16 進値を表す 2 文字のストリング。 この 16 進コードは、SSL プロトコルで定義されている値に相当します。

詳しくは、「*z/OS Cryptographic Services System Secure Sockets Layer* プログラミング」(SD88-6252) の API リファレンスの章にある gsk_environment_open() の説明を参照してください。 ここには、2 桁の 16 進数コードの形式でサポートされるすべての SSL V3.0 および TLS V1.0 暗号仕様のリストがあります。

### WebSphere MQ クラスターの考慮事項

WebSphere MQ クラスターを使用する場合は、217 ページの『CipherSpec の指定』 にある CipherSpec 名を使用するのが無難です。 代替指定を使用する場合は、他のプラットフォームではその指定は無効な場合があることに注意してください。 詳細については、250 ページの『SSL とクラスター』を参照してください。

## IBM WebSphere MQ MQI クライアント用の CipherSpec の指定

IBM WebSphere MQ MQI クライアントの CipherSpec を指定するには、3 つのオプションがあります。

以下のオプションがあります。

- チャネル定義テーブルを使用する
- MQCONNX 呼び出しで、MQCD_VERSION_7 以降の MQCD 構造の SSLCipherSpec フィールドを使用する。
- Active Directory を使用する (Active Directory サポートを備えた Windows システム上)

## IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS を使用した CipherSuite の指定

IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS は、他のプラットフォームとは異なる方法で CipherSuites を指定します。

IBM WebSphere MQ classes for Java で CipherSuite を指定する方法については、 Secure Sockets Layer (SSL) サポートを参照してください。

IBM WebSphere MQ classes for JMS での CipherSuite の指定については、 WebSphere MQ classes for JMS での Secure Sockets Layer (SSL) の使用を参照してください。

# SSL および TLS 秘密鍵のリセット

IBM WebSphere MQ では、キュー・マネージャーおよびクライアントでの秘密鍵のリセットがサポートされています。

特定バイト数の暗号化されたデータがチャネルを流れた場合、またはチャネルが一定期間使用されなかった場合に、秘密鍵がリセットされます。

鍵リセット値は、MQ チャネルの開始側が常に設定します。

## キュー・マネージャー

キュー・マネージャーの場合、コマンド **ALTER QMGR** をパラメーター **SSLRKEYC** と共に使用して、鍵の再ネゴシエーション中に使用する値を設定します。

## MQI クライアント

デフォルトでは、MQI クライアントは秘密鍵の再ネゴシエーションは行いません。 3 つの方法のいずれかによって、MQI クライアントで鍵の再ネゴシエーションを実行できます。 次のリストでは、優先度の高い順に方法を示しています。 複数の値を指定している場合は、最高の優先度の値が使用されます。

1. MQCONNX 呼び出しで、MQSCO 構造体の KeyResetCount フィールドを使用する方法
2. 環境変数 MQSSLRESET を使用する方法
3. SSLKeyResetCount 属性を MQI クライアント構成ファイルに設定する方法

これらの変数は、0 から 999 999 999 の範囲の整数に設定することができ、SSL または TLS 秘密鍵が再ネゴシエーションされる前に、SSL または TLS 会話内で送受信される暗号化されていないバイト数を表します。 0 の値を指定すると、SSL または TLS 秘密鍵は絶対に再ネゴシエーションされません。 SSL または TLS 秘密鍵のリセット・カウントを 1 バイトから 32 KB の範囲で指定すると、SSL または TLS チャネルは 32 KB の秘密鍵リセット・カウントを使用します。 これは、SSL または TLS 秘密鍵のリセット値が小さい場合に生じる可能性のある、鍵の過度のリセットを防ぐためです。

このチャネルに対して、ゼロよりも大きな値が指定され、チャネルのハートビートが有効化されている場合、チャネル・ハートビートに続けてメッセージ・データが送受信される前に、秘密鍵も再ネゴシエーションされます。

再ネゴシエーションが成功するごとに、次の秘密鍵の再ネゴシエーションまでのバイト数がリセットされます。

MQSCO 構造体の詳細については、KeyResetCount (MQLONG) を参照してください。 MQSSLRESET の詳細については、MQSSLRESET を参照してください。 クライアント構成ファイルでの SSL または TLS の使用方法について詳しくは、クライアント構成ファイルの SSL スタンザを参照してください。

## JAVA

IBM WebSphere MQ classes for Java の場合、次のいずれかの方法によって、アプリケーションで秘密鍵をリセットできます。

- MQEnvironment クラスの sslResetCount フィールドを設定する方法。
- Hashtable オブジェクトの環境プロパティー MQC.SSL_RESET_COUNT_PROPERTY を設定する。 この方法では、アプリケーションによって、MQEnvironment クラスの `properties` フィールドにハッシュ・テーブルが割り当てられるか、またはそのコンストラクターで MQQueueManager オブジェクトにハッシュ・テーブルが受け渡されます。

アプリケーションでこれらの方法を複数使用する場合、通常の優先順位ルールが適用されます。 優先順位ルールについては、クラス com.ibm.mq.MQEnvironment を参照してください。

sslReset カウント・フィールドまたは環境プロパティー MQC.SSL_RESET_COUNT_PROPERTY の値は、秘密鍵が再ネゴシエーションされる前に WebSphere MQ classes for Java クライアント・コードによって送受信される合計バイト数を表します。 送信バイト数は暗号化前の数であり、受信バイト数は暗号化解除された後の数です。 バイト数には、 WebSphere MQ classes for Java クライアントによって送受信される制御情報も含まれます。

リセット・カウントがゼロ (デフォルト値) の場合、秘密鍵は再ネゴシエーションされません。 CipherSuite が指定されていない場合、リセット・カウントは無視されます。

## JMS

IBM WebSphere MQ classes for JMS の場合、SSLRESETCOUNT プロパティーは、暗号化に使用される秘密鍵が再ネゴシエーションされるまでに接続で送受信される合計バイト数を表します。 送信バイト数は暗号化前の数であり、受信バイト数は暗号化解除された後の数です。 バイト数には、IBM WebSphere MQ

classes for JMS によって送受信される制御情報も含まれています。 例えば、4 MB のデータが流れた後に再ネゴシエーションされる秘密鍵を使用して、SSL または TLS が使用可能になっている MQI チャネルでの接続を作成するために使用できる ConnectionFactory オブジェクトを構成するには、JMSAdmin に次のコマンドを発行します。

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

SSLRESETCOUNT の値がゼロ (デフォルト値) の場合、秘密鍵の再ネゴシエーションは行われません。 SSLCIPHERSUITE が設定されていない場合、SSLRESETCOUNT プロパティーは無視されます。

### .NET

.NET の非管理対象クライアントの場合、整数プロパティー SSLKeyResetCount は、秘密鍵が再ネゴシエーションされるまでに SSL または TLS 会話内で送受信される暗号化されていないバイト数を示します。

IBM WebSphere MQ classes for .NET のオブジェクト・プロパティーの使用法については、属性値の取得と設定を参照してください。

### XMS .NET

XMS .NET の非管理対象クライアントについては、IBM WebSphere MQ キュー・マネージャーとのセキュア接続を参照してください。

**関連資料**
ALTER QMGR
DISPLAY QMGR

## ユーザー出口プログラムでの機密性の実装

### セキュリティー出口による機密性の実装

セキュリティー出口は、チャネル上を流れるデータの暗号化と復号用に、対称鍵を生成し、配布することによって、機密性サービスで役割を果たすことができます。 これを行うための一般的な手法では、PKI テクノロジーが使用されます。

あるセキュリティー出口は、ランダム・データ値を生成し、相手側セキュリティー出口が代理をするキュー・マネージャーまたはユーザーの公開鍵を使用して、そのデータ値を暗号化し、暗号化されたデータをセキュリティー・メッセージ内で相手側に送信します。 相手側のセキュリティー出口は、代理をするキュー・マネージャーまたはユーザーの秘密鍵を使用して、ランダム・データ値を復号します。 これで、各セキュリティー出口は、両方のセキュリティー出口が認識するアルゴリズムを使用することによって、このランダム・データ値を使用して、相手側とは無関係に、対称鍵を入手することができるようになります。 代わりの方法として、ランダム・データ値を鍵として使用することもできます。

この時点までに最初のセキュリティー出口が相手側のセキュリティー出口を認証しなかった場合、相手側によって送信される次のセキュリティー・メッセージには、対称鍵を使用して暗号化される期待値が入っている場合があります。 最初のセキュリティー出口は、相手側セキュリティー出口が期待値を正しく暗号化できたかどうかを調べることによって、相手側のセキュリティー出口を認証することができます。

また、複数のアルゴリズムが使用可能である場合、セキュリティー出口は、この機会を使用して、チャネル上で流れるデータの暗号化と復号用のアルゴリズムについて合意することもできます。

### メッセージ出口による機密性の実装

チャネルの送信側にあるメッセージ出口は、メッセージ内のアプリケーション・データを暗号化し、チャネルの受信側にある別のメッセージ出口は、そのデータを復号することができます。 パフォーマンス上の理由から、通常、対称鍵アルゴリズムがこの目的に使用されます。 対称鍵の生成と配布方法の詳細については、226 ページの『ユーザー出口プログラムでの機密性の実装』を参照してください。

組み込みメッセージ記述子が含まれている伝送キュー見出し MQXQH のようなメッセージ内の見出しは、メッセージ出口によって暗号化してはなりません。 これは、メッセージ見出しのデータ変換が、送信側で

メッセージ出口が呼び出された後か、受信側でメッセージ出口が呼び出 される前のどちらかで行われるからです。 見出しが暗号化されると、データ変換が失敗し、チャネルが停止します。

### 送信出口と受信出口による機密性の実装

送信出口と受信出口は、チャネル上で流れるデータの暗号化と復号に使用できます。 これらの出口は、次の理由でこのサービスを提供する場合に、メッセージ出口よりも適しています。

- メッセージ・チャネル上で、メッセージ見出しが、メッセージ内のアプリケーション・データとともに暗号化できる。
- 送信出口と受信出口が、メッセージ・チャネルだけでなく、MQI チャネル上でも使用できる。 MQI 呼び出しのパラメーターには、MQI チャネル上を通過する間に保護が必要な機密アプリケーション・データが入っている場合があります。 したがって、両方のチャネル上で同じ送信出口と受信出口を使用できます。

### API 出口と API 交差出口による機密性の実装

送信側のアプリケーションがメッセージを書き込むときには、API 出口または API 交差出口によって、そのメッセージのアプリケーション・データを暗号化し、受信側のアプリケーションがそのメッセージを取り出すときには、2 つ目の出口によってそのデータを復号できます。 パフォーマンス上の理由から、通常、対称鍵アルゴリズムがこの目的に使用されます。 しかし、多数のユーザーが相互にメッセージを送信するアプリケーション・レベルでは、問題は、メッセージの所定の受信側だけがメッセージを復号できることを確実にするにはどうべきかということです。 1 つの解決法は、メッセージを相互に送信するユーザーのペアごとに、別々の対称鍵を使用することです。 しかし、この解決法は、特にユーザーが別々の組織に属している場合は、管理が難しく、時間がかかります。 この問題の標準的な解決方法は、デジタル・エンベロープ と呼ばれ、PKI テクノロジーを使用します。

アプリケーションがキューにメッセージを書き込むときには、API 出口または API 交差出口によって、ランダムな対称鍵を生成し、そのキーでメッセージのアプリケーション・データを暗号化します。 さらに、その出口は、対象の受信側の公開鍵を使用して対称鍵を暗号化します。 次に、メッセージ内のアプリケーション・データを、暗号化されたアプリケーション・データおよび暗号化された対称鍵で置き換えます。 このようにして、所定の受信側だけが、対称鍵を復号でき、したがってアプリケーション・データを復号できます。 暗号化されたメッセージの対象になり得る受信側が複数存在する場合、その出口によって、対象の受信側ごとに、対称鍵のコピーを暗号化できます。

アプリケーション・データの暗号化と復号のために使用できるアルゴリズムが複数存在する場合は、その出口で使用したアルゴリズムの名前をその出口の中に格納できます。

## メッセージのデータ保全性

データ保全性を維持するには、さまざまなタイプのユーザー出口プログラムを使用して、メッセージのメッセージ・ダイジェストまたはデジタル署名を提供できます。

### データ整合性

**メッセージによるデータ保全性の実装**

SSL または TLS を使用する場合は、どの CipherSpec を選択するかによって企業内のデータ保全性のレベルが決まります。 WebSphere MQ Advanced Message Service (AMS) を使用する場合は、固有メッセージの保全性を指定することができます。

**メッセージ出口によるデータ保全性の実装**

チャネルの送信側のメッセージ出口によって、メッセージをデジタル署名することができます。 その後、メッセージが意図的に変更されたかどうかを検出するために、このデジタル署名を、チャネルの受信側のメッセージ出口によって検査することができます。

デジタル署名の代わりに、メッセージ・ダイジェストを使用して、保護を行うこともできます。 メッセージ・ダイジェストは、不用意による改ざん、または無差別の改ざんに対して有効ですが、知識のある人物が、メッセージの変更または置き換えを行ったり、まったく新しいダイジェストを生成したりす

ることは防止できません。 これは、メッセージ・ダイジェストの生成に使用されるアルゴリズムが、既知のアルゴリズムである場合は、特に当てはまります。

**送信出口と受信出口によるデータ保全性の実装**

メッセージ・チャネル上では、このサービスの提供には、メッセージ出口の方が適切です。これは、メッセージ出口がメッセージ全体にアクセスできるからです。 MQI チャネル上では、MQI 呼び出しのパラメーターには、保護が必要なアプリケーション・データが入っている場合があります。この保護を提供できるのは、送信出口と受信出口だけです。

**API 出口または API 交差出口によるデータ保全性の実装**

送信側のアプリケーションがメッセージを書き込むときには、API 出口または API 交差出口によって、そのメッセージにデジタル署名を追加できます。 受信側のアプリケーションがそのメッセージを取得するときには、メッセージが意図的に変更されたかどうかを検出するために、2 つ目の出口によってそのデジタル署名を検査できます。

デジタル署名の代わりに、メッセージ・ダイジェストを使用して、保護を行うこともできます。 メッセージ・ダイジェストは、不用意による改ざん、または無差別の改ざんに対して有効ですが、知識のある人物が、メッセージの変更または置き換えを行ったり、まったく新しいダイジェストを生成したりすることは防止できません。 これは、メッセージ・ダイジェストの生成に使用されるアルゴリズムが、既知のアルゴリズムである場合は、特に当てはまります。

## SSL または TLS による 2 つのキュー・マネージャーの接続

SSL または TLS 暗号セキュリティー・プロトコルを使用するセキュア通信では、 通信チャネルをセットアップし、認証に使用するデジタル証明書を管理する必要があります。

SSL または TLS インストール環境をセットアップするには、SSL または TLS を使用するようにチャネルを定義する必要があります。 また、デジタル証明書を取得し、管理することも必要です。 テスト・システムでは、自己署名証明書か、ローカル認証局 (CA) で発行された証明書を使用できます。 実動システムでは、自己署名証明書を使用しないでください。 細については、 ../zs14140_.dita を参照してください。

証明書の作成と管理の詳細については、114 ページの『UNIX, Linux, and Windows システムでの SSL または TLS の取り扱い』を参照してください。

このトピック集では、SSL 通信のセットアップに関連したタスクを取り上げ、それらのタスクを実行するための段階的な手順を説明します。

また、プロトコルのオプション部分である SSL または TLS クライアント認証をテストすることもできます。SSL または TLS ハンドシェーク中に、SSL または TLS クライアントは常にサーバーからデジタル証明書を取得し検証します。 WebSphere MQ の実装では、SSL または TLS サーバーは、常にクライアントから証明書を要求します。

**注 :**

1. この状況では、SSL クライアントはハンドシェークを開始する接続を参照します。
2. 詳細については、用語集を参照してください。

UNIX、Linux、および Windows システムでは、SSL または TLS クライアントは、正しい WebSphere MQ 形式 (`ibmwebspheremq` の後に続けて、小文字にされたキュー・マネージャー名) のラベルが付いた証明書が存在する場合に限って証明書を送信します。 例えば、`QM1` の場合は、`ibmwebspheremqqm1` です。

WebSphere MQ は、他の製品の証明書との混同を避けるために、`ibmwebspheremq` という接頭部をラベルに付けます。 証明書ラベル全体を小文字で指定してください。

SSL または TLS サーバーは、クライアント証明書が送信される場合は、常にそのクライアント証明書を検証します。 クライアントが証明書を送信しない場合に認証が失敗するのは、チャネルの SSL または TLS サーバー側の定義で、SSLCAUTH パラメーターが REQUIRED に設定されている場合、または SSLPEER パラメーター値が設定されている場合に限られます。 匿名によるキュー・マネージャーへの接続 (つまり、SSL クライアントまたは TLS クライアントから証明書を送信しない場合) について詳しくは、211 ページの『片方向認証による 2 つのキュー・マネージャーの接続』を参照してください。

# デジタル証明書ラベルの要件に関する説明

デジタル証明書を使用するように SSL および TLS をセットアップする際には、使用しているプラットフォームや接続方式に応じて、特定のラベル要件に従わなければなりません。

## このタスクについて

### 証明書ラベルに関する説明

証明書ラベルは、鍵リポジトリーに格納されているデジタル証明書を表す固有 ID で、便利で人間が理解できる名前があり、この名前を使用して鍵管理機能の実行時に特定の証明書が参照されます。 証明書ラベルは、初めて証明書を鍵リポジトリーに追加する際に割り当てます。

証明書ラベルは、証明書の「サブジェクト識別名」フィールドや「サブジェクト共通名」フィールドとは別のものです。 「サブジェクト識別名」と「サブジェクト共通名」は証明書自体のフィールドであることに注意してください。 これらのフィールドは、証明書の作成時に定義され、変更できません。 しかし、デジタル証明書に関連付けられているラベルは、必要に応じて変更できます。

### 証明書ラベルの使用法

IBM WebSphere MQ は、証明書ラベルを使用して、SSL ハンドシェーク中に送信される個人証明書を見つけます。 したがって、鍵リポジトリーに複数の個人証明書があっても、あいまいになりません。

証明書ラベルは命名規則に従います。したがって、使用しているプラットフォームに対応する正しいラベル命名規則を使用していることを確認する必要があります。

このコンテキストで、SSL または TLS クライアントは、ハンドシェークを開始する接続パートナーを意味します。このパートナーは、IBM WebSphere MQ クライアントや別のキュー・マネージャーの可能性があります。

SSL または TLS ハンドシェーク中に、SSL または TLS クライアントは常にサーバーからデジタル証明書を取得し検証します。 IBM WebSphere MQ 実装環境で、SSL または TLS サーバーは常にクライアントからの証明書を要求し、クライアントは証明書があれば常にサーバーに証明書を提供します。 クライアントが個人証明書を見つけられない場合は、クライアントは `no certificate` 応答をサーバーに送信します。

SSL または TLS サーバーは、クライアント証明書が送信される場合は、常にそのクライアント証明書を検証します。 クライアントが証明書を送信しない場合に認証が失敗するのは、チャネルの SSL または TLS サーバー側の定義で、SSLCAUTH パラメーターが REQUIRED に設定されている場合、または SSLPEER パラメーター値が設定されている場合です。

片方向の認証を使用したキュー・マネージャーへの接続 (つまり、SSL クライアントまたは TLS クライアントから証明書を送信しない場合) について詳しくは、211 ページの『片方向認証による 2 つのキュー・マネージャーの接続』を参照してください。

## 、 *UNIX, Linux, and Windows システム*

## このタスクについて

、UNIX, Linux, and Windows システムでは、サーバーが正しい IBM WebSphere MQ 形式でラベル付けされた証明書を検出した場合にのみ、SSL または TLS サーバーがクライアントに証明書を送信します。 これらのシステムでは、正しい形式は `ibmwebspheremq` の後に、小文字に変換されたキュー・マネージャーの名前が続きます。

例えば、`QM1` という名前のキュー・マネージャーの場合、証明書ラベルの要件は以下のようになります。

```
ibmwebspheremqqm1
```

キュー・マネージャーの鍵リポジトリー内に、大/小文字および形式が正しく、要求されたラベルに一致する証明書がない場合、エラーが発生し、SSL または TLS ハンドシェークは失敗します。

### *IBM WebSphere MQ クライアント*

## このタスクについて

IBM WebSphere MQ クライアント・アプリケーションから接続している場合、SSL または TLS クライアントは、クライアント・アプリケーション・プロセスを実行しているユーザー名を ibmwebspheremq の後に追加した形式のラベルが付いた証明書が存在する場合に限って、証明書を送信します。

例えば、ユーザー名 wasadmin の場合、証明書ラベルの要件は示されているとおりで、小文字に変換されます。

```
ibmwebspheremqwasadmin
```

上記のラベル要件は、C または C++ と .NET のメッセージ・サービス・クライアントに適用されます。

### *IBM WebSphere MQ Java または IBM WebSphere MQ JMS クライアント*

## このタスクについて

IBM WebSphere MQ Java または IBM WebSphere MQ JMS クライアントは、Java Secure Socket Extension (JSSE) プロバイダーの機構を使用して、SSL または TLS ハンドシェーク中に個人証明書を選択するので、証明書ラベルの要件の対象外です。

デフォルトの動作は、JSSE クライアントが鍵リポジトリー全体で証明書を順番に検索し、最初に検出された受け入れ可能な個人証明書を選択するという動作です。 しかし、この動作はデフォルトにすぎないので、JSSE プロバイダーの実装環境に応じて異なります。

さらに、構成により、またはアプリケーションで実行時に直接アクセスして、JSSE インターフェースを大幅にカスタマイズできます。 特定の詳細情報については、JSSE プロバイダーによって提供される資料を参照してください。

トラブルシューティング用に、または IBM WebSphere MQ Java クライアント・アプリケーションと特定の JSSE プロバイダーを組み合わせて実行されるハンドシェークに関する理解を深めるために、以下のように設定してデバッグを使用可能にすることができます。

```
javax.net.debug=ssl
```

この設定は、JVM 環境で行います。

コマンド・ラインで -Djavax.net.debug=ssl を使用したり、この変数をアプリケーション内で設定したり、構成により設定したりできます。

**関連概念**

134 ページの『個人証明書を鍵リポジトリーにインポートする操作 (UNIX, Linux, and Windows システム)』
個人証明書をインポートする手順を取り上げます。

# 2 つのキュー・マネージャーの相互認証への自己署名証明書の使用

自己署名 SSL または TLS 証明書を使用して 2 つのキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

## このタスクについて

シナリオ

- 安全に通信する必要のあるキュー・マネージャーが 2 つ (QM1 と QM2) あります。 QM1 と QM2 の間で、相互認証を実行する必要があります。
- そこで、自己署名証明書を使用して安全な通信をテストすることにしました。

構成の結果は次のようになります。

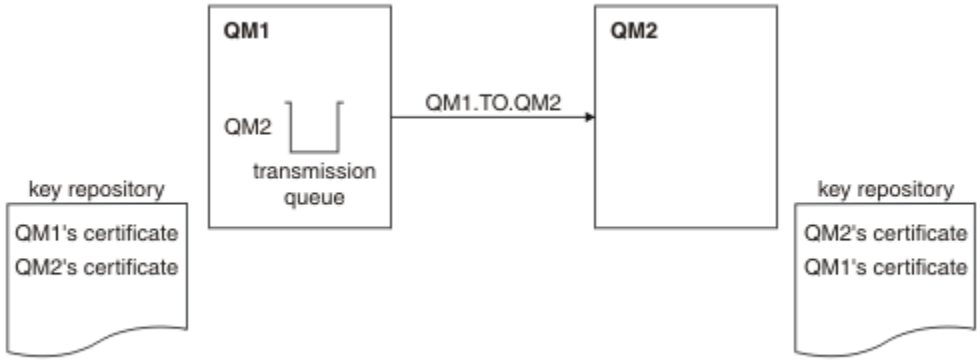



図 *20.* このタスクで実装する構成

207 ページの図 14 にあるとおり、QM1 の鍵リポジトリーには QM1 の証明書と QM2 の公開証明書を格納します。 QM2 の鍵リポジトリーには、QM2 の証明書と QM1 の公開証明書を格納します。

## 手順

1. オペレーティング・システムに従って、各キュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。
2. キュー・マネージャーごとに自己署名証明書を作成します。
   - UNIX、Linux、および Windows システムの場合。
3. 各証明書のコピーを取り出します。
   - UNIX、Linux、および Windows システムの場合。
4. FTP などのユーティリティーを使用して QM1 証明書の公開部分を QM2 システムに転送し、またその逆を実行します。
5. キュー・マネージャーごとにパートナーの証明書を鍵リポジトリーに追加します。
   - UNIX、Linux、および Windows システムの場合。
6. QM1 で以下の例のようなコマンドを実行して、送信側チャネルとそれに関連する伝送キューを定義します。

   ```
   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)
   ```

   この例では、CipherSpec RC4_MD5 を使用します。 チャネルの両端の CipherSpec は同じでなければなりません。
7. QM2 で以下の例のようなコマンドを実行して、受信側チャネルを定義します。

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   このチャネルの名前は、ステップ 6 で定義した送信側チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。
8. チャネルを開始します。

## タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、207 ページの図 14 です。

### 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

キュー・マネージャー QM2 から以下のコマンドを入力します。

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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( )
```

いずれの場合も、SSLPEER の値は、ステップ 2 で作成したパートナー証明書の DN の値と一致している必要があります。 この証明書は自己署名証明書なので、発行者の名前はピアの名前と一致しています。

SSLPEER はオプションです。 これを指定する場合は、パートナー証明書 (ステップ 2 で作成された) の DN が許可されるように値を設定する必要があります。 SSLPEER の使用について詳しくは、「WebSphere MQ rules for SSLPEER values」を参照してください。

## 2 つのキュー・マネージャーの相互認証への CA 署名証明書の使用

CA 署名 SSL または TLS 証明書を使用して 2 つのキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

### このタスクについて

シナリオ

- 安全に通信する必要のあるキュー・マネージャーが 2 つ (QMA と QMB) あります。 QMA と QMB の間で、相互認証を実行する必要があります。
- 将来的にはこのネットワークを実稼働環境で使用することを計画しているので、最初から CA 署名証明書を使用することにしました。

構成の結果は次のようになります。

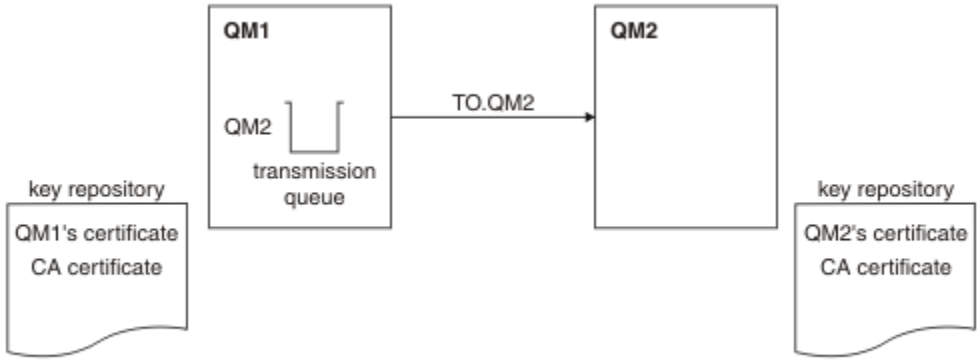



図 *21.* このタスクで実装する構成

209 ページの図 15 にあるとおり、QMA の鍵リポジトリーには、QMA の証明書と CA 証明書を格納します。 QMB の鍵リポジトリーには、QMB の証明書と CA 証明書を格納します。 この例では、QMA の証明書と QMB の証明書の両方が同じ CA から発行されました。 QMA の証明書と QMB の証明書が異なる CA から発行された場合は、QMA と QMB の鍵リポジトリーに両方の CA 証明書が含まれている必要があります。

## 手順

1. オペレーティング・システムに従って、各キュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。
2. キュー・マネージャーごとに CA 署名証明書を要求します。
   2 つのキュー・マネージャーで異なる CA を使用することができます。
   - UNIX、Linux、および Windows システムの場合。
3. キュー・マネージャーごとに認証局証明書を鍵リポジトリーに追加します。
   これらのキュー・マネージャーがそれぞれ異なる認証局を使用する場合は、各認証局の CA 証明書を両方の鍵リポジトリーに追加する必要があります。
   - UNIX、Linux、および Windows システムの場合。
4. キュー・マネージャーごとに CA 署名証明書を鍵リポジトリーに追加します。
   - UNIX、Linux、および Windows システムの場合。
5. QMA で以下の例のようなコマンドを発行して、送信側チャネルとそれに関連する伝送キューを定義します。

   ```
   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)
   ```

   この例では、CipherSpec RC4_MD5 を使用します。 チャネルの両端の CipherSpec は同じでなければなりません。
6. QMB で以下の例のようなコマンドを発行して、受信側チャネルを定義します。

   ```
   DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
   ```

   このチャネルの名前は、ステップ 6 で定義した送信側チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。
7. チャネルを開始します。

## タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、209 ページの図 15 です。

### 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QMA から以下のコマンドを入力します。

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

キュー・マネージャー QMB から以下のコマンドを入力します。

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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( )
```

いずれの場合も、SSLPEER の値は、ステップ 2 で作成したパートナー証明書の識別名 (DN) の値と一致している必要があります。 発行者名は、ステップ 4 で追加された個人証明書に署名した CA 証明書の識別名と一致します。

## 片方向認証による 2 つのキュー・マネージャーの接続

相互認証を使用するシステムを変更して、キュー・マネージャーが片方向認証で別のキュー・マネージャーに接続できるようにするには (つまり、SSL クライアントまたは TLS クライアントが証明書を送信しない場合)、以下のサンプル手順に従ってください。

### このタスクについて

シナリオ

- 209 ページの『2 つのキュー・マネージャーの相互認証への CA 署名証明書の使用』で、2 つのキュー・マネージャー (QM1 と QM2) がセットアップされました。
- 片方向認証を使用して QM2 に接続されるように QM1 を変更する必要があるとします。

構成の結果は次のようになります。

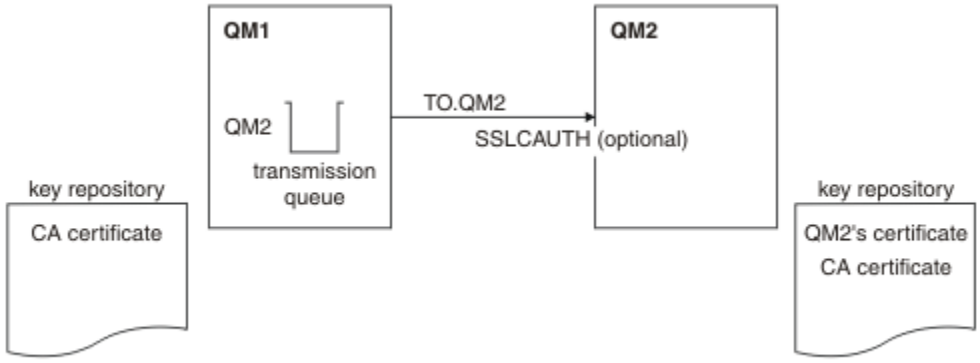



図 *22.* 片方向認証を許可するキュー・マネージャー

## 手順

1. オペレーティング・システムに従って、QM1 の個人証明書を鍵リポジトリーから削除します。
   - UNIX、Linux、および Windows システムの場合。 この証明書には、以下のようなラベルが付いています。
     - `ibmwebspheremq` の後に、小文字に変換されたキュー・マネージャーの名前が続きます。 例えば、QM1 の場合は、`ibmwebspheremqqm1` です。
2. オプション: QM1 で、SSL または TLS チャネルが以前に実行されている場合は、の説明に従って、SSL または TLS 環境をリフレッシュします。
3. 受信側で匿名接続を許可します。

## タスクの結果

変更した鍵リポジトリーとチャネルを図にまとめたのが、211 ページの図 16 です。

## 次のタスク

送信側チャネルの稼働中に REFRESH SECURITY TYPE(SSL) コマンドを実行した場合は (手順 2)、チャネルが自動的に再始動されます。 送信側チャネルが稼働していなかった場合は、送信側チャネルを始動します。

チャネルのサーバー側で、チャネル状況表示にピア名パラメーター値が表示される場合は、クライアント証明書が流れたことを意味します。

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

QM1 キュー・マネージャーから、次のコマンドを入力します。

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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 キュー・マネージャーから、次のコマンドを入力します。

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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( )
```

QM2 で SSLPEER フィールドが空になっています。これは、QM1 が証明書を送信しなかったことを示しています。 QM1 では、SSLPEER の値が QM2 の個人証明書の DN の値と一致しています。

# キュー・マネージャーへのクライアントのセキュア接続

SSL または TLS 暗号セキュリティー・プロトコルを使用するセキュア通信では、 通信チャネルをセットアップし、認証に使用するデジタル証明書を管理する必要があります。

SSL または TLS インストール環境をセットアップするには、SSL または TLS を使用するようにチャネルを定義する必要があります。 また、デジタル証明書を取得し、管理することも必要です。 テスト・システムでは、自己署名証明書か、ローカル認証局 (CA) で発行された証明書を使用できます。 実動システムでは、自己署名証明書を使用しないでください。 細については、 ../zs14140_.dita を参照してください。

証明書の作成と管理の詳細については、114 ページの『UNIX, Linux, and Windows システムでの SSL または TLS の取り扱い』を参照してください。

このトピック集では、SSL 通信のセットアップに関連したタスクを取り上げ、それらのタスクを実行するための段階的な手順を説明します。

また、プロトコルのオプション部分である SSL または TLS クライアント認証をテストすることもできます。 SSL または TLS ハンドシェーク中に、SSL または TLS クライアントは常にサーバーからデジタル証明書を取得し検証します。 WebSphere MQ の実装では、SSL または TLS サーバーは、常にクライアントから証明書を要求します。

UNIX, Linux, and Windows システムでは、SSL または TLS クライアントは、正しい WebSphere MQ 形式 (`ibmwebspheremq` の後に続けて、小文字にされたログオン・ユーザー ID。例えば、`ibmwebspheremqmyuserid`) のラベルが付いた証明書が存在する場合に限って証明書を送信します。

WebSphere MQ は、他の製品の証明書との混同を避けるために、`ibmwebspheremq` という接頭部をラベルに付けます。 証明書ラベル全体を小文字で指定してください。

SSL または TLS サーバーは、クライアント証明書が送信される場合は、常にそのクライアント証明書を検証します。 クライアントが証明書を送信しない場合に認証が失敗するのは、チャネルの SSL または TLS サーバー側の定義で、SSLCAUTH パラメーターが REQUIRED に設定されている場合、または SSLPEER パラメーター値が設定されている場合に限られます。 匿名でのキュー・マネージャーの接続について詳しくは、216 ページの『キュー・マネージャーへのクライアントの匿名接続』を参照してください。

## クライアントとキュー・マネージャーの相互認証への自己署名証明書の使用

自己署名 SSL または TLS 証明書を使用してクライアントとキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

### このタスクについて

シナリオ

- クライアント C1、およびキュー・マネージャー QM1 が存在し、これらは安全に通信する必要があります。 C1 と QM1 の間で、相互認証を実行する必要があります。
- そこで、自己署名証明書を使用して安全な通信をテストすることにしました。

IBM i での DCM は自己署名証明書をサポートしていないため、このタスクは IBM i システムでは適用されません。

構成の結果は次のようになります。

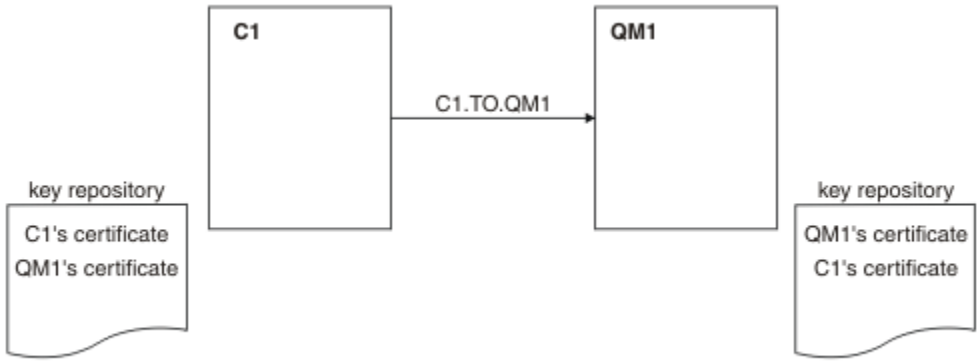



図 *23.* このタスクで実装する構成

213 ページの図 17 にあるとおり、QM1 の鍵リポジトリーには QM1 の証明書と C1 の公開証明書を格納します。 C1 の鍵リポジトリーには、C1 の証明書と QM1 の公開証明書を格納します。

## 手順

1. オペレーティング・システムに従って、クライアントとキュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。
2. クライアントおよびキュー・マネージャー用の自己署名証明書を作成します。
   - UNIX、Linux、および Windows システムの場合。
3. 各証明書のコピーを取り出します。
   - UNIX、Linux、および Windows システムの場合。
4. FTP などのユーティリティーを使用して C1 証明書の公開部分を QM1 システムに転送し、またその逆を実行しますを参照)。
5. パートナーの証明書を、クライアントとキュー・マネージャーの鍵リポジトリーに追加します。
   - UNIX、Linux、および Windows システムの場合。
6. キュー・マネージャーでコマンド REFRESH SECURITY TYPE(SSL) を実行します。
7. 以下のいずれかの方法で、クライアント接続チャネルを定義します。
   - C1 で MQSCO 構造を持つ MQCONNX 呼び出しを使用します (WebSphere MQ MQI クライアントでのクライアント接続チャネルの作成を参照)。
   - サーバーでのサーバー接続定義およびクライアント接続定義の作成 の説明に従って、クライアント・チャネル定義テーブルを使用する。
8. QM1 で以下の例のようなコマンドを実行して、サーバー接続チャネルを定義します。

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   このチャネルの名前は、ステップ 6 で定義したクライアント接続チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。

### タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、213 ページの図 17 です。

### 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

オプションで、チャネル定義の SSLPEER フィルター属性を設定できます。 チャネル定義 SSLPEER が設定されている場合、その値は、ステップ 2 で作成したパートナー証明書のサブジェクト DN と一致している必要があります。 接続が成功すると、DISPLAY CHSTATUS 出力の SSLPEER フィールドに、リモート・クライアント証明書の対象 DN が表示されます。

## クライアントとキュー・マネージャーの相互認証への CA 署名証明書の使用

CA 署名 SSL または TLS 証明書を使用してクライアントとキュー・マネージャーの間で相互認証を実装するための手順の例を取り上げます。

### このタスクについて

シナリオ

- クライアント C1、およびキュー・マネージャー QM1 が存在し、これらは安全に通信する必要があります。 C1 と QM1 の間で、相互認証を実行する必要があります。
- 将来的にはこのネットワークを実稼働環境で使用することを計画しているので、最初から CA 署名証明書を使用することにしました。

構成の結果は次のようになります。

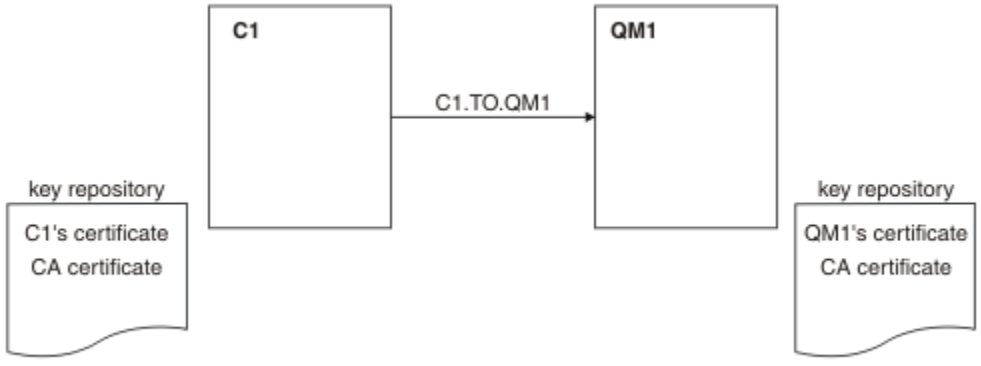



図 *24.* このタスクで実装する構成

215 ページの図 18 にあるとおり、C1 の鍵リポジトリーには、C1 の証明書と CA 証明書を格納します。 QM1 の鍵リポジトリーには、QM1 の証明書と CA 証明書を格納します。 この例では、C1 の証明書と QM1 の証明書の両方が同じ CA から発行されました。 C1 の証明書と QM1 の証明書が異なる CA から発行された場合は、C1 と QM1 の鍵リポジトリーに両方の CA 証明書が含まれている必要があります。

## 手順

1. オペレーティング・システムに従って、クライアントとキュー・マネージャーで鍵リポジトリーを準備します。
   - UNIX、Linux、および Windows システムの場合。
2. クライアントおよびキュー・マネージャー用の CA 署名証明書を要求します。
   クライアントとキュー・マネージャーで異なる CA を使用することがあります。
   - UNIX、Linux、および Windows システムの場合。
3. クライアントとキュー・マネージャーの鍵リポジトリーに認証局証明書を追加します。
   クライアントとキュー・マネージャーがそれぞれ異なる認証局を使用する場合は、各認証局の CA 証明書を両方の鍵リポジトリーに追加する必要があります。
   - UNIX、Linux、および Windows システムの場合。
4. クライアントとキュー・マネージャーの鍵リポジトリーに CA 署名証明書を追加します。
   - UNIX、Linux、および Windows システムの場合。
5. 以下のいずれかの方法で、クライアント接続チャネルを定義します。
   - C1 で MQSCO 構造を持つ MQCONNX 呼び出しを使用します (WebSphere MQ MQI クライアントでのクライアント接続チャネルの作成を参照)。
   - サーバーでのサーバー接続定義およびクライアント接続定義の作成 の説明に従って、クライアント・チャネル定義テーブルを使用する。
6. QM1 で以下の例のようなコマンドを実行して、サーバー接続チャネルを定義します。

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   このチャネルの名前は、ステップ 6 で定義したクライアント接続チャネルと同じ名前でなければならず、使用する CipherSpec も同じでなければなりません。

## タスクの結果

作成した鍵リポジトリーとチャネルを図にまとめたのが、215 ページの図 18 です。

## 次のタスク

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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 出力の SSLPEER フィールドには、ステップ 2 で作成されたリモート・クライアント証明書の識別名が表示されます。 発行者名は、ステップ 4 で追加された個人証明書に署名した CA 証明書の識別名と一致します。

## キュー・マネージャーへのクライアントの匿名接続

相互認証のシステムを変更して、キュー・マネージャーが別のキュー・マネージャーに匿名で接続できるようにするための手順の例を取り上げます。

### このタスクについて

シナリオ

- 214 ページの『クライアントとキュー・マネージャーの相互認証への CA 署名証明書の使用』で、キュー・マネージャーとクライアント (QM1 と C1) がセットアップされました。
- C1 が QM1 に匿名接続されるように変更する必要があるとします。

構成の結果は次のようになります。

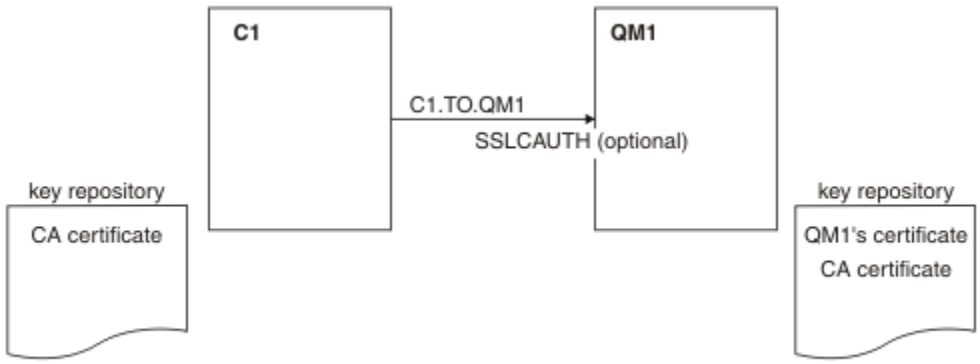



図 *25.* 匿名接続を可能にするクライアントとキュー・マネージャー

### 手順

1. オペレーティング・システムに従って、個人証明書を C1 の鍵リポジトリーから削除します。
   - UNIX、Linux、および Windows システムの場合。 この証明書には、以下のようなラベルが付いています。
     - `ibmwebspheremq` の後に、小文字に変換されたログオン・ユーザー ID が続きます (例: `ibmwebspheremqmyuserid`)。
2. クライアント・アプリケーションを再始動します。 あるいは、クライアント・アプリケーションを閉じ、すべての SSL または TLS 接続を再開します。
3. 次のコマンドを発行して、キュー・マネージャーでの匿名接続を許可します。

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

### タスクの結果

変更した鍵リポジトリーとチャネルを図にまとめたのが、216 ページの図 19 です。

### 次のタスク

チャネルのサーバー側で、チャネル状況表示にピア名パラメーター値が表示される場合は、クライアント証明書が流れたことを意味します。

DISPLAY コマンドを使用して、タスクが正常に完了していることを確認します。 タスクが正常に完了していれば、以下の例のような出力が表示されます。

キュー・マネージャー QM1 から以下のコマンドを入力します。

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

結果の出力は、以下の例のようになります。

```
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)
```

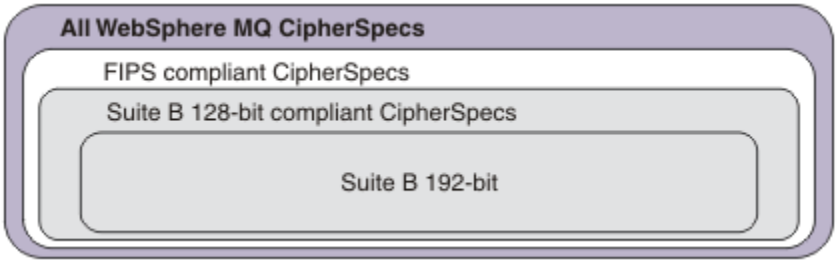
SSLCERTI フィールドと SSLPEER フィールドは空です。 これは、C1 が証明書を送信しなかったことを示しています。

## CipherSpec の指定

**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのいずれかで **SSLCIPH** パラメーターを使用して、 CipherSpec を指定します。

IBM WebSphere MQ とともに使用できる CipherSpec の一部は FIPS 準拠です。 それ以外 (NULL_MD5 など) は準拠していません。 同様に FIPS 準拠の CipherSpec の一部は Suite B 準拠でもありますが、それ以外は準拠していません。 Suite B 準拠の CipherSpec はすべて、FIPS 準拠でもあります。 Suite B 準拠の CipherSpec はすべて、128 ビット (ECDHE_ECDSA_AES_128_GCM_SHA256 など) と 192 ビット (ECDHE_ECDSA_AES_256_GCM_SHA384 など) の 2 つのグループに分けられます。

次の図は、これらのサブセットの関係を表しています。




次の表に、IBM WebSphere MQ SSL および TLS サポートとともに使用できる Cipher 仕様をリストします。 個人用証明書を要求するときに、公開鍵と秘密鍵のペアの鍵サイズを指定します。 SSL ハンドシェーク時に使用される鍵のサイズは、表の注記のとおり、CipherSpec によって決定されている場合を除き、証明書に保管されているサイズです。

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|
| NULL_MD5 [a] | SSL 3.0 | MD5 | なし | 0 | いいえ | いいえ | いいえ |
| NULL_SHA [a] | SSL 3.0 | SHA-1 | なし | 0 | いいえ | いいえ | いいえ |

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|
| RC4_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC4 | 40 | いいえ | いいえ | いいえ |
| RC4_MD5_US [a] | SSL 3.0 | MD5 | RC4 | 128 | いいえ | いいえ | いいえ |
| RC4_SHA_US [a] | SSL 3.0 | SHA-1 | RC4 | 128 | いいえ | いいえ | いいえ |
| RC2_MD5_EXPORT [2] [a] | SSL 3.0 | MD5 | RC2 | 40 | いいえ | いいえ | いいえ |
| DES_SHA_EXPORT [2] [a] | SSL 3.0 | SHA-1 | DES | 56 | いいえ | いいえ | いいえ |
| RC4_56_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | RC4 | 56 | いいえ | いいえ | いいえ |
| DES_SHA_EXPORT1024 [3] [b] | SSL 3.0 | SHA-1 | DES | 56 | いいえ | いいえ | いいえ |
| TLS_RSA_WITH_AES_128_CBC_SHA [a] | TLS 1.0 | SHA-1 | AES | 128 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_256_CBC_SHA [4] [a] | TLS 1.0 | SHA-1 | AES | 256 | はい | いいえ | いいえ |
| TLS_RSA_WITH_DES_CBC_SHA [a] | TLS 1.0 | SHA-1 | DES | 56 | いいえ [5] | いいえ | いいえ |
| FIPS_WITH_DES_CBC_SHA [B] | SSL 3.0 | SHA-1 | DES | 56 | いいえ [6] | いいえ | いいえ |
| TLS_RSA_WITH_AES_128_GCM_SHA256 [B] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_256_GCM_SHA384 [B] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_128_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 128 | はい | いいえ | いいえ |
| TLS_RSA_WITH_AES_256_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 256 | はい | いいえ | いいえ |
| ECDHE_ECDSA_RC4_128_SHA256 [B] | TLS 1.2 | SHA-1 | RC4 | 128 | いいえ | いいえ | いいえ |

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|
| ECDHE_RSA_RC4_128_SHA256 [B] | TLS 1.2 | SHA_1 | RC4 | 128 | いいえ | いいえ | いいえ |
| ECDHE_ECDSA_AES_128_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 128 | はい | いいえ | いいえ |
| ECDHE_ECDSA_AES_256_CBC_SHA384 [B] | TLS 1.2 | SHA-384 | AES | 256 | はい | いいえ | いいえ |
| ECDHE_RSA_AES_128_CBC_SHA256 [B] | TLS 1.2 | SHA-256 | AES | 128 | はい | いいえ | いいえ |
| ECDHE_RSA_AES_256_CBC_SHA384 [B] | TLS 1.2 | SHA-384 | AES | 256 | はい | いいえ | いいえ |
| ECDHE_ECDSA_AES_128_GCM_SHA256 [B] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | はい | はい | いいえ |
| ECDHE_ECDSA_AES_256_GCM_SHA384 [B] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | はい | いいえ | はい |
| ECDHE_RSA_AES_128_GCM_SHA256 [B] | TLS 1.2 | AEAD AES-128 GCM | AES | 128 | はい | いいえ | いいえ |
| ECDHE_RSA_AES_256_GCM_SHA384 [B] | TLS 1.2 | AEAD AES-256 GCM | AES | 256 | はい | いいえ | いいえ |
| TLS_RSA_WITH_NULL_SHA256 [B] | TLS 1.2 | SHA-256 | なし | 0 | いいえ | いいえ | いいえ |
| ECDHE_RSA_NULL_SHA256 [B] | TLS 1.2 | SHA-1 | なし | 0 | いいえ | いいえ | いいえ |
| ECDHE_ECDSA_NULL_SHA256 [B] | TLS 1.2 | SHA-1 | なし | 0 | いいえ | いいえ | いいえ |
| TLS_RSA_WITH_NULL_NULL [B] | TLS 1.2 | なし | なし | 0 | いいえ | いいえ | いいえ |
| TLS_RSA_WITH_RC4_128_SHA256 [B] | TLS 1.2 | SHA-1 | RC4 | 128 | いいえ | いいえ | いいえ |

| CipherSpec 名 | 使用されるプロトコル | MAC アルゴリズム | 暗号化アルゴリズム | 暗号化ビット数 | FIPS [1] | Suite B 128 ビット | Suite B 192 ビット |
|---|---|---|---|---|---|---|---|

**注 :**

1. FIPS 認定プラットフォーム上の FIPS 認定 CipherSpec であるかどうかを示しています。 FIPS の説明については、連邦情報処理標準 (FIPS) を参照してください。
2. ハンドシェークの最大鍵サイズは 512 ビットです。 SSL ハンドシェーク時に交換されるどちらかの証明書の鍵サイズが 512 ビットより大きい場合は、ハンドシェーク時に使用するための 512 ビットの一時鍵が生成されます。
3. ハンドシェークの鍵サイズは 1024 ビットです。
4. WebSphere MQ エクスプローラーが使用する JRE に対して適切な無制限のポリシー・ファイルが適用されていない場合には、この CipherSpec を使用して、WebSphere MQ エクスプローラーからキュー・マネージャーへの安全な接続を確立することはできません。
5. この CipherSpec は、2007 年 5 月 19 日より前は FIPS 140-2 で認証されていました。
6. この CipherSpec は、2007 年 5 月 19 日より前は FIPS 140-2 で認証されていました。 FIPS_WITH_DES_CBC_SHA という名前は歴史的なものであり、この CipherSpec がかつて FIPS 準拠であった という事実を反映しています (ただし、現在は準拠していません)。 この CipherSpec は非推奨となりました。使用することはお勧めしません。
7. この CipherSpec を使用して最大 32 GB までデータを転送できますが、それを超えるとエラー AMQ9288 を出して接続が終了します。 このエラーを回避するために、Triple-DES を使用しないか、またはこの CipherSpec を使用する際に秘密鍵リセットを有効にします。

**プラットフォームのサポート :**

- a サポート対象のすべてのプラットフォームで使用可能
- b UNIX, Linux, and Windows プラットフォームのみで使用可能

**関連概念**

34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』
このトピックでは、IBM WebSphere MQ における CipherSpec とデジタル証明書の関係を概説することにより、個々のセキュリティー・ポリシーに適した CipherSpec とデジタル証明書を選択する方法を説明します。

**関連資料**

DEFINE CHANNEL
ALTER CHANNEL

## IBM WebSphere MQ Explorer を使用した CipherSpecs に関する情報の取得

IBM WebSphere MQ Explorer を使用して、CipherSpec の説明を表示できます。

217 ページの『CipherSpec の指定』 の CipherSpec についての情報を取得するには、次の手順を実行してください。

1. **IBM WebSphere MQ エクスプローラー**を開き、**Queue Managers** フォルダーを展開する。
2. キュー・マネージャーを開始したことを確認する。
3. 処理したいキュー・マネージャーを選択して 「**チャネル**」をクリックします。
4. 処理するチャネルを右クリックし、「**プロパティー**」を選択する。
5. 「**SSL**」プロパティー・ページを選択する。
6. 処理したい CipherSpec を、リストの中から選択する。 リストの下のウィンドウに、説明が表示されます。

## CipherSpec の代替指定方法

オペレーティング・システムの SSL サポートを使用できるプラットフォームの場合は、システムで新しい CipherSpec に対応できる可能性があります。 新しい CipherSpec を指定するには、SSLCIPH パラメーターを使用しますが、指定する値は、ご使用のプラットフォームによって異なります。

**注 :** このセクションは、UNIX、Linux、または Windows システムには適用されません。これは、CipherSpec が WebSphere MQ 製品に付属しているので、新しい CipherSpec が出荷後に使用可能にならないからです。

オペレーティング・システムが SSL サポートを提供するプラットフォームの場合、ご使用のシステムが、217 ページの『CipherSpec の指定』 に含まれていない新しい CipherSpec をサポートする場合があります。新しい CipherSpec を指定するには、SSLCIPH パラメーターを使用しますが、指定する値は、ご使用のプラットフォームによって異なります。 いずれの場合も、CipherSpec の指定は、システムが実行している SSL のバージョンによってサポートされ、かつ有効である SSL CipherSpec と対応している必要があります。

**IBM i**

16 進値を表す 2 文字のストリング。

許可される値について詳しくは、該当する製品資料を参照してください ( IBM i 製品資料で *cipher_spec* を検索してください)。

次のように CHGMQMCHL コマンドまたは CRTMQMCHL コマンドを使用すると、値を指定できます。

```
CRTMQMCHL CHLNAME('channel name') SSLCIPH('hexadecimal value')
```

また、ALTER QMGR MQSC コマンドを使用して SSLCIPH パラメーターを設定することもできます。

**z/OS**

16 進値を表す 2 文字のストリング。 この 16 進コードは、SSL プロトコルで定義されている値に相当します。

詳しくは、「*z/OS Cryptographic Services System Secure Sockets Layer* プログラミング」(SD88-6252) の API リファレンスの章にある gsk_environment_open() の説明を参照してください。 ここには、2 桁の 16 進数コードの形式でサポートされるすべての SSL V3.0 および TLS V1.0 暗号仕様のリストがあります。

## WebSphere MQ クラスターの考慮事項

WebSphere MQ クラスターを使用する場合は、217 ページの『CipherSpec の指定』 にある CipherSpec 名を使用するのが無難です。 代替指定を使用する場合は、他のプラットフォームではその指定は無効な場合があることに注意してください。 詳細については、250 ページの『SSL とクラスター』 を参照してください。

## IBM WebSphere MQ MQI クライアント用の CipherSpec の指定

IBM WebSphere MQ MQI クライアントの CipherSpec を指定するには、3 つのオプションがあります。

以下のオプションがあります。

- チャネル定義テーブルを使用する
- MQCONNX 呼び出しで、MQCD_VERSION_7 以降の MQCD 構造の SSLCipherSpec フィールドを使用する。
- Active Directory を使用する (Active Directory サポートを備えた Windows システム上)

## IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS を使用した CipherSuite の指定

IBM WebSphere MQ classes for Java および IBM WebSphere MQ classes for JMS は、他のプラットフォームとは異なる方法で CipherSuites を指定します。

IBM WebSphere MQ classes for Java で CipherSuite を指定する方法については、 Secure Sockets Layer (SSL) サポートを参照してください。

IBM WebSphere MQ classes for JMS での CipherSuite の指定については、WebSphere MQ classes for JMS での Secure Sockets Layer (SSL) の使用を参照してください。

# 監査

イベント・メッセージを使用して、セキュリティー侵入あるいは侵入試行がないかどうかを調べることができます。 さらに、IBM WebSphere MQ Explorer を使用して、システムのセキュリティーを調べることもできます。

キュー・マネージャーへの接続など、許可されていないアクションを実行しようとしたり、あるいはキューにメッセージを書き込もうとしたりしていないか検出するには、キュー・マネージャーによって生成されるイベント・メッセージ (特に権限イベント・メッセージ) を調べます。 キュー・マネージャーのイベント・メッセージについて詳しくは、キュー・マネージャー・イベント、一般的なイベント・モニターについて詳しくは、イベント・モニターを参照してください。

# クラスターのセキュリティーの確保

キュー・マネージャーがクラスターを結合したり、クラスター・キュー上にメッセージを書き込んだりすることを許可あるいは禁止します。 キュー・マネージャーをクラスターから強制的に退去させます。 クラスター用に SSL を構成する場合、一部の追加の考慮事項を検討してください。

## 無許可キュー・マネージャーのメッセージ送信の停止

チャネル・セキュリティー出口を使用して、無許可キュー・マネージャーが自分のキュー・マネージャーにメッセージを送信できないようにします。

### 始める前に

クラスタリングは、セキュリティー出口が作動する方法には何の影響も与えません。 キュー・マネージャーへのアクセス権の制限は、分散キューイング環境で行うのと同じ方法で行えます。

### このタスクについて

選択したキュー・マネージャーが自分のキュー・マネージャーにメッセージを送信できないようにします。

### 手順

1. CLUSRCVR チャネル定義でチャネル・セキュリティー出口プログラムを定義します。
2. クラスター受信側チャネルでメッセージを送信しようとするキュー・マネージャーの認証を行い、無許可であればアクセスを拒否するプログラムを作成します。

### 次のタスク

チャネル・セキュリティー出口プログラムは、MCA の開始時および終了時に呼び出されます。

## 無許可キュー・マネージャーから自分のキューへのメッセージ書き込みの停止

クラスター受信側チャネルでチャネルの書き込み権限属性を使用して、無許可のキュー・マネージャーがキューにメッセージを書き込めないようにします。 リモート・キュー・マネージャーを許可するには、RACF (z/OS の場合) あるいは OAM (他のプラットフォームの場合) を使用してメッセージ内のユーザー ID を検査します。

### このタスクについて

プラットフォームのセキュリティー機能および WebSphere MQ のアクセス制御メカニズムを使用して、キューへのアクセスを制御します。

### 手順

1. 特定のキュー・マネージャーがメッセージをキューに書き込むのを防止するには、ご使用のプラットフォームで使用可能なセキュリティー機能を使用します。

   以下に例を示します。

   - RACF またはその他の外部セキュリティー・マネージャー (WebSphere MQ for z/OS の場合)
   - オブジェクト権限マネージャー (OAM) (他のプラットフォームの場合)。
2. CLUSRCVR チャネル定義で書き込み権限 (PUTAUT) 属性を使用します。

   PUTAUT 属性により、メッセージをキューに書き込むための権限を設定するために使用するユーザー ID を指定できます。

   PUTAUT 属性のオプションは次のとおりです。

   **DEF**
   デフォルトのユーザー ID を使用します。 z/OS の場合、この検査では、このネットワークから受け取ったユーザー ID および MCAUSER から派生したユーザー ID の両方が使用されます。

   **CTX**
   メッセージに関連したコンテキスト情報に含まれるユーザー ID を使用します。 z/OS では、この検査では、ネットワークから受け取ったユーザー ID、または MCAUSER から派生したユーザー ID のいずれか、あるいはその両方が使用されます。 リンクが信頼でき、かつ認証されている場合に、このオプションを使用します。

   **ONLYMCA (z/OS のみ)**
   DEF と同様ですが、ネットワークから受け取るユーザー ID は使用されません。 リンクが信頼できない場合に、このオプションを使用します。 そのリンクに対して特定の操作 (MCAUSER に定義される) のセットだけを許可します。

   **ALTMCA (z/OS のみ)**
   CTX と同様ですが、ネットワークから受け取るユーザー ID は使用されません。

## リモート・クラスター・キューへのメッセージ書き込み権限の付与

使用するプラットフォームで、キュー・マネージャーへの接続と、そのキュー・マネージャーのキューに対する書き込みのためのアクセス権限を与えます。

### このタスクについて

デフォルトの動作では、SYSTEM.CLUSTER.TRANSMIT.QUEUE に対するアクセス制御を実行します。 この動作は、複数の伝送キューを使用している場合でも適用されることに注意してください。

このトピックで説明する特定の動作は、 「セキュリティー・スタンザ」 トピックで説明されているように qm.ini ファイル内の **ClusterQueueAccessControl** 属性を *RQMName* に構成し、キュー・マネージャーを再始動した場合にのみ適用されます。

### 手順

- UNIX、 Linux 、および Windows システムの場合は、以下のコマンドを発行します。

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
  ```

  ユーザーは指定されたクラスター・キューにのみメッセージを書き込むことができ、他のクラスター・キューには書き込めません。

  変数名の意味は次のとおりです。

  **QMgrName**
  キュー・マネージャーの名前。

**GroupName**
アクセス権を付与されるグループの名前。

**QueueName**
権限を変更するキューまたは総称プロファイルの名前。

### 次のタスク

クラスター・キューでメッセージを書き込む際、応答先キューを指定する場合は、応答を送信する権限がコンシューム・アプリケーションに必要です。 193 ページの『リモート・クラスター・キューにメッセージを書き込むための権限の付与』の説明に従って、この権限を設定してください。

**関連情報**
qm.ini ファイル内の Security スタンザ

## キュー・マネージャーのクラスターへの参加の防止

キュー・マネージャーがクラスターに参加する場合、受け取らせたくないメッセージをキュー・マネージャーが受け取れないようにするのは困難です。

### 手順

特定の許可キュー・マネージャーのみがクラスターに参加できるようにする場合、以下の 3 つの技法の選択肢があります。

- チャネル認証レコードを使用して、リモート・システムによって提供されるリモート IP アドレス、リモート・キュー・マネージャー名、または SSL/TLS 識別名に基づいて、クラスター・チャネル接続をブロックできます。
- 権限のないキュー・マネージャーが SYSTEM.CLUSTER.COMMAND.QUEUE に書き込むことを防止するための出口プログラムを作成します。 SYSTEM.CLUSTER.COMMAND.QUEUE へのアクセス権を制限して、どのキュー・マネージャーもそれに書き込むことができないようにはしないでください。もしそうするなら、どのキュー・マネージャーもクラスターに参加できなくなります。
- CLUSRCVR チャネル定義でのチャネル・セキュリティー出口プログラム。

## クラスター・チャネルでのセキュリティー出口

クラスター・チャネルでセキュリティー出口を使用する場合の追加の考慮事項

### このタスクについて

クラスター送信側チャネルは、初めて始動する時に、システム管理者が手動で定義した属性を使用します。チャネルが停止および再始動する時には、対応するクラスター受信側チャネル定義から属性を取り出します。 元のクラスター送信側チャネル定義は、SecurityExit 属性も含め、新規の属性で上書きされます。

### 手順

1. チャネルのクラスター送信側およびクラスター受信側の両方で、セキュリティー出口を定義する必要があります。

   セキュリティー出口名はクラスター受信側定義から送信されますが、それでも初期接続はセキュリティー出口ハンドシェークによって確立する必要があります。
2. セキュリティー出口の MQCXP 構造体で PartnerName を検証します。

   出口は、パートナーのキュー・マネージャーに権限がある場合にのみ、チャネルを始動する許可を与える必要があります。
3. クラスター受信側定義でセキュリティー出口を受信側から開始するよう設計します。
4. 送信側から開始するよう設計すると、セキュリティー検査が実行されないため、セキュリティー出口を持たない無許可のキュー・マネージャーがクラスターに参加できることになります。

チャネルの停止と再始動が済んではじめて、SCYEXIT 名をクラスター受信側定義から送信でき、十分なセキュリティー検査を実行できます。

5. 現在使用されているクラスター送信側チャネル定義を表示するには、次のコマンドを使用します。

```
DISPLAY CLUSQMGR(queue manager) ALL
```

このコマンドは、クラスター受信側定義から送信された属性を表示します。

6. 元の定義を表示するには、次のコマンドを使用します。

```
DISPLAY CHANNEL(channel name) ALL
```

7. それぞれのキュー・マネージャーが異なるプラットフォーム上にある場合には、クラスター送信側キュー・マネージャーにチャネルの自動定義出口 (CHADEXIT) を定義する必要があります。

チャネルの自動定義出口を使用して、SecurityExit 属性を宛先プラットフォームに適合する形式に設定します。

8. セキュリティー出口をデプロイおよび構成します。

Windows UNIX Linux **Windows、 UNIX and Linux システム**

- セキュリティー出口のダイナミック・リンク・ライブラリーは、チャネル定義の SCYEXIT 属性で指定されたパスになければなりません。
- チャネル自動定義出口のダイナミック・リンク・ライブラリーは、キュー・マネージャー定義の CHADEXIT 属性で指定されたパスになければなりません。

# 不必要なキュー・マネージャーをクラスターから退去させる

完全リポジトリー・キュー・マネージャーで RESET CLUSTER コマンドを実行することによって、不必要なキュー・マネージャーをクラスターから退去させます。

## このタスクについて

不必要なキュー・マネージャーをクラスターから退去させることができます。 これは例えば、あるキュー・マネージャーが削除されたが、そのクラスター受信側チャネルが引き続きそのクラスターに定義されているような場合に実行します。 タイディアップ (整理) を行うこともできます。

完全リポジトリー・キュー・マネージャーだけがクラスターからのキュー・マネージャーの排除を許可されます。

以下の手順を実行して、キュー・マネージャー OSLO をクラスター NORWAY から排除します。

## 手順

1. 完全リポジトリー・キュー・マネージャーで、以下のコマンドを実行します。

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. あるいは、以下のようにコマンド内で QMNAME ではなく QMID を使用します。

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

## タスクの結果

強制的に除去されるキュー・マネージャーは変更されず、そのローカル・クラスター定義ではまだクラスターに含まれているように表示されます。 他のすべてのキュー・マネージャーの定義では、これはクラスターに含まれているように表示されません。

# キュー・マネージャーのメッセージ受信の防止

出口プログラムを使用することによって、受信する権限のないメッセージをクラスター・キュー・マネージャーが受信できないようにすることができます。

## このタスクについて

クラスターのメンバーとなっているキュー・マネージャーによるキューの定義を防ぐのは困難です。 悪質なキュー・マネージャーがクラスターに加わり、クラスター内のいずれかのキューのインスタンスを独自に定義してしまう、という危険性があります。 こうなると、受信する権限がないメッセージを受信できるようになってしまいます。 キュー・マネージャーがメッセージを受信するのを防ぐために、手順で指定されている以下のいずれかのオプションを使用します。

## 手順

- それぞれのクラスター送信側チャネル上のチャネル出口プログラム。 この出口プログラムは、接続名を使用して、メッセージ送信の宛先となるキュー・マネージャーが適正かを判別します。
- クラスター・ワークロード出口プログラム。これは、宛先レコードを使用して、メッセージの送信先となる宛先キューおよびキュー・マネージャーが適正かを判別します。

# SSL とクラスター

クラスターの SSL を構成する場合、CLUSRCVR チャネル定義は自動定義 CLUSSDR チャネルとして他のキュー・マネージャーに伝搬されることにご注意ください。 CLUSRCVR チャネルが SSL を使用する場合、チャネルを使用して通信するすべてのキュー・マネージャー上に SSL を構成する必要があります。

SSL の詳細については、WebSphere MQ での SSL および TLS のサポートを参照してください。 その資料中のアドバイスは一般にクラスター・チャネルにあてはまりますが、特に以下の事柄を考慮する必要があります。

IBM WebSphere MQ クラスターでは、特別な CLUSRCVR チャネル定義が、他の多数のキュー・マネージャーに頻繁に伝搬されます。伝搬先でのそのチャネル定義は、自動定義 CLUSSDR に変換されます。 その後、自動定義 CLUSSDR が使用されて、CLUSRCVR へのチャネルが始動します。 CLUSRCVR が SSL 接続用に構成されている場合、以下の考慮事項が適用されます。

- この CLUSRCVR との通信を希望するすべてのキュー・マネージャーには、SSL サポートへのアクセス権が必要です。 この SSL プロビジョンは、チャネル用の CipherSpec をサポートする必要があります。
- 自動定義クラスター送信側チャネルの伝搬先である各種キュー・マネージャーには、それぞれ異なる識別名が関連付けられています。 識別名の対等検査が CLUSRCVR で使用される場合には、受信する可能性のあるすべての識別名が正しくマッチングされるようにセットアップする必要があります。

  例えば、特定の CLUSRCVR に接続するクラスター送信側チャネルをホストするキュー・マネージャーすべてに、関連する証明書があるとします。 また、これらの証明書すべてにある識別名が、国を UK、組織を IBM、組織単位を IBM WebSphere MQ Development、と定義しており、すべてに DEVT.QMnnn (nnn は数値) という形式の共通名があるとします。

  この場合、CLUSRCVR 上の C=UK, O=IBM, OU=WebSphere MQ Development, CN=DEVT.QM* という SSLPEER 値により、必要なすべてのクラスター送信側チャネルが正常に接続される一方で、不要なクラスター送信側チャネルは接続を妨げられます。
- カスタム CipherSpec ストリングが使用される場合、カスタム・ストリング・フォーマットが必ずしもすべてのプラットフォームで使用できるわけではないことにご注意ください。 例えば、CipherSpec ストリング RC4_SHA_US は、IBM i 上では値 05 ですが、UNIX、Linux または Windows システム上では有効な仕様ではありません。 そのため、カスタム SSLCIPH パラメーターが CLUSRCVR で使用される場合、作成されるすべての自動定義クラスター送信側チャネルは、基盤 SSL サポートがこの CipherSpec を実装し、かつそれをカスタム値で指定できるプラットフォーム上に存在する必要があります。 クラスター全体で理解される SSLCIPH パラメーターの値を選択できない場合には、チャネル自動定義出口によって、使用しているプラットフォームが理解できるものに変更する必要があります。 できれば、テキスト形式の CipherSpec ストリングを使用してください (例えば RC4_MD5_US)。

SSLCRLNL パラメーターは、個々のキュー・マネージャーに適用されますが、クラスター内の他のキュー・マネージャーには伝搬しません。

# クラスター化されたキュー・マネージャーおよびチャネルの SSL へのアップグレード

CLUSSDR チャネルの前にすべての CLUSRCVR チャネルを変更して、クラスター・チャネルを一度に 1 つずつアップグレードします。

## 始める前に

クラスター用の CipherSpec の選択に影響する可能性があるので、以下の考慮事項を検討してください。

- 一部のプラットフォームでは使用できない CipherSpec もあります。 クラスター内のすべてのキュー・マネージャーでサポートされている CipherSpec を選択するよう注意してください。
- 現行の WebSphere MQ リリースの新機能として提供されている CipherSpec については、旧リリースではサポートされない場合があります。 クラスターに含まれているキュー・マネージャーが異なる複数の MQ リリースで実行されている場合、クラスターでは、各リリースでサポートされている CipherSpec のみ使用できます。

  クラスター内で新しい CipherSpec を使用するには、最初にすべてのクラスター・キュー・マネージャーを現行リリースにマイグレーションする必要があります。
- CipherSpec によっては (特に、楕円曲線暗号を使用している場合)、特定のタイプのデジタル証明書を使用する必要があります。

クラスター内のすべてのキュー・マネージャーを WebSphere MQ V6 以降にアップグレードします (まだそのレベルになっていない場合)。 それぞれのキュー・マネージャーから SSL が作動するよう、証明書および鍵を配布します。

## このタスクについて

CLUSRCVR は一度に 1 つずつ変更し、変更がクラスター全体に行き渡ってから、次のものの変更を始めてください。 切り替え中のチャネルの変更がクラスター全体に広がるまで、リバース・パスは絶対に変更しないでください。

## 手順

1. CLUSRCVR チャネルを任意の順番で SSL に切り替えます。

   変更は、SSL に変更されていないチャネル上を反対方向に流れます。
2. すべての手動 CLUSSDR チャネルを SSL に切り替えます。

   このことは、REFRESH CLUSTER コマンドを REPOS(YES) オプションを指定して使用しない限り、クラスターの操作に影響を与えません。

   **注 :** 大規模クラスターでは、稼働中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、そのクラスターに悪影響が及ぶ可能性があります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性がある を参照してください。

**関連概念**

217 ページの『CipherSpec の指定』
**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのいずれかで **SSLCIPH** パラメーターを使用して、 CipherSpec を指定します。

34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』
このトピックでは、IBM WebSphere MQ における CipherSpec とデジタル証明書の関係を概説することにより、個々のセキュリティー・ポリシーに適した CipherSpec とデジタル証明書を選択する方法を説明します。

**関連情報**
クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

### クラスター化されたキュー・マネージャーおよびチャネルで SSL または TLS を無効にする

SSL または TLS をオフにするには、 SSLCIPH パラメーターを ' 'に設定します。 すべてのクラスター受信側のチャネルを変更してからクラスター送信側チャネルを変更して、クラスター・チャネル上の TLS を個別に無効にします。

#### このタスクについて

クラスター受信側チャネルは一度に 1 つずつ変更し、変更がクラスター全体に行き渡ってから、次のものの変更を始めてください。

**重要 :** 切り替え中のチャネルの変更がクラスター全体に広がるまで、リバース・パスは決して変更しないでください。

#### 手順

1. SSLCIPH パラメーターの値を ' '、単一引用符で囲んだ空ストリング に設定します。

   クラスター受信側チャネル上の SSL または TLS は任意の順番でオフにすることができます。

   変更は、SSL または TLS がアクティブのままになっているチャネル上を反対方向に流れることに注意してください。
2. **DISPLAY CLUSQMGR(*)** ALL コマンドを使用して、その他のすべてのキュー・マネージャーで新しい値が反映されているか確認します。
3. すべての手動クラスター送信側チャネル上の SSL または TLS をオフにします。

   このことは、**REFRESH CLUSTER** コマンドを REPOS(YES) オプションを指定して使用しない限り、クラスターの操作に影響を与えません。

   大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが定期的な間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。 詳細については、大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。
4. クラスター送信側チャネルを停止してから再始動します。

## パブリッシュ/サブスクライブのセキュリティー

パブリッシュ/サブスクライブに関係するコンポーネントおよび相互作用について、概要を示し、その後に詳細な説明と例を示します。

トピックへのパブリッシュ/サブスクライブには多くのコンポーネントが関わっています。 それらの間のセキュリティー関係のいくつかを 253 ページの図 26 に示し、続いて例を挙げて説明します。

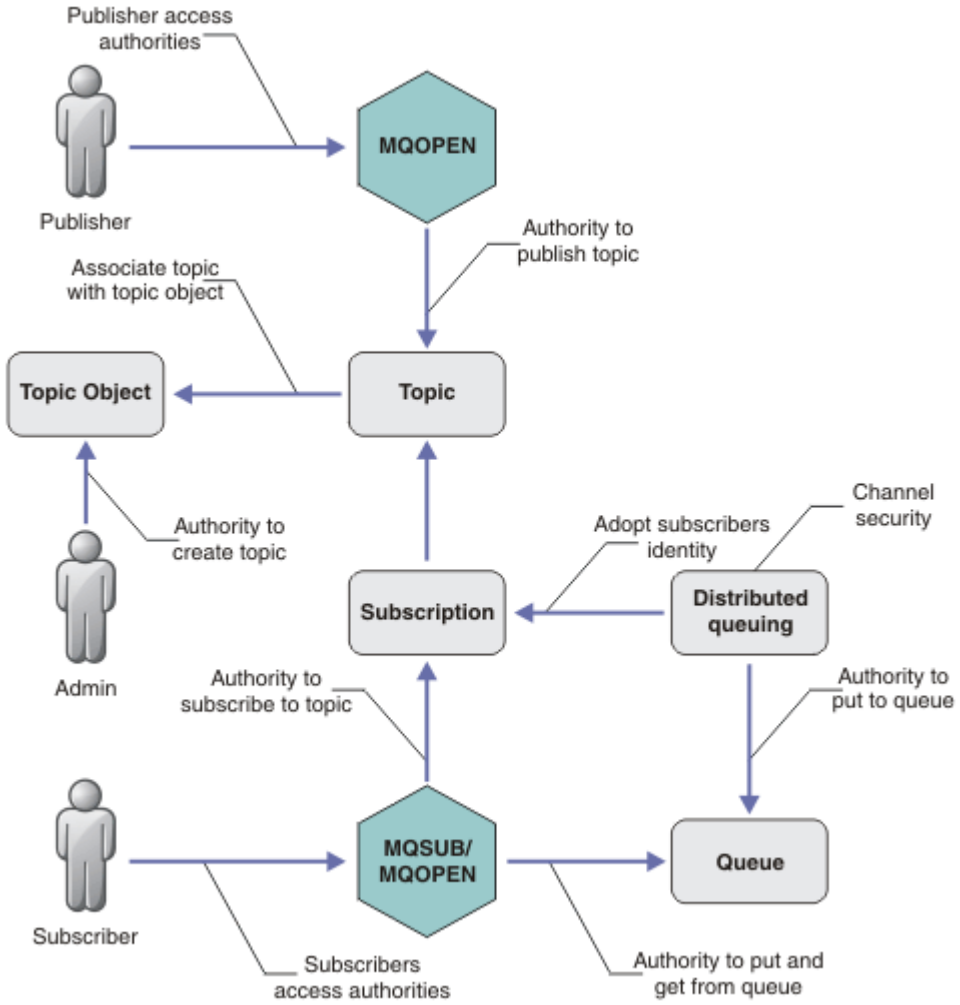



図 *26.* パブリッシュ/サブスクライブのセキュリティーの関係

**トピック**

トピックはトピック・ストリングによって識別され、通常はツリーに編成されます。「トピック・ツリー」 を参照してください。 トピックへのアクセスを制御するために、トピックとトピック・オブジェクトを関連付ける必要があります。255 ページの『トピック・セキュリティー・モデル』 トピック・オブジェクトを使用してトピックを保護する方法について説明します。

**管理トピック・オブジェクト**

管理トピック・オブジェクトのリストを指定したコマンド **setmqaut** を使用して、トピックにアクセスする人および目的を制御できます。 260 ページの『トピックにサブスクライブするアクセス権限をユーザーに付与する』および 265 ページの『トピックにパブリッシュするアクセス権限をユーザーに付与する』の例を参照してください。

**サブスクリプション**

トピック・ストリングを提供するサブスクリプションを作成することによって、1 つ以上のトピックにサブスクライブします。このトピック・ストリングにはワイルドカードを含めることができ、パブリケーションのトピック・ストリングに対して突き合わせを行います。 詳細については、以下を参照してください。

**トピック・オブジェクトを使用したサブスクライブ**

256 ページの『トピック・オブジェクト名を使用したサブスクライブ操作』

**トピックを使用したサブスクライブ**

257 ページの『トピック・ストリングを使用したサブスクライブ操作 (トピック・ノードが存在しない場合)』

**ワイルドカードが含まれているトピックを使用したサブスクライブ**
258 ページの『ワイルドカード文字を含むトピック・ストリングを使用したサブスクライブ操作』

サブスクリプションには、サブスクライバーの ID、パブリケーションを配置する宛先キューの ID についての情報が含まれます。 また、パブリケーションを宛先キューに配置する方法についての情報も含まれます。

特定のトピックにサブスクライブする権限を持つサブスクライバーを定義するだけでなく、サブスクリプションが個別のサブスクライバーにより使用されるよう制限することもできます。 パブリケーションを宛先キューに配置するときに、キュー・マネージャーがサブスクライバーに関するどの情報を使用するかも制御できます。 270 ページの『サブスクリプションのセキュリティー』を参照してください。

## キュー

宛先キューは保護のための重要なキューです。 このキューはサブスクライバーに対してローカルに位置し、サブスクリプションに一致したパブリケーションがこのキューに入れられます。 宛先キューへのアクセスは、次の 2 つの観点から検討する必要があります。

1. 宛先キューへのパブリケーションの配置。
2. 宛先キューからのパブリケーションの取得。

キュー・マネージャーは、サブスクライバーから提供される ID を使ってパブリケーションを宛先キューに書き込みます。 パブリケーションの取得タスクを代行しているサブスクライバーまたはプログラムが、メッセージをキューから取り出します。 258 ページの『宛先キューに対する権限』を参照してください。

トピック・オブジェクトの別名はありませんが、トピック・オブジェクトの別名として別名キューを使用できます。 使用する場合、キュー・マネージャーは、パブリッシュまたはサブスクライブ用にトピックを使用する権限を検査するだけでなく、キューを使用する権限も検査します。

## キュー・マネージャー間におけるパブリッシュ/サブスクライブのセキュリティー

トピックにパブリッシュまたはサブスクライブする権限は、ローカル・キュー・マネージャーでローカル ID および許可を使用して検査されます。 許可は、トピックが定義されているかどうかにも、どこに定義されているかにも左右されません。 したがって、クラスター化されたトピックを使用するときには、クラスター内のキュー・マネージャーごとにトピックの許可を実行する必要があります。

**注 :** トピックのセキュリティー・モデルは、キューのセキュリティー・モデルとは異なります。 クラスター化されたキューごとにローカルにキュー別名を定義することで、キューについて同じ結果が得られます。

キュー・マネージャーは、クラスター内でサブスクリプションを交換します。 ほとんどの WebSphere MQ クラスター構成では、チャネルは PUTAUT=DEF で構成されており、チャネル・プロセスの権限を使用してターゲット・キューにメッセージを配置します。 PUTAUT=CTX を使用するようにチャネル構成を変更し、クラスター内の別のキュー・マネージャーにサブスクリプションを伝搬するための権限をサブスクライブ・ユーザーが持つように要求できます。

キュー・マネージャー間におけるパブリッシュ/サブスクライブのセキュリティーでは、クラスター内の他のサーバーにサブスクリプションを伝搬できるユーザーを制御するためにチャネル定義を変更する方法について説明しています。

## 許可

キューおよび他のオブジェクトと同様にトピック・オブジェクトに許可を適用できます。 トピックのみに適用できる許可操作には pub、sub、および resume の 3 つがあります。 詳細については、種々のオブジェクト・タイプについての権限の指定で説明しています。

## 関数呼び出し

パブリッシュおよびサブスクライブのプログラムでは、キュー型プログラムと同様に、オブジェクトがオープン、作成、変更、または削除されるときに許可検査が行われます。 MQPUT または MQGET MQI 呼び出しによってパブリケーションの配置および取得を行う場合は、検査は行われません。

トピックをパブリッシュするには、トピック上で MQOPEN を実行します (そこで許可検査が実行されます)。 MQPUT コマンドを使ってメッセージをトピック・ハンドルにパブリッシュします (このコマンドは許可検査を実行しません)。

トピックにサブスクライブするには、通常は MQSUB コマンドを実行して、サブスクリプションを作成または再開し、パブリケーションを受け取る宛先キューもオープンします。 あるいは、別の MQOPEN を実行して宛先キューをオープンしてから、MQSUB を実行してサブスクリプションを作成または再開します。

いずれの呼び出しを使用しても、キュー・マネージャーは、ユーザーがトピックにサブスクライブして、生成されるパブリケーションを宛先キューから取得できるかどうかを検査します。 宛先キューが非管理対象である場合も、キュー・マネージャーが宛先キューにパブリケーションを配置できるかどうかの許可検査が行われます。 この場合、一致するサブスクリプションから採用された ID が使われます。 これは、キュー・マネージャーが管理対象の宛先キューにパブリケーションを常に配置できることを前提としています。

**ロール**

ユーザーは、パブリッシュ/サブスクライブ・アプリケーションの実行時に次の 4 つの役割を果たします。

1. パブリッシャー
2. サブスクライバー
3. トピック管理者
4. WebSphere MQ 管理者 - グループ mqm のメンバー

パブリッシュ、サブスクライブ、およびトピック管理役割に対応する適切な許可を使ってグループを定義してください。 その後、プリンシパルをこれらのグループに割り当てることで、特定のパブリッシュ/サブスクライブ・タスクの実行を許可できます。

さらに、パブリケーションとサブスクリプションの移動を行うキューとチャネルの管理者にまで、管理操作の許可を拡張する必要があります。

## トピック・セキュリティー・モデル

定義済みのトピック・オブジェクトだけが、関連するセキュリティー属性を持つことができます。 トピック・オブジェクトの説明については、「管理トピック・オブジェクト」 を参照してください。 セキュリティー属性では、指定のユーザー ID またはセキュリティー・グループが、それぞれのトピック・オブジェクトに対するサブスクライブ操作またはパブリッシュ操作を実行する権限を持っているかどうかを指定します。

セキュリティー属性は、トピック・ツリー内の適切な管理ノードに関連付けられます。 サブスクライブ操作中またはパブリッシュ操作中に特定のユーザー ID に関する権限検査が行われる場合、関連するトピック・ツリー・ノードのセキュリティー属性に基づいて権限が付与されます。

セキュリティー属性はアクセス制御リストであり、特定のオペレーティング・システム・ユーザー ID またはセキュリティー・グループが、トピック・オブジェクトに対して、どの権限を持っているかを示します。

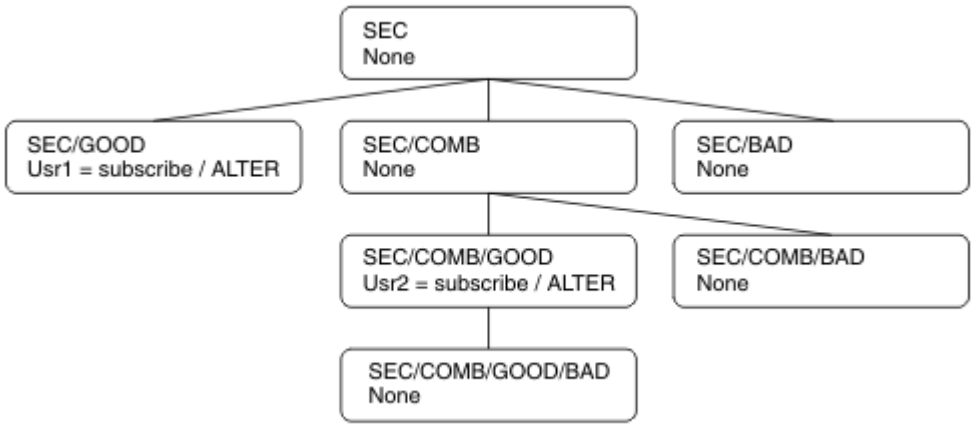
以下のようにセキュリティー属性または権限が定義されたトピック・オブジェクトの例を考えてみます。

表 *17.* トピック・オブジェクトの権限の例

| トピック名 | トピック・ストリング | 権限 - z/OS 以外 | z/OS の権限 |
|---|---|---|---|
| SECROOT | SEC | なし | なし |
| SECGOOD | SEC/GOOD | usr1+subscribe | ALTER<br>HLQ.SUBSCRIBE.SECGOOD |
| SECBAD | SEC/BAD | なし | なし<br>HLQ.SUBSCRIBE.SECBAD |
| SECCOMB | SEC/COMB | なし | なし<br>HLQ.SUBSCRIBE.SECCOMB |

| 表 17. トピック・オブジェクトの権限の例 (続き) | | | |
|---|---|---|---|
| **トピック名** | **トピック・ストリング** | **権限 - z/OS 以外** | **z/OS の権限** |
| SECCOMBB | SEC/COMB/GOOD/BAD | なし | なし<br>HLQ.SUBSCRIBE.SECCOMBB |
| SECCOMBG | SEC/COMB/GOOD | usr2+subscribe | ALTER<br>HLQ.SUBSCRIBE.SECCOMBG |
| SECCOMBN | SEC/COMB/BAD | なし | なし<br>HLQ.SUBSCRIBE.SECCOMBN |

各ノードにセキュリティー属性を関連付けたトピック・ツリーを図で表すと、以下のようになります。




この例では、権限は以下のようになっています。

- ツリー /SEC のルート・ノードでは、そのノードに対する権限を持っているユーザーはいません。
- usr1 は、オブジェクト /SEC/GOOD に対するサブスクライブ権限を付与されています。
- usr2 は、オブジェクト /SEC/COMB/GOOD に対するサブスクライブ権限を付与されています。

## トピック・オブジェクト名を使用したサブスクライブ操作

MQCHAR48 名を指定してトピック・オブジェクトにサブスクライブする場合、トピック・ツリー内の該当するノードが検出されます。 そのノードに関連付けられているセキュリティー属性で、そのユーザーがサブスクライブする権限を持っていることが確認されれば、アクセスが認められます。

ユーザーがアクセスを認められない場合は、ツリー内の親ノードで、そのユーザーが親ノード・レベルでサブスクライブする権限を持っているかどうかが確認されます。 持っていれば、アクセスが認められます。持っていない場合、そのノードの親で権限の確認が行われます。 サブスクライブ権限をユーザーに付与するノードが見つかるまで、再帰が続行されます。 ルート・ノードで権限の確認が行われても権限が付与されない場合、再帰が停止します。 この場合、アクセスが拒否されます。

つまり、パス内のいずれかのノードで、そのユーザーまたはアプリケーションにサブスクライブ権限を付与する場合は、サブスクライバーが、そのノードまたはトピック・ツリー内でそのノードよりも下位にあるすべてのノードにサブスクライブすることが許可されます。

例では、SEC がルート・ノードになっています。

ユーザーにサブスクライブ権限が付与されるのは、アクセス制御リストで、そのユーザー ID 自身が権限を持っているか、そのユーザー ID がメンバーであるオペレーティング・システムのセキュリティー・グループが権限を持っていることが示されている場合です。

例えば、以下のようになります。

- usr1 がトピック・ストリング SEC/GOOD を使用してサブスクライブしようとした場合は、そのユーザー ID がそのトピックに関連付けられているノードに対するアクセス権限を持っているので、そのサブスクリプションは許可されます。 一方、usr1 がトピック・ストリング SEC/COMB/GOOD を使用してサブスクライブしようとした場合は、そのユーザー ID がそのトピックに関連付けられているノードに対するアクセス権限を持っていないので、そのサブスクリプションは許可されません。
- usr2 がトピック・ストリング SEC/COMB/GOOD を使用してサブスクライブしようとした場合は、そのユーザー ID がそのトピックに関連付けられているノードに対するアクセス権限を持っているので、そのサブスクリプションは許可されます。 一方、usr2 が SEC/GOOD にサブスクライブしようとした場合は、そのユーザー ID がそのトピックに関連付けられているノードに対するアクセス権限を持っていないので、そのサブスクリプションは許可されません。
- usr2 がトピック・ストリング SEC/COMB/GOOD/BAD を使用してサブスクライブしようとした場合は、そのユーザー ID が親ノード SEC/COMB/GOOD に対するアクセス権限を持っているので、そのサブスクリプションは許可されます。
- usr1 または usr2 がトピック・ストリング /SEC/COMB/BAD を使用してサブスクライブしようとした場合は、そのどちらも、そのトピックに関連付けられているトピック・ノードに対しても、そのトピックの親ノードに対してもアクセス権限を持っていないので、いずれのサブスクリプションも許可されません。

存在しないトピック・オブジェクトの名前を指定したサブスクライブ操作は、MQRC_UNKNOWN_OBJECT_NAME エラーになります。

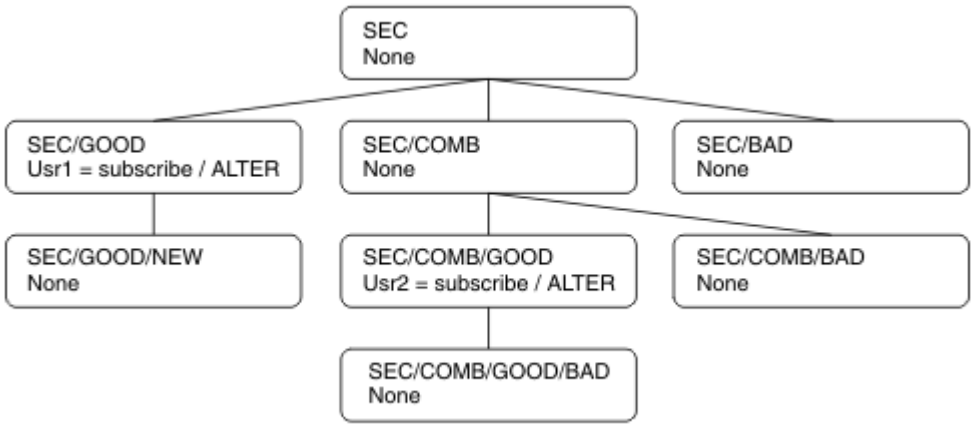
## トピック・ストリングを使用したサブスクライブ操作 (トピック・ノードが存在する場合)

MQCHAR48 オブジェクト名でトピックを指定する場合と同じ動作です。

## トピック・ストリングを使用したサブスクライブ操作 (トピック・ノードが存在しない場合)

アプリケーションが、トピック・ツリーに現在存在しないトピック・ノードを表すトピック・ストリングを指定してサブスクライブするケースを想定します。 前のセクションで示したように権限検査が実行されます。 トピック・ストリングによって表されるノードの親ノードから検査が始まります。 権限が認められると、そのトピック・ストリングを表す新しいノードがトピック・ツリー内に作成されます。

例えば、usr1 がトピック SEC/GOOD/NEW へのサブスクライブを試みたとします。 usr1 は親ノード SEC/GOOD へのアクセス権限を持っているため、権限が付与されます。 以下の図に示すように、新しいトピック・ノードがツリー内に作成されます。 新しいトピック・ノードは、トピック・オブジェクトではないので、セキュリティー属性が直接関連付けられていません。セキュリティー属性は、親から継承します。



## ワイルドカード文字を含むトピック・ストリングを使用したサブスクライブ操作

ワイルドカード文字を含むトピック・ストリングを使用してサブスクライブするケースを想定します。 トピック・ツリー内でそのトピック・ストリングの完全修飾部分に一致するノードに対して権限検査が行われます。

つまり、アプリケーションが SEC/COMB/GOOD/* にサブスクライブすると、トピック・ツリー内の SEC/COMB/GOOD ノードで、前の 2 つのセクションで概説されているとおりに権限検査が行われます。

同様に、アプリケーションが SEC/COMB/*/GOOD にサブスクライブする必要がある場合は、SEC/COMB ノードで権限検査が行われます。

## 宛先キューに対する権限

トピックにサブスクライブする場合、パラメーターの 1 つは、パブリケーションを受け取るために出力用としてオープンされたキューのハンドル hobj になります。

hobj が指定されず、ブランクであれば、以下の条件に該当する場合に管理対象キューが作成されます。

- MQSO_MANAGED オプションが指定されている。
- サブスクリプションが存在しない。
- Create が指定されている。

hobj がブランクの場合に、既存のサブスクリプションを変更または再開すると、それまでに指定した宛先キューは管理対象になる場合も、非管理対象になる場合もあります。

MQSUB 要求を実行するアプリケーションまたはユーザーは、自身で用意した宛先キューにメッセージを書き込む権限 (つまり、パブリッシュされたメッセージをそのキューに書き込む権限) を持っていなければなりません。 権限検査は、キューのセキュリティー検査のための既存のルールに基づきます。

セキュリティー検査には、必要に応じて、代替ユーザー ID 検査とコンテキスト・セキュリティー検査も含まれます。 いずれかの ID コンテキスト・フィールドを設定できるようにするには、MQSO_SET_IDENTITY_CONTEXT オプションだけでなく、MQSO_CREATE オプションまたは MQSO_ALTER オプションも指定する必要があります。 MQSO_RESUME 要求では、いずれの ID コンテキスト・フィールドも設定できません。

宛先が管理対象キューであれば、その管理対象宛先に対するセキュリティー検査は行われません。 トピックへのサブスクライブ権限を持つユーザーは、管理対象宛先を使用できると見なされます。

## トピック名またはトピック・ストリングを使用したパブリッシュ操作 (トピック・ノードが存在する場合)

パブリッシュのセキュリティー・モデルは、ワイルドカード以外はサブスクライブの場合と同じです。 パブリケーションにはワイルドカードは含まれません。 そのため、考慮が必要な、ワイルドカードが含まれるトピック・ストリングの事例はありません。

パブリッシュする権限とサブスクライブする権限は別個のものです。 ユーザーまたはグループは、必ずしも両方の操作を実行できる必要はなく、片方の操作を実行する権限のみを持つことができます。

MQCHAR48 名またはトピック・ストリングを指定してトピック・オブジェクトにパブリッシュする場合、トピック・ツリー内の該当するノードが検出されます。 トピック・ノードに関連付けられているセキュリティー属性で、そのユーザーがパブリッシュする権限を持っていることが確認されれば、アクセスが認められます。

アクセスが認められない場合は、ツリー内の親ノードで、そのユーザーが親ノードのレベルでパブリッシュする権限を持っているかどうかが確認されます。 持っていれば、アクセスが認められます。 持っていなければ、パブリッシュ権限をユーザーに付与するノードが見つかるまで、再帰が続行されます。 ルート・ノードで権限の確認が行われても権限が付与されない場合、再帰が停止します。 この場合、アクセスが拒否されます。

つまり、パス内のいずれかのノードで、そのユーザーまたはアプリケーションにパブリッシュ権限を付与する場合は、パブリッシャーが、そのノードまたはトピック・ツリー内でそのノードよりも下位にあるすべてのノードにパブリッシュすることが許可されます。

## トピック名またはトピック・ストリングを使用したパブリッシュ操作 (トピック・ノードが存在しない場合)

サブスクライブ操作の場合と同様に、アプリケーションが、トピック・ツリー内に現在存在しないトピック・ノードを表すトピック・ストリングを指定してパブリッシュすると、そのトピック・ストリングが表すノードの親から権限検査が開始されます。 権限が認められると、そのトピック・ストリングを表す新しいノードがトピック・ツリー内に作成されます。

## トピック・オブジェクトに解決される別名キューを使用したパブリッシュ操作

トピック・オブジェクトに解決される別名キューを使用してパブリッシュする場合は、別名キューと、解決される元のトピックの両方でセキュリティー検査が行われます。

別名キューのセキュリティー検査では、ユーザーがその別名キューにメッセージを書き込む権限を持っているかどうかが検証され、トピックのセキュリティー検査では、ユーザーがそのトピックにパブリッシュできるかどうかが検証されます。 別名キューが別のキューに解決される場合、元のキューに対しては検査が行われません。 トピックとキューでは、異なる方法で権限検査が実行されます。

## サブスクリプションのクローズ

このハンドルでサブスクリプションを作成していない場合に、MQCO_REMOVE_SUB オプションを使用してそのサブスクリプションをクローズすると、追加のセキュリティー検査が行われます。

この操作を実行すると、サブスクリプションが削除されるため、セキュリティー検査では、ユーザーがこの操作を行う適切な権限を持っているかどうかが確認されます。 そのトピック・ノードに関連付けられているセキュリティー属性で、ユーザーが権限を持っていることが確認されれば、アクセスが認められます。確認されない場合は、ツリー内の親ノードで、そのユーザーがそのサブスクリプションをクローズする権限を持っているかどうかが確認されます。 権限が付与されるか、ルート・ノードに到達するまで、再帰が続行されます。

## サブスクリプションの定義、変更、削除

MQSUB API 要求を使用してではなく、管理的にサブスクリプションが作成されるときには、サブスクライブ・セキュリティー検査が実行されません。 管理者には、コマンドを介して、既にこの権限が付与されています。

サブスクリプションに関連付けられている宛先キューにパブリケーションを書き込むことができるかどうかを確認するためのセキュリティー検査が行われます。 MQSUB 要求に関しても、同じように検査が実行されます。

それらのセキュリティー検査で使用されるユーザー ID は、実行されるコマンドによって異なります。 **SUBUSER** パラメーターが指定される場合、259 ページの表 18 に示すように、検査の実行方法に影響を与えます。

表 *18.* コマンドのセキュリティー検査に使用されるユーザー *ID*

| コマンド | SUBUSER が指定され、ブランクになっている | SUBUSER が指定され、値が設定されている | SUBUSER が指定されていない |
|---|---|---|---|
| | 管理者の ID が使用されます | | 管理者の ID が使用されます |

| 表 18. コマンドのセキュリティー検査に使用されるユーザー ID (続き) | | | |
|---|---|---|---|
| **コマンド** | **SUBUSER が指定され、ブランクになっている** | **SUBUSER が指定され、値が設定されている** | **SUBUSER が指定されていない** |
| | 管理者の ID が使用されます | | 既存のサブスクリプションのユーザー ID が使用されます |

DELETE SUB コマンドを使用してサブスクリプションを削除するときに行われるセキュリティー検査は、コマンドのセキュリティー検査のみです。

# パブリッシュ/サブスクライブのセキュリティー・セットアップの例

このセクションでは、必要に応じてセキュリティー管理を適用することが可能な方法で、トピックのアクセス制御をセットアップするシナリオについて説明します。

## トピックにサブスクライブするアクセス権限をユーザーに付与する

ここでは、トピックに対するアクセス権限を複数のユーザーに付与するための最初の作業を取り上げます。

### このタスクについて

この作業は、管理トピック・オブジェクトが存在しないことと、サブスクリプションまたはパブリケーションのプロファイルが一切定義されていないことを前提にしています。 アプリケーションは、既存のサブスクリプションを再開するのではなく、新しいサブスクリプションを作成し、そのためにトピック・ストリングだけを使用します。

アプリケーションでサブスクリプションを作成するときには、トピック・オブジェクトを指定することも、トピック・ストリングを指定することも、その両方を組み合わせて指定することもできます。 アプリケーションでどの方法を選択するにしても、結果として、トピック・ツリー内の特定のポイントでサブスクリプションが作成されることになります。 トピック・ツリー内のそのポイントが管理トピック・オブジェクトで表されている場合、そのトピック・オブジェクトの名前に基づいてセキュリティー・プロファイルがチェックされます。

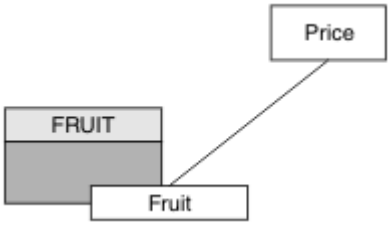



図 27. トピック・オブジェクトのアクセス権限の例

| 表 19. トピック・オブジェクトのアクセス権限の例 | | |
|---|---|---|
| **トピック** | **必要なサブスクライブ・アクセス権限** | **トピック・オブジェクト** |
| Price | ユーザーなし | なし |
| Price/Fruit | USER1 | FRUIT |

以下のようにして、新しいトピック・オブジェクトを定義します。

### 手順

1. MQSC コマンド DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit') を発行します。
2. 以下のようにしてアクセス権限を付与します。
   - その他のプラットフォーム:

     トピック "Price/Fruit" にサブスクライブするアクセス権限を USER1 に付与するために、FRUIT オブジェクトに対するアクセス権限をそのユーザーに付与します。 そのために、プラットフォームに応じて以下の許可コマンドを使用します。

     
     **Windows、 UNIX and Linux システム**

     ```
     setmqaut -t topic -n FRUIT -p USER1 +sub
     ```

### タスクの結果

USER1 がトピック "Price/Fruit" へサブスクライブしようとすると、成功します。

USER2 がトピック "Price/Fruit" へサブスクライブしようとした場合、MQRC_NOT_AUTHORIZED メッセージが出て失敗し、さらに以下のような結果になります。

- 
  他のプラットフォームでは、以下の許可イベントが発生します。

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

ただし、ここに示すのはすべてのフィールドではなく、実際に表示される内容であることに注意してください。

## ツリー内の下位トピックにサブスクライブするアクセス権限をユーザーに付与する

ここでは、トピックに対するアクセス権限を複数のユーザーに付与するための 2 番目の作業を取り上げます。

### 始める前に

このトピックで使用するセットアップについては、260 ページの『トピックにサブスクライブするアクセス権限をユーザーに付与する』を参照してください。

### このタスクについて

アプリケーションによってサブスクリプションが作成されたトピック・ツリー内のポイントが管理トピック・オブジェクトで表されていない場合は、直近の親管理トピック・オブジェクトの場所までツリーを上がっていきます。 そのトピック・オブジェクトの名前に基づいてセキュリティー・プロファイルがチェックされます。

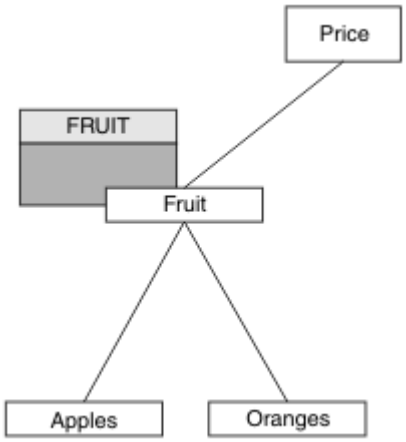



図 *28.* トピック・ツリー内のトピックへのアクセス権限を付与する例

表 *20.* サンプル・トピックおよびトピック・オブジェクトでのアクセス権限の要件

| トピック | 必要なサブスクライブ・アクセス権限 | トピック・オブジェクト |
|---|---|---|
| Price | ユーザーなし | なし |
| Price/Fruit | USER1 | FRUIT |
| Price/Fruit/Apples | USER1 | |
| Price/Fruit/Oranges | USER1 | |

前のタスクでは、 z/OS 上の hlq.SUBSCRIBE.FRUIT プロファイルに対するアクセス権限と、他のプラットフォーム上の FRUIT プロファイルに対するサブスクライブ・アクセス権限を USER1 に付与することによって、トピック "Price/Fruit" にサブスクライブする権限が付与されました。 また、その 1 つのプロファイルによって、USER1 には、"Price/Fruit/Apples"、"Price/Fruit/Oranges"、および "Price/Fruit/#" にサブスクライブするアクセス権限も与えられます。

USER1 がトピック "Price/Fruit/Apples" へサブスクライブしようとすると、成功します。

USER2 がトピック "Price/Fruit/Apples" へサブスクライブしようとした場合、MQRC_NOT_AUTHORIZED メッセージが出て失敗し、さらに以下のような結果になります。

- z/OS では、トピック・ツリー内で試行された完全セキュリティー・パスを示した以下のメッセージがコンソールに表示されます。

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- 他のプラットフォームでは、以下の許可イベントが発生します。

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

次の事項に注意してください。

- z/OS で受け取るメッセージは、前の作業で受け取ったメッセージと同じです。同じトピック・オブジェクトと同じプロファイルがアクセス権限を制御しているからです。
- 他のプラットフォームで受け取るイベント・メッセージは、前の作業で受け取ったイベント・メッセージと類似していますが、実際のトピック・ストリングが異なっています。

## ツリー内の下位トピックだけにサブスクライブするアクセス権限を別のユーザーに付与する

ここでは、トピックにサブスクライブするアクセス権限を複数のユーザーに付与するための 3 番目の作業を取り上げます。

### 始める前に

このトピックで使用するセットアップについては、261 ページの『ツリー内の下位トピックにサブスクライブするアクセス権限をユーザーに付与する』を参照してください。

### このタスクについて

前の作業では、USER2 は、トピック "Price/Fruit/Apples" に対するアクセスを拒否されました。 ここでは、そのトピックだけに対するアクセス権限を付与して、他のトピックに対するアクセス権限を付与しない方法を示します。

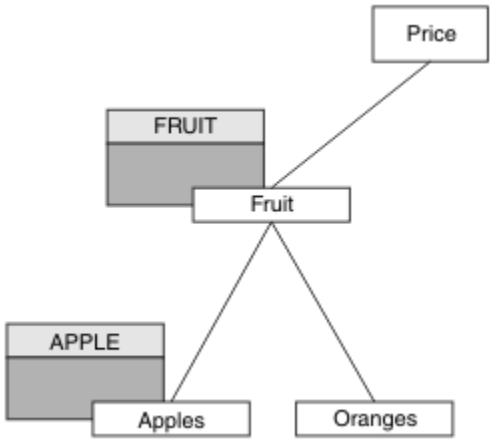



図 *29.* トピック・ツリー内の特定のトピックへのアクセス権限の付与

表 *21.* サンプル・トピックおよびトピック・オブジェクトでのアクセス権限の要件

| トピック | 必要なサブスクライブ・アクセス権限 | トピック・オブジェクト |
|---|---|---|
| Price | ユーザーなし | なし |
| Price/Fruit | USER1 | FRUIT |
| Price/Fruit/Apples | USER1 および USER2 | APPLE |
| Price/Fruit/Oranges | USER1 | |

以下のようにして、新しいトピック・オブジェクトを定義します。

### 手順

1. MQSC コマンド `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')` を発行します。

2. 以下のようにしてアクセス権限を付与します。

   - その他のプラットフォーム:

     前の作業では、トピック "Price/Fruit/Apples" にサブスクライブするアクセス権限を USER1 に与えるために、FRUIT プロファイルに対するサブスクライブ・アクセス権限をそのユーザーに与えました。

     その 1 つのプロファイルによって、USER1 には、"Price/Fruit/Oranges" と "Price/Fruit/#" にサブスクライブするアクセス権限も付与されます。新しいトピック・オブジェクトおよび関連するプロファイルが追加されても、そのアクセス権限は残ります。

     トピック "Price/Fruit/Apples" にサブスクライブするアクセス権限を USER2 に付与するために、APPLE プロファイルに対するサブスクライブ・アクセス権限をそのユーザーに付与します。 そのために、プラットフォームに応じて以下の許可コマンドを使用します。

     Windows UNIX Linux **Windows、 UNIX and Linux システム**

     ```
     setmqaut -t topic -n APPLE -p USER2 +sub
     ```

## タスクの結果

z/OS では、USER1 がトピック "Price/Fruit/Apples" にサブスクライブしようとすると、hlq.SUBSCRIBE.APPLE プロファイルでの最初のセキュリティー検査は失敗しますが、ツリーの上の方にある hlq.SUBSCRIBE.FRUIT プロファイルでは USER1 のサブスクライブが許可されているので、サブスクリプションは成功し、MQSUB 呼び出しに戻りコードが送信されることはありません。 ただし、最初の検査については、RACF ICH メッセージが生成されます。

```
    ICH408I USER(USER1   ) ...
      hlq.SUBSCRIBE.APPLE ...
```

USER2 がトピック "Price/Fruit/Apples" にサブスクライブしようとした場合、最初のプロファイルでセキュリティー検査に合格するので、操作は成功します。

USER2 がトピック "Price/Fruit/Oranges" へサブスクライブしようとした場合、MQRC_NOT_AUTHORIZED メッセージが出て失敗し、さらに以下のような結果になります。

- Windows UNIX Linux Windows、UNIX、および Linux プラットフォームでは、以下の許可イベントが発生します。

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

このセットアップの場合、z/OS では、追加の ICH メッセージがコンソールに表示されるという欠点があります。 別の方法でトピック・ツリーのセキュリティーを確保すれば、この問題は回避できます。

# 追加のメッセージを回避するためにアクセス制御を変更する

このトピックは、複数のユーザーによってトピックにサブスクライブするためのアクセス権限を付与する方法、および z/OS に追加の RACF ICH408I メッセージが表示されないようにするためのタスクのリストの 4 番目です。

## 始める前に

このトピックでは、263 ページの『ツリー内の下位トピックだけにサブスクライブするアクセス権限を別のユーザーに付与する』のセットアップを拡張して、追加のエラー・メッセージが出ないようにします。

## このタスクについて

ここでは、ツリー内の下位トピックに対するアクセス権限を付与する方法と、ユーザーが必要としないツリーの下位トピックに対するアクセス権限を除去する方法を示します。

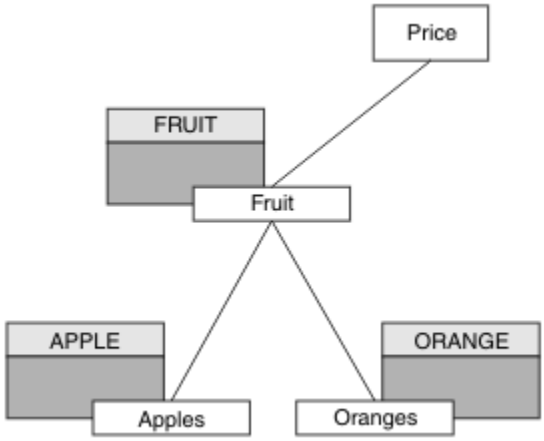



図 *30.* 追加のメッセージを回避するためにアクセス制御を付与する例。

以下のようにして、新しいトピック・オブジェクトを定義します。

## 手順

1. MQSC コマンド `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')` を発行します。
2. 以下のようにしてアクセス権限を付与します。
   - その他のプラットフォーム:

     プラットフォームに応じた許可コマンドを使用し、同等のアクセス権限をセットアップします。

     Windows UNIX Linux **Windows、 UNIX and Linux システム**

     ```
     setmqaut -t topic -n ORANGE -p USER1 +sub
     setmqaut -t topic -n APPLE -p USER1 +sub
     ```

## タスクの結果

z/OS では、USER1 がトピック "Price/Fruit/Apples" にサブスクライブしようとした場合、hlq.SUBSCRIBE.APPLE プロファイルでの最初のセキュリティー検査が成功します。

同じように、USER2 がトピック "Price/Fruit/Apples" にサブスクライブしようとした場合、最初のプロファイルでセキュリティー検査に合格するので、操作は成功します。

USER2 がトピック "Price/Fruit/Oranges" へサブスクライブしようとした場合、MQRC_NOT_AUTHORIZED メッセージが出て失敗し、さらに以下のような結果になります。

- Windows UNIX Linux 他のプラットフォームでは、以下の許可イベントが発生します。

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

# トピックにパブリッシュするアクセス権限をユーザーに付与する

ここでは、トピックにパブリッシュするアクセス権限を複数のユーザーに付与するための最初の作業を取り上げます。

## このタスクについて

この作業は、トピック・ツリーの右側に管理トピック・オブジェクトが存在しないことと、パブリケーションのプロファイルが何も定義されていないことを前提にしています。 この前提では、パブリッシャーがトピック・ストリングだけを使用します。

アプリケーションでトピックにパブリッシュするときには、トピック・オブジェクトを指定することも、トピック・ストリングを指定することも、その両方を組み合わせて指定することもできます。 アプリケーションでどの方法を選択するにしても、結果として、トピック・ツリー内の特定のポイントでパブリッシュされることになります。 トピック・ツリー内のそのポイントが管理トピック・オブジェクトで表されている場合、そのトピック・オブジェクトの名前に基づいてセキュリティー・プロファイルがチェックされます。 以下に例を示します。

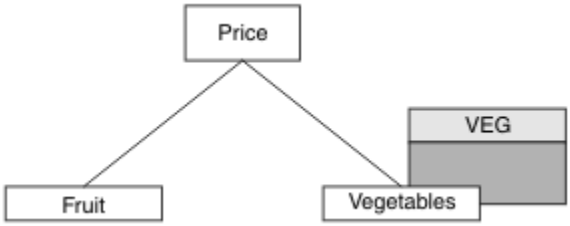



図 *31.* トピックへのパブリッシュ・アクセス権限の付与

表 *22.* パブリッシュ・アクセス権限の要件の例

| トピック | 必要なパブリッシュ・アクセス権限 | トピック・オブジェクト |
|---|---|---|
| Price | ユーザーなし | なし |
| Price/Vegetables | USER1 | VEG |

以下のようにして、新しいトピック・オブジェクトを定義します。

## 手順

1. MQSC コマンド `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')` を発行します。
2. 以下のようにしてアクセス権限を付与します。
   - その他のプラットフォーム:

     トピック `"Price/Vegetables"` にパブリッシュするアクセス権限を USER1 に付与するために、VEG プロファイルに対するアクセス権限をそのユーザーに付与します。 そのために、プラットフォームに応じて以下の許可コマンドを使用します。

     Windows UNIX Linux **Windows、 UNIX and Linux システム**

     ```
     setmqaut -t topic -n VEG -p USER1 +pub
     ```

## タスクの結果

USER1 がトピック `"Price/Vegetables"` にパブリッシュしようとすると、成功します (つまり、MQOPEN 呼び出しは成功します)。

USER2 がトピック `"Price/Vegetables"` にパブリッシュしようとした場合、MQOPEN 呼び出しは `MQRC_NOT_AUTHORIZED` メッセージが出て失敗し、さらに以下のような結果になります。

- Windows UNIX Linux 他のプラットフォームでは、以下の許可イベントが発生します。

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
```

```
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

ただし、ここに示すのはすべてのフィールドではなく、実際に表示される内容であることに注意してください。

## ツリー内の下位トピックにパブリッシュするアクセス権限をユーザーに付与する

ここでは、トピックにパブリッシュするアクセス権限を複数のユーザーに付与するための 2 番目の作業を取り上げます。

### 始める前に

このトピックで使用するセットアップについては、265 ページの『トピックにパブリッシュするアクセス権限をユーザーに付与する』を参照してください。

### このタスクについて

アプリケーションによってパブリッシュされたトピック・ツリー内のポイントが管理トピック・オブジェクトで表されていない場合は、直近の親の管理トピック・オブジェクトの場所までツリーを上がっていきます。 そのトピック・オブジェクトの名前に基づいてセキュリティー・プロファイルがチェックされます。

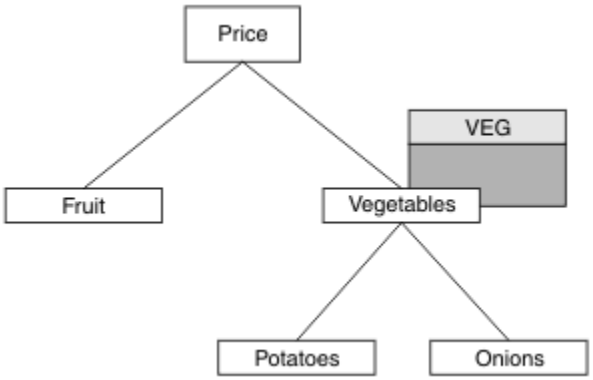



図 *32.* トピック・ツリー内のトピックへのパブリッシュ・アクセス権限の付与

表 *23.* パブリッシュ・アクセス権限の要件の例

| トピック | 必要なサブスクライブ・アクセス権限 | トピック・オブジェクト |
|---|---|---|
| Price | ユーザーなし | なし |
| Price/Vegetables | USER1 | VEG |
| Price/Vegetables/Potatoes | USER1 | |
| Price/Vegetables/Onions | USER1 | |

前のタスクでは、 z/OS 上の `hlq.PUBLISH.VEG` プロファイルへのアクセス権限、または他のプラットフォーム上の VEG プロファイルへのパブリッシュ・アクセス権限を USER1 に付与することによって、トピック `"Price/Vegetables/Potatoes"` をパブリッシュする権限が付与されました。 この単一プロファイルは、 `"Price/Vegetables/Onions"`で公開するための USER1 アクセス権限も付与します。

USER1 がトピック `"Price/Vegetables/Potatoes"` にパブリッシュしようとすると、成功します (つまり、MQOPEN 呼び出しは成功します)。

USER2 がトピック "Price/Vegetables/Potatoes" にサブスクライブしようとした場合、失敗します (つまり、MQOPEN 呼び出しは MQRC_NOT_AUTHORIZED メッセージが出て失敗し、さらに以下のような結果になります)。

- z/OS では、トピック・ツリー内で試行された完全セキュリティー・パスを示した以下のメッセージがコンソールに表示されます。

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- 他のプラットフォームでは、以下の許可イベントが発生します。

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

次の事項に注意してください。

- z/OS で受け取るメッセージは、前の作業で受け取ったメッセージと同じです。同じトピック・オブジェクトと同じプロファイルがアクセス権限を制御しているからです。
- 他のプラットフォームで受け取るイベント・メッセージは、前の作業で受け取ったイベント・メッセージと類似していますが、実際のトピック・ストリングが異なっています。

## パブリッシュとサブスクライブのためのアクセス権限を付与する

ここでは、トピックにパブリッシュおよびサブスクライブするアクセス権限を複数のユーザーに付与するための最後の作業を取り上げます。

### 始める前に

このトピックで使用するセットアップについては、267 ページの『ツリー内の下位トピックにパブリッシュするアクセス権限をユーザーに付与する』を参照してください。

### このタスクについて

前の作業では、トピック "Price/Fruit" にサブスクライブするアクセス権限を USER1 に与えました。ここでは、そのトピックにパブリッシュするアクセス権限をそのユーザーに付与する方法を示します。

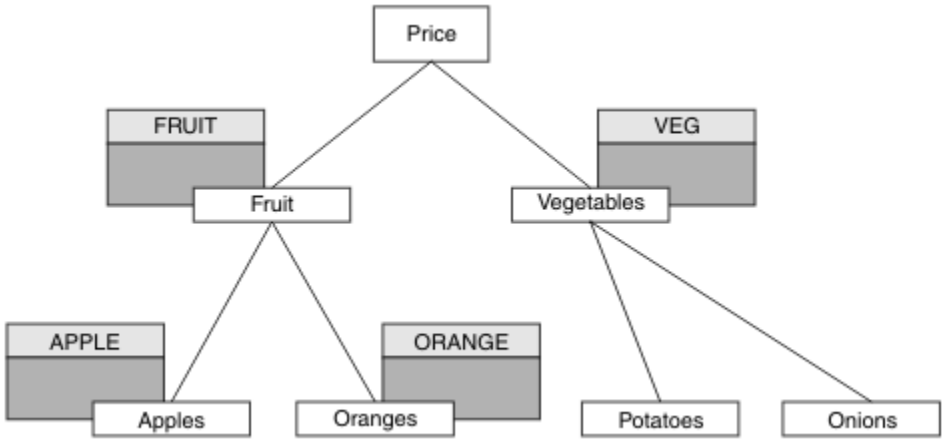



図 *33.* パブリッシュおよびサブスクライブのためのアクセス権限の付与

表 24. パブリッシュおよびサブスクライブのアクセス権限の要件の例

| トピック | 必要なサブスクライブ・アクセス権限 | 必要なパブリッシュ・アクセス権限 | トピック・オブジェクト |
|---|---|---|---|
| Price | ユーザーなし | ユーザーなし | なし |
| Price/Fruit | USER1 | USER1 | FRUIT |
| Price/Fruit/Apples | USER1 および USER2 | | APPLE |
| Price/Fruit/Oranges | USER1 | | ORANGE |

## 手順

以下のようにしてアクセス権限を付与します。

- その他のプラットフォーム:

  トピック "Price/Fruit" にパブリッシュするアクセス権限を USER1 に付与するために、FRUIT プロファイルに対するパブリッシュ・アクセス権限をそのユーザーに付与します。 そのために、プラットフォームに応じて以下の許可コマンドを使用します。

  Windows UNIX Linux **Windows、 UNIX and Linux システム**

```
setmqaut -t topic -n FRUIT -p USER1 +pub
```

## タスクの結果

z/OS では、USER1 がトピック "Price/Fruit" にパブリッシュしようとすると、MQOPEN 呼び出しのセキュリティー検査に合格します。

USER2 がトピック "Price/Fruit" にパブリッシュしようとした場合、MQRC_NOT_AUTHORIZED メッセージが出て失敗し、さらに以下のような結果になります。

- Windows UNIX Linux Windows、UNIX、および Linux プラットフォームでは、以下の許可イベントが発生します。

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

以下のリストでは、この一連の作業が完了した時点で、どのトピックにパブリッシュおよびサブスクライブするアクセス権限が USER1 および USER2 に付与されるかが示されています。

表 25. セキュリティーの例でのアクセス権限の完全なリスト

| トピック | 必要なサブスクライブ・アクセス権限 | 必要なパブリッシュ・アクセス権限 | トピック・オブジェクト |
|---|---|---|---|
| Price | ユーザーなし | ユーザーなし | なし |
| Price/Fruit | USER1 | USER1 | FRUIT |

表 25. セキュリティーの例でのアクセス権限の完全なリスト (続き)

| トピック | 必要なサブスクライブ・アクセス権限 | 必要なパブリッシュ・アクセス権限 | トピック・オブジェクト |
|---|---|---|---|
| Price/Fruit/Apples | USER1 および USER2 | | APPLE |
| Price/Fruit/Oranges | USER1 | | ORANGE |
| Price/Vegetables | | USER1 | VEG |
| Price/Vegetables/Potatoes | | | |
| Price/Vegetables/Onions | | | |

トピック・ツリー内のレベルごとにセキュリティー・アクセス要件がそれぞれ異なる場合は、入念な計画により、z/OS のコンソール・ログに余計なセキュリティー警告が表示されないようにすることができます。ツリー内の正しいレベルでセキュリティーをセットアップすれば、混乱を招くセキュリティー・メッセージを回避できます。

# サブスクリプションのセキュリティー

## MQSO_ALTERNATE_USER_AUTHORITY

AlternateUserId フィールドは、この MQSUB 呼び出しの妥当性検査に使用するユーザー ID を格納します。この呼び出しが成功するのは、指定されたアクセス・オプションでトピックにサブスクライブする権限がこの AlternateUserId にある場合だけです。アプリケーションの実行に使用されているユーザー ID がこの許可を持っているかどうかは関係ありません。

## MQSO_SET_IDENTITY_CONTEXT

サブスクリプションは、PubAccountingToken フィールドと PubApplIdentityData フィールドで提供されるアカウンティング・トークンとアプリケーション ID データを使用します。

このオプションを指定すると、MQOO_SET_IDENTITY_CONTEXT を指定した MQOPEN 呼び出しを使用して宛先キューがアクセスされた場合と同じ許可検査が実行されます。ただし、MQSO_MANAGED オプションも使用する場合は例外で、この場合は宛先キューに関する許可検査は行われません。

このオプションを指定しないと、以下のように、このサブスクライバーに送信されるパブリケーションにデフォルトのコンテキスト情報が関連付けられます。

表 26. デフォルトのパブリケーション・コンテキスト情報

| MQMD のフィールド | 使用される値 |
|---|---|
| *UserIdentifier* | パブリケーションの作成時にサブスクリプションに関連付けられたユーザー ID (DISPLAY SBSTATUS 上の SUBUSER フィールドを参照)。 |
| *AccountingToken* | 環境から判別できる場合は判別されます。判別できない場合は MQACT_NONE に設定されます。 |
| *ApplIdentityData* | ブランクに設定されます。 |

このオプションは、MQSO_CREATE および MQSO_ALTER と併用する場合のみ有効です。 MQSO_RESUME と併用すると、PubAccountingToken および PubApplIdentityData フィールドは無視されるので、このオプションは無効になります。

以前にサブスクリプションで ID コンテキスト情報が提供された場合に、このオプションを使用しないでそのサブスクリプションを変更すると、変更されたサブスクリプションに関するデフォルトのコンテキスト情報が生成されます。

サブスクリプションで、さまざまなユーザー ID がオプション MQSO_ANY_USERID を指定してそのサブスクリプションを使用することを許可している場合、別のユーザー ID がそのサブスクリプションを再開すると、現在のそのサブスクリプションの所有者となるその新しいユーザー ID に関するデフォルトの ID コンテキストが生成され、送達されるそれ以降のパブリケーションにはその新しい ID コンテキストが含まれるようになります。

### AlternateSecurityId

これは、AlternateUserId と共に許可サービスに渡されて、適切な許可検査を実行できるようにするセキュリティー ID です。 AlternateSecurityId が使用されるのは、MQSO_ALTERNATE_USER_AUTHORITY が指定されており、AlternateUserId フィールドが最初のヌル文字かフィールドの終わりまですべてブランクでない場合のみです。

## MQSO_ANY_USERID サブスクリプション・オプション

MQSO_ANY_USERID を指定すると、サブスクライバーの ID は単一のユーザー ID に制限されなくなります。 そのため、ユーザーは適切な権限を持っていれば、サブスクリプションの変更や再開を行うことができます。 一度に 1 人のユーザーだけがサブスクリプションを持つことができます。 別のアプリケーションで現在使用中のサブスクリプションの使用を再開しようとすると、MQRC_SUBSCRIPTION_IN_USE で呼び出しが失敗します。

このオプションを既存のサブスクリプションに追加するには、(MQSO_ALTER を使用する) MQSUB 呼び出しを元のサブスクリプションと同じユーザー ID から行わなければなりません。

MQSO_ANY_USERID が設定された既存のサブスクリプションを MQSUB 呼び出しが参照する際に、元のサブスクリプションとユーザー ID が違う場合は、この呼び出しが成功するのは、トピックにサブスクライブする権限が新しいユーザー ID にある場合に限られます。 正常終了後は、このサブスクライバーへのパブリケーションはサブスクライバーのキューに書き込まれ、パブリケーションに新しいユーザー ID が設定されます。

### MQSO_FIXED_USERID

MQSO_FIXED_USERID を指定すると、単一の所有ユーザー ID のみがサブスクリプションの変更や再開を行うことができます。 このユーザー ID は、前回サブスクリプションに対してこのオプションを設定して MQSO_ANY_USERID オプションを除去する変更を行ったユーザー ID か、または、変更が行われていない場合は、サブスクリプションを作成したユーザー ID です。

MQSUB verb が、MQSO_ANY_USERID を設定した既存のサブスクリプションを参照し、(MQSO_ALTER を使用して) オプション MQSO_FIXED_USERID を使用するようにサブスクリプションを変更すると、この時点でサブスクリプションのユーザー ID はこの新しいユーザー ID で固定されます。 このトピックにサブスクライブする権限が新しいユーザー ID にある場合にのみ、呼び出しは成功します。

サブスクリプションを所有していると記録されているのと別のユーザー ID が MQSO_FIXED_USERID サブスクリプションの再開か変更を行おうとすると、呼び出しは MQRC_IDENTITY_MISMATCH で失敗します。 サブスクリプションの所有者になっているユーザー ID は、DISPLAY SBSTATUS コマンドを使用して表示できます。

MQSO_ANY_USERID と MQSO_FIXED_USERID のどちらも指定しないと、デフォルトは MQSO_FIXED_USERID になります。

# IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security (AMS) は、別個にライセンス交付される IBM WebSphere MQ Advanced Message Security のコンポーネントです。これを使用すると、末端のアプリケーションに影響を与えずに、IBM WebSphere MQ Advanced Message Security ネットワーク経由で流れる機密データを高水準で保護できます。

## IBM WebSphere MQ Advanced Message Security 概要

IBM WebSphere MQ アプリケーションは、IBM WebSphere MQ Advanced Message Security を使用して、高価値の金融取引情報や個人情報などの機密データを送信できます。その際、公開鍵暗号モデルを使用して、さまざまなレベルの保護を提供します。

**関連資料**

IBM WebSphere MQ AMS メッセージで使用される GSKit 戻りコード

### バージョン 7.0.1 とバージョン 7.5 の間の動作の変更点

IBM Advanced Message Security が WebSphere MQ 7.5 のコンポーネントになると、 IBM WebSphere MQ AMS 機能のいくつかの側面が変更され、既存のアプリケーション、管理スクリプト、または管理手順に影響を与える可能性があります。

キュー・マネージャーをバージョン 7.5 にアップグレードする前に、変更点に関する以下のリストをよく確認してください。 IBM WebSphere MQ バージョン 7.5 へのシステムの移行を開始する前に、既存のアプリケーション、スクリプト、およびプロシージャーの変更を計画する必要があるかどうかを判断してください。

- IBM WebSphere MQ AMS のインストールは、WebSphere MQ インストール・プロセスの一部です。
- IBM WebSphere MQ AMS のセキュリティー機能はそのインストールによって有効になり、セキュリティー・ポリシーで制御されます。 IBM WebSphere MQ AMS がデータのインターセプトを開始できるようにするために、インターセプターを有効にする必要はありません。
- WebSphere MQ バージョン 7.5 の IBM WebSphere MQ AMS では、スタンドアロン・バージョンの IBM WebSphere MQ AMS のように **cfgmqs** コマンドを使用する必要はありません。

### IBM WebSphere MQ Advanced Message Security のフィーチャーおよび機能

Advanced Message Security は、WebSphere MQ セキュリティー・サービスを拡張して、データの署名および暗号化をメッセージ・レベルで提供します。 拡張されたサービスは、メッセージ・データが最初にキューに入れられてから取り出されるまでの間にメッセージ・データが変更されていないことを保証します。さらに、IBM WebSphere MQ AMS は、メッセージ・データの送信者が、署名されたメッセージをターゲット・キューに入れる権限を持っていることを確認します。

以下に、IBM WebSphere MQ AMS の機能の完全なリストを示します。

- WebSphere MQ で処理される重要トランザクションまたは高価値トランザクションを保護する。
- 不正メッセージまたは無許可メッセージが受信アプリケーションによって処理される前に、それらのメッセージを検出して削除する。
- キュー間での転送中にメッセージが変更されていないことを検証する。
- ネットワークを流れるときだけでなく、キューに入っているときにもデータを保護する。
- WebSphere MQ 用の既存のプロプラエタリー・アプリケーションおよびカスタマー作成アプリケーションを保護する。

### エラーの処理

Advanced Message Security では、エラーを含むメッセージや保護を解除できないメッセージを管理するためのエラー処理キューが定義されています。

問題のあるメッセージは、例外ケースとして処理されます。 受信されたメッセージがキューのセキュリティー要件 (例えば、暗号化時にメッセージが署名されているかどうか、暗号化解除または署名検証が失敗す

るかどうか) を満たしていない場合、メッセージはエラー処理キューに送信されます。 メッセージは、以下のような理由でエラー処理キューに送信されます。

- 保護品質の不一致 - 受信したメッセージとセキュリティー・ポリシーの QOP 定義との間に保護品質 (QOP) の不一致が存在します。
- 暗号化解除エラー - メッセージを暗号化解除できません。
- PDMQ ヘッダー・エラー - WebSphere MQ AMS メッセージ・ヘッダーにアクセスできません。
- サイズの不一致 - 暗号化解除後のメッセージの長さが、予期される値と異なります。
- 暗号化アルゴリズム強度の不一致 - メッセージの暗号化アルゴリズムが要件よりも弱いです。
- 不明のエラー - 予期しないエラーが発生しました。

WebSphere MQ AMS は SYSTEM.PROTECTION.ERROR.QUEUE 。 IBM WebSphere MQ AMS によって SYSTEM.PROTECTION.ERROR.QUEUE の前には MQDLH ヘッダーがあります。

WebSphere MQ 管理者は、 SYSTEM.PROTECTION.ERROR.QUEUE 。

## 基本概念

Advanced Message Security の基本概念を学んで、ツールの機能と、ツールを効果的に管理する方法について理解してください。

### 公開鍵インフラストラクチャー

公開鍵インフラストラクチャー (PKI) とは、安全に通信を行うために公開鍵暗号の使用をサポートする機構、ポリシー、およびサービスの体系のことです。

公開鍵インフラストラクチャーの構成要素を定義する単一の規格があるわけではありませんが、PKI は一般に公開鍵証明書の使用が関係し、以下のサービスを提供する認証局 (CA) とその他の登録局 (RA) で構成されます。

- デジタル証明書を発行する
- デジタル証明書を検証する
- デジタル証明書を取り消す
- 証明書を配布する

ユーザーおよびアプリケーションの ID は、署名されたメッセージまたは暗号化されたメッセージに関連付けられている証明書内の**識別名 (DN)** フィールドによって表されています。 Advanced Message Security は、ユーザーまたはアプリケーションを表すためにこの ID を使用します。 この ID を認証するために、ユーザーまたはアプリケーションは、証明書および関連付けられている秘密鍵が格納されている鍵ストアに対するアクセス権限を持っている必要があります。 各証明書は、鍵ストア内のラベルによって表されています。

#### 関連概念

297 ページの『鍵ストアおよび証明書の使用』
WebSphere MQ アプリケーションにトランスペアレントな暗号保護を提供するために、 Advanced Message Security は鍵ストア・ファイルを使用します。このファイルには、公開鍵証明書と秘密鍵が格納されています。

### デジタル証明書

Advanced Message Security は、ユーザーおよびアプリケーションを X.509 規格のデジタル証明書に関連付けます。 X.509 証明書は、一般に信頼できる認証局 (CA) によって署名され、暗号化と復号に使用される秘密鍵と公開鍵を必要とします。

デジタル証明書は、公開鍵をその所有者にバインドすることによって偽名の使用を防止し、この所有者が個人であるか、キュー・マネージャーであるか、その他のエンティティーであるかは関係ありません。 デジタル証明書は、非対称鍵体系を使用する場合に公開鍵の所有権を保証するので、公開鍵証明書とも呼ばれます。 この体系では、1 つのアプリケーションに対して、1 つの公開鍵と 1 つの秘密鍵を生成する必要があります。 公開鍵で暗号化されたデータは、対応する秘密鍵を使用することでのみ復号でき、秘密鍵で暗号化されたデータは、対応する公開鍵を使用することでのみ復号できます。 秘密鍵は、パスワード保護

された鍵データベース・ファイルに格納されます。 秘密鍵の所有者のみが、対応する公開鍵を使用して暗号化されたメッセージを復号するための秘密鍵にアクセスできます。

公開鍵が、所有者によって別のエンティティーに直接送信される場合、メッセージが傍受され、公開鍵が別のものに置き換えられる危険性があります。 これは、中間者攻撃と呼ばれます。 解決方法は、信頼のおける第三者機関を通じて公開鍵を交換し、公開鍵が通信相手のエンティティーに属しているという確かな保証をユーザーに与えるというものです。 公開鍵を直接送信する代わりに、公開鍵をデジタル証明書に組み込むように、信頼のおける第三者機関に依頼します。 デジタル証明書を発行する信頼のおける第三者機関は、認証局 (CA) と呼ばれます。

デジタル証明書について詳しくは、デジタル証明書の内容を参照してください。

デジタル証明書は、エンティティーの公開鍵を含んでいて、公開鍵がそのエンティティーに属していることを示します。

- 証明書が個人エンティティーの証明書である場合、個人用証明書 またはユーザー証明書 と呼ばれます。
- 証明書が認証局の証明書である場合、*CA* 証明書 または署名者証明書 と呼ばれます。

**注 :** Advanced Message Security は、Java およびネイティブ・アプリケーションの両方で自己署名証明書をサポートします。

**関連概念**

7 ページの『暗号化方式』

暗号化方式とは、平文 と呼ばれる可読テキストと、暗号文 と呼ばれる非可読形式との間で変換を行うプロセスです。

## オブジェクト権限マネージャー

オブジェクト権限マネージャー (OAM) は、WebSphere MQ 製品で提供されている許可サービス・コンポーネントです。

Advanced Message Security エンティティーへのアクセスは、WebSphere MQ ユーザー・グループおよび OAM によって制御されます。 管理者はコマンド・ライン・インターフェースを使用して、必要に応じて許可を与えたり取り消したりすることができます。 同じオブジェクトに対して、ユーザーのグループごとに異なる種類のアクセス権限を与えることができます。 例えば、あるグループには特定のキューに対する PUT 操作と GET 操作の両方の実行を許可し、別のグループにはキューのブラウズのみを許可することができます。 同様に、一部のグループには、キューに対する GET 権限と PUT 権限は与えるが、そのキューの変更または削除の権限は与えないこともできます。

OAM により、以下を制御することができます。

- MQI を介した Advanced Message Security オブジェクトへのアクセス。 アプリケーション・プログラムがオブジェクトにアクセスしようとすると、OAM は、要求された操作に対する許可を要求元のユーザー・プロファイルが持っているかどうかを調べます。 これはキューおよびキュー上のメッセージを無許可アクセスから保護することを意味します。
- PCF および MQSC コマンドの使用許可。

**関連概念**

オブジェクト権限マネージャー

## サポートされるテクノロジー

Advanced Message Security は、いくつかのテクノロジー・コンポーネントに依存してセキュリティー・インフラストラクチャーを提供します。

Advanced Message Security は、以下の WebSphere MQ アプリケーション・プログラミング・インターフェース (API) をサポートしています。

- Message Queue Interface (MQI)
- WebSphere MQ Java Message Service (JMS) 1.0.2 および 1.1。
- WebSphere MQ Java の基本クラス
- WebSphere MQ classes for .Net (非管理対象モード)

**注 :** Advanced Message Security は、X.509 準拠の認証局をサポートしています。

### 既知の制限事項

IBM WebSphere MQ Advanced Message Security の制限について確認します。

- サポートされていない IBM WebSphere MQ のオプションは、以下のとおりです。
  - パブリッシュ/サブスクライブ。
  - チャネル・データの変換。
  - 配布リスト。
  - アプリケーション・メッセージのセグメンテーション
  - HP-UX プラットフォームでの API 出口を使用した非スレッド化アプリケーションの使用。
  - 管理対象ノードでの IBM WebSphere MQ classes for .NET (クライアントまたはバインディング接続)。
  - Message Service Client for .NET (XMS) アプリケーション。
  - Message Service Client for C/C++ (XMS supportPac IA94) アプリケーション。
- すべての Java アプリケーションは、IBM Java ランタイムに依存しています。

  IBM WebSphere MQ Advanced Message Security は、他のベンダーが提供する JRE をサポートしていません。
- クライアント・モードで IBM WebSphere MQ Advanced Message Security を使用する JMS および Java クライアント・アプリケーション。

  JMS または Java クライアント・アプリケーション ( IBM WebSphere MQ Explorer および IBM WebSphere MQ Managed File Transfer エージェントを含む) は、Version 7.5 より前の WebSphere MQ キュー・マネージャーで IBM WebSphere MQ Advanced Message Security をクライアント・モードで使用することはできません。

  メッセージ保護ポリシーを使用するには、これらのアプリケーションが IBM WebSphere MQ Version 7.5 キュー・マネージャーと対話するか、ローカル・バインディング・モードでアプリケーションと同じマシン上のキュー・マネージャーに接続する必要があります。
- 同じ識別名を持つ複数の証明書を 1 つの鍵ストア・ファイルに置かないでください。IBM WebSphere MQ Advanced Message Security インターセプターは、このような証明書を扱うときに不定の動作となってしまうためです。
- IBM WebSphere MQ Version 7.5 リソース・アダプターは IBM WebSphere MQ Advanced Message Security をサポートしていません。 アプリケーション・サーバー環境内で実行されている IBM WebSphere MQ classes for JMS または IBM WebSphere MQ classes for Java アプリケーションでメッセージ保護を使用する必要がある場合、以下のようにします。
  - Version 8.0 以降のリソース・アダプターを使用するようにアプリケーション・サーバーを構成します。
  - または、メッセージ・チャネル・エージェント (MCA) インターセプトを使用する必要があります。

## ユーザー・シナリオ

有効なシナリオに習熟して、 Advanced Message Security で実現できるビジネス目標について理解してください。

### *Windows* プラットフォームのクイック・スタート・ガイド

このガイドでは、Windows プラットフォーム上でメッセージ・セキュリティーを提供するよう IBM Advanced Message Security を迅速に構成する方法について説明します。 このガイドを完了することにより、ユーザー ID を検証するための鍵データベースが作成され、キュー・マネージャーの署名/暗号化ポリシーが定義されます。

### 始める前に

少なくとも以下のフィーチャーがシステムにインストールされていなければなりません。

- サーバー

- 開発ツールキット (サンプル・プログラム用)
- Advanced Message Security

詳しくは、 Windows システムの IBM WebSphere MQ 機能 を参照してください。

**setmqenv** コマンドを使用して現行環境を初期化し、オペレーティング・システムが適切な WebSphere MQ コマンドを見つけて実行できるようにする方法については、 setmqenv を参照してください。

*1.* キュー・マネージャーおよびキューの作成

## このタスクについて

以下のすべての例では、アプリケーション間でメッセージをやり取りするために TEST.Q という名前のキューを使用します。 Advanced Message Security インターセプターを使用して、標準の WebSphere MQ インターフェースを介して WebSphere MQ インフラストラクチャーに入った時点でメッセージの署名と暗号化を行います。 基本的なセットアップは WebSphere MQ で行い、以下のステップで構成されます。

WebSphere MQ エクスプローラーを使用して、すべてのデフォルト・ウィザード設定を使用して TEST.Q というキュー・マネージャー QM_VERIFY_AMS とそのローカル・キューを作成することも、 \WebSphere MQ\bin にあるコマンドを使用することもできます。 以下の管理コマンドを実行するには、mqm ユーザー・グループのメンバーでなければなりません。

## 手順

1. キュー・マネージャーの作成

```
crtmqm QM_VERIFY_AMS
```

2. キュー・マネージャーを開始する

```
strmqm QM_VERIFY_AMS
```

3. キュー・マネージャー QM_VERIFY_AMS の **runmqsc** に次のコマンドを入力して、TEST.Q というキューを作成します

```
DEFINE QLOCAL(TEST.Q)
```

## タスクの結果

この手順を完了すると、**runmqsc** に以下のコマンドを入力することで、TEST.Q に関する詳細を表示できます。

```
DISPLAY Q(TEST.Q)
```

*2.* ユーザーの作成と許可

## このタスクについて

この例では、送信者の alice と受信者の bob という 2 人のユーザーが登場します。 アプリケーション・キューを使用するには、その使用権限がこれらのユーザーに対し付与されている必要があります。 また、定義する保護ポリシーを正常に使用するには、これらのユーザーに対し、一部のシステム・キューにアクセスするための権限が付与されている必要があります。 **setmqaut** コマンドの詳細については、 **setmqaut** を参照してください。

## 手順

1. 2 人のユーザーを作成し、両方のユーザーに HOMEPATH および HOMEDRIVE を設定します。
2. これらのユーザーにキュー・マネージャーへの接続およびキューでの作業を行う許可を付与します。

```
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. また、2 人のユーザーに対し、システム・ポリシー・キューの参照およびエラー・キューへのメッセージの書き込みも許可する必要があります。

```
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
```

## タスクの結果
ユーザーが作成され、必要な権限が付与されました。

## 次のタスク
ステップが正しく実行されたことを確認するには、280 ページの『7. セットアップのテスト』のセクションに説明されているとおりに、amqsput および amqsget のサンプルを使用します。

*3.* 鍵データベースと証明書の作成

## このタスクについて
インターセプターでメッセージを暗号化するには、送信側ユーザーの公開鍵が必要です。 したがって、公開鍵および秘密鍵にマップされたユーザー ID の鍵データベースを作成する必要があります。 ユーザーおよびアプリケーションが複数のコンピューターに分散している実際のシステムでは、各ユーザーが自分専用の鍵ストアを持っています。 同様に、このガイドでは、alice と bob のための鍵データベースを作成し、両者の間でユーザー証明書を共有します。

**注 :** このガイドでは、ローカル・バインディングを使用して接続する、C 言語で作成されたサンプル・アプリケーションを使用しています。 クライアント・バインディングを使用する Java アプリケーションを使用する予定の場合は、JRE の一部である **keytool** コマンドを使用して JKS 鍵ストアと証明書を作成する必要があります (詳しくは、286 ページの『Java クライアントのクイック・スタート・ガイド』 を参照してください)。 他のすべての言語の場合、およびローカル・バインディングを使用する Java アプリケーションの場合は、このガイドのステップが正しいことになります。

## 手順
1. IBM 鍵管理 GUI (strmqikm.exe) を使用して、ユーザー alice 用の新規の鍵データベースを作成します。

```
Type:       CMS
Filename:   alicekey.kdb
Location:   C:/Documents and Settings/alice/AMS
```

   **注 :**

   - 強いパスワードを使用して、データベースを保護することをお勧めします。
   - 「**パスワードをファイルに隠す**」チェック・ボックスが選択されていることを確認してください。
2. 鍵データベースのコンテンツ・ビューを「**個人証明書**」に変更します。
3. 「**新規自己署名**」を選択します。このシナリオでは、自己署名証明書を使用します。
4. 暗号化で使用するために、以下のフィールドを使用して、ユーザー alice を識別する証明書を作成します。

```
Key label: Alice_Cert
Common Name: alice
Organisation: IBM
Country: GB
```

   **注 :**

- このガイドでは、認証局を利用することなく作成できる自己署名証明書を使用します。 実動システムの場合、自己署名証明書を使用するのではなく、認証局が署名した証明書を信頼することをお勧めします。
- **Key label** パラメーターは、インターセプターが必要な情報を受信するためにルックアップする証明書の名前を指定します。
- **Common Name** パラメーターおよびオプション・パラメーターは、ユーザーごとに固有でなければならない**識別名** (DN) の詳細を指定します。

5. ユーザー bob について、ステップ 1 から 4 までを繰り返します。

## タスクの結果
2 人のユーザー alice および bob は、それぞれ自己署名証明書を保持するようになりました。

*4. keystore.conf の作成*

## このタスクについて
Advanced Message Security インターセプターに、鍵データベースと証明書が located.This は、その情報をプレーン・テキスト形式で保持する keystore.conf ファイルを介して実行されます。 各ユーザーは、それぞれの keystore.conf ファイルを持つ必要があります。 このステップは、 alice と bob の両方に対して実行する必要があります。

keystore.conf の内容は、以下のような形式でなければなりません。

```
cms.keystore = <dir>/keystore_file
cms.certificate = certificate_label
```

### 例
このシナリオでは、keystore.conf の内容は以下のようになります。

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

**注 :**

- 鍵ストア・ファイルへのパスは、ファイル拡張子なしで指定する必要があります。
- 証明書ラベルにはスペースを含めることができるため、例えば「Alice_Cert」と「Alice_Cert 」は 2 つの別々の証明書のラベルとして認識されます。 しかし、混乱しないように、ラベルの名前にスペースを使用しないことをお勧めします。
- 鍵ストアの形式には、CMS (Cryptographic Message Syntax)、JKS (Java Keystore)、および JCEKS (Java Cryptographic Extension Keystore) があります。 詳細については、297 ページの『鍵ストア構成ファイル (keystore.conf) の構造』を参照してください。
- *%HOMEDRIVE%\%HOMEPATH%*\.mqs\keystore.conf (例: C: ¥ Documents and Settings\alice\.mqs\keystore.conf) は、Advanced Message Security が keystore.conf ファイルを検索するデフォルトの場所です。 keystore.conf をデフォルト以外の場所で使用する方法については、297 ページの『鍵ストアおよび証明書の使用』を参照してください。
- .mqs ディレクトリーを作成するには、コマンド・プロンプトを使用する必要があります。

*5.* 証明書の共有

## このタスクについて
各ユーザーが互いを正しく識別できるように、2 つの鍵データベース間で証明書を共有します。 そのために、各ユーザーの公開証明書をファイルに抽出し、そのファイルを他のユーザーの鍵データベースに追加します。

**注 :** エクスポート・オプションではなく、必ず抽出 オプションを使用してください。 抽出 はユーザーの公開鍵を取得しますが、エクスポート は公開鍵と秘密鍵の両方を取得します。 誤ってエクスポート を使用すると、秘密鍵が人手に渡り、アプリケーションのセキュリティーが完全に侵害される可能性があります。

## 手順

1. alice を識別する証明書を外部ファイルに抽出します。

```
runmqakm -cert -extract -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. 証明書を bob's 鍵ストアに追加します。

```
runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. bob について、ステップを繰り返します。

```
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
```

## タスクの結果

2 人のユーザー alice および bob は、自己署名証明書を作成して共有することで、互いを正しく識別できるようになります。

## 次のタスク

GUI を使用して参照するか、詳細を出力する次のコマンドを実行することで、証明書が鍵ストアに置かれていることを確認します。

```
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.* キュー・ポリシーの定義

## このタスクについて

キュー・マネージャーの作成とインターセプターの準備が完了し、メッセージをインターセプトして暗号化鍵にアクセスできるようになったら、setmqspl コマンドを使用して QM_VERIFY_AMS での保護ポリシーの定義を開始できます。 このコマンドの詳細については、setmqspl を参照してください。 各ポリシー名は、適用先のキュー名と同じでなければなりません。

### 例

これは、TEST.Q キューに対して定義されたポリシーの例です。 この例では、メッセージは SHA1 アルゴリズムで署名され、AES256 アルゴリズムで暗号化されます。 このキューでは、alice が唯一の有効な送信者であり、bob が唯一のメッセージ受信者です。

```
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"
```

**注 :** DN は、鍵データベースからの各ユーザーの証明書に指定された DN と正確に一致します。

## 次のタスク

定義したポリシーを検証するには、以下のコマンドを実行します。

```
dspmqspl -m QM_VERIFY_AMS
```

一連の setmqspl コマンドとしてポリシーの詳細を出力するには、-export フラグを指定します。 これにより、既に定義されているポリシーが格納されます。

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

*7.* セットアップのテスト

## このタスクについて

さまざまなプログラムをさまざまなユーザーのもとで実行することによって、アプリケーションが正しく構成されているかどうかを確認できます。

## 手順

1. ユーザーを切り替えて、ユーザー alice として実行します。

   cmd.exe を右クリックして、「**実行...**」を選択します。 プロンプトが表示されたら、ユーザー alice としてログインします。
2. ユーザー alice として、サンプル・アプリケーションを使用してメッセージを配置します。

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. メッセージのテキストを入力して、Enter キーを押します。
4. ユーザーを切り替えて、ユーザー bob として実行します。

   cmd.exe を右クリックして「**実行...**」を選択し、別のウィンドウを開きます。 プロンプトが表示されたら、ユーザー bob としてログインします。
5. ユーザー Bob として、サンプル・アプリケーションを使用してメッセージを取得します。

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

## タスクの結果

両方のユーザーでアプリケーションが正しく構成されている場合、bob が取得アプリケーションを実行したときにユーザー alice のメッセージが表示されます。

*8.* 暗号化のテスト

## このタスクについて

暗号化が正しく行われていることを検証するには、元のキュー TEST.Q を参照する別名キューを作成します。 この別名キューにはセキュリティー・ポリシーがないため、メッセージを復号するための情報を持つユーザーは存在しません。これにより、暗号化されたデータが示されることになります。

## 手順

1. キュー・マネージャー QM_VERIFY_AMS に対して **runmqsc** コマンドを使用して、別名キューを作成します。

   ```
   DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
   ```
2. bob アクセス権を付与して、別名キューから参照できるようにします。

   ```
   setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
   ```
3. ユーザー alice として、前述のステップと同様にサンプル・アプリケーションを使用して別のメッセージを配置します。

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. 次に、ユーザー bob として、別名キュー経由でサンプル・アプリケーションを使用してメッセージを参照します。

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. ユーザー bob として、ローカル・キューからサンプル・アプリケーションを使用してメッセージを取得します。

```
amqsget TEST.Q QM_VERIFY_AMS
```

## タスクの結果

amqsbcg アプリケーションの出力に、キュー内の暗号化されたデータが表示され、メッセージが暗号化されていることを証明します。

## *UNIX* プラットフォームのクイック・スタート・ガイド

このガイドを使用して、 UNIX プラットフォームでメッセージ・セキュリティーを提供するように IBM Advanced Message Security を素早く構成します。 このガイドを完了することにより、ユーザー ID を検証するための鍵データベースが作成され、キュー・マネージャーの署名/暗号化ポリシーが定義されます。

## 始める前に

少なくとも以下のコンポーネントがシステムにインストールされていなければなりません。

- のランタイム
- サーバー
- サンプル・プログラム
- IBM グローバル・セキュリティー・キット
- MQ Advanced Message Security

特定の各プラットフォームでのコンポーネント名については、以下のトピックを参照してください。

- Linux システム用の IBM WebSphere MQ コンポーネント
- HP-UX システム用の IBM WebSphere MQ コンポーネント
- AIX システム用の IBM WebSphere MQ コンポーネント
- Solaris システム用の IBM WebSphere MQ コンポーネント

*1.* キュー・マネージャーおよびキューの作成

## このタスクについて

以下のすべての例では、アプリケーション間でメッセージをやり取りするために TEST.Q という名前のキューを使用します。 Advanced Message Security インターセプターを使用して、標準の WebSphere MQ インターフェースを介して WebSphere MQ インフラストラクチャーに入った時点でメッセージの署名と暗号化を行います。 基本的なセットアップは WebSphere MQ で行い、以下のステップで構成されます。

WebSphere MQ エクスプローラーを使用して、すべてのデフォルト・ウィザード設定を使用して TEST.Q というキュー・マネージャー QM_VERIFY_AMS とそのローカル・キューを作成することも、 <MQ_INSTALL_PATH>/bin にあるコマンドを使用することもできます。 以下の管理コマンドを実行するには、mqm ユーザー・グループのメンバーでなければなりません。

## 手順

1. キュー・マネージャーの作成

```
crtmqm QM_VERIFY_AMS
```

2. キュー・マネージャーを開始する

```
strmqm QM_VERIFY_AMS
```

3. キュー・マネージャー QM_VERIFY_AMS の **runmqsc** に次のコマンドを入力して、TEST.Q というキューを作成します

```
DEFINE QLOCAL(TEST.Q)
```

## タスクの結果

この手順を正常に完了すると、**runmqsc** に以下のコマンドを入力することで、TEST.Q に関する詳細を表示できます。

```
DISPLAY Q(TEST.Q)
```

*2.* ユーザーの作成と許可

## このタスクについて

この例では、送信者の alice と受信者の bob という 2 人のユーザーが登場します。 アプリケーション・キューを使用するには、その使用権限がこれらのユーザーに対し付与されている必要があります。 また、定義する保護ポリシーを正常に使用するには、これらのユーザーに対し、一部のシステム・キューにアクセスするための権限が付与されている必要があります。 **setmqaut** コマンドについて詳しくは、「**setmqaut**」を参照してください。

## 手順

1. 2 人のユーザーを作成します。

```
useradd alice
useradd bob
```

2. これらのユーザーにキュー・マネージャーへの接続およびキューでの作業を行う許可を付与します。

```
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. また、2 人のユーザーに対し、システム・ポリシー・キューの参照およびエラー・キューへのメッセージの書き込みも許可する必要があります。

```
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
```

## タスクの結果

ユーザー・グループが作成され、必要な権限が付与されます。 このように、これらのグループに割り当てられたユーザーにも、キュー・マネージャーに接続するための権限と、キューに対して PUT および GET を行う権限が付与されます。

## 次のタスク

ステップが正しく実行されたかどうかを確認するには、286 ページの『8. 暗号化のテスト』のセクションで説明されているように、 amqsput サンプルと amqsget サンプルを使用します。

*3.* 鍵データベースと証明書の作成

## このタスクについて

メッセージを暗号化するには、インターセプターに送信側ユーザーの秘密鍵と受信者側の公開鍵が必要です。 したがって、公開鍵および秘密鍵にマップされたユーザー ID の鍵データベースを作成する必要があり

ます。 ユーザーおよびアプリケーションが複数のコンピューターに分散している実際のシステムでは、各ユーザーが自分専用の鍵ストアを持っています。 同様に、このガイドでは、alice と bob のための鍵データベースを作成し、両者の間でユーザー証明書を共有します。

**注 :** このガイドでは、ローカル・バインディングを使用して接続する、C 言語で作成されたサンプル・アプリケーションを使用しています。 クライアント・バインディングを使用する Java アプリケーションを使用する予定の場合は、JRE の一部である **keytool** コマンドを使用して JKS 鍵ストアと証明書を作成する必要があります (詳しくは、 286 ページの『Java クライアントのクイック・スタート・ガイド』 を参照してください)。 他のすべての言語の場合、およびローカル・バインディングを使用する Java アプリケーションの場合は、このガイドのステップが正しいことになります。

## 手順

1. ユーザー alice の新規の鍵データベースを作成します。

   ```
   mkdir /home/alice/.mqs -p
   runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
   ```

   **注 :**

   - 強いパスワードを使用して、データベースを保護することをお勧めします。
   - stash パラメーターは、インターセプターがデータベースを開くために使用できる key.sth ファイルにパスワードを格納します。
2. 鍵データベースが読み取り可能であることを確認します。

   ```
   chmod +r /home/alice/.mqs/alicekey.kdb
   ```

3. 暗号化で使用するために、ユーザー alice を識別する証明書を作成します。

   ```
   runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd
   -label Alice_Cert -dn "cn=alice,o=IBM,c=GB" -default_cert yes
   ```

   **注 :**

   - このガイドでは、認証局を利用することなく作成できる自己署名証明書を使用します。 実動システムの場合、自己署名証明書を使用するのではなく、認証局が署名した証明書を信頼することをお勧めします。
   - label パラメーターは、インターセプターが必要な情報を受信するためにルックアップする証明書の名前を指定します。
   - DN パラメーターは、ユーザーごとに固有でなければならない**識別名** (DN) の詳細を指定します。
4. 鍵データベースが作成されます。この所有権を設定し、他のすべてのユーザーがこれを読めないようにします。

   ```
   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. ユーザー bob について、ステップ 1 から 4 までを繰り返します。

## タスクの結果

2 人のユーザー alice および bob は、それぞれ自己署名証明書を保持するようになりました。

*4. keystore.conf の作成*

## このタスクについて

鍵データベースと証明書が置かれているディレクトリーを参照するように Advanced Message Security インターセプターに指示する必要があります。 これは、keystore.conf ファイルを介して行われ、このファイルはその情報をプレーン・テキスト形式で保持します。 各ユーザーは、.mqs フォルダーにそれぞれの

keystore.conf ファイルを持つ必要があります。 このステップは、 alice と bob の両方に対して実行する必要があります。

keystore.conf の内容は、以下のような形式でなければなりません。

```
cms.keystore = <dir>/keystore_file
cms.certificate = certificate_label
```

**例**

このシナリオでは、keystore.conf の内容は以下のようになります。

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

**注 :**

- 鍵ストア・ファイルへのパスは、ファイル拡張子なしで指定する必要があります。
- 鍵ストアの形式には、CMS (Cryptographic Message Syntax)、JKS (Java Keystore)、および JCEKS (Java Cryptographic Extension Keystore) があります。 詳細については、297 ページの『鍵ストア構成ファイル (keystore.conf) の構造』を参照してください。
- HOME/.mqs/keystore.conf は、 Advanced Message Security が keystore.conf ファイルを検索するデフォルトの場所です。 keystore.conf をデフォルト以外の場所で使用する方法については、297 ページの『鍵ストアおよび証明書の使用』を参照してください。

*5.* 証明書の共有

## このタスクについて

各ユーザーが互いを正しく識別できるように、2 つの鍵データベース間で証明書を共有します。 そのために、各ユーザーの公開証明書をファイルに抽出し、そのファイルを他のユーザーの鍵データベースに追加します。

**注 :** エクスポート・オプションではなく、必ず抽出 オプションを使用してください。 抽出 はユーザーの公開鍵を取得しますが、エクスポート は公開鍵と秘密鍵の両方を取得します。 誤ってエクスポート を使用すると、秘密鍵が人手に渡り、アプリケーションのセキュリティーが完全に侵害される可能性があります。

## 手順

1. alice を識別する証明書を外部ファイルに抽出します。

   ```
   runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
   -target alice_public.arm
   ```

2. 証明書を bob's 鍵ストアに追加します。

   ```
   runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
   alice_public.arm
   ```

3. bob について、ステップを繰り返します。

   ```
    runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
   bob_public.arm
   ```

4. bob の証明書を alice's 鍵ストアに追加します。

   ```
   runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
   bob_public.arm
   ```

## タスクの結果

2 人のユーザー alice および bob は、自己署名証明書を作成して共有することで、互いを正しく識別できるようになります。

## 次のタスク

詳細を出力する次のコマンドを実行して、証明書が鍵ストアに置かれていることを確認します。

```
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.* キュー・ポリシーの定義

## このタスクについて

キュー・マネージャーの作成とインターセプターの準備が完了し、メッセージをインターセプトして暗号化鍵にアクセスできるようになったら、setmqspl コマンドを使用して QM_VERIFY_AMS での保護ポリシーの定義を開始できます。 このコマンドの詳細については、setmqspl を参照してください。 各ポリシー名は、適用先のキュー名と同じでなければなりません。

### 例

これは、TEST.Q キューに対して定義されたポリシーの例です。 この例では、メッセージは SHA1 アルゴリズムを使用してユーザー alice によって署名され、256 ビット AES アルゴリズムを使用して暗号化されます。このキューでは、alice が唯一の有効な送信者であり、bob が唯一のメッセージ受信者です。

```
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"
```

**注 :** DN は、鍵データベースからの各ユーザーの証明書に指定された DN と正確に一致します。

## 次のタスク

定義したポリシーを検証するには、以下のコマンドを実行します。

```
dspmqspl -m QM_VERIFY_AMS
```

一連の setmqspl コマンドとしてポリシーの詳細を出力するには、-export フラグを指定します。 これにより、既に定義されているポリシーが格納されます。

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

*7.* セットアップのテスト

## このタスクについて

さまざまなプログラムをさまざまなユーザーのもとで実行することによって、アプリケーションが正しく構成されているかどうかを確認できます。

## 手順

1. サンプルが存在するディレクトリーに移動します。 MQ がデフォルト以外の場所にインストールされている場合、別の場所である可能性があります。

   ```
   cd /opt/mqm/samp/bin
   ```

2. ユーザーを切り替えて、ユーザー alice として実行します。

   ```
   su alice
   ```

3. ユーザー alice として、サンプル・アプリケーションを使用してメッセージを配置します。

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. メッセージのテキストを入力して、Enter キーを押します。
5. ユーザー alice として実行を停止します。

```
exit
```

6. ユーザーを切り替えて、ユーザー bob として実行します。

```
su bob
```

7. ユーザー bob として、サンプル・アプリケーションを使用してメッセージを取得します。

```
./amqsget TEST.Q QM_VERIFY_AMS
```

### タスクの結果

両方のユーザーでアプリケーションが正しく構成されている場合、bob が取得アプリケーションを実行したときにユーザー alice のメッセージが表示されます。

*8.* 暗号化のテスト

### このタスクについて

暗号化が正しく行われていることを検証するには、元のキュー TEST.Q を参照する別名キューを作成します。 この別名キューにはセキュリティー・ポリシーがないため、メッセージを復号するための情報を持つユーザーは存在しません。これにより、暗号化されたデータが示されることになります。

### 手順

1. キュー・マネージャー QM_VERIFY_AMS に対して **runmqsc** コマンドを使用して、別名キューを作成します。

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. bob アクセス権を付与して、別名キューから参照できるようにします。

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. ユーザー alice として、前述のステップと同様にサンプル・アプリケーションを使用して別のメッセージを配置します。

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. 次に、ユーザー bob として、別名キュー経由でサンプル・アプリケーションを使用してメッセージを参照します。

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. ユーザー bob として、ローカル・キューからサンプル・アプリケーションを使用してメッセージを取得します。

```
./amqsget TEST.Q QM_VERIFY_AMS
```

### タスクの結果

amqsbcg アプリケーションの出力に、キュー内の暗号化されたデータが表示され、メッセージが暗号化されていることを証明します。

## *Java* クライアントのクイック・スタート・ガイド

このガイドを使用して、クライアント・バインディングを使用して接続する Java アプリケーションのメッセージ・セキュリティーを提供するように IBM Advanced Message Security を素早く構成します。 このガイドを完了することにより、ユーザー ID を検証するための鍵ストアが作成され、キュー・マネージャーの署名/暗号化ポリシーが定義されます。

## 始める前に

**クイック・スタート・ガイド** (Windows または UNIX) で説明されているように、適切なコンポーネントがインストールされていることを確認してください。

*1.* キュー・マネージャーおよびキューの作成

## このタスクについて

以下のすべての例では、アプリケーション間でメッセージをやり取りするために TEST.Q という名前のキューを使用します。 Advanced Message Security インターセプターを使用して、標準の WebSphere MQ インターフェースを介して WebSphere MQ インフラストラクチャーに入った時点でメッセージの署名と暗号化を行います。 基本的なセットアップは WebSphere MQ で行い、以下のステップで構成されます。

## 手順

1. キュー・マネージャーの作成

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. キュー・マネージャーを開始する

   ```
   strmqm QM_VERIFY_AMS
   ```

3. キュー・マネージャー QM_VERIFY_AMS の **runmqsc** に次のコマンドを入力して、リスナーを作成および始動します。

   ```
   DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
   START LISTENER(AMS.LSTR)
   ```

4. キュー・マネージャー QM_VERIFY_AMS の **runmqsc** に次のコマンドを入力して、アプリケーションが接続時に使用するチャネルを作成します。

   ```
   DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
   ```

5. キュー・マネージャー QM_VERIFY_AMS の **runmqsc** に次のコマンドを入力して、TEST.Q というキューを作成します

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

## タスクの結果

この手順を正常に完了すると、**runmqsc** に以下のコマンドを入力することで、TEST.Q に関する詳細を表示できます。

```
DISPLAY Q(TEST.Q)
```

*2.* ユーザーの作成と許可

## このタスクについて

このシナリオには送信者の alice と受信者の bob という 2 人のユーザーが登場します。 アプリケーション・キューを使用するには、その使用権限がこれらのユーザーに対し付与されている必要があります。 また、定義する保護ポリシーを正常に使用するには、これらのユーザーに対し、一部のシステム・キューにアクセスするための権限が付与されている必要があります。 **setmqaut** コマンドについて詳しくは、「**setmqaut**」を参照してください。

## 手順

1. 使用しているプラットフォームの**クイック・スタート・ガイド** (Windows または UNIX) で説明されているように、2 人のユーザーを作成します。
2. これらのユーザーにキュー・マネージャーへの接続およびキューでの作業を行う許可を付与します。

```
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. また、2 人のユーザーに対し、システム・ポリシー・キューの参照およびエラー・キューへのメッセージの書き込みも許可する必要があります。

```
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
```

## タスクの結果
ユーザーが作成され、必要な権限が付与されました。

## 次のタスク
ステップが正しく実行されたかどうかを確認するには、290 ページの『7. セットアップのテスト』のセクションで説明されているように、 JmsProducer サンプルと JmsConsumer サンプルを使用します。

*3.* 鍵データベースと証明書の作成

## このタスクについて
インターセプターがメッセージを暗号化するには、送信側ユーザーの公開鍵が必要です。 したがって、公開鍵および秘密鍵にマップされたユーザー ID の鍵データベースを作成する必要があります。 ユーザーおよびアプリケーションが複数のコンピューターに分散している実際のシステムでは、各ユーザーが自分専用の鍵ストアを持っています。 同様に、このガイドでは、alice と bob のための鍵データベースを作成し、両者の間でユーザー証明書を共有します。

**注 :** このガイドでは、クライアント・バインディングを使用して接続する Java で作成されたサンプル・アプリケーションを使用します。 ローカル・バインディングを使用する Java アプリケーションまたは C アプリケーションを使用する予定の場合、**runmqakm** コマンドを使用して CMS 鍵ストアおよび証明書を作成する必要があります。 これは、**クイック・スタート・ガイド** (Windows または UNIX) で示されています。

## 手順
1. 鍵ストアを作成するディレクトリー (例えば、/home/alice/.mqs) を作成します。 使用しているプラットフォームの**クイック・スタート・ガイド** (Windows または UNIX) で使用されているものと同じディレクトリーに鍵ストアを作成することもできます。

   **注 :** 以下の手順では、このディレクトリーを *keystore-dir* と表しています。
2. 暗号化で使用するために、ユーザー alice を識別する新規の鍵ストアおよび証明書を作成します。

   **注 : keytool** コマンドは、JRE の一部です。

```
keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks
-storepass passw0rd
-dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
```

   **注 :**

   - *keystore-dir* にスペースが含まれている場合、鍵ストアの絶対パス名を引用符で囲む必要があります。
   - 強いパスワードを使用して、鍵ストアを保護することをお勧めします。
   - このガイドでは、認証局を利用することなく作成できる自己署名証明書を使用します。 実動システムの場合、自己署名証明書を使用するのではなく、認証局が署名した証明書を信頼することをお勧めします。
   - alias パラメーターは、インターセプターが必要な情報を受信するためにルックアップする証明書の名前を指定します。
   - dname パラメーターは、ユーザーごとに固有でなければならない**識別名** (DN) の詳細を指定します。
3. UNIX の場合、鍵ストアが読み取り可能であることを確認します。

```
chmod +r keystore-dir/keystore.jks
```

4. ユーザー bob

## タスクの結果

2 人のユーザー alice および bob は、それぞれ自己署名証明書を保持するようになりました。

*4. keystore.conf の作成*

## このタスクについて

鍵データベースと証明書が置かれているディレクトリーを参照するように Advanced Message Security インターセプターに指示する必要があります。 これは、この情報を平文形式で保持する keystore.conf ファイルを介して行います。 各ユーザーは、それぞれの keystore.conf ファイルを持つ必要があります。 このステップは、alice および bob の両方に対して行う必要があります。

### 例

このシナリオでは、alice の keystore.conf の内容は以下のようになります。

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

このシナリオでは、bob の keystore.conf の内容は以下のようになります。

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

**注 :**

- 鍵ストア・ファイルへのパスは、ファイル拡張子なしで指定する必要があります。
- 「**クイック・スタート・ガイド**」(Windows または UNIX) に従って既に keystore.conf を使用している場合は、既存のものを編集して上記の行に追加することができます。
- 詳しくは、297 ページの『鍵ストア構成ファイル (keystore.conf) の構造』を参照してください。

*5.* 証明書の共有

## このタスクについて

各ユーザーが互いを正しく識別できるように、2 つの鍵ストア間で証明書を共有します。 各ユーザーの証明書を抽出し、他のユーザーの鍵ストアにインポートすることで、共有できます。

**注 :** 抽出 とエクスポート という言葉の意味は、証明書ツールによって異なります。 例えば、IBM GSKit Keyman (ikeyman) ツールでは、抽出 するものは証明書 (公開鍵) であり、エクスポート するものは秘密鍵であるというように区別しています。 両方のオプションが用意されているツールを使用する場合は、この区別が非常に重要です。誤ってエクスポート を使用すると、秘密鍵が人手に渡り、アプリケーションのセキュリティーが完全に侵害される可能性があるからです。 この区別は非常に重要であるため、WebSphere MQ の資料では、これらの言葉を一貫して使用するように努めています。 一方、Java の keytool の *exportcert* というコマンド・ライン・オプションは、公開鍵のみを抽出します。 このため、以下の手順では、*exportcert* オプションを使用する場合に、証明書を抽出する という言葉を使用しています。

## 手順

1. alice を識別する証明書を抽出します。

```
keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
-alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
```

2. alice を識別する証明書を、bob が使用する鍵ストアにインポートします。 プロンプトが表示されたら、この証明書を信頼することを指定します。

```
keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
-keystore bob-keystore-dir/keystore.jks -storepass passw0rd
```

3. bob

## タスクの結果

2 人のユーザー alice および bob は、自己署名証明書を作成して共有することで、互いを正しく識別できるようになります。

## 次のタスク

詳細を出力する次のコマンドを実行して、証明書が鍵ストアに置かれていることを確認します。

```
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.* キュー・ポリシーの定義

## このタスクについて

キュー・マネージャーの作成とインターセプターの準備が完了し、メッセージをインターセプトして暗号化鍵にアクセスできるようになったら、setmqspl コマンドを使用して QM_VERIFY_AMS での保護ポリシーの定義を開始できます。 このコマンドの詳細については、setmqspl を参照してください。 各ポリシー名は、適用先のキュー名と同じでなければなりません。

### 例

これは、TEST.Q キューで定義されるポリシーの例で、SHA1 アルゴリズムを使用してユーザー alice によって署名され、ユーザー bob のために 256 ビットの AES アルゴリズムを使用して暗号化されています。

```
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"
```

**注 :** DN は、鍵データベースからの各ユーザーの証明書に指定された DN と正確に一致します。

## 次のタスク

定義したポリシーを検証するには、以下のコマンドを実行します。

```
dspmqspl -m QM_VERIFY_AMS
```

一連の setmqspl コマンドとしてポリシーの詳細を出力するには、-export フラグを指定します。 これにより、既に定義されているポリシーが格納されます。

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

*7.* セットアップのテスト

## 始める前に

使用している Java のバージョンに、無制限 JCE ポリシー・ファイルがインストールされていることを確認します。

**注 :** WebSphere MQ のインストールで提供される Java のバージョンには、既にこのポリシー・ファイルが存在します。 これは、*MQ_INSTALLATION_PATH*/java/bin にあります。

## このタスクについて

さまざまなプログラムをさまざまなユーザーのもとで実行することによって、アプリケーションが正しく構成されているかどうかを確認できます。 異なる複数のユーザー下でプログラムを実行する方法の詳細については、使用しているプラットフォームの **クイック・スタート・ガイド** (Windows または UNIX) を参照してください。

## 手順

1. これらの JMS サンプル・アプリケーションを実行するには、IBM WebSphere MQ classes for JMS で使用される環境変数に示されているように、プラットフォームの CLASSPATH 設定を使用して、サンプル・ディレクトリーが含まれるようにしてください。
2. クライアントとして接続するサンプル・アプリケーションを使用し、ユーザー `alice` としてメッセージを配置します。

```
java JMSProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
```

3. クライアントとして接続するサンプル・アプリケーションを使用し、ユーザー bob としてメッセージを取得します。

```
java JMSConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
```

## タスクの結果

両方のユーザーでアプリケーションが正しく構成されている場合、bob が取得アプリケーションを実行したときにユーザー `alice` のメッセージが表示されます。

## リモート・キューの保護

リモート・キュー接続を完全に保護するには、メッセージの送信先のリモート・キューとローカル・キューに同じポリシーを設定する必要があります。

メッセージがリモート・キューに入ると、 Advanced Message Security は操作をインターセプトして、リモート・キューに設定されているポリシーに従ってメッセージを処理します。 例えば、暗号化ポリシーの場合、メッセージは WebSphere MQ に渡されて処理される前に暗号化されます。 リモート・キューに入れられたメッセージを Advanced Message Security が処理すると、WebSphere MQ は、そのメッセージを関連する伝送キューに入れ、ターゲット・キュー・マネージャーとターゲット・キューに転送します。

GET 操作がローカル・キューに対して実行されると、 Advanced Message Security は、ローカル・キューに設定されたポリシーに従ってメッセージをデコードしようとします。 この操作が成功するためには、メッセージの復号に使用されるポリシーが、メッセージの暗号化に使用されたポリシーと同じでなければなりません。 相違があれば、メッセージは拒否されます。

何らかの理由で両方のポリシーを同時に設定できない場合は、ステージングされたロールアウト・サポートが提供されます。 ポリシーは、容認フラグをオンにしてローカル・キュー上に設定できます。これは、このキューからメッセージを取得しようとした場合に、セキュリティー・ポリシーが設定されていないメッセージがあれば、キューに関連付けられたポリシーを無視できることを示します。 この場合、GET はメッセージを復号しようとしますが、非暗号化メッセージの送信を許可します。 このような方法で、ローカル・キューが保護 (およびテスト) された後で、リモート・キュー上のポリシーを設定できます。

**要確認 :** 容認フラグは、 Advanced Message Security ロールアウトが完了した後に除去してください。

**関連資料**

setmqspl (セキュリティー・ポリシーの設定)

## 保護されたメッセージの *WebSphere Message Broker* を使用した経路指定

IBM Advanced Message Security は、WebSphere Message Broker バージョン 8.0.0.1 (またはそれ以降) がインストールされているインフラストラクチャーでメッセージを保護できます。 WebSphere Message Broker 環境でセキュリティーを適用する前に、両方の製品の性質を理解する必要があります。

## このタスクについて

Advanced Message Security メッセージ・ペイロードにエンドツーエンドのセキュリティーを提供します。これは、メッセージの正当な送信者および受信者として指定されている当事者のみが、そのメッセージを作成または受信できるという意味です。 これは、 WebSphere Message Broker を流れるメッセージを保護するために、 WebSphere Message Broker がメッセージの内容を知らずにメッセージを処理できるようにするか ( シナリオ 1)、または許可ユーザーがメッセージを送受信できるようにすることができるようにするか ( シナリオ 2) を意味します。

シナリオ *1 - Message Broker* がメッセージの内容を認識しない

## 始める前に

WebSphere Message Broker を既存のキュー・マネージャーに接続しておく必要があります。 以下で示しているコマンドの *QMgrName* を、この既存のキュー・マネージャー名で置き換えます。

## このタスクについて

このシナリオでは、Alice が入力キュー QIN に保護メッセージを入れます。 メッセージ・プロパティー routeTo に基づいて、メッセージは *bob* (QBOB) にルーティングされます。[1](QCECIL)、またはデフォルト (QDEF) キュー。 このような経路指定が可能なのは、 Advanced Message Security が、ヘッダーとプロパティーではなくメッセージ・ペイロードのみを保護するためです。このヘッダーとプロパティーは保護されないため、WebSphere Message Broker が読み取ることができます。 Advanced Message Security *alice*、*bob* 、および *cecil* でのみ使用されます。 これをインストールしたり、WebSphere Message Broker 用に構成したりする必要はありません。

WebSphere Message Broker は、当該メッセージが復号試行されるのを防ぐために、無保護の別名キューから保護されたメッセージを受け取ります。 保護されたキューを直接使用する場合、当該メッセージは復号不可メッセージとして送達不能キューに送られます。 このメッセージは WebSphere Message Broker によってルーティングされ、変更されずにターゲット・キューに到達します。 そのため、メッセージは元の作成者によって署名されたままとなり (*bob* と *cecil* は、いずれも *alice* が送信したメッセージのみを受け入れる)、元のまま保護されています (*bob* と *cecil* のみがメッセージを読むことができる)。 WebSphere Message Broker は、ルーティングされたメッセージを無保護別名として配置します。 受信者は、保護された出力キューからこのメッセージを受け取ります。このキューでは、IBM WebSphere MQ AMS がメッセージを透過的に復号します。

## 手順

1. 「**クイック・スタート・ガイド** 」(Windows または UNIX) の説明に従って、Advanced Message Security を使用するように *alice*、 *bob* 、および *cecil* を構成します。

   以下のステップが完了していることを確認します。

   - ユーザーの作成と許可
   - 鍵データベースと証明書の作成
   - keystore.conf の作成
2. *alice* の証明書を *bob* と *cecil* に提供し、メッセージのデジタル署名の検証時に *alice* を識別できるようにします。

   そのために、*alice* を識別する証明書を外部ファイルに抽出し、抽出した証明書を *bob* と *cecil* の鍵ストアに追加します。 **タスク 5 で説明されている方法を使用することが重要です。 クイック・スタート・ガイド (Windows または UNIX) 内の 証明書の共有** 。
3. *bob* と *cecil* の証明書を *alice* に提供し、*alice* が *bob* と *cecil* のために暗号化されたメッセージを送信できるようにします。

   これは、前の手順で示した方法で行います。
4. キュー・マネージャーで、ローカル・キュー (QIN、QBOB、QCECIL、および QDEF) を定義します。

[1] cecil's

```
DEFINE QLOCAL(QIN)
```

5. QIN キューのセキュリティー・ポリシーを適格な構成にセットアップします。 QBOB、QCECIL、および QDEF キューにも、同じセットアップを使用します。

```
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"
```

このシナリオでは、*alice* が唯一の許可された送信者で、*bob* と *cecil* が受信者であるというセキュリティー・ポリシーを想定しています。

6. それぞれローカル・キュー (QIN、QBOB、および QCECIL) を参照する別名キュー (AIN、ABOB、および ACECIL) を定義します。

```
DEFINE QALIAS(AIN) TARGET(QIN)
```

7. 前のステップで指定した別名のセキュリティー構成がないことを確認します。セキュリティー構成がある場合は、そのポリシーを NONE に設定します。

```
dspmqspl -m QMgrName -p AIN
```

8. WebSphere Message Broker で、AIN 別名キューに到達したメッセージを、メッセージの routeTo プロパティーに従って BOB、CECIL、または DEF ノードにルーティングするメッセージ・フローを作成します。 これを行うには、以下のようにします。
   a) MQInput ノードを IN という名前で作成し、AIN という別名をキュー名と割り当てます。
   b) MQOutput ノード (BOB、CECIL、および DEF) を作成し、それぞれのキュー名として別名キュー (ABOB、ACECIL、および ADEF) を割り当てます。
   c) 経路ノードを作成して、TEST という名前を付けます。
   d) IN ノードを TEST ノードの入力ターミナルに接続します。
   e) TEST ノード用に bob および cecil 出力ターミナルを作成します。
   f) bob 出力ターミナルを BOB ノードに接続します。
   g) cecil 出力ターミナルを CECIL ノードに接続します。
   h) DEF ノードをデフォルトの出力ターミナルに接続します。
   i) 以下のルールを適用します。

```
$Root/MQRFH2/usr/routeTo/text()="bob"
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. メッセージ・フローを WebSphere Message Broker ランタイム・コンポーネントにデプロイします。
10. 実行ユーザー Alice として、値が bob または cecil であるメッセージ・プロパティー routeTo も含まれるメッセージを配置します。 サンプル・アプリケーション **amqsstm** を実行することで、これを行うことができます。

```
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. 実行ユーザー *bob* として、サンプル・アプリケーション **amqsget** を使用してキュー QBOB からメッセージを取得します。

## タスクの結果

*alice* が QIN キューにメッセージを配置すると、メッセージは保護されます。 このメッセージは、WebSphere Message Broker によって、AIN 別名キューから保護された形式で取得されます。 WebSphere Message Broker 暗号化されていない routeTo プロパティー (すべてのプロパティーも同様) を読み取って、メッセージの経路指定先を決定します。 WebSphere Message Broker は、適切な無保護の別名キューにメッセージを入れ、メッセージが重複して保護されないようにします。 *bob* または *cecil* がキューからメッセージを受信すると、メッセージは復号され、デジタル署名が検査されます。

シナリオ *2 - Message Broker* がメッセージの内容を認識する

## このタスクについて

このシナリオでは、個人のグループが WebSphere Message Broker にメッセージを送信することができます。 別のグループに、WebSphere Message Broker によって作成されたメッセージを受け取る許可が付与されます。 当事者と WebSphere Message Broker との間の伝送が盗聴されることはありません。

WebSphere Message Broker が保護ポリシーと証明書を読み取るのはキューが開かれたときのみであるため、保護ポリシーを更新した場合は、実行グループを再ロードして変更を有効にする必要があります。

```
mqsireload execution-group-name
```

WebSphere Message Broker が、メッセージ・ペイロードの読み取りまたは署名を許可された許可されたパーティーであると見なされる場合は、 WebSphere Message Broker サービスを開始するユーザー用に Advanced Message Security を構成する必要があります。 キューに対してメッセージを PUT または GET するユーザーや、WebSphere Message Broker アプリケーションを作成およびデプロイするユーザーと、必ずしも同じユーザーである必要はありません。

## 手順

1. 「**クイック・スタート・ガイド**」(Windows または UNIX) の説明に従って、 Advanced Message Security を使用するように *alice*、 *bob*、 *cecil* 、 *dave* 、および WebSphere Message Broker サービス・ユーザーを構成します。

   以下のステップが完了していることを確認します。

   - ユーザーの作成と許可
   - 鍵データベースと証明書の作成
   - keystore.conf の作成
2. *alice*、*bob*、*cecil* および *dave* の証明書を WebSphere Message Broker サービス・ユーザーに提供します。

   そのために、*alice*、*bob*、*cecil*、*dave* の身元に関する各証明書を外部ファイルに抽出し、抽出した証明書を WebSphere Message Broker の鍵ストアに追加します。 **タスク 5 で説明されている方法を使用することが重要です。 クイック・スタート・ガイド (Windows または UNIX) 内の 証明書の共有** 。
3. WebSphere Message Broker サービス・ユーザーの証明書を *alice*、*bob*、*cecil*、および *dave* に提供します。

   これは、前の手順で示した方法で行います。

   **注 :** *Alice* と *bob* は、メッセージを正しく暗号化するために、WebSphere Message Broker サービス・ユーザーの証明書を必要とします。 WebSphere Message Broker サービス・ユーザーは、メッセージの作成者を検証するために、*alice* と *bob* の証明書を必要とします。 WebSphere Message Broker サービス・ユーザーは、*cecil* と *dave* のメッセージを暗号化するために、この両者の証明書を必要とします。 *cecil* と *dave* は、メッセージが WebSphere Message Broker から送信されたことを検証するために、WebSphere Message Broker サービス・ユーザーの証明書を必要とします。
4. ローカル・キュー IN を定義し、*alice* と *bob* を作成者として、また WebSphere Message Broker のサービス・ユーザーを受信者として指定した、セキュリティー・ポリシーを定義します。

   ```
   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. ローカル・キュー OUT を定義し、WebSphere Message Broker のサービス・ユーザーを作成者として、また *cecil* と *dave* を受信者として指定した、セキュリティー・ポリシーを定義します。

```
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 Broker で、MQInput ノードおよび MQOutput ノードを使用してメッセージ・フローを作成します。 IN キューを使用するように MQInput ノードを構成し、OUT キューを使用するように MQOutput ノードを構成します。
7. メッセージ・フローを WebSphere Message Broker ランタイム・コンポーネントにデプロイします。
8. 実行ユーザー *alice* または *bob* として、サンプル・アプリケーション **amqsput** を使用してキュー IN にメッセージを配置します。
9. 実行ユーザー *cecil* または *dave* として、サンプル・アプリケーション **amqsget** を使用してキュー OUT からメッセージを取得します。

## タスクの結果

*alice* または *bob* が入力キュー IN に送信するメッセージは暗号化され、WebSphere Message Broker のみがそのメッセージを読み取ることができます。 WebSphere Message Broker は、*alice* と *bob* からのメッセージのみを受け入れ、他のユーザーからのメッセージは拒否します。 受け入れられたメッセージは適切に処理され、*cecil* と *dave* の鍵で署名および暗号化されてから、出力キュー OUT に配置されます。 *cecil* と *dave* のみが、 そのメッセージを読み取ることができ、WebSphere Message Broker によって署名されていないメッセージは拒否されます。

# *IBM WebSphere MQ Advanced Message Security* と *IBM WebSphere MQ Managed File Transfer* の使用

このシナリオでは、IBM WebSphere MQ Managed File Transfer を介して送信されるデータのメッセージ・プライバシーを提供するように Advanced Message Security を構成する方法について説明します。

## 始める前に

保護する IBM WebSphere MQ Managed File Transfer によって使用されるキューをホストする Advanced Message Security コンポーネントが WebSphere MQ インストール済み環境にインストールされていることを確認します。

IBM WebSphere MQ Managed File Transfer エージェントがバインディング・モードで接続している場合、そのローカル・インストール済み環境に GSKit コンポーネントもインストールされていることを確認してください。

## このタスクについて

2 つの IBM WebSphere MQ Managed File Transfer エージェント間のデータ転送が中断した場合、転送の管理に使用されている基礎の WebSphere MQ キューで、機密データが無保護のままになっていた可能性があります。 このシナリオでは、 Advanced Message Security を構成および使用して、IBM WebSphere MQ Managed File Transfer キューでそのようなデータを保護する方法について説明します。

このシナリオでは、シナリオ「スクリプトを使用した基本的なファイル転送」で説明されているように、2 つの IBM WebSphere MQ Managed File Transfer キューと 2 つのエージェント AGENT1 および AGENT2 が 1 つのキュー・マネージャー hubQM を共有する 1 つのマシンで構成される単純なトポロジーを検討します。 どちらのエージェントも、同じ方法で接続されています。この方法は、バインディング・モードまたはクライアント・モードのいずれかです。

*1.* 証明書の作成

## 始める前に

このシナリオでは、グループ FTAGENTS 内のユーザー ftagent を使用して IBM WebSphere MQ Managed File Transfer エージェント・プロセスを実行する単純なモデルを使用します。 独自のユーザー名やグループ名を使用している場合は、それに応じてコマンドを変更してください。

## このタスクについて

Advanced Message Security は、保護されたキューのメッセージに対して署名および暗号化、またはそのいずれかを実行するために、公開鍵暗号を使用します。

**注 :**

- IBM WebSphere MQ Managed File Transfer エージェントがバインディング・モードで実行されている場合、CMS (Cryptographic Message Syntax) 鍵ストアの作成に使用するコマンドについては、ご使用のプラットフォームの**クイック・スタート・ガイド** (Windows または UNIX) で詳述されています。
- IBM WebSphere MQ Managed File Transfer エージェントがクライアント・モードで実行されている場合、JKS (Java Keystore) の作成に必要なコマンドは、286 ページの『Java クライアントのクイック・スタート・ガイド』で詳述されています。

## 手順

1. 該当するクイック・スタート・ガイドで説明されているように、ユーザー `ftagent` を識別する自己署名証明書を作成します。
   次のように識別名 (DN) を使用します。

   ```
   CN=ftagent, OU=MFT, O=IBM, L=Hursley, ST=Hampshire, C=GB
   ```

2. 該当するクイック・スタート・ガイドで説明されているように、鍵ストアおよびその中の証明書が存在する場所を識別する `keystore.conf` ファイルを作成します。

*2.* メッセージ保護の構成

## このタスクについて

**setmqspl** コマンドを使用して、AGENT2 によって使用されるデータ・キューのセキュリティー・ポリシーを定義する必要があります。 ここで示すシナリオでは、同じユーザーを使用して両方のエージェントを開始するので、署名者と受信者の DN は同じになり、生成した証明書と一致します。

## 手順

1. **fteStopAgent** コマンドを使用して、保護の準備として IBM WebSphere MQ Managed File Transfer エージェントをシャットダウンします。
2. SYSTEM.FTE.DATA.AGENT2 キューを保護するセキュリティー・ポリシーを作成します。

   ```
   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 エージェント・プロセスを実行しているユーザーに、システム・ポリシー・キューの参照およびエラー・キューへのメッセージの書き込みを行うためのアクセス権があることを確認します。

   ```
   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** コマンドを使用して、 IBM WebSphere MQ Managed File Transfer エージェントを再始動します。
5. **fteListAgents** コマンドを使用して、エージェントが READY 状況になっていることを確認することで、それらのエージェントが正常に再始動したことを確認します。

## タスクの結果

これで、AGENT1 から AGENT2 に転送を行うことができます。ファイル内容は、2 つのエージェント間で保護されて送信されます。

# のインストール IBM WebSphere MQ Advanced Message Security

各種プラットフォームに IBM WebSphere MQ Advanced Message Security コンポーネントをインストールします。

## このタスクについて

詳細なインストール手順については、IBM WebSphere MQ Advanced Message Security のインストールを参照してください。

**関連タスク**

のアンインストール IBM WebSphere MQ Advanced Message Security

# 鍵ストアおよび証明書の使用

WebSphere MQ アプリケーションにトランスペアレントな暗号保護を提供するために、 Advanced Message Security は鍵ストア・ファイルを使用します。このファイルには、公開鍵証明書と秘密鍵が格納されています。

Advanced Message Security では、ユーザーおよびアプリケーションは、公開鍵インフラストラクチャー (PKI) ID によって表されます。 このタイプの ID は、メッセージの署名と暗号化に使用されます。 PKI ID は、署名および暗号化されたメッセージに関連付けられている証明書内のサブジェクトの**識別名 (DN)** フィールドによって表されています。 ユーザーまたはアプリケーションがメッセージを暗号化するには、証明書および関連付けられている秘密鍵と公開鍵が格納されている鍵ストア・ファイルに対するアクセス権限が必要です。

鍵ストアの場所は鍵ストア構成ファイル (デフォルトでは `keystore.conf`) に指定されます。 鍵ストア・ファイルを指す鍵ストア構成ファイルは、 Advanced Message Security ユーザーごとに必要です。 Advanced Message Security `.kdb`、 `.jceks`、 `.jks` の形式の鍵ストア・ファイルを受け入れます。

`keystore.conf` ファイルのデフォルトの場所は、以下のとおりです。

- UNIX プラットフォームの場合: `$HOME/.mqs/keystore.conf`
- Windows プラットフォームの場合: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

指定した鍵ストアのファイル名と場所を使用している場合は、以下のコマンドを使用する必要があります。

- Java の場合: `java -D` *`MQS_KEYSTORE_CONF=path/filename app_name`*
- C クライアントおよびサーバーの場合:
  - UNIX and Linux の場合、`export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windows の場合、`set MQS_KEYSTORE_CONF=`*`path\filename`*

    **注 :** 複数のドライブ名が使用可能な場合は、Windows 上のパスでドライブ名を指定できます。使用する場合は、指定する必要があります。

**関連概念**

310 ページの『送信側の識別名』
送信側の識別名 (DN) は、メッセージをキューに置く権限が付与されているユーザーを識別します。

311 ページの『受信側の識別名』
受信者識別名 (DN) は、キューからメッセージを取り出す権限が付与されているユーザーを識別します。

## 鍵ストア構成ファイル (keystore.conf) の構造

鍵ストア構成ファイル (`keystore.conf`) は、 Advanced Message Security が適切な鍵ストアの場所を指すようにします。

CMS と Java の構成には 2 つのタイプ (JKS と JCEKS) があります。 鍵ストアのタイプに応じて、CMS 構成エントリーには `cms.` という接頭部が付き、Java には `jks.` または `jceks.` という接頭部が付きます。

構成ファイルは、そのタイプに応じて以下のいずれかの構造になります。

```
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
```

構成ファイル・パラメーターは以下のように定義されます。

**keystore**
鍵ストア・ファイルへのパス。

**重要 :**

- 鍵ストア・ファイルへのパスには、拡張子を含まないようにする必要があります。
- Java 鍵ストア・ファイルの場合、 IBM WebSphere MQ AMS はファイル・フォーマット .jks、 .jceks、 .jck をサポートします。

**certificate**
証明書ラベル。

**encrypted**
パスワードの状況。

**keystore_pass**
鍵ストア・ファイルのパスワード。

**注 :**

- CMS 鍵ストアの場合、IBM WebSphere MQ AMS は stash ファイル (.sth) に依存しますが、JKS および JCEKS では、証明書とユーザー秘密鍵の両方のパスワードが必要となる場合があります。
- パスワードを平文で格納すると、セキュリティー・リスクになります。

**key_pass**
ユーザーの秘密鍵のパスワード。

**重要 :** パスワードを平文形式で格納すると、セキュリティー・リスクが生じる可能性があります。

**provider**
鍵ストア証明書に必要な暗号アルゴリズムを実装する Java セキュリティー・プロバイダー。

**注 :** 現在、IBMJCE が、 Advanced Message Security でサポートされている唯一のプロバイダーです。

**重要 :** 鍵ストアに格納される情報は、WebSphere MQ を使用して送信されるデータの安全なフローにとって不可欠です。セキュリティー管理者が、これらのファイルに対するファイル許可を割り当てる際に特に注意を払う必要があるのは、このような理由によります。

以下に keystore.conf ファイルの例を示します。

```
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
```

**関連タスク**

308 ページの『Java でのパスワードの保護』

鍵ストアと秘密鍵のパスワードを平文で格納するとセキュリティー・リスクが発生するため、 Advanced Message Security には、ユーザーの鍵 (鍵ストア・ファイル内で取得可能) を使用してこれらのパスワードの順序を変えることができるツールが用意されています。

## メッセージ・チャネル・エージェント (MCA) インターセプト

MCA インターセプトにより、 IBM WebSphere MQ の下で実行されるキュー・マネージャーは、サーバー接続チャネルに適用するポリシーを選択的に使用可能にすることができます。

MCA インターセプトを使用することで、IBM WebSphere MQ AMS の外部にあるクライアントは、引き続きキュー・マネージャーに接続し、メッセージを暗号化および復号できます。

MCA インターセプトの目的は、クライアントで IBM WebSphere MQ AMS を有効にできない場合に IBM WebSphere MQ AMS の機能を利用できるようにすることです。 MCA インターセプトと IBM WebSphere MQ AMS 対応クライアントを使用すると、メッセージの保護が二重になり、受信側のアプリケーションにとって問題になる可能性があります。

Version 7.5 以降のクライアントを使用して旧バージョンの製品からキュー・マネージャーに接続している場合に 2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告される場合は、クライアントで IBM WebSphere MQ Advanced Message Security を無効にする必要があります。 詳しくは、301 ページの『クライアントでの IBM WebSphere MQ Advanced Message Security の無効化』を参照してください。

### 鍵ストア構成ファイル

デフォルトでは、MCA インターセプトの鍵ストア構成ファイルは keystore.conf で、キュー・マネージャーまたはリスナーを開始したユーザーの HOME ディレクトリー・パスの .mqs ディレクトリーに配置されます。 鍵ストアは、MQS_KEYSTORE_CONF 環境変数を使用して構成することもできます。 IBM WebSphere MQ AMS 鍵ストアの構成について詳しくは、297 ページの『鍵ストアおよび証明書の使用』を参照してください。

MCA インターセプトを有効にするには、使用するチャネル名を鍵ストア構成ファイルに指定する必要があります。 MCA インターセプトでは、CMS タイプの鍵ストアのみ使用可能です。

MCA インターセプトのセットアップの例については、299 ページの『IBM WebSphere MQ AMS MCA インターセプトの例』を参照してください。

**重要 :** 許可されたクライアントのみが接続してこの機能を使用できるようにするために、SSL と SSLPEER または CHLAUTH TYPE (SSLPEERMAP) を使用するなどして、選択したチャネルのクライアント認証と暗号化を完了する必要があります。

### *IBM WebSphere MQ AMS MCA インターセプトの例*

IBM WebSphere MQ AMS MCA インターセプトのセットアップ方法に関するタスク例。

### 始める前に

**重要 :** 許可されたクライアントのみが接続してこの機能を使用できるようにするために、SSL と SSLPEER または CHLAUTH TYPE (SSLPEERMAP) を使用するなどして、選択したチャネルのクライアント認証と暗号化を完了する必要があります。

### このタスクについて

このタスクでは、MCA インターセプトを使用するようにシステムをセットアップし、そのセットアップを検証するプロセスについて説明します。

**注 :** IBM WebSphere MQ Version 7.5 より前は、IBM WebSphere MQ AMS は、別途インストールが必要なアドオン製品であり、アプリケーションを保護するためにインターセプターを構成する必要がありました。 Version 7.5 以降、インターセプターは、MQ のクライアントとサーバーのランタイム環境に自動的に組み込まれ、動的に有効になります。 この MCA インターセプトの例では、インターセプターがチャネルのサーバー側に用意されており、古いクライアント・ランタイムを使用して (手順 12)、チャネル経由で無保護メ

ッセージを書き込むので、MCA インターセプターでメッセージが保護される様子を確認できます。 この例で Version 7.5 以降のクライアントを使用した場合、メッセージは 2 回保護されることになります。メッセージが MQ に到着するときに MQ クライアント・ランタイムのインターセプターと MCA インターセプターの両方によって保護されるからです。

**重要 :** コード内の userID はご使用のユーザー ID に置き換えてください。

## 手順

1. 以下のコマンドを使用してシェル・スクリプトを作成することによって、鍵データベースと証明書を作成します。

   また、**INSTLOC** と **KEYSTORELOC** を変更するか、必要なコマンドを実行してください。 bob 用の証明書は作成する必要がない場合もあります。

```
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. 各ユーザーが互いを正しく識別できるように、2 つの鍵データベース間で証明書を共有します。

   **タスク 5 で説明されている方法を使用することが重要です。 クイック・スタート・ガイド (Windows または UNIX) 内の 証明書の共有** 。

3. 次の構成を使用して keystore.conf を作成します: Keystore.conf location: /home/userID/ssl/ams1/

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

4. キュー・マネージャー AMSQMGR1 を作成して開始します。
5. *port* 14567 および *control* QMGR でリスナーを定義します。
6. チャネル権限を無効にするか、チャネル権限のルールを設定します。

   詳しくは、SET CHLAUTH を参照してください。

7. キュー・マネージャーを停止させます。
8. 鍵ストアを設定します。

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. 同じシェルでキュー・マネージャーを開始します。
10. セキュリティー・ポリシーを設定して検証します。

```
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
```

    詳しくは、setmqspl および dspmqspl を参照してください。

11. チャネル構成を設定します。

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. MCA インターセプターを自動的に有効にしない MQ クライアント ( IBM WebSphere MQ Version 7.1 以前のクライアントなど) から **amqsputc** を実行します。 以下の 2 つのメッセージを書き込みます。

```
/opt/mqm/samp/bin/amqsputc TESTQ TESTQMGR
```

13. セキュリティー・ポリシーを除去し、その結果を検証します。

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

14. IBM WebSphere MQ Version 7.5 のインストール環境からキューを参照します。

```
/opt/mq75/samp/bin/amqsbcg TESTQ AMSQMGR1
```

その参照出力には、暗号化された形式のメッセージが表示されます。

15. セキュリティー・ポリシーを設定し、その結果を検証します。

```
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 インストール済み環境から **amqsgetc** を実行します。

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

**関連タスク**

286 ページの『Java クライアントのクイック・スタート・ガイド』
このガイドを使用して、クライアント・バインディングを使用して接続する Java アプリケーションのメッセージ・セキュリティーを提供するように IBM Advanced Message Security を素早く構成します。 このガイドを完了することにより、ユーザー ID を検証するための鍵ストアが作成され、キュー・マネージャーの署名/暗号化ポリシーが定義されます。

**関連資料**

275 ページの『既知の制限事項』
IBM WebSphere MQ Advanced Message Security の制限について確認します。

## クライアントでの IBM WebSphere MQ Advanced Message Security の無効化

Version 7.5 以降のクライアントを使用して旧バージョンの製品からキュー・マネージャーに接続していて、2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告される場合は、クライアント側で IBM WebSphere MQ Advanced Message Security (AMS) を無効にする必要があります。

### このタスクについて

Version 7.5 以降、 IBM WebSphere MQ Advanced Message Security (AMS) は IBM WebSphere MQ クライアントで自動的に有効になるため、デフォルトでは、クライアントはキュー・マネージャーでオブジェクトのセキュリティー・ポリシーを検査しようとします。 ただし、以前のバージョンの製品のサーバー (Version 7.1 など) では、 AMS が有効になっていないため、 2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告されます。

このエラーが報告された場合は、旧バージョンの製品からキュー・マネージャーに接続しようとするときに、以下のようにしてクライアントで AMS を無効にすることができます。

- Java クライアントの場合は、以下のいずれかの方法で行います。
  - V7.5.0.4 環境変数 AMQ_DISABLE_CLIENT_AMS を設定します。
  - V7.5.0.4 Java システム・プロパティー com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS を設定します。
  - V7.5.0.5 `mqclient.ini` ファイルの **Security** スタンザの下で DisableClientAMS プロパティーを使用します。
- C クライアントの場合は、以下のいずれかの方法で行います。
  - V7.5.0.4 環境変数 AMQ_DISABLE_CLIENT_AMS を設定します。

– V7.5.0.5 mqclient.ini ファイルの **Security** スタンザの下で DisableClientAMS プロパティーを使用します。

## 手順

- クライアントで AMS を無効にするには、以下のいずれかのオプションを使用します。

  **V7.5.0.4 AMQ_DISABLE_CLIENT_AMS 環境変数**
  以下のケースでは、この変数を設定する必要があります。

  – IBM Java ランタイム環境 (JRE) 以外の Java ランタイム環境 (JRE) を使用している場合
  – Version 7.5 以降、IBM WebSphere MQ classes for Java または IBM WebSphere MQ classes for JMS クライアントを使用している場合。

  AMQ_DISABLE_CLIENT_AMS を使用して、C クライアントの AMS 機能を無効にすることもできます。

  AMQ_DISABLE_CLIENT_AMS 環境変数を作成し、アプリケーションが実行されている環境で、この環境変数を TRUE に設定します。 以下に例を示します。

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  **V7.5.0.4 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS システム・プロパティー**
  IBM WebSphere MQ classes for JMS および IBM WebSphere MQ classes for Java クライアントの場合、 Java アプリケーションの Java システム・プロパティー com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS の値を TRUE に設定します。

  例えば、Java コマンドを呼び出すときに、Java システム・プロパティーを -D オプションとして設定できます。

  ```
  java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/java/lib/
  com.ibm.mqjms.jar  my.java.applicationClass
  ```

  また、アプリケーションがこのファイルを使用する場合、Java システムプロパティを JMS 構成ファイルの jms.config 内で指定することもできます。

  **V7.5.0.5 mqclient.ini ファイル内の 「DisableClientAMS」 プロパティー**
  IBM WebSphere MQ classes for JMS および IBM WebSphere MQ classes for Java クライアントの場合、および C クライアントの場合は、以下の例に示すように、 **Security** スタンザの下にプロパティー名 DisableClientAMS を mqclient.ini ファイルの下に追加します。

  ```
  Security:
  DisableClientAMS=Yes
  ```

  以下の例に示すように、AMS を有効化することもできます。

  ```
  Security:
  DisableClientAMS=No
  ```

## 次のタスク

AMS に保護されたキューを開く際の問題について詳しくは、325 ページの『JMS 使用時の、保護されたキューを開く際の問題』を参照してください。

**関連概念**
299 ページの『メッセージ・チャネル・エージェント (MCA) インターセプト』
MCA インターセプトにより、 IBM WebSphere MQ の下で実行されるキュー・マネージャーは、サーバー接続チャネルに適用するポリシーを選択的に使用可能にすることができます。

**関連タスク**
構成ファイルを使用したクライアントの構成

**関連資料**
IBM WebSphere MQ classes for JMS 構成ファイル

## AMS の証明書の要件

証明書を Advanced Message Security で使用するには RSA 公開鍵が必要です。

さまざまな公開鍵のタイプの詳細とその作成方法については、34 ページの『IBM WebSphere MQ におけるデジタル証明書と CipherSpec の互換性』を参照してください。

### 鍵用途拡張

鍵用途拡張を使用すると、証明書の使用方法がさらに制限されます。

Advanced Message Security では、鍵用途は次のように設定する必要があります。つまり、保護品質の整合性を保つために使用される X.509 V3 以降の規格の証明書では、鍵用途拡張を設定する場合、少なくとも以下のいずれか一方が鍵用途拡張に含まれている必要があります。

- **nonRepudiation**
- **digitalSignature**

保護品質プライバシーの場合、鍵用途拡張を設定する場合は、**keyEncipherment** 拡張も鍵用途拡張に含まれている必要があります。

**関連概念**
313 ページの『保護品質』
Advanced Message Security データ保護ポリシーは、保護品質 (QOP) を意味します。

## IBM WebSphere MQ Advanced Message Security での証明書の検証

セキュリティー規格を満たしていない証明書を使用してキュー上のメッセージが保護されることがないように、IBM WebSphere MQ Advanced Message Security を使用して、失効した証明書を検出および拒否することができます。

IBM WebSphere MQ AMS では、Online Certificate Status Protocol (OCSP) か証明書取り消しリスト (CRL) を使用して、証明書の有効期間を検証することができます。

IBM WebSphere MQ AMS は、OCSP 検査または CRL 検査 (またはその両方) 用に構成することができます。両方の方式を有効にする場合、パフォーマンス上の理由で、IBM WebSphere MQ AMS はまず OCSP を使用して失効状況を確認します。 OCSP 検査の実行後も証明書の失効状況を判別できない場合、IBM WebSphere MQ AMS は CRL 検査を使用します。

**関連概念**
303 ページの『Online Certificate Status Protocol (OCSP)』
Online Certificate Status Protocol (OCSP) は、証明書が失効しているかどうかを判別するため、証明書を信頼できるかどうかを判別するのに役立ちます。

305 ページの『証明書取り消しリスト (CRL)』
CRL には、秘密鍵が失われたり暗号漏えいしたりしているなどのさまざまな理由で、信頼できなくなったとして、認証局 (CA) によりマークが付けられた証明書のリストが保持されます。

### *Online Certificate Status Protocol (OCSP)*

Online Certificate Status Protocol (OCSP) は、証明書が失効しているかどうかを判別するため、証明書を信頼できるかどうかを判別するのに役立ちます。

ネイティブ・インターセプターでの *OCSP* 検査の有効化
Advanced Message Security での Online Certificate Status Protocol (OCSP) 検査を有効にするには、鍵ストア構成ファイルを変更する必要があります。

## 手順

鍵ストア構成ファイルに、以下のオプションを追加します。

**注 :** 表に示されている個々のオプションの値はデフォルトです。

以下のいずれかの値を指定してください。

- ocsp.enable=on
- ocsp.url=<*reposnder_URL*>
- ocsp.http.proxy.host=<*OCSP_proxy*>

| オプション | 説明 |
|---|---|
| ocsp.enable=off | 検査する証明書に認証局情報アクセス拡張がある場合、OCSP 応答側が位置する URI が含まれている PKIX_AD_OCSP アクセス方式で OCSP 検査を有効にします。<br>指定可能な値: on/off。 |
| ocsp.url=<*reposnder_URL*> | OCSP 応答側の URL アドレス。 |
| ocsp.http.proxy.host=<*OCSP_proxy*> | OCSP プロキシー・サーバーの URL アドレス。 |
| ocsp.http.proxy.port=<*port_number*> | OCSP プロキシー・サーバーのポート番号。 |
| ocsp.nonce.generation=on/off | OCSP の照会時に nonce を生成します。<br>デフォルト値は off です。 |
| ocsp.nonce.check=on/off | OCSP からの応答の受信後に nonce を検査します。<br>デフォルト値は off です。 |
| ocsp.nonce.size=8 | nonce のサイズ (バイト)。 |
| ocsp.http.get=on/off | 要求方式として HTTP GET を指定します。 このオプションを off に設定すると、HTTP POST が使用されます。 |
| ocsp.max_response_size=20480 | OCSP 応答側からの応答の最大サイズ (バイト単位で指定)。 |
| ocsp.cache_size=100 | 内部 OCSP 応答キャッシングを有効にし、キャッシュ項目の数に限度を設定します。 |
| ocsp.timeout=30 | サーバー応答の待ち時間 (秒)。これを超えると、Advanced Message Security がタイムアウトになります。 |

*Java での OCSP 検査の有効化*
Advanced Message Security で Java の OCSP チェックインを有効にするには、java.security ファイルまたは鍵ストア構成ファイルを変更します。

## このタスクについて

Advanced Message Security で OCSP 検査を有効にする方法には、以下の 2 つの方法があります。

*java.security を使用する*
証明書で認証局情報アクセス (AIA) がセットアップされているかどうかを確認します。

## 手順

1. AIA がセットアップされていない場合、または証明書をオーバーライドする場合は、以下のプロパティーを指定して $JAVA_HOME/lib/security/java.security ファイルを編集します。

```
ocsp.responderURL=http://url.to.responder:port
ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
```

次に、以下の行を指定して $JAVA_HOME/lib/security/java.security ファイルを編集し、OCSP 検査を有効にします。

```
ocsp.enable=true
```

2. AIA がセットアップされている場合は、以下の行を指定して $JAVA_HOME/lib/security/java.security ファイルを編集し、OCSP 検査を有効にします。

```
ocsp.enable=true
```

## 次のタスク

Java Security Manager を使用している場合は、構成を完了しすぎて、以下の Java アクセス権を lib/security/java.policy に追加します。

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

*keystore.conf* を使用する

## 手順

構成ファイルに以下の属性を追加します。

```
ocsp.enable=true
```

**重要 :** この属性を構成ファイルに設定すると、java.security 設定がオーバーライドされます。

## 次のタスク

構成を完了するには、以下の Java アクセス権を lib/security/java.policy に追加します。

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

## 証明書取り消しリスト *(CRL)*

CRL には、秘密鍵が失われたり暗号漏えいしたりしているなどのさまざまな理由で、信頼できなくなったとして、認証局 (CA) によりマークが付けられた証明書のリストが保持されます。

証明書を検証するために、 Advanced Message Security は証明書チェーンを構成します。これは、トラスト・アンカーに至るまでの署名者の証明書と認証局 (CA) の証明書のチェーンで構成されています。 トラスト・アンカーとは、証明書の信頼性の表明に使用する信頼証明書やトラステッド・ルート証明書が入ったトラステッド鍵ストア・ファイルのことです。IBM WebSphere MQ AMS PKIX 検証アルゴリズムを使用して証明書パスを検証します。 チェーンが作成され検証されると、IBM WebSphere MQ AMS は、証明書の妥当性検査をすべて実行します。これには、チェーン内の各証明書の発行日付と有効期限日付を現在の日付に照らして確認することや、鍵用途拡張がエンド・エンティティー証明書に存在するかどうか検査することが含まれます。 この拡張を証明書に追加すると、IBM WebSphere MQ AMS は、**digitalSignature** または **nonRepudiation** も設定されているかどうかを検証します。 設定されていない場合は、MQRC_SECURITY_ERROR が報告されてログに記録されます。 次に、IBM WebSphere MQ AMS は、構成ファイルに指定されている値に基づいて、CRL をファイルまたは LDAP からダウンロードします。 DER 形式でエンコードされている CRL のみが、IBM WebSphere MQ AMS でサポートされています。 鍵ストア構成ファイル内で CRL 関連の構成が見つからない場合、IBM WebSphere MQ AMS は CRL 妥当性検査を実行しません。 CA 証明書ごとに、IBM WebSphere MQ AMS は、CRL を検索するための CA の識別名を使用して、LDAP の中で CRL を照会します。 LDAP 照会には、以下の属性が組み込まれます。

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
```

```
deltaRevocationList
deltaRevocationList;binary,
```

**注 :** deltaRevocationList は、配布ポイントとして指定されている場合にのみサポートされます。

ネイティブ・インターセプターでの証明書の検証および証明書取り消しリスト・サポートの有効化
鍵ストア構成ファイルを変更して、 Advanced Message Security が Lightweight Directory Access Protocol (LDAP) サーバーから CLR をダウンロードできるようにする必要があります。

## 手順

構成ファイルに、以下のオプションを追加します。

**注 :** 表に示されている個々のオプションの値はデフォルトです。

| オプション | 説明 |
|---|---|
| crl.ldap.host=<*host_name*> | LDAP サーバーのホスト名。 |
| crl.ldap.port=<*port_number*> | LDAP サーバーのポート番号。<br>最大 11 台のサーバーを指定できます。 LDAP 接続が失敗した場合は、トランスペアレントなフェイルオーバーを実現するために複数の LDAP ホストが使用されます。 すべての LDAP サーバーはレプリカであり、同じデータを含んでいることが期待されています。 IBM WebSphere MQ AMS Java インターセプターは、LDAP サーバーに正常に接続すると、提供されている残りのサーバーから CRL をダウンロードしようとしません。 |
| crl.cdp=off | このオプションは、証明書内の CRLDistributionPoints 拡張を確認または使用する場合に使用します。 |
| crl.ldap.version=3 | LDAP プロトコルのバージョン番号。 指定可能な値は、2 または 3 です。 |
| crl.ldap.user=cn=<*username*> | LDAP サーバーにログインします。 この値を指定しない場合、LDAP 内の CRL 属性は world-readable でなければなりません。 |
| crl.ldap.pass=<*password*> | LDAP サーバーのパスワード。 |
| crl.ldap.cache_lifetime=0 | LDAP キャッシュの存続期間 (秒)。 指定可能な値は、0-86400 です。 |
| crl.ldap.cache_size=50 | LDAP キャッシュ・サイズ。 このオプションは、crl.ldap.cache_lifetime 値が 0 より大きい場合にのみ指定できます。 |
| crl.http.proxy.host=some.host.com | CDP CRL 検索用の Http プロキシー・サーバー・ポート。 |
| crl.http.proxy.port=8080 | HTTP プロキシー・サーバーのポート番号。 |
| crl.http.max_response_size=204800 | GSKit によって受け入れられている HTTP サーバーから取り出せる CRL の最大サイズ (バイト)。 |
| crl.http.timeout=30 | サーバー応答の待ち時間 (秒)。これを超えると、IBM WebSphere MQ AMS がタイムアウトになります。 |
| crl.http.cache_size=0 | HTTP キャッシュ・サイズ (バイト)。 |

*Java* での証明書失効リスト・サポートの有効化
Advanced Message Security で CRL サポートを有効にするには、鍵ストア構成ファイルを変更して、IBM WebSphere MQ AMS が Lightweight Directory Access Protocol (LDAP) サーバーから CRL をダウンロードできるようにし、java.security ファイルを構成します。

## 手順

1. 構成ファイルに、以下のオプションを追加します。

| ヘッダー | 説明 |
| --- | --- |
| crl.ldap.host=<*host_name*> | LDAP ホスト名。 |
| crl.ldap.port=<*port_number*> | LDAP サーバーのポート番号。<br>最大 11 台のサーバーを指定できます。 LDAP 接続が失敗した場合は、トランスペアレントなフェイルオーバーを実現するために複数の LDAP ホストが使用されます。 すべての LDAP サーバーはレプリカであり、同じデータを含んでいることが期待されています。 IBM WebSphere MQ AMS Java インターセプターは、LDAP サーバーに正常に接続すると、提供されている残りのサーバーから CRL をダウンロードしようとしません。<br>Java は crl.ldap.user および crl.ldaworldp.pass 値を使用しません。 LDAP サーバーへの接続時に、ユーザーおよびパスワードを使用しません。 したがって、LDAP 内の CRL 属性は world-readable でなければなりません。 |
| crl.cdp=on/off | このオプションは、証明書内の CRLDistributionPoints 拡張を確認または使用する場合に使用します。 |

2. 以下のプロパティーを使用して、JRE/lib/security/java.security ファイルを変更します。

| プロパティー名 | 説明 |
| --- | --- |
| com.ibm.security.enableCRLDP | このプロパティーには、true、false の値を指定できます。<br>true に設定する場合、証明書の失効検査を実行すると、証明書の CRL 配布ポイント拡張の URL を使用して CRL がロードされます。<br>このプロパティーを設定しない場合、または false に設定する場合、CRL 配布ポイント拡張を使用した CRL の検査は無効になります。 |
| ibm.security.certpath.ldap.cache.lifetime | このプロパティーは、LDAP CertStore のメモリー・キャッシュ内の項目の存続期間を秒単位の値に設定するために使用できます。 0 の値はキャッシュを無効にし、-1 の値は無制限の存続期間を意味します。 設定しない場合、デフォルトの存続期間は 30 です。 |

| プロパティー名 | 説明 |
| --- | --- |
| com.ibm.security.enableAIAEXT | このプロパティーには、true、false の値を指定できます。<br>true に設定する場合、構築中の証明書パスの証明書内で見つかった認証局情報アクセス拡張が調べられ、LDAP URI が含まれているかどうか判別されます。 見つかった LDAP URI ごとに、LDAPCertStore オブジェクトが作成され、証明書パスの構築に必要な他の証明書を探すために使用する CertStore のコレクションに追加されます。<br>このプロパティーを設定しない場合、または false に設定する場合、追加の LDAPCertStore オブジェクトは作成されません。 |

## Java でのパスワードの保護

鍵ストアと秘密鍵のパスワードを平文で格納するとセキュリティー・リスクが発生するため、 Advanced Message Security には、ユーザーの鍵 (鍵ストア・ファイル内で取得可能) を使用してこれらのパスワードの順序を変えることができるツールが用意されています。

### 始める前に

keystore.conf ファイル所有者は、ファイル所有者のみがファイルの読み取り資格を持つようにする必要があります。 この章で説明するパスワード保護は、補足的な保護手段に過ぎません。

### 手順

1. keystore.conf ファイルを編集して、鍵ストアおよびユーザー・ラベルへのパスを組み込みます。

```
jceks.keystore = c:/Documents and Setting/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.provider = IBMJCE
```

2. このツールを実行するには、次のコマンドを発行します。

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password
```

暗号化パスワードを含む出力が生成され、keystore.conf ファイルにコピーできるようになります。

出力を keystore.conf ファイルに自動的にコピーするには、以下を実行します。

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password >> ~/<path_to_keystore>/keystore.conf
```

**注:**

各種プラットフォームにおける keystore.conf のデフォルトの場所については、297 ページの『鍵ストアおよび証明書の使用』を参照してください。

#### 例

以下に、出力例を示します。

```
#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 セキュリティー・ポリシーの管理

IBM WebSphere MQ Advanced Message Security は、セキュリティー・ポリシーを使用して、キュー間を流れるメッセージを暗号化して認証するための暗号アルゴリズムと署名アルゴリズムを指定します。

## セキュリティー・ポリシーの概要

IBM Advanced Message Security セキュリティー・ポリシーは、メッセージが暗号化および署名される方法を記述する概念的なオブジェクトです。

セキュリティー・ポリシーの属性の詳細については、以下のサブトピックを参照してください。

**関連概念**

313 ページの『保護品質』
Advanced Message Security データ保護ポリシーは、保護品質 (QOP) を意味します。

312 ページの『セキュリティー・ポリシー属性』
Advanced Message Security を使用して、データを保護するための特定のアルゴリズムまたはメソッドを選択できます。

### ポリシー名

ポリシー名は、特定の Advanced Message Security ポリシーとそれが適用されるキューを識別する固有名です。

ポリシー名は、適用されるキュー名と同じでなければなりません。 Advanced Message Security (IBM WebSphere MQ AMS) ポリシーとキューの間には 1 対 1 のマッピングがあります。

キューと同じ名前のポリシーを作成することにより、そのキューに対してポリシーをアクティブ化します。一致するポリシー名がないキューは、IBM WebSphere MQ AMS によって保護されません。

ポリシーの有効範囲は、ローカル・キュー・マネージャーとそのキューに関係します。 リモート・キュー・マネージャーには、管理するキューに対する独自のローカル定義ポリシーが必要です。

### 署名アルゴリズム

署名アルゴリズムとは、データ・メッセージに署名するときに使用されることになっているアルゴリズムです。

有効な値は、以下のとおりです。

- MD5
- SHA-1
- SHA-2 ファミリー:
  - SHA256
  - SHA384 (許容される鍵の最小長 - 768 ビット)
  - SHA512 (許容される鍵の最小長 - 768 ビット)

署名アルゴリズムを指定しない、またはアルゴリズムに NONE を指定するポリシーは、ポリシーに関連付けられたキューに配置されるメッセージには署名されないことを暗黙に示します。

**注 :** メッセージの PUT および GET 関数に使用される保護品質は、同じでなければなりません。 キューとキュー内のメッセージとの間でポリシー保護品質の不一致がある場合、メッセージは受け入れられず、エラー処理キューに送信されます。 このルールは、ローカル・キューとリモート・キューの両方に適用されます。

## 暗号化アルゴリズム

暗号化アルゴリズムとは、ポリシーに関連付けられたキューに配置されるデータ・メッセージを暗号化するときに使用されることになっているアルゴリズムです。

有効な値は、以下のとおりです。

- RC2
- DES
- 3DES
- AES128
- AES256

暗号化アルゴリズムを指定しない、つまりアルゴリズムに NONE を指定するポリシーは、ポリシーに関連付けられたキューに配置されるメッセージを暗号化しないことを暗黙に示します。

Advanced Message Security 暗号化メッセージには署名も行われるため、NONE 以外の暗号化アルゴリズムを指定するポリシーは、少なくとも 1 人の受信者 DN と署名アルゴリズムも指定する必要があります。

**重要 :** メッセージの PUT および GET 関数に使用される保護品質は、同じでなければなりません。 キューとキュー内のメッセージとの間でポリシー保護品質の不一致がある場合、メッセージは受け入れられず、エラー処理キューに送信されます。 このルールは、ローカル・キューとリモート・キューの両方に適用されます。

## 容認

容認属性は、IBM Advanced Message Security が、セキュリティー・ポリシーの指定されていないメッセージを受け入れるかどうかを指定します。

メッセージを暗号化するポリシーが設定されているキューからメッセージを取得する場合、メッセージが暗号化されていなければ、そのメッセージは呼び出し側のアプリケーションに返されます。 有効な値は、以下のとおりです。

**0**
使用しません (**デフォルト**)。

**1**
はい。

容認値を指定しないか、または 0 を指定するポリシーは、ポリシーに関連付けられたキューに配置されるメッセージがポリシー・ルールに一致する必要があることを意味します。

容認はオプションであり、ポリシーが適用されているキューに、セキュリティー・ポリシーの指定されていないメッセージが既に含まれている場合の構成ロールアウトを容易にするものです。

## 送信側の識別名

送信側の識別名 (DN) は、メッセージをキューに置く権限が付与されているユーザーを識別します。

IBM Advanced Message Security (IBM WebSphere MQ AMS) は、メッセージが取り出されるまで、有効なユーザーによってメッセージがデータ保護キューに置かれているかどうかを検査しません。 その時点で、1 つ以上の有効な送信側がポリシーに明記されている場合、そのメッセージをキューに入れたユーザーが有効な送信側のリストに含まれていなければ、IBM WebSphere MQ AMS はメッセージ取得側のアプリケーションにエラーを返し、そのメッセージをエラー・キューに入れます。

ポリシーには、ゼロ以上の送信側 DN を指定することができます。 ポリシーに送信側 DN が指定されていない場合、ユーザーの証明書が信頼できるものであれば、すべてのユーザーがデータ保護されたメッセージをキューに登録することができます。

送信側の識別名の形式は、次のようになります。

```
CN=Common Name,O=Organization,C=Country
```

**重要 :**

- すべての DN は大文字でなければなりません。 DN 内のすべてのコンポーネント名 ID は、以下の表に示す順序で指定する必要があります。

| コンポーネント名 | 値 |
|---|---|
| CN | この DN の対象の共通名 (フルネーム、またはデバイスの目的など)。 |
| OU | DN の対象が関連付けられている組織内の単位 (会社の部門や製品名など)。 |
| O | DN の対象が関連付けられている組織 (会社など)。 |
| L | DN の対象が置かれている場所 (都市や自治体など)。 |
| ST | DN の対象が置かれている都道府県の名前。 |
| C | 識別名 (DN) の対象が置かれている国。 |

- ポリシーに 1 つ以上の送信側 DN が指定されている場合、それらのユーザーのみが、ポリシーに関連付けられたキューにメッセージを登録することができます。
- 送信側 DN を指定する場合、その DN は、メッセージを登録するユーザーに関連付けられたデジタル証明書に含まれている DN と正確に一致する必要があります。
- IBM WebSphere MQ AMS は、Latin 1 文字セットのみを使用した値を持つ DN をサポートしています。 この文字セットを使用して DN を作成するには、UTF-8 コーディングがオンになっている UNIX プラットフォームか、iKeyman ユーティリティーを使用して、UTF-8 コーディングで作成された DN を持つ証明書を作成する必要があります。 次に、UTF-8 コーディングがオンになっている UNIX プラットフォームから、または WebSphere MQ に対する IBM WebSphere MQ AMS プラグインを使用して、ポリシーを作成する必要があります。

**関連概念**

311 ページの『受信側の識別名』

受信者識別名 (DN) は、キューからメッセージを取り出す権限が付与されているユーザーを識別します。

## 受信側の識別名

受信者識別名 (DN) は、キューからメッセージを取り出す権限が付与されているユーザーを識別します。

ポリシーには、ゼロ個以上の受信者 DN を指定できます。 受信者の識別名の形式は、次のようになります。

```
CN=Common Name,O=Organization,C=Country
```

**重要 :**

- すべての DN は大文字でなければなりません。 DN 内のすべてのコンポーネント名 ID は、以下の表に示す順序で指定する必要があります。

| コンポーネント名 | 値 |
|---|---|
| CN | この DN の対象の共通名 (フルネーム、またはデバイスの目的など)。 |
| OU | DN の対象が関連付けられている組織内の単位 (会社の部門や製品名など)。 |
| O | DN の対象が関連付けられている組織 (会社など)。 |
| L | DN の対象が置かれている場所 (都市や自治体など)。 |
| ST | DN の対象が置かれている都道府県の名前。 |
| C | 識別名 (DN) の対象が置かれている国。 |

- ポリシーに受信側 DN が指定されていない場合、ポリシーに関連付けられたキューから、すべてのユーザーがメッセージを取り出すことができます。
- ポリシーに 1 つ以上の受信側 DN が指定されている場合、それらのユーザーのみが、ポリシーに関連付けられたキューからメッセージを取得できます。
- 受信側 DN を指定する場合、その DN は、メッセージを取得するユーザーに関連付けられたデジタル証明書に含まれている DN と正確に一致する必要があります。
- Advanced Message Security は、Latin 1 文字セットのみを使用した値を持つ DN をサポートしています。この文字セットを使用して DN を作成するには、UTF-8 コーディングがオンになっている UNIX プラットフォームか、iKeyman ユーティリティーを使用して、UTF-8 コーディングで作成された DN を持つ証明書を作成する必要があります。 次に、UTF-8 コーディングがオンになっている UNIX プラットフォームから、または WebSphere MQ に対する Advanced Message Security プラグインを使用して、ポリシーを作成する必要があります。

**関連概念**

310 ページの『送信側の識別名』
送信側の識別名 (DN) は、メッセージをキューに置く権限が付与されているユーザーを識別します。

## セキュリティー・ポリシー属性

Advanced Message Security を使用して、データを保護するための特定のアルゴリズムまたはメソッドを選択できます。

セキュリティー・ポリシーは、メッセージが暗号化および署名される方法を記述した概念的なオブジェクトです。 以下の表は、 Advanced Message Security でのセキュリティー・ポリシー属性を示しています。

| 属性 | 説明 |
|---|---|
| ポリシー名 | キュー・マネージャーのポリシーの固有の名前。 |
| 署名アルゴリズム | 送信前にメッセージに署名を行うために使用される暗号アルゴリズム。 |
| 暗号化アルゴリズム | 送信前にメッセージを暗号化するために使用される暗号アルゴリズム。 |
| 宛先リスト | メッセージの潜在的な受信者の証明書識別名 (DN) のリスト。 |
| 署名 DN チェックリスト | メッセージの取得中に検証する署名 DN のリスト。 |

Advanced Message Security では、メッセージは対称鍵で暗号化され、対称鍵は受信者の公開鍵で暗号化されます。 公開鍵は RSA アルゴリズムで暗号化され、鍵の有効長は最大 2048 ビットです。 実際の非対称鍵暗号化は、証明書の鍵の長さに依存しています。

サポートされる対称鍵アルゴリズムは、以下のとおりです。

- RC2
- DES
- 3DES
- AES128
- AES256

Advanced Message Security は、以下の暗号ハッシュ関数もサポートしています。

- MD5
- SHA-1
- SHA-2 ファミリー:
  - SHA256
  - SHA384 (許容される鍵の最小長 - 768 ビット)

– SHA512 (許容される鍵の最小長 - 768 ビット)

**注 :** メッセージの PUT および GET 関数に使用される保護品質は、同じでなければなりません。 キューとキュー内のメッセージとの間でポリシー保護品質の不一致がある場合、メッセージは受け入れられず、エラー処理キューに送信されます。 このルールは、ローカル・キューとリモート・キューの両方に適用されます。

### 保護品質

Advanced Message Security データ保護ポリシーは、保護品質 (QOP) を意味します。

Advanced Message Security での、メッセージに対する署名および暗号化に使用される暗号アルゴリズムに基づく 3 つの保護品質レベルは、以下のとおりです。

- プライバシー - キューに入れられるメッセージは、署名および暗号化される必要があります。
- 整合性 - キューに入れられるメッセージは、送信者によって署名される必要があります。
- なし - データ保護は適用されません。

メッセージがキューに入れられるときに署名される必要があると規定しているポリシーの QOP は「整合性」です。 「整合性」の QOP は、ポリシーが署名アルゴリズムを規定しているが、暗号化アルゴリズムを規定していないことを意味します。 整合性が保護されたメッセージは、「署名済み」とも呼ばれます。

メッセージがキューに入れられるときに署名および暗号化される必要があると規定しているポリシーの QOP は「プライバシー」です。 「プライバシー」の QOP は、ポリシーが署名アルゴリズムと暗号化アルゴリズムを規定しているということを意味します。 プライバシーが保護されたメッセージは、「シール済み」とも呼ばれます。

署名アルゴリズムまたは暗号化アルゴリズムを規定していないポリシーの QOP は「なし」です。 Advanced Message Security QOP が NONE のポリシーを持つキューにデータ保護を提供しません。

## セキュリティー・ポリシーの管理

セキュリティー・ポリシーは、メッセージが暗号化および署名される方法を記述した概念的なオブジェクトです。

セキュリティー・ポリシーに関連したすべての管理用タスクは、以下の場所から実行されます。

- UNIX プラットフォームの場合: `<MQInstallRoot>/bin`
- Windows プラットフォームの場合: PATH 環境変数はインストール時に更新されるため、管理タスクはどの場所からでも実行できます。

**関連タスク**

313 ページの『セキュリティー・ポリシーの作成』
セキュリティー・ポリシーは、メッセージの書き込み時にメッセージが保護される方法、またはメッセージの受信時にメッセージがどのように保護されている必要があるかを定義します。

314 ページの『セキュリティー・ポリシーの変更』
Advanced Message Security を使用して、既に定義済みのセキュリティー・ポリシーの詳細を変更できます。

315 ページの『セキュリティー・ポリシーの表示およびダンプ』
`dspmqspl` コマンドを使用して、提供したコマンド・ライン・パラメーターに基づいて、すべてのセキュリティー・ポリシーのリスト、または指定したポリシーの詳細を表示します。

316 ページの『セキュリティー・ポリシーの削除』
Advanced Message Security でセキュリティー・ポリシーを削除するには、`setmqspl` コマンドを使用する必要があります。

### セキュリティー・ポリシーの作成

セキュリティー・ポリシーは、メッセージの書き込み時にメッセージが保護される方法、またはメッセージの受信時にメッセージがどのように保護されている必要があるかを定義します。

## 始める前に

セキュリティー・ポリシーを作成する場合に満たす必要がある入り口条件がいくつかあります。

- キュー・マネージャーが実行中でなければなりません。
- セキュリティー・ポリシーの名前は、WebSphere MQ オブジェクトの命名規則に従う必要があります。
- セキュリティー・ポリシーを作成するために必要な +connect +inq +chg 権限を持っている必要があります。 許可変更コマンドの完全な構文については、「**setmqaut**」を参照してください。
- WebSphere MQ キューおよびキュー・マネージャーを操作するために必要な許可があることを確認します。 詳しくは、318 ページの『OAM 許可の付与』を参照してください。

### 例

キュー・マネージャー QMGR 上にポリシーを作成する方法の例を示します。 このポリシーは、DN が CN=joe、O=IBM、C=US および CN=jane、O=IBM、C=US である証明書について、メッセージが SHA1 アルゴリズムを使用して署名され、AES256 アルゴリズムを使用して暗号化されることを指定しています。 このポリシーは MY.QUEUE に付加されます。

```
$ setmqspl -m QMGR -p MY.QUEUE -s SHA1 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

キュー・マネージャー QMGR 上にポリシーを作成する方法の例を示します。 このポリシーは、DN が CN=john、O=IBM、C=US および CN=jeff、O=IBM、C=US である証明書について、メッセージが DES アルゴリズムを使用して暗号化され、DN が CN=phil、O=IBM、C=US である証明書について、メッセージが MD5 アルゴリズムを使用して署名されることを指定しています。

```
$ 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
```

**注 :**

- メッセージの PUT および GET に使用されている保護品質は、同じでなければなりません。 メッセージに定義されているポリシー保護品質が、キューに定義されているポリシーより弱い場合、メッセージがエラー処理キューに送信されます。 このポリシーは、ローカル・キューとリモート・キューの両方で有効です。

### 関連資料

setmqspl コマンド属性の完全なリスト

## セキュリティー・ポリシーの変更

Advanced Message Security を使用して、既に定義済みのセキュリティー・ポリシーの詳細を変更できます。

## 始める前に

- 操作を行うキュー・マネージャーが実行されている必要があります。
- セキュリティー・ポリシーを作成するには、必要な +connect +inq +chg 権限を持っている必要があります。 許可変更コマンドの完全な構文については、「**setmqaut**」を参照してください。

## このタスクについて

セキュリティー・ポリシーを変更するには、新しい属性を指定した setmqspl コマンドを既に存在しているポリシーに対して適用します。

### 例

QMGR という名前のキュー・マネージャー上に MYQUEUE という名前のポリシーを作成する方法の例を示します。ここでは、DN が CN=bob、O=IBM、C=US である証明書について、RC2 アルゴリズムを使用してメ

ッセージが暗号化され、DN が CN=jeff、O=IBM、C=US である証明書について、SHA1 アルゴリズムを使用してメッセージが署名されることを指定しています。

```
setmqspl -m QMGR -p MYQUEUE -e RC2 -s SHA1 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

このポリシーを変更するには、例で示すすべての属性を使用して、変更対象の値のみを変更して setmqspl コマンドを発行します。 この例では、以前に作成したポリシーが新しいキューに付加され、その暗号化アルゴリズムが AES256 に変更されます。

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA1 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

**関連資料**
setmqspl

## セキュリティー・ポリシーの表示およびダンプ

dspmqspl コマンドを使用して、提供したコマンド・ライン・パラメーターに基づいて、すべてのセキュリティー・ポリシーのリスト、または指定したポリシーの詳細を表示します。

## 始める前に

- セキュリティー・ポリシーの詳細を表示するには、キュー・マネージャーが存在していて、実行されている必要があります。
- セキュリティー・ポリシーを表示およびダンプするには、必要な +connect +inq +dsp 権限がキュー・マネージャーに適用されている必要があります。 許可変更コマンドの完全な構文については、「**setmqaut**」を参照してください。

## このタスクについて

以下に、dspmqspl コマンド・フラグのリストを示します。

表 *27. dspmqspl* コマンド・フラグ

| **コマンド・フラグ** | **説明** |
|---|---|
| -m | キュー・マネージャー名 (**必須**)。 |
| -p | ポリシー名。 |
| -export | このフラグを追加すると、別のキュー・マネージャーに簡単に適用できる出力が生成されます。 |

### 例

次の例では、venus.queue.manager に関する 2 つのセキュリティー・ポリシーを作成します。

```
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
```

次の例は、venus.queue.manager に定義されているすべてのポリシーの詳細と、生成される出力を表示するコマンドを示しています。

```
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
```

次の例は、venus.queue.manager に定義されている特定のセキュリティー・ポリシーの詳細と、生成される出力を表示するコマンドを示しています。

```
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
```

次の例では、まずセキュリティー・ポリシーを作成してから、-export フラグを使用してこのポリシーをエクスポートします。

```
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]
```

セキュリティー・ポリシーをインポートするには、以下のようにします。

- Windows プラットフォームの場合は、policies.bat を実行します。
- UNIX プラットフォームの場合:
  1. mqm WebSphere MQ 管理グループに属するユーザーとしてログオンします。
  2. . policies.sh を実行します。

**関連資料**
dspmqspl コマンド属性の完全なリスト

## セキュリティー・ポリシーの削除
Advanced Message Security でセキュリティー・ポリシーを削除するには、setmqspl コマンドを使用する必要があります。

## 始める前に
セキュリティー・ポリシーを管理する場合に満たされる必要がある入り口条件がいくつかあります。

- キュー・マネージャーが実行中でなければなりません。
- セキュリティー・ポリシーを作成するには、必要な +connect +inq +chg 権限を持っている必要があります。 許可変更コマンドの完全な構文については、「**setmqaut**」を参照してください。

## このタスクについて
-remove オプションを指定した setmqspl コマンドを使用します。

### 例
ポリシーを削除する方法の例を以下に示します。

```
$ setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

**関連資料**
setmqspl コマンド属性の完全なリスト

## システム・キューの保護

システム・キューは、WebSphere MQ と補助アプリケーションとの通信を有効にします。 キュー・マネージャーが作成されるたびに、WebSphere MQ 内部メッセージおよびデータを格納するためのシステム・キューも作成されます。 許可ユーザーのみがシステム・キューにアクセスしたり復号したりすることができるように、 Advanced Message Security を使用してシステム・キューを保護できます。

システム・キューを保護するには、通常のキューを保護するのと同じ方法に従います。 313 ページの『セキュリティー・ポリシーの作成』を参照してください。

Windows プラットフォーム上でシステム・キューを保護するには、以下のディレクトリーに keystore.conf ファイルをコピーします。

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

SYSTEM.ADMIN.COMMAND.QUEUE を保護するには、コマンド・サーバーが keystore および keystore.conf に対するアクセス権限を持っている必要があります。これらのファイルには、コマンド・サーバーが鍵と証明書にアクセスできるようにする鍵と構成が格納されています。 SYSTEM.ADMIN.COMMAND.QUEUE のセキュリティー・ポリシーに対して変更を行うと、コマンド・サーバーを再始動する必要があります。

コマンド・キューとの間で送受信されるすべてのメッセージは、ポリシー設定に従って、署名されるか、署名および暗号化されます。 管理者が許可済み署名者を定義する場合、署名者の識別名 (DN) 検査に合格しないコマンド・メッセージは、コマンド・サーバーによって実行されず、 Advanced Message Security エラー処理キューに経路指定されません。 WebSphere MQ Explorer の一時動的キューに対する応答として送信されるメッセージは、WebSphere MQ AMS によって保護されません。

Advanced Message Security セキュリティー・ポリシーを変更すると、WebSphere MQ コマンド・サーバーを再始動する必要があります。

セキュリティー・ポリシーは、以下の SYSTEM キューに対して影響を与えることはありません。

- 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 許可の付与

ファイル許可により、すべてのユーザーが `setmqspl` コマンドと `dspmqspl` コマンドの実行を許可されます。 ただし、IBM Advanced Message Security はオブジェクト権限マネージャー (OAM) に依存しているため、mqm グループ (WebSphere MQ 管理グループ) に属していないユーザーによるこれらのコマンドの実行試行、または付与されているセキュリティー・ポリシー設定を読み取る許可を持たないユーザーによるこれらのコマンドの実行試行は、すべてエラーとなります。

### 手順

必要な許可をユーザーに付与するには、以下を実行します。

```
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
```

## コマンド・イベントと構成イベント

Advanced Message Security を使用すると、コマンド・イベント・メッセージと構成イベント・メッセージを生成できます。これらは、ログに記録することが可能で、監査用のポリシー変更の記録として役立ちます。

WebSphere MQ によって生成されるコマンド・イベントと構成イベントは、専用キューに送信される PCF 形式のメッセージです。

構成イベント・メッセージは、イベントが発生するキュー・マネージャー上の SYSTEM.ADMIN.CONFIG.EVENT キューに送信されます。

コマンド・イベント・メッセージは、イベントが発生するキュー・マネージャー上の SYSTEM.ADMIN.COMMAND.EVENT キューに送信されます。

イベントは、 Advanced Message Security セキュリティー・ポリシーを管理するために使用しているツールに関係なく生成されます。

Advanced Message Security では、セキュリティー・ポリシーに対する各種アクションごとに、以下の 4 つのタイプのイベントが生成されます。

- 313 ページの『セキュリティー・ポリシーの作成』。以下の 2 つの WebSphere MQ イベント・メッセージを生成します。
  - 構成イベント
  - コマンド・イベント

- 314 ページの『セキュリティー・ポリシーの変更』。以下の 3 つの WebSphere MQ イベント・メッセージを生成します。
  - 古いセキュリティー・ポリシーの値が入っている構成イベント
  - 新しいセキュリティー・ポリシーの値が入っている構成イベント
  - コマンド・イベント
- 315 ページの『セキュリティー・ポリシーの表示およびダンプ』。以下の 1 つの WebSphere MQ イベント・メッセージを生成します。
  - コマンド・イベント
- 316 ページの『セキュリティー・ポリシーの削除』。以下の 2 つの WebSphere MQ イベント・メッセージを生成します。
  - 構成イベント
  - コマンド・イベント

## イベント・ログの有効化および無効化

キュー・マネージャー属性の CONFIGEV および CMDEV を使用して、コマンド・イベントと構成イベントを制御します。 これらのイベントを有効にするには、該当するキュー・マネージャー属性を ENABLED に設定します。 これらのイベントを無効にするには、該当するキュー・マネージャー属性を DISABLED に設定します。

## 手順

**構成イベント**

構成イベントを有効にするには、CONFIGEV を ENABLED に設定します。 構成イベントを無効にするには、CONFIGEV を DISABLED に設定します。 構成イベントを有効にするには、次のような MQSC コマンドを使用します。

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

**コマンド・イベント**

コマンド・イベントを有効にするには、CMDEV を ENABLED に設定します。 DISPLAY MQSC コマンドおよび Inquire PCF コマンドを除くコマンドのコマンド・イベントを 有効にするには、CMDEV を NODISPLAY に設定します。 コマンド・イベントを無効にするには、CMDEV を DISABLED に設定します。 コマンド・イベントを有効にするには、次のような MQSC コマンドを使用します。

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

**関連タスク**

Websphere MQ での構成イベント、コマンド・イベント、およびロガー・イベントの制御

## コマンド・イベント・メッセージ形式

コマンド・イベント・メッセージは、MQCFH 構造と、それに続く PCF パラメーターで構成されます。

選択された MQCFH 値は、以下のとおりです。

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

**注 :** ParameterCount 値は、MQCFGR タイプ (グループ) のパラメーターが常に 2 つあるため、2 になっています。 各グループは、適切なパラメーターで構成されます。 イベント・データは、CommandContext と CommandData の 2 つの グループから成ります。

CommandContext の内容は、以下のとおりです。

***EventUserID***

説明: コマンドを発行したユーザー ID、またはイベントを生成した呼び出し。 (これは、コマンドまたは呼び出しを発行する権限の検査に使用するものと同じユーザー ID です。キューから受け取ったコマンドの場合、これはコマンド・メッセージの MD からのユーザー ID (UserIdentifier) でもあります。)

ID: MQCACF_EVENT_USER_ID

データ型: MQCFST

最大長: MQ_USER_ID_LENGTH

戻り: 常時。

***EventOrigin***

説明: イベントを引き起こしたアクションの発信元。

ID: MQIACF_EVENT_ORIGIN

データ型: MQCFIN

値:

**MQEVO_CONSOLE**
コンソール・コマンド - コマンド・ライン

**MQEVO_MSG**
WebSphere MQ エクスプローラー・プラグインからのコマンド・メッセージ

戻り: 常時。

***EventQMgr***

説明: コマンドまたは呼び出しが入れられたキュー・マネージャー。 (コマンドが実行されたキュー・マネージャー、およびイベントを生成したキュー・マネージャーは、イベント・メッセージの MD にあります。)

ID: MQCACF_EVENT_Q_MGR

データ型: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 常時。

***EventAccountingToken***

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアカウンティング・トークン (AccountingToken)。

ID: MQBACF_EVENT_ACCOUNTING_TOKEN

データ型: MQCFBS

最大長: MQ_ACCOUNTING_TOKEN_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

***EventIdentityData***

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーション識別データ (ApplIdentityData)。

ID: MQCACF_EVENT_APPL_IDENTITY

データ型: MQCFST

最大長: MQ_APPL_IDENTITY_DATA_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

***EventApplType***

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーションのタイプ (PutApplType)。

ID: MQIACF_EVENT_APPL_TYPE

データ型: MQCFIN

戻り: EventOrigin が MQEVO_MSG の場合のみ。

***EventApplName***

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーションの名前 (PutApplName)。

ID: MQCACF_EVENT_APPL_NAME

データ型: MQCFST

最大長: MQ_APPL_NAME_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

***EventApplOrigin***

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーションの発信元データ (ApplOriginData)。

ID: MQCACF_EVENT_APPL_ORIGIN

データ型: MQCFST

最大長: MQ_APPL_ORIGIN_DATA_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

***Command***

説明: コマンド・コード。

ID: MQIACF_COMMAND

データ型: MQCFIN

値: **MQCMD_INQUIRE_PROT_POLICY 数値 205**
**MQCMD_CREATE_PROT_POLICY 数値 206**
**MQCMD_DELETE_PROT_POLICY 数値 207**
**MQCMD_CHANGE_PROT_POLICY 数値 208**
これらは、WebSphere MQ 7.5 `cmqcfc.h` で定義されています

戻り: 常時。

CommandData には、PCF コマンドを構成する PCF エレメントが含まれます。

## 構成イベント・メッセージ形式

構成イベントは、標準の Advanced Message Security 形式の PCF メッセージです。

MQMD メッセージ記述子に指定できる値については、イベント・メッセージ MQMD (メッセージ記述子)を参照してください。

選択された MQMD 値は、以下のとおりです。

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR                        //for both CLI and command server
```

メッセージ・バッファーは、MQCFH 構造とそれに続くパラメーター構造で構成されます。 可能な MQCFH 値については、 イベント・メッセージ MQCFH (PCF ヘッダー)を参照してください。

選択された MQCFH 値は、以下のとおりです。

```
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}
```

MQCFH に続くパラメーターは、以下のとおりです。

***EventUserID***

説明: コマンドを発行したユーザー ID、またはイベントを生成した呼び出し。 (これは、コマンドまたは呼び出しを発行する権限の検査に使用するものと同じユーザー ID です。キューから受け取ったコマンドの場合、これはコマンド・メッセージの MD からのユーザー ID (UserIdentifier) でもあります。)

ID: **MQCACF_EVENT_USER_ID**

データ型: MQCFST

最大長: MQ_USER_ID_LENGTH

戻り: 常時。

***SecurityId***

説明: コマンド・サーバー・メッセージの場合は MQMD.AccountingToken の値、またはローカル・コマンドの場合は Windows SID。

ID: **MQBACF_EVENT_SECURITY_ID**

データ型: MQCBS.

最大長: MQ_SECURITY_ID_LENGTH

戻り: 常時。

***EventOrigin***

説明: イベントを引き起こしたアクションの発信元。

ID: **MQIACF_EVENT_ORIGIN**

データ型: MQCFIN

値:

**MQEVO_CONSOLE**
コンソール・コマンド - コマンド・ライン

**MQEVO_MSG**
WebSphere MQ エクスプローラー・プラグインからのコマンド・メッセージ

戻り: 常時。

***EventQMgr***

説明: コマンドまたは呼び出しが入れられたキュー・マネージャー。(コマンドが実行されたキュー・マネージャー、およびイベントを生成したキュー・マネージャーは、イベント・メッセージの MD にあります。)

ID: **MQCACF_EVENT_Q_MGR**

データ型: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 常時。

***ObjectType***

説明: オブジェクト・タイプ

ID: **MQIACF_OBJECT_TYPE**

データ型: MQCFIN

値: **MQOT_PROT_POLICY**
Advanced Message Security 保護ポリシー。 **1019** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

戻り: 常時。

***PolicyName***

説明: Advanced Message Security ポリシー名。

ID: **MQCA_POLICY_NAME**

データ型: MQCFST

値: **2112** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

最大長: MQ_OBJECT_NAME_LENGTH

戻り: 常時。

***PolicyVersion***

説明: Advanced Message Security ポリシーのバージョン。

ID: **MQIA_POLICY_VERSION**

データ型: MQCFIN

値 **238** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

戻り: 常時

***TolerateFlag***

説明: Advanced Message Security ポリシーの容認フラグ。

ID: **MQIA_TOLERATE_UNPROTECTED**

データ型: MQCFIN

値 **235** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

戻り: 常時。

***SignatureAlgorithm***

説明: Advanced Message Security ポリシーの署名アルゴリズム。

ID: **MQIA_SIGNATURE_ALGORITHM**

データ型: MQCFIN

値: **236** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

戻り: Advanced Message Security ポリシーに署名アルゴリズムが定義されている場合

***EncryptionAlgorithm***

説明: Advanced Message Security ポリシーの暗号化アルゴリズム。

ID: **MQIA_ENCRYPTION_ALGORITHM**

データ型: MQCFIN

値: **237** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

戻り: WebSphere MQ ポリシーに暗号化アルゴリズムが定義されている場合

***SignerDNs***

説明: 許可された署名者のサブジェクト識別名。

ID: **MQCA_SIGNER_DN**

データ型: MQCFSL

値: **2113** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

最大長: ポリシー内の最長の署名者 DN (MQ_DISTINGUISHED_NAME_LENGTH 以内)

戻り: WebSphere MQ ポリシーに定義されている場合

***RecipientDNs***

説明: 許可された署名者のサブジェクト識別名。

ID: **MQCA_RECIPIENT_DN**

データ型: MQCFSL

値: **2114** - WebSphere MQ 7.5 または cmqc.h ファイルに定義されている数値

最大長: ポリシー内の最長の受信者 DN (MQ_DISTINGUISHED_NAME_LENGTH 以内)

戻り: WebSphere MQ ポリシーに定義されている場合

# 問題および解決策

このセクションでは、IBM 製品のインストールに関連して発生する問題を解決する方法について説明します。この情報を使用して、 Advanced Message Security に関連する問題を識別して解決してください。

## com.ibm.security.pkcsutil.PKCSException: Error encrypting contents

エラー com.ibm.security.pkcsutil.PKCSException: Error encrypting contents は、 IBM Advanced Message Security が暗号アルゴリズムへのアクセスに問題があることを示しています。

次のエラーが Advanced Message Security によって返された場合:

```
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 の JCE セキュリティー・ポリシーが、MQ AMS ポリシーで使用されている署名アルゴリズムへのアクセス権を付与しているかどうか確認してください。

使用する署名アルゴリズムが現在のセキュリティー・ポリシーに指定されていない場合は、以下の場所から正しい Java ポリシー・ファイルをダウンロードします。

- IBM SDK Policy files for Java 1.4.2。
- IBM SDK Policy files for Java 5.0。
- IBM SDK Policy files for Java 6.0。
- IBM SDK Policy files for Java 7.0。

## OSGi サポート

IBM Advanced Message Security で OSGi バンドルを使用するには、追加のパラメーターが必要です。

OSGi バンドル開始中に以下のパラメーターを実行します。

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7
```

keystore.conf で暗号化されたパスワードを使用している場合、OSGi バンドルの実行時に以下のステートメントを追加する必要があります。

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7,com.ibm.misc
```

**制約事項 :** IBM WebSphere MQ AMS は、OSGi バンドル内から保護されたキューについて、MQ ベース Java クラスのみを使用した通信をサポートします。

## JMS 使用時の、保護されたキューを開く際の問題

IBM WebSphere MQ Advanced Message Security を使用している場合、保護されたキューを開く際にさまざまな問題が発生する可能性があります。

JMS を実行していて、エラー 2085 (MQRC_UNKNOWN_OBJECT_NAME) とエラー JMSMQ2008 を同時に受け取ります。

286 ページの『Java クライアントのクイック・スタート・ガイド』 の説明に従って IBM WebSphere MQ Advanced Message Security がセットアップされていることを確認しました。

考えられる原因は、IBM 以外の Java ランタイム環境を使用していることです。 これは、275 ページの『既知の制限事項』 で説明されている既知の制約事項です。

AMQ_DISABLE_CLIENT_AMS 環境変数を設定していません。

### 問題の解決方法

この問題の対応策には、以下の 4 とおりのオプションがあります。

1. サポートされる IBM Java ランタイム環境 (JRE) の下で JMS アプリケーションを開始します。
2. アプリケーションを、キュー・マネージャーが実行されているのと同じマシンに移動して、バインディング・モード接続を使用して接続する。

   バインディング・モード接続は、プラットフォーム・ネイティブのライブラリーを使用して IBM WebSphere MQ API 呼び出しを実行します。 したがって、AMS 操作の実行にはネイティブの AMS インターセプターが使用され、JRE の機能には依存しません。
3. MCA インターセプターを使用する。そうすることによって、メッセージがキュー・マネージャーに到着してすぐに、クライアントが AMS 処理を実行しなくても、そのメッセージの署名と暗号化が行えるようになるためです。

   保護がキュー・マネージャーで適用される場合、代わりのメカニズムを使用して、クライアントからキュー・マネージャーに転送中のメッセージを保護する必要があります。 通常、これを行うには、アプリケーションが使用するサーバー接続チャネルで SSL/TLS 暗号化を構成します。
4. IBM WebSphere MQ Advanced Message Security を使用しない場合は、AMQ_DISABLE_CLIENT_AMS 環境変数を設定する。

詳しくは、299 ページの『メッセージ・チャネル・エージェント (MCA) インターセプト』を参照してください。

**注 :** MCA インターセプターがメッセージを送信するキューごとに、セキュリティー・ポリシーを設定する必要があります。 つまり、ターゲット・キューには、MCA インターセプターに割り当てられた証明書の識別名 (DN) に対応する署名者と受信者の DN を付けて AMS セキュリティー・ポリシーを配置する必要があります。 つまり、キュー・マネージャーが使用する `keystore.conf` 内の `cms.certificate.channel.SYSTEM.DEF.SVRCONN` プロパティーによって指定された証明書の DN です。

# 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。 日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。 本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。 これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。 ただし、IBM 以外の製品とプログラムの操作またはサービスの 評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。 本書の提供は、お客様にこれらの特許権について 実施権を許諾することを意味するものではありません。 実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510
東京都中央区日本橋箱崎町 19 番 21 号
日本アイ・ビー・エム株式会社
日本アイ・ビー・エム株式会社
法務・知的財産
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
〒 103-8510
103-8510
東京 103-8510、日本

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。 "" 国または地域によっては、法律の強行規定により、保証責任の制限が 禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。 本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものでは ありません。 それらの Web サイトにある資料は、この IBM 製品の資料の一部では ありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号
日本アイ・ビー・エム株式会社
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用すること ができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他の ライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、 IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で 決定されたものです。 そのため、他の操作環境で得られた結果は、異なる可能性があります。 一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。 さらに、一部の測定値が、推定値である可能性があります。 実際の結果は、異なる可能性があります。 お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、 もしくはその他の公に利用可能なソースから入手したものです。 IBM は、それらの製品のテストは行っておりません。 したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、 予告なしに変更または撤回される場合があり、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。 より具体性を与えるために、それらの例には、個人、企業、ブランド、 あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在して いるとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。 お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。 このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。 従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが IBM WebSphere MQ のサービスを使用するためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。 診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

**重要 :** この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

## 商標

IBM、 IBM ロゴ、 ibm.com®は、世界の多くの国で登録された IBM Corporation の商標です。 現時点での IBM の商標リストについては、 "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml をご覧ください。 他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

この製品には、 Eclipse Project (http://www.eclipse.org/) により開発されたソフトウェアが含まれています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

部品番号:

(1P) P/N: