9.1

IBM 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、651 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® MQ バージョン 9 リリース 1、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

セキュリティー IBM MQ

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

セキュリティーの更新

セキュア・ゾーン内とオペレーター・ワークステーション上にあるすべてのハードウェアとソフトウェアについて、サポート・ライフサイクルの期間中であること、必須のソフトウェア更新によるアップグレードが実行されていること、セキュリティー更新がすぐに適用されていることを確認してください。

以下に関するセキュリティー更新の詳細を参照できます。

- すべてのプラットフォーム: IBM Security Bulletins
- z/OS® のセキュリティーとシステム保全性の APAR: IBM Z System Integrity ポータル

セキュリティーの概要

このトピック集では、IBM 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 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 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。

- 6 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『データ保全性』

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』
- 14 ページの『公開鍵インフラストラクチャー (PKI)』

IBM 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、78 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

270 ページの『SSL/TLS の取り扱い』
これらのトピックでは、IBM MQ での 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。

関連タスク

TLS による 2 つのキュー・マネージャーの接続

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 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 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。

Advanced Message Security を使用する IBM MQ は、基本機能の一部として否認防止サービスを提供しません。 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、IBM 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 MQ による識別と認証』
IBM 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 MQ での許可』
許可を使用して、IBM MQ 環境で特定の個別のユーザーまたはアプリケーションが行え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、81 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 MQ での監査』
IBM 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。

データ保全性

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、IBM 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、IBM MQ MQI client、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 MQ での暗号化』
IBM MQ は、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。

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/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。 共有キー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 は、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 は非対称鍵暗号方式を示し、これは 公開鍵暗号方式とも呼ばれます。

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。

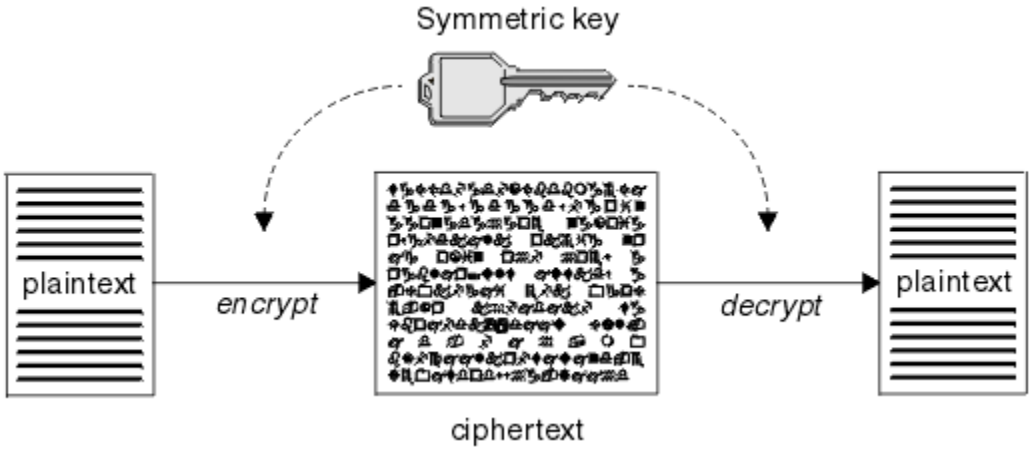


図 *1.* 対称鍵暗号化方式

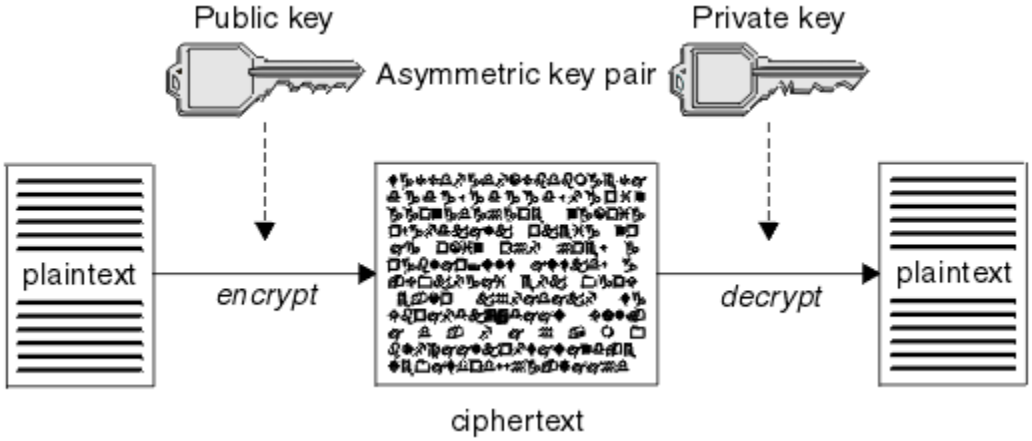


図 *2.* 非対称鍵暗号化方式

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、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・ダイジェストはハッシュ関数によって計算され、これを暗号化してデジタル署名を作成することができます。

メッセージ・ダイジェストの計算に使用するハッシュ関数は、次の 2 つの基準を満たしている必要があります。

- 片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/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。

IBM 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、システムがそのフィールドを認識できない場合、それは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

IBM 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 MQ はピアツーピア・システムであるため、SSL/TLS 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/TLS 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 MQ での証明書妥当性検査ポリシーの詳細については、42 ページの『IBM MQ における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、334 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。

重要 : SSLPEER フィルターでは、以下の表に挙げる属性だけを使用できます。 証明書 DN には他の属性を含めることができますが、これらの属性でのフィルタリングは許可されていません。

表 *1. SSLPEER* フィルターで使用できる *DN* 内の属性タイプ

属性タイプ	**説明**
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 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 IBM 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、システムで使用する以下のよう な証明書管理ツールによって生成されます。

- Multi iKeyman ツール (マルチプラットフォーム)
- z/OS z/OS 上の RACF®。

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。

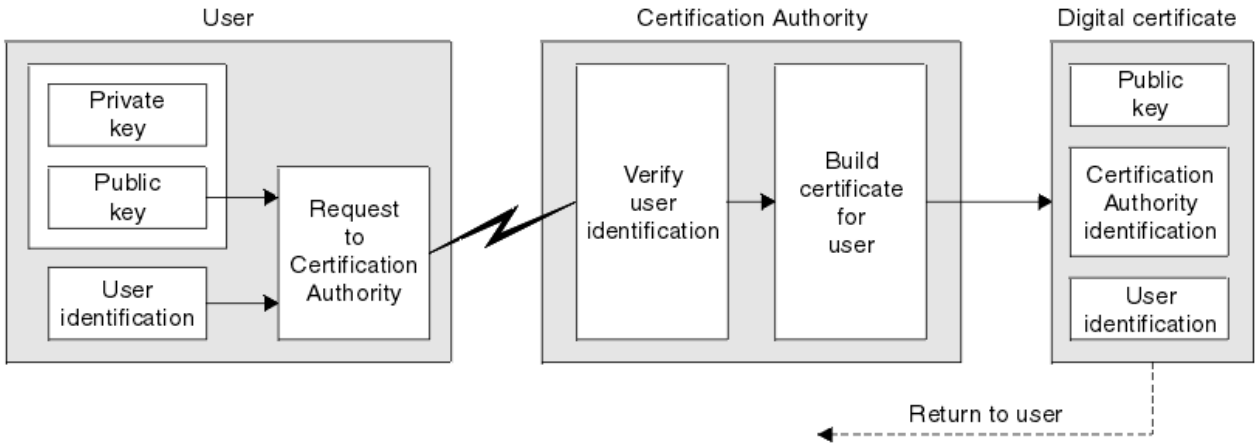


図 3. デジタル証明書の取得

図の説明:

- ユーザー ID 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。

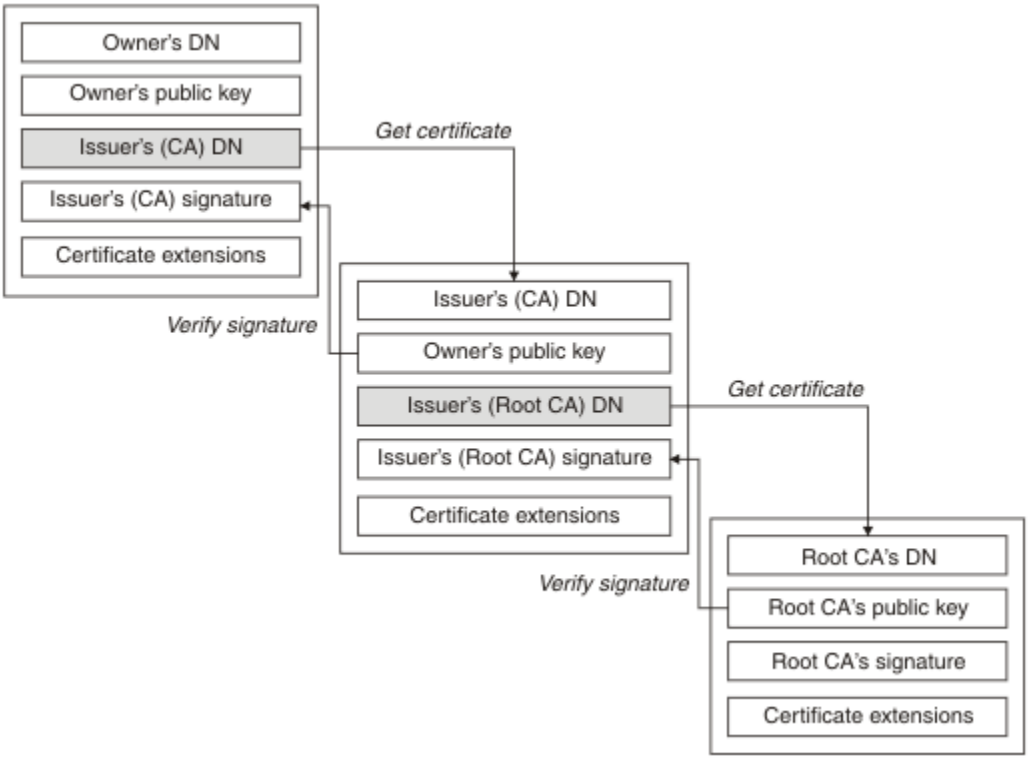


図 *4.* トラストのチェーン

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。

IBM MQ では、Online Certificate Status Protocol (OCSP) の応答側に要求を送信することによって、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®, and Windows のみ)。 あるいは、LDAP サーバー上の証明書取り消しリスト (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、334 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: 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 MQ は、TLS 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 MQ での TLS セキュリティー・プロトコル』
IBM MQ は、Transport Layer Security (TLS) プロトコルをサポートし、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。

Transport Layer Security (TLS) の概念

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。

TLS プロトコルは、インターネットでの通信セキュリティーを提供し、クライアント/サーバー・アプリケーションが、機密性の保たれた、信頼できる方法で通信できるようにします。 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。

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 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、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、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 MQ でサポートされる CipherSpecs の全リストは、416 ページの『CipherSpecs の有効化』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/TLS ハンドシェークの概要

SSL/TLS ハンドシェークにより、TLS クライアントと 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､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、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。

TLS 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. TLS クライアントは、TLS バージョンなどの暗号情報をリストした "クライアント・ハロー" メッセージを送信し、クライアントの優先順位では、クライアントがサポートする CipherSuites をリストします。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. TLS サーバーは、クライアントによって提供されるリストによって選択された CipherSuite、セッション ID、および別のランダム・バイト・ストリングサーバー を含む "サーバー・ハロー" メッセージを使用して応答します。 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. 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。 詳細については、16 ページの『TLS による識別、認証、機密性、保全性』を参照してください。
4. 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. TLS サーバーが"クライアント証明書要求"を送信すると、クライアントは、クライアントの秘密鍵を使用して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または "デジタル証明書のアラートなし"とともに送信します。 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. TLS サーバーはクライアントの証明書を検査します。 詳細については、16 ページの『TLS による識別、認証、機密性、保全性』を参照してください。
7. TLS クライアントは、ハンドシェークのクライアント部分が完了していることを示す、秘密鍵で暗号化された "終了済み" メッセージをサーバーに送信します。
8. TLS サーバーは、ハンドシェークのサーバー部分が完了していることを示す、秘密鍵で暗号化された "終了済み" メッセージをクライアントに送信します。
9. 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、TLS ハンドシェークを示しています。

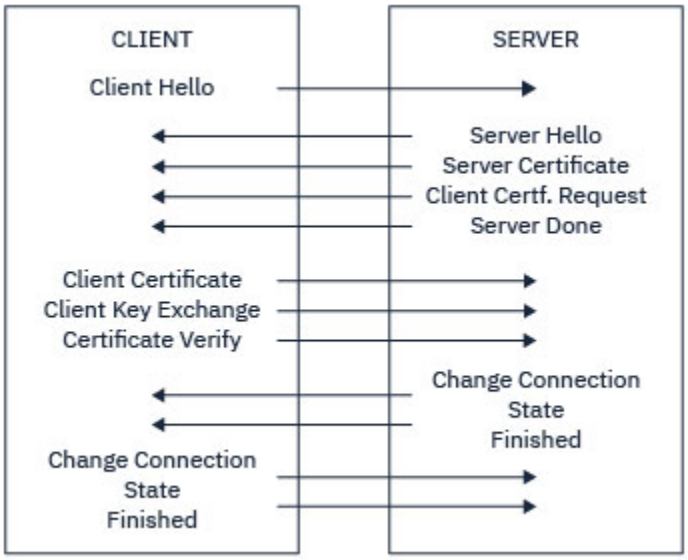


図 *5. TLS* ハンドシェークの概要

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、 15 ページの『SSL/TLS ハンドシェークの概要』 を参照してください。

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。

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、TLS クライアントに証明書を発行し、CA Y 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、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

TLS クライアントは次のものが必要です。

- CA Y の CA 証明書

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り、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

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、TLS クライアントはサーバーの証明書を検査し、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/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 (13 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 (334 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

TLS ハンドシェーク中に、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。

TLS での機密性

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。 TLS ハンドシェーク時に、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。 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。 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。

TLS での保全性

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、453 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。

TLS を使用するとデータ保全性が確保されます (ただし、416 ページの『CipherSpecs の有効化』 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。

IBM MQ は TLS 1.2 プロトコルをサポートします。 ただし、必要がある場合は、推奨されていない CipherSpecs を有効にすることもできます。

以下については、416 ページの『CipherSpecs の有効化』 を参照してください。

- IBM MQ によってサポートされる CipherSpec
- 推奨されない SSL 3.0 および TLS 1.0 の CipherSpec を有効にする方法

重要 : IBM 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、416 ページの『CipherSpecs の有効化』 を参照してください。

CipherSuite 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 TLS_RSA_WITH_AES_128_CBC_SHA 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 ビット鍵および暗号化ブロック・チェーン (CBC) モードを使用する AES 暗号化アルゴリズム
- SHA-1 メッセージ認証コード (MAC)

SSL/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。

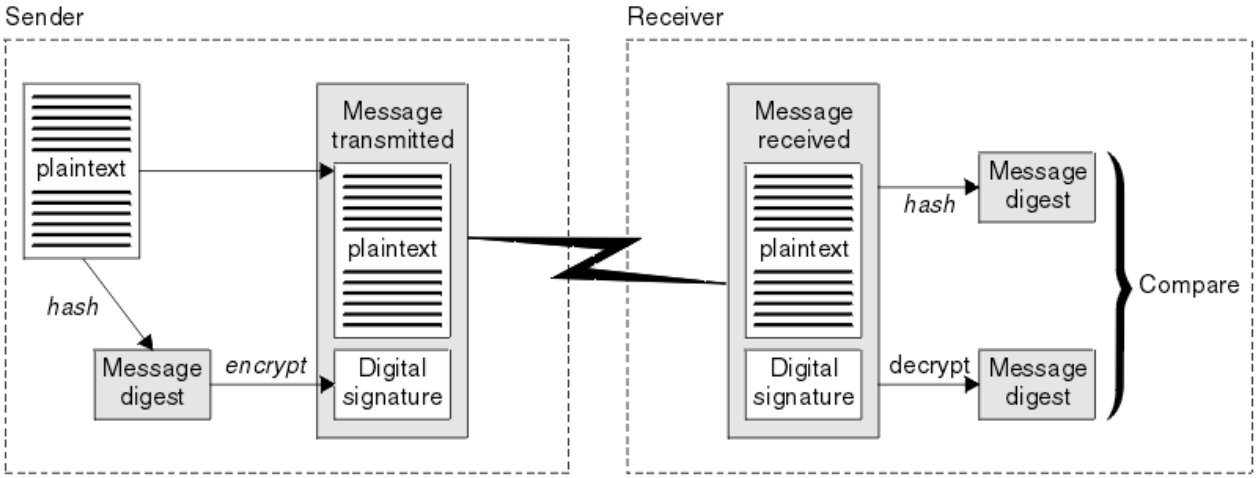


図 6. デジタル署名のプロセス

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 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。

関連概念

19 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。

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 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、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 1.0 または TLS 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 and Linux の各システムでは、IBM MQ を Suite B 準拠 TLS 1.2 プロファイルに適合するよう構成することは可能です が、Suite B 暫定プロファイルはサポートされていません。 詳しくは、39 ページの『IBM MQ における NSA Suite B 暗号方式』を参照してください。

関連資料

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 MQ セキュリティー・メカニズム

このトピック集では、IBM 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 MQ は、5 ページの『セキュリティーの概念とメカニズム』で導入された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を提供します。 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 MQ による識別と認証

IBM 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 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

6 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 MQ での許可

許可を使用して、IBM MQ 環境で特定の個別のユーザーまたはアプリケーションが行えることを制限できます。

IBM MQ 環境における許可の例を、以下で説明します。

- 許可された管理者だけが、IBM MQ リソースを管理するコマンドを実行することを許可する。
- 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、81 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 MQ での監査

IBM 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 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 MQ での機密性

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

IBM MQ 環境 では、以下のように機密性を確保することができます。

- 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 MQ が 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 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、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。
- V 9.1.4 z/OS ローカル・キューに保管されているメッセージは、z/OS データ・セット暗号化を使用して保存状態のまま暗号化できます。

データ・セット暗号化による IBM MQ for z/OS での保存データの機密性 のセクションを参照してください。 を参照してください。

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 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 MQ 環境 では、以下のようにデータ保全性を確保することができます。

- 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。 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 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)。

  IBM MQ は、変更されたメッセージであることを、メッセージを受け取ってすぐに検出します。 変更されたメッセージを受け取ると、IBM MQ は AMQ9661 エラー・メッセージをスローし、チャネルが停止します。
- 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意図的に変更できないようにするには、IBM 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、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。

  変更されたメッセージを検出すると、メッセージを受け取ろうとしているアプリケーションは戻りコード 2063 を受け取ります。 MQGET 呼び出しを使用している場合、そのメッセージは SYSTEM.PROTECTION.ERROR.QUEUE に移されます。

関連概念

7 ページの『データ保全性』
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 MQ での暗号化

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

詳しくは、22 ページの『IBM MQ での TLS 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、IBM MQ に該当する暗号方式の概念を取り上げます。

IBM MQ での TLS セキュリティー・プロトコル

IBM MQ は、Transport Layer Security (TLS) プロトコルをサポートし、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、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 TLS クライアントであり、応答側 MCA が TLS サーバーです。 IBM MQ は、TLS 1.0 および TLS 1.2 をサポートします。 チャネル定義の一環として CipherSpec によって提供される 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できます。

注 : IBM MQ 8.0.0 Fix Pack 2 以降、SSLv3 プロトコルおよびいくつかの IBM MQ CipherSpecs の使用が推奨されなくなりました。 詳しくは、Deprecation: SSLv3 protocol を参照してください。

SECPROT および SSLCIPH パラメーターを使用して、チャネル上で使用中のセキュリティー・プロトコルおよび CipherSpec を表示できます。

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。 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 IBM MQ コードは、IBM 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。 TLS ハンドシェーク時に、この IBM 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・マネージャーと IBM MQ クライアントのユーザーは、TLS クライアントとして操作するときでも、自身に関連付けられている個人デジタル証明書を持っていなければならないわけではありません (ただし、チャネルのサーバー・サイドで SSLCAUTH(REQUIRED) が指定されている場合は別です)。

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・マネージャーの属性 **SSLKeyRepository** は、キュー・マネージャーのデジタル証明書が入っている鍵リポジトリーの位置を指定します。 IBM MQ クライアント・システム上では、MQSSLKEYR 環境変数が、ユーザーのデジタル証明書が入っている鍵リポジトリーの位置を指定します。 または、IBM MQ クライアント・アプリケーションは、MQCONNX 呼び出しで、TLS 構成オプション構造である MQSCO の **KeyRepository** フィールドで、その位置を指定できます。 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。

TLS のサポート

IBM MQ は、ご使用のプラットフォームに応じて、TLS 1.0 と TLS 1.2 をサポートします。 TLS プロトコルと TLS プロトコルの詳細については、サブトピックの情報を参照してください。

IBM i
TLS サポートは、IBM i オペレーティング・システムに不可欠の要素として組み込まれています。

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 TLS サポートを提供します。

UNIX, Linux, and Windows システム
TLS サポートは、IBM MQ と共にインストールされます。

z/OS
TLS サポートは、z/OS オペレーティング・システムに不可欠の要素として組み込まれています。 z/OS の TLS のサポートのことをシステム *SSL* といいます。

IBM MQ の TLS と TLS のサポートの前提条件については、IBM MQ のシステム要件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: 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 MQ は、TLS をサポートしています。

SSL/TLS 鍵リポジトリー
相互認証される 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、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。 鍵リポジトリーは、TLS をサポートする異なるプラットフォームおよび環境ごとに、異なる名前で呼ばれます。

- IBM i IBM i では: 証明書ストア
- Java および JMS では: 鍵ストア およびトラストストア
- ULW UNIX, Linux, and Windows では: 鍵データベース・ファイル
- z/OS z/OS では: 鍵リング

詳しくは、9 ページの『デジタル証明書』および 14 ページの『Transport Layer Security (TLS) の概念』を参照してください。

相互認証される 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、次のような証明書および要求が含まれることがあります。

- 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 IBM MQ MQI client 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 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 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。

IBM i **IBM i**

鍵リポジトリーは、証明書ストアです。 デフォルトのシステム証明書ストアは、統合ファイル・システム (IFS) の `/QIBM/UserData/ICSS/Cert/Server/Default` にあります。 IBM MQ は証明書ストアのパスワードをパスワード・スタッシュ・ファイル に格納します。 例えば、キュー・マネージャー QM1 のスタッシュ・ファイル は `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth` です。

あるいは、IBM i システム証明書ストアが代わりに使用されるよう指定することもできます。 そのためには、キュー・マネージャーの **SSLKEYR** 属性の値を *SYSTEM に変更します。 その値は、キュー・マネージャーがシステム証明書ストアを使用しなければならないことと、キュー・マネージャーが使用可能なアプリケーションとしてデジタル証明書マネージャー (DCM) に登録されていることを示す値です。

証明書ストアにはキュー・マネージャーの秘密鍵も格納されます。

ULW **UNIX, Linux, and Windows システム**

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・ファイルの名前は、`.kdb` のファイル拡張子を持っている必要があります。 例えば、UNIX and Linux 上で、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 MQ がデフォルト・ロケーションにインストールされている場合、Windows 上の同等のパスは `C:&#xa5;ProgramData&#xa5;IBM&#xa5;MQ\Qmgrs\QM1\ssl\key.kdb` です。

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 : PKCS #11 暗号ハードウェア・カードには、他の方法では鍵データベース・ファイルに保持される証明書と鍵を入れることができます。 証明書と鍵が PKCS #11 カード上に保持される場合、IBM 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。

UNIX および Windows システムでは、キュー・マネージャーまたは IBM MQ MQI client 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。

z/OS **z/OS**

証明書は、z/OS 内の鍵リングに保持されます。

その他の外部セキュリティー・マネージャー (ESM) も、証明書の保管に鍵リングを使用します。

秘密鍵は、RACF によって管理されます。

IBM MQ の鍵リポジトリーの保護

IBM 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 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。

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 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。

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 **Subject Distinguished Name** または **Subject Common Name** フィールドとは別のものです。 **Subject Distinguished Name** および **Subject Common Name** 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。

証明書ラベルの構文

証明書ラベルには、次の条件で文字、数字、および句読点を含めることができます。

- Multi 証明書ラベルには、最大 64 文字を使用できます。
- z/OS 証明書ラベルには、最大 32 文字を使用できます。
- 証明書ラベルにはスペースを含めることができます。
- ラベルでは、大/小文字の区別があります。
- EBCDIC カタカナを使用するシステムでは、小文字を使用することはできません。

証明書ラベルの値に関する追加の要件を以下のセクションに示します。

証明書ラベルの使用法

IBM MQ は、証明書ラベルを使用して、TLS 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。

証明書ラベルは自分で選択した値に設定できます。 値を設定しない場合、使用しているプラットフォームに応じた命名規則に従ったデフォルトのラベルが使用されます。 詳細については、後述する特定のプラットフォームに関するセクションを参照してください。

注 :

1. Java または JMS のシステムでは、証明書ラベルを自分で設定することはできません。
2. チャネル自動定義 (CHAD) 出口によって作成される自動定義されたチャネルは、証明書ラベルを設定できません。これは、チャネルが作成される時点までに TLS ハンドシェークが既に発生しているためです。 インバウンド・チャネル用に CHAD 出口で証明書ラベルを設定しても効果がありません。

このコンテキストで、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 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。

TLS ハンドシェーク中に、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 MQ 実装環境で、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。

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 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** パラメーターの値が設定されている場合です。

インバウンド・チャネル (受信側チャネル、要求側チャネル、クラスター受信側チャネル、非修飾サーバー・チャネル、およびサーバー接続チャネルを含む) は、リモート・ピアの IBM MQ のバージョンが証明書ラベルの構成を完全にサポートしており、チャネルが TLS CipherSpec を使用している場合にのみ、構成済みの証明書を送信する点に注意してください。

修飾されていないサーバー・チャネルとは、CONNAME フィールドが設定されていないチャネルです。

それ以外の場合はすべて、キュー・マネージャーの **CERTLABL** パラメーターによって、送信される証明書が決定されます。 特に、以下のものは、チャネル固有のラベル設定に関係なく、キュー・マネージャーの **CERTLABL** パラメーターによって構成された証明書のみを受け取ります。

- IBM MQ 9.1.1 より前のバージョンでは、現在のすべての Java および JMS クライアント。
- V 9.1.1 Server Name Indication (SNI) をサポートする IBM MQ 9.1.1、Java 、および JMS クライアント (チャネルごとのチャネル上の証明書) から。
- IBM MQ より前の IBM MQ 8.0 のバージョン。
- 管理対象 .NET クライアント

また、チャネルにより使用される証明書は、チャネルの CipherSpec に適したものでなければなりません。詳細については、43 ページの『IBM 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。

IBM MQ 8.0 以降では、チャネル定義の **CERTLABL** 属性を使用して指定されたチャネルごとの証明書ラベルを使用して、同じキュー・マネージャー上で複数の証明書を使用できます。 キュー・マネージャーへのインバウンド・チャネル (サーバー接続や受信側など) は、キュー・マネージャーからの正しい証明書を提示するために、TLS Server Name Indication (SNI) を使用したチャネル名の検出に依存します。

チャネルが IBM MQ Internet Pass-Thru (MQIPT) を介して宛先キュー・マネージャーに接続され、 MQIPT 経路に **SSLServer** と **SSLClient** の両方が設定されている場合、エンドポイント間には 2 つの別個の TLS セッションが存在し、SNI データはセッション・ブレークを通過しません。 これにより、MQIPT とキュー・マネージャーとの間の TLS 接続のために、宛先キュー・マネージャーでのチャネルごとの証明書が使用されなくなります。 MQIPT を経由する TLS 接続の場合、宛先キュー・マネージャーでチャネルごとの証明書を使用するには、 MQIPT 経路で TLS プロキシー・モードを使用する必要があります。このモードでは、SNI 名を含むすべての TLS 制御フローがそのまま転送されます。 MQIPT での TLS サポートについて詳しくは、SSL/TLS サポートを参照してください。

MQIPT によって終了または開始される TLS 接続に使用される証明書は、経路ごとに個別に構成できます。例えば、**SSLServerSiteLabel** および **SSLClientSiteLabel** 経路プロパティーを使用します。

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、TLS 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、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。

Multiplatforms システム

Multi

マルチプラットフォーム では、TLS のサーバーからクライアントに証明書が送信されます。

キュー・マネージャーとクライアントはそれぞれ、以下のソースを順に検索して空ではない値を見つけます。 最初に見つけた空ではない値により、証明書ラベルが決まります。 証明書ラベルは鍵リポジトリーに存在していなければなりません。 ラベルと大/小文字および形式が正しく一致する証明書が見つからない場合、エラーが発生し、TLS ハンドシェークは失敗します。

キュー・マネージャー

1. チャネル証明書ラベル属性 **CERTLABL**。
2. キュー・マネージャー証明書ラベル属性 **CERTLABL**。
3. デフォルト。すなわち、ibmwebspheremq にキュー・マネージャーの名前をすべて小文字で付加した形式。 例えば、QM1 という名前のキュー・マネージャーの場合、デフォルトの証明書ラベルは ibmwebspheremqqm1 になります。

IBM MQ クライアント

1. CLNTCONN チャネル定義の証明書ラベル属性 **CERTLABL**。
2. MQSCO 構造 **CertificateLabel** 属性。
3. 環境変数 **MQCERTLABL**。
4. クライアントの .ini ファイルの (SSL セクションにある) **CertificateLabel** 属性。
5. デフォルト。すなわち、ibmwebspheremq にクライアント・アプリケーションを実行しているユーザー ID をすべて小文字で付加した形式。 例えば、USER1 というユーザー ID の場合、デフォルトの証明書ラベルは ibmwebspheremquser1 になります。

z/OS システム

z/OS

IBM MQ クライアントは z/OS ではサポートされません。 しかし、z/OS キュー・マネージャーが、接続の開始時には TLS クライアントの役割を果たし、接続要求の受諾時には TLS サーバーの役割を果たすことができます。 これらの両方の役割では、z/OS キュー・マネージャーの証明書ラベルの要件が適用され、この要件はマルチプラットフォーム上の要件とは異なります。

キュー・マネージャーとクライアントはそれぞれ、以下のソースを順に検索して空ではない値を見つけます。 最初に見つけた空ではない値により、証明書ラベルが決まります。 証明書ラベルは鍵リポジトリーに存在していなければなりません。 ラベルと大/小文字および形式が正しく一致する証明書が見つからない場合、エラーが発生し、TLS ハンドシェークは失敗します。

1. チャネル証明書ラベル属性 **CERTLABL**。
2. 共有されている場合、キュー共有グループの証明書ラベル属性 **CERTQSGL**。

   共有されていない場合、キュー・マネージャーの証明書ラベル属性 **CERTLABL**。
3. デフォルト。すなわち、ibmWebSphereMQ にキュー・マネージャーまたはキュー共有グループの名前を付加した形式。 このストリングは大/小文字が区別され、表示のとおりに書きこむ必要があることに注意してください。 例えば、QM1 という名前のキュー・マネージャーの場合、デフォルトの証明書ラベルは ibmWebSphereMQQM1 になります。
4. オプション 27 ページの『3』 の形式で証明書が見つからない場合、IBM MQ は、鍵リング内でデフォルトとしてマークされた証明書を使用しようとします。

キー・リポジトリーを表示する方法については、317 ページの『z/OS 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』を参照してください。

IBM MQ Java および IBM MQ JMS クライアント

IBM MQ Java および IBM MQ JMS のクライアントは、Java Secure Socket Extension (JSSE) プロバイダーの機構を使用して 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の場合、または IBM 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、JVM 環境に `javax.net.debug=ssl` を設定することにより、デバッグを使用可能にすることができます。

この変数は、アプリケーション内で構成を使用するか、またはコマンド・ラインで `-Djavax.net.debug=ssl` を入力することにより、設定できます。

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 IBM MQ Explorer 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。 詳しくは、 MQCMD_REFRESH_SECURITY コマンド およびこの製品資料の「 IBM MQ Explorer 」セクションのトピック「 *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
28 ページの『SSL/TLS キー・リポジトリーの内容と SSL/TLS 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/TLS キー・リポジトリーの内容と *SSL/TLS* 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。

IBM 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
28 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。

MQCSP パスワード保護
IBM MQ 8.0 以降、パスワードは、MQCSP 構造に組み込んで、IBM MQ 機能を使用して保護するか、または TLS 暗号化を使用するかのどちらかにより送信することができます。

重要 : MQCSP パスワード保護は TLS 暗号化より設定が簡単なので、テストや開発目的に役立ちます。ただし、それほど安全ではありません。 実動用としては、特にクライアントとキュー・マネージャーの間のネ

ットワークが信頼できない場合は、IBM MQ パスワード保護よりも TLS 暗号化を優先して使用してください。TLS 暗号化の方が安全性が高いためです。

どの暗号化を使用し、それによってどの程度の保護が提供されるのかが重要な場合は、フル TLS 暗号化を使用する必要があります。 この場合、アルゴリズムが公開されているので、**SSLCIPH** チャネル属性を使用して自社に適したものを選択できます。

MQCSP 構造の詳細については、MQCSP 構造を参照してください。

パスワード保護は、以下のすべての条件が満たされる場合に使用されます。

- 接続の両端が IBM MQ 8.0 以降を使用している。
- チャネルが TLS 暗号化を使用していない。 チャネルの **SSLCIPH** 属性がブランクであるか、**SSLCIPH** 属性が暗号化を提供しない CipherSpec に設定されている場合、チャネルは TLS 暗号化を使用していません。 NULL_SHA などのヌル暗号は、暗号化を提供しません。
- **MQCSP** を設定します。**AuthenticationType** を MQCSP_AUTH_USER_ID_AND_PWD に設定します。この値を設定すると、より多くの検査が評価され、パスワード保護が行われるかどうかを判別できます。**MQCSP** のデフォルト値。**AuthenticationType** は MQCSP_AUTH_NONE です。 デフォルト設定の場合、パスワード保護は行われません。 詳しくは、 **AuthenticationType** を参照してください。
- クライアントが IBM MQ エクスプローラーであり、ユーザー ID の互換モードが有効でない場合。これはデフォルトではありません。 この条件は、IBM MQ エクスプローラーにのみ適用されます。

これらの条件が満たされない場合、**PasswordProtection** 構成設定によって禁止されていない限り、パスワードはプレーン・テキストで送信されます。

PasswordProtection 構成設定

クライアントおよびキュー・マネージャーの .ini 構成ファイルの Channels セクションにある **PasswordProtection** 属性は、パスワードがプレーン・テキストで送信されるのを防ぐことができます。この属性は、次のいずれかの値にできます。 デフォルト値は `compatible` です。

compatible
キュー・マネージャーまたはクライアントのいずれかが IBM MQ 8.0 より前のバージョンの IBM MQ を実行している場合は、パスワードをプレーン・テキストで送信できます。 つまり、互換性のためにプレーン・テキストのパスワードが許可されます。

したがって、

- TLS 暗号化が使用されて CipherSpec が非ヌルの場合、パスワードは TLS CipherSpec で暗号化されて送信されます。
- キュー・マネージャーまたはクライアントのいずれかが IBM MQ 8.0 より前のバージョンの IBM MQ を実行しており、TLS 暗号化が使用されていない場合、パスワードはプレーン・テキストで送信されます。 パスワードはプレーン・テキストで送信されます。 IBM MQ 8.0 より前のバージョンの IBM MQ では、プレーン・テキストでのみパスワードを送信できます。
- キュー・マネージャーとクライアントの両方が IBM MQ 8.0 以降のバージョンの IBM MQ を実行しており、ヌルの CipherSpec が使用されているか、TLS 暗号化が使用されていない場合、パスワードは保護されて送信されます。 **MQCSPAuthenticationType** は MQCSP_AUTH_USER_ID_AND_PWD に設定する必要があります。
- キュー・マネージャーとクライアントの両方が IBM MQ 8.0 以降のバージョンの IBM MQ 、および **MQCSP** を実行している場合、パスワードが送信される前に接続が失敗します。**AuthenticationType** が MQCSP_AUTH_USER_ID_AND_PWD に設定されていない。

always
パスワードは、ヌルの CipherSpec ではない CipherSpec を使用して暗号化するか、 **MQCSP** で暗号化する必要があります。**AuthenticationType** は MQCSP_AUTH_USER_ID_AND_PWD に設定する必要があります。 そうしないと、接続は失敗します。 つまり、プレーン・テキストのパスワードは許可されません。

したがって、

- TLS 暗号化が使用されて CipherSpec が非ヌルの場合、パスワードは TLS CipherSpec で暗号化されて送信されます。
- キュー・マネージャーとクライアントの両方が IBM MQ 8.0 以降のバージョンの IBM MQ を実行しており、TLS 暗号化が使用されていないか、ヌルの CipherSpec が使用されている場合、パスワードは保護されて送信されます。 **MQCSPAuthenticationType** は MQCSP_AUTH_USER_ID_AND_PWD に設定する必要があります。
- キュー・マネージャーまたはクライアントのいずれかが IBM MQ 8.0 より前のバージョンの IBM MQ を実行しており、TLS 暗号化が使用されていない場合、パスワードが送信される前に接続が失敗します。 IBM MQ 8.0 より前のバージョンの IBM MQ はプレーン・テキストでのみパスワードを送信でき、 `always` はパスワードを暗号化または保護する必要があるため、接続は失敗します。

オプション
パスワードはオプションで保護して送信できますが、 **MQCSP** の場合はプレーン・テキストで送信されます。**AuthenticationType** が MQCSP_AUTH_USER_ID_AND_PWD に設定されていない。 つまり、どのクライアントもプレーン・テキストのパスワードを送信することが許可されます。

したがって、

- TLS 暗号化が使用されて CipherSpec が非ヌルの場合、パスワードは TLS CipherSpec で暗号化されて送信されます。
- ヌルの CipherSpec が使用され、 **MQCSP** である場合、パスワードはプレーン・テキストで送信されます。**AuthenticationType** が MQCSP_AUTH_USER_ID_AND_PWD に設定されていない。
- キュー・マネージャーまたはクライアントのいずれかが IBM MQ 8.0 より前のバージョンの IBM MQ を実行しており、TLS 暗号化が使用されていない場合、パスワードはプレーン・テキストで送信されます。 パスワードはプレーン・テキストで送信されます。 IBM MQ 8.0 より前のバージョンの IBM MQ では、プレーン・テキストでのみパスワードを送信できます。
- キュー・マネージャーとクライアントの両方が IBM MQ 8.0 以降のバージョンの IBM MQ を実行している場合、TLS 暗号化が使用されていない場合、またはヌルの CipherSpec が使用されている場合、パスワードは保護されます。 **MQCSPAuthenticationType** は MQCSP_AUTH_USER_ID_AND_PWD に設定されます。

warn
どのクライアントもプレーン・テキストのパスワードを送信することが許可されます。 プレーン・テキストのパスワードが受信されると、キュー・マネージャーのエラー・ログに警告メッセージ (AMQ9297) が書き込まれます。

Java クライアントおよび JMS クライアントの場合、**PasswordProtection** 属性の振る舞いは、次のように、互換モードと MQCSP モードのどちらを使用することを選択するかによって異なります。

- Java クライアントおよび JMS クライアントが互換モードで動作している場合、MQCSP 構造は接続処理中にフローされません。 したがって、 **PasswordProtection** 属性の動作は、 IBM MQ 8.0 より前のバージョンの IBM MQ を実行しているクライアントの場合と同じ動作になります。
- Java クライアントおよび JMS クライアントが MQCSP モードで動作している場合、**PasswordProtection** 属性の振る舞いは説明どおりの振る舞いになります。

Java クライアントおよび JMS クライアントでの接続認証について詳しくは、76 ページの『Java クライアントを使用した接続認証』を参照してください。

Digital Certificate Manager (DCM)
IBM i の DCM を使用してデジタル証明書および秘密鍵を管理します。

DCM (Digital Certificate Manager) を使用すると、デジタル証明書を管理したり、IBM i サーバーのセキュア・アプリケーションでデジタル証明書を使用したりすることができます。 DCM により、認証局 (CA) またはその他のサード・パーティーのデジタル証明書を要求および処理することができます。 また、ローカル認証局としてユーザー用のデジタル証明書の作成および管理を行うこともできます。

DCM では、証明書失効リスト (CRL) を使用して証明書とアプリケーションを検証する強力なプロセスもサポートされています。 DCM を使用することにより、LDAP サーバーで特定の認証局 CRL が存在するロケーションを定義できるため、IBM MQ は特定の証明書が失効していないことを確認できます。

DCM は、さまざまな形式の証明書をサポートしており、それらを自動的に検出することができます。 DCM が PKCS #12 エンコードの証明書 または暗号化されたデータを含む PKCS #7 証明書を検出すると、証明書の暗号化に使用されたパスワードを入力するように求める プロンプトが自動的に表示されます。 暗号化されたデータを含まない PKCS #7 の場合は、DCM はプロンプトを表示しません。

DCM には ブラウザー・ベースのユーザー・インターフェースが用意されており、このインターフェースからアプリケーションおよびユーザーの デジタル証明書を管理できます。 ユーザー・インターフェースは、ナビゲーション・フレームとタスク・フレームの 2 つのメインフレームに分かれています。

ナビゲーション・フレームは、証明書を管理するためのタスクまたはそれらを使用するアプリケーションの 選択に使用します。 一部のタスクは、メイン・ナビゲーション・フレームに直接表示されますが、ナビゲーション・フレームのほとんどのタスクはカテゴリー別に編成されています。 例えば、「Manage Certificates (証明書の管理)」というタスク・カテゴリーには、証明書の表示、証明書の更新、証明書のインポートなど、さまざまなガイド付きタスクが含まれています。 ナビゲーション・フレームの項目が複数のタスクを含むカテゴリーである場合は、左側に矢印が表示されます。 矢印は、そのカテゴリー・リンクを選択すると展開したタスクのリストが表示され、実行するタスクを選択できることを示します。

DCM に関する重要な情報について、以下の IBM Redbooks® 資料を参照してください。

- 「*IBM i Wired Network Security: OS/400 V5R1 DCM and Cryptographic Enhancements*」(SG24-6168)。 特に、付録を参照して、IBM i システムをローカル CA としてセットアップする場合の基本的な情報を確認してください。
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*、SG24-5659。 特に第 5 章を参照してください。 「*Digital Certificate Manager for AS/400*」 は、AS/400 DCM について説明しています。

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 TLS 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。

この情報は、次のプラットフォームに当てはまります。

- ULW UNIX, Linux, and Windows
- z/OS z/OS

ULW UNIX, Linux, and Windows での IBM MQ TLS 接続の FIPS 140-2 準拠について詳しくは、 32 ページの『UNIX, Linux, and Windows での連邦情報処理標準 (FIPS)』を参照してください。

z/OS z/OS での IBM MQ TLS 接続の FIPS 140-2 準拠について詳しくは、 34 ページの『z/OS で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 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 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

266 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。

283 ページの『runmqckm、runmqakm、および strmqikm を使用したデジタル証明書の管理』
UNIX, Linux, and Windows システムでは、**strmqikm** (iKeyman) GUI またはコマンド行から **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) を使用して、鍵およびデジタル証明書を管理します。

関連タスク

IBM MQ classes for Java での TLS の使用可能化

IBM MQ classes for JMS での Transport Layer Security (TLS) の使用

関連資料

JMS オブジェクトの TLS プロパティー

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。

ULW *UNIX, Linux, and Windows* での連邦情報処理標準 *(FIPS)*

Windows システムや UNIX and Linux システム上の SSL/TLS チャネルで暗号化が必要な場合、IBM MQ は IBM Crypto for C (ICC) と呼ばれる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、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 IBM MQ TLS 接続の FIPS 140-2 準拠は、以下のとおりです。

- すべての IBM 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 MQ MQI client・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 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ているなら、接続は JRE の TLS 実装および TLS 実装を使用し、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、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 (JRE) が FIPS 準拠である。
  - 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 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 TLS 実装および TLS 実装を使用し、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、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 (JRE) が FIPS 準拠である。
  - 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。
- 管理対象ではない .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。
- 管理対象ではない 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。

すべてのサポート対象プラットフォームは、FIPS 140-2 の認定を受けています (ただし、それぞれの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 ファイルに注記がある場合は別です)。

GSkit を使用する 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 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

IBM 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、IBM 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、442 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。

IBM MQ は、既にルール 1 とルール 2 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

266 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。

283 ページの『runmqckm、runmqakm、および strmqikm を使用したデジタル証明書の管理』
UNIX, Linux, and Windows システムでは、**strmqikm** (iKeyman) GUI またはコマンド行から **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) を使用して、鍵およびデジタル証明書を管理します。

関連タスク

IBM MQ classes for Java での TLS の使用可能化
IBM MQ classes for JMS での Transport Layer Security (TLS) の使用

関連資料

JMS オブジェクトの TLS プロパティー
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。

z/OS *z/OS での連邦情報処理標準 (FIPS)*

z/OS 上の SSL/TLS チャネル上で暗号化が必要な場合、IBM MQ は System SSL というサービスを使用します。 System SSL の目的は、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) を順守するために設計されたモードを安全に実行する機能を提供することです。

IBM MQ の TLS 接続または TLS 接続を使用して FIPS 140-2 準拠の接続を実装するときは、以下のように考慮すべき点がいくつかあります。

- IBM MQ メッセージ・チャネルを FIPS 準拠にするには、以下の条件を満たしておく必要があります。
  - System SSL Security Level 3 FMID がインストールされて構成されている (IBM MQ インストールの計画を参照)。
  - System SSL モジュールが検証されている。
  - キュー・マネージャーの SSLFIPS 属性が **YES** に設定されている。

System SSL は、FIPS モードで実行されるときに、CP Assist for Cryptographic Function (CPACF) が使用可能な場合は、それを活用します。 非 FIPS モードで実行されるときに ICSF がサポートされているハードウェアによって実行される暗号機能は、FIPS モードで実行されるときにも引き続き活用されますが、ソフトウェアで実行される必要がある RSA 署名生成は例外です。

表 2. *FIPS* モードと非 *FIPS* モードとのアルゴリズム・サポートの違い

	非 FIPS		FIPS	
アルゴリズム	**鍵サイズ**	**ハードウェア**	**鍵サイズ**	**ハードウェア**
RC2	40 および 128			
RC4	40 および 128			

表 2. *FIPS* モードと非 *FIPS* モードとのアルゴリズム・サポートの違い (続き)				
	非 FIPS		**FIPS**	
アルゴリズム	**鍵サイズ**	**ハードウェア**	**鍵サイズ**	**ハードウェア**
DES	56	x		
TDES	168	x	168	x
AES	128 および 256	x	128 および 256	x
MD5	48			
SHA-1	160	x	160	x
SHA-2	224、256、384 および 512	x	224、256、384 および 512	x
RSA	512-4096	x	1024-4096	x
DSA	512-1024		1024	
DH	512-2048		2048	

FIPS モードでは、System SSL は、表 1 に示されているアルゴリズムおよびキー・サイズを使用する証明書のみを使用できます。 X.509 証明書の妥当性検査中に、FIPS モードと互換性のないアルゴリズムが検出された場合、証明書を使用することができず、無効として処理されます。

IBM MQ 内でクライアント・モードを使用する WebSphere® Application Server クラス・アプリケーションの場合、連邦情報処理標準サポートを参照してください。

System SSL モジュールの設定については、System SSL モジュール検証設定を参照してください。

関連資料

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。

Multi *mqcertck* を使用したキュー・マネージャーの *TLS* 構成の検査

MQCERTCK コマンドは、キュー・マネージャーの TLS 構成でよくある誤りを探し、それらの問題を解決するためのいくつかの提案を提供するツールです。

概要

mqcertck コマンドは、次のことを検査します。

- キュー・マネージャーの鍵リポジトリーの存在と許可。キュー・マネージャーの **SSLKEYR** 属性で参照されます。
- キュー・マネージャー証明書の証明書の存在と妥当性。キュー・マネージャーの **CERTLABL** 属性で参照されます。
- TLS 対応チャネルの **CERTLABL** 属性で参照されるすべての証明書の存在と妥当性。
- クライアント・アプリケーションの鍵リポジトリーと証明書 (証明書がキュー・マネージャーで許可されていることの検査を含む)。

注 : **mqcertck** コマンドは、z/OS または IBM i では使用できません。

使用法

mqcertck コマンドを使用するには、コマンド行からコマンド mqcertck に必須パラメーターと必要なオプション・パラメーターを指定して実行します。

このコマンドと、このコマンドが取るパラメーターの説明については、mqcertck を参照してください。

例

キュー・マネージャーの SVRCONN チャネルに接続するクライアントからの TLS 接続を許可するように、キュー・マネージャー QM1 のセットアップが完了したところです。

複数の証明書機能を使用しているため、キュー・マネージャーとチャネルの両方の **CERTLABL** 属性で、証明書ラベルが指定されています。 チャネルの作成中にチャネルの **CERTLABL** 属性の指定で間違えたため、クライアントが接続を試みると、キュー・マネージャーが MQRC_SSL_INITIALIZATION_ERROR の 2393 の戻りコードを返します。

キュー・マネージャーをアクティブ化する前に、**mqcertck** コマンドを使用してキュー・マネージャーの TLS 構成を検査します。

コマンド `mqcertck QM1` を実行して、次の出力を受け取ります。

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

この出力では、サーバー接続チャネル MQCERTCK.CHANNEL のチャネル定義 を確認するように求められます。 ここで、誤りを確認してエラーを修正してから、`mqcertck` コマンドを再度実行して、その問題が解決したことを検証します。

クライアント接続の検査

mqcertck コマンドには、クライアントの鍵リポジトリーと、キュー・マネージャーの TLS 構成を検査する機能があります。 この検査を行うには、**mqcertck** が、キュー・マネージャーを実行しているマシンからクライアントの鍵リポジトリーにアクセスできる必要があります。

mqcertck コマンドの実行時に、**-clientkeyr** パラメーターにクライアントの鍵リポジトリーのロケーション (拡張子を除く) を指定すると、**mqcertck** はこの鍵リポジトリーをキュー・マネージャーと照合して検査します。

クライアントがキュー・マネージャーへの接続に使用するチャネルが分かっている場合は、**-clientchannel** フラグを使用してこのチャネルを指定できます。

クライアントが相互認証を使用してキュー・マネージャーに接続する場合は、**-clientusername** パラメーターまたは **-clientlabel** パラメーターを使用して、クライアントの鍵リポジトリーで使用する証明書を **mqcertck** コマンドに指示できます。

デフォルトの証明書を使用し、クライアント・アプリケーションに証明書ラベルを指定しない場合は、このアプリケーションを実行する **-clientusername** パラメーターと **username** パラメーターを使用できます。

mqcertck コマンドの操作中に、このコマンドは証明書ラベルの ibmwebspheremqXXXX を生成します。ここで、XXXX は **-clientusername** パラメーターで渡される値です。

クライアントの鍵リポジトリーを完全に検査するために、**mqcertck** コマンドでは GSKit を使用してダミーの接続を作成します。 この作成を行うには、このコマンドで、クライアントの検査時にバインドできるポートを使用可能にしておく必要があります。 使用されるデフォルトのポートは 5857 ですが、これが既に使用中の場合は、クライアントの検査時に使用する別のポートを指定できます。

注 : mqcertck コマンドはポートにバインドされますが、**mqcertck** によって外部との通信は使用されず、すべての検査がローカルで実行されます。

IBM MQ MQI client での SSL/TLS

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

IBM MQ は、 IBM MQ MQI clients on Windows、 UNIX and Linux システムの TLS サポートを提供します。 IBM MQ classes for Java を使用する場合は、IBM MQ classes for Java の使用を参照し、IBM MQ classes for JMS を使用する場合は、IBM 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 または JMS 環境には適用されません。

IBM MQ MQI client の鍵リポジトリーは、IBM 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 TLS 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 TLS 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 TLS が指定されていない限り、TLS なしで既存の IBM MQ MQI client・アプリケーションをそのまま実行できます。

TLS 鍵リポジトリーの内容、TLS 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 TLS 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、TLS チャネルを再始動します。 新規 TLS 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 MQ MQI client を実行する Windows および UNIX and Linux システムに暗号ハードウェアがある場合は、MQSSLCRYP 環境変数で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。

TLS 秘密鍵リセットおよび FIPS は、IBM MQ MQI clients でサポートされています。 詳しくは、442 ページの『SSL および TLS 秘密鍵のリセット』および 32 ページの『UNIX, Linux, and Windows での連邦情報処理標準 (FIPS)』を参照してください。

IBM MQ MQI clients の TLS サポートについて詳しくは、 266 ページの『IBM MQ MQI client・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/TLS* を使用するよう指定する
MQI チャネルで TLS 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 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. IBM MQ MQI client・アプリケーションで、MQCONNX 呼び出しが発行される場合。

   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、IBM 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、TLS 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
431 ページの『IBM MQ MQI client 用の CipherSpec の指定』
IBM MQ MQI client の CipherSpec を指定するためのオプションが 3 つあります。

IBM MQ での CipherSpec と CipherSuite
IBM MQ は TLS 1.2 CipherSpec と、RSA および Diffie-Hellman アルゴリズムをサポートしています。 ただし、必要がある場合は、推奨されていない CipherSpecs を有効にすることもできます。

以下については、416 ページの『CipherSpecs の有効化』 を参照してください。

- IBM MQ によってサポートされる CipherSpec。
- 推奨されない SSL 3.0 および TLS 1.0 の CipherSpec を有効にする方法。

IBM 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。 TLS 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、CipherSpec の一部には、ハンドシェーク鍵サイズの仕様が含まれているものがあります。 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 MQ における *NSA Suite B* 暗号方式

このトピックでは、Suite B 準拠の TLS 1.2 プロファイルに準拠するように Windows、 Linux、および UNIX 上で IBM 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 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 MQ 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 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、Linux の各システムでは、IBM MQ を、表 1 に示す各セキュリティー・レベルで Suite B 準拠 TLS 1.2 プロファイルに適合するよう構成できます。

表 *3. 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 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA と SHA-256 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 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA と SHA-256 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 MQ のデフォルトの動作は Suite B 標準に準拠しませんが、 Windows、 UNIX and 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 IBM MQ を構成できます。 Suite B を使用するように IBM 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。

IBM MQ CipherSpecs の詳細については、416 ページの『CipherSpecs の有効化』 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、IBM 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 コマンドについて詳しくは、 runmqckm および runmqakm オプションを参照してください。

注 : runmqckm および **strmqikm** コマンドは、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、290 ページの『UNIX, Linux, and Windows での自己署名個人証明書の作成』を参照してください。

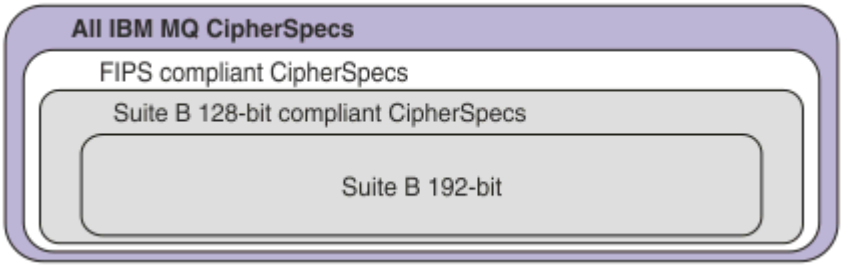
Suite B の実動用に CA 署名デジタル証明書を要求する場合は、294 ページの『UNIX, Linux, and Windows での個人証明書の要求』を参照してください。

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 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、IBM 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。

以下の図は、これらのサブセット間の関係を示しています。



IBM MQ を Suite B 準拠操作用に構成する

Windows UNIX and Linux で Suite B 準拠操作用に IBM MQ を構成する方法については、 41 ページの『Suite B 用 IBM MQ の構成』を参照してください。

IBM 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。 IBM MQ 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

266 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 MQ の構成

IBM MQ は、Windows、UNIX and Linux の各プラットフォーム上で、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 MQ は、セキュリティー・レベルを強化するため、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 の詳細については、19 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 TLS チャネルでのその影響についての詳細は、39 ページの『IBM 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。

注 : キュー・マネージャーの Suite B 設定を変更した場合に、設定内容を有効にするには、MQXR サービスを再始動しなければなりません。

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 の使い方について詳しくは、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 環境変数の使用について詳しくは、 環境変数の説明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 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。

AMQP

キュー・マネージャーの Suite B 属性設定は、そのキュー・マネージャー上の AMQP チャネルに適用されます。 キュー・マネージャーの Suite B 設定を変更した場合に、変更内容を有効にするには、AMQP サービスを再始動しなければなりません。

IBM MQ における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。
- IBM i システムの場合、証明書妥当性検査ポリシーは、オペレーティング・システムから提供されるセキュア・ソケット・ライブラリー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、オペレーティング・システムのドキュメンテーションを参照してください。
- z/OS システムの場合、証明書妥当性検査ポリシーは、オペレーティング・システムから提供されるシステム SSL 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、オペレーティング・システム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 での証明書妥当性検査ポリシーの構成方法については、42 ページの『IBM 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。 証明書妥当性検査の基本ポリシーと標準ポリシーの間の相違点については、UNIX, Linux, and Windows 上での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。

IBM 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 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。 この属性の設定の詳細については、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。 この変数の使用の詳細については、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、42 ページの『IBM MQ における証明書妥当性検査ポリシー』を参照してください。

IBM 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 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。

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 を使用するには、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。 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** コマンドを使用します。

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

ここで、*cert_label* は、表示するデジタル署名アルゴリズムの証明書ラベルです。 詳細については、デジタル証明書ラベルを参照してください。

注 : runmqckm (iKeycmd) および **strmqikm** (iKeyman) GUI 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生成されます。

```
Label : ibmmqexample
Key Size : 1024
Version : X509 V3
Serial : 4e4e93f1
Issuer : CN=Old Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : August 19, 2011 5:48:49 PM GMT+01:00
Not After : August 18, 2012 5:48:49 PM GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 98 5A 7A
    F0 18 21 EE E4 8A 6E DE C8 01 4B 3A 1E 41 90 3D
    CE 01 3F E6 32 30 6C 23 59 F0 FE 78 6D C2 80 EF
    BC 83 54 7A EB 60 80 62 6B F1 52 FE 51 9D C1 61
    80 A5 1C D4 F0 76 C7 15 6D 1F 0D 4D 31 3E DC C6
    A9 20 84 6E 14 A1 46 7D 4C F5 79 4D 37 54 0A 3B
    A9 74 ED E7 8B 0F 80 31 63 1A 0B 20 A5 99 EE 0A
    30 A6 B6 8F 03 97 F6 99 DB 6A 58 89 7F 27 34 DE
    55 08 29 D8 A9 6B 46 E6 02 17 C3 13 D3 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    09 4E 4F F2 1B CB C1 F4 4F 15 C9 2A F7 32 0A 82
    DA 45 92 9F
Fingerprint : MD5 :
    44 54 81 7C 58 68 08 3A 5D 75 96 40 D5 8C 7A CB
Fingerprint : SHA256 :
    3B 47 C6 E7 7B B0 FF 85 34 E7 48 BE 11 F2 D4 35
```

```
    B7 9A 79 53 2B 07 F5 E7 65 E8 F7 84 E0 2E 82 55
Signature Algorithm : MD5WithRSASignature (1.2.840.113549.1.1.4)
Value
    3B B9 56 E6 F2 77 94 69 5B 3F 17 EA 7B 19 D0 A2
    D7 10 38 F1 88 A4 44 1B 92 35 6F 3B ED 99 9B 3A
    A5 A4 FC 72 25 5A A9 E3 B1 96 88 FC 1E 9F 9B F1
    C5 E8 8E CF C4 8F 48 7B 0E A6 BB 13 AE 2B BD D8
    63 2C 03 38 EF DC 01 E1 1F 7A 6F FB 2F 65 74 D0
    FD 99 94 BA B2 3A D5 B4 89 6C C1 2B 43 6D E2 39
    66 6A 65 CB C3 C4 E2 CC F5 49 39 A3 8B 93 5A DD
    B0 21 0B A8 B2 59 5B 24 59 50 44 89 DC 78 19 51
Trust Status : Enabled
```

Signature Algorithm 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

V 9.1.4 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。 CipherSpecs 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対する制限が異なります。 これらの制限は、IBM MQ のすべての 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 *4. 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 RSA デジタル署名アルゴリズムを使用する CipherSpec。	RSA	RSA	ECDHE
V 9.1.4 3	(すべての TLS 1.3 CipherSpecs)	CipherSpecs 。これは、楕円曲線または RSA 公開鍵、楕円曲線秘密鍵、および楕円曲線または RSA デジタル署名アルゴリズムを使用します。	楕円曲線 (Elliptic Curve) または RSA	ECDSA または RSA	ECDHE または RSA
4	(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 CipherSpecs は、 IBM i プラットフォーム上の IBM MQ キュー・マネージャーおよび MQI クライアントではサポートされません。

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。

楕円曲線 (EC) 証明書を必要とする CipherSpec と RSA 証明書の証明書ラベルの両方を使用してチャネルを構成できます (またはその逆の方法で)。 証明書ラベルに指定された証明書がチャネルの CipherSpec に適切であることを確認する必要があります。

IBM MQ を正しく構成したと仮定すると、以下を実現できます。

- RSA と EC の証明書を併せ持つ単一のキュー・マネージャー。

- RSA または EC のいずれかの証明書を使用する同じキュー・マネージャー上に存在する異なる複数のチャネル。

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 IBM 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。

同じキュー・マネージャーまたはクライアント・アプリケーションで異なるタイプの CipherSpec を使用する必要がある場合は、適切な証明書ラベルと CipherSpec の組み合わせをクライアント定義に構成してください。

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 TLS 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・マネージャー上の TO.QM1 という名前の受信側チャネルに対して ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec を使用することを選択した場合、受信側は楕円曲線鍵と ECDSA ベースのデジタル署名を持つ個人証明書を持つ必要があります。 受信側チャネルがこれらの要件を満たしていない場合、チャネルは開始できません。

キュー・マネージャー QM1 に接続する他のチャネルは他の CipherSpec を使用できます。 ただし、それぞれのチャネルがそのチャネルの CipherSpec に対する正しいタイプの証明書を使用する場合に限ります。 例えば、QM1 が TO.QM2 という名前の送信側チャネルを使用して、メッセージを QM2 という名前の別のキュー・マネージャーに送信するとします。 RSA 公開鍵を持つ証明書をチャネルの両端で使用する場合に限り、チャネル TO.QM2 は Type 3 の CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 を使用できます。 各チャネルに対して別の証明書を構成するために、証明書ラベル・チャネル属性を使用できます。

IBM MQ ネットワークを計画する際には、どのチャネルが TLS を必要とするかを慎重に検討し、各チャネルで使用される証明書のタイプが、そのチャネルの CipherSpec で使用するのに適していることを確認してください。

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。 詳細については、デジタル証明書ラベルを参照してください。

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 : ibmmqexample
Key Size : 384
Version : X509 V3
Serial : 9ad5eeef5d756f41
Issuer : CN=Example Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : 21 August 2011 13:10:24 GMT+01:00
Not After : 21 August 2012 13:10:24 GMT+01:00
Public Key
```

```
    30 76 30 10 06 07 2A 86 48 CE 3D 02 01 06 05 2B
    81 04 00 22 03 62 00 04 3E 6F A9 06 B6 C3 A0 11
    F8 D6 22 78 FE EF 0A FE 34 52 C0 8E AB 5E 81 73
    D0 97 3B AB D6 80 08 E7 31 E9 18 3F 6B DE 06 A7
    15 D6 9D 5B 6F 56 3B 7F 72 BB 6F 1E C9 45 1C 46
    60 BE F2 DC 1B AD AC EC 64 4C 0E 06 65 6E ED 93
    B8 F5 95 E0 F9 2A 05 D6 21 02 BD FB 06 63 A1 CC
    66 C6 8A 0A 5C 3F F7 D3
Public Key Type : EC_ecPublicKey (1.2.840.10045.2.1)
Fingerprint : SHA1 :
    3C 34 58 04 5B 63 5F 5C C9 7A E7 67 08 2B 84 43
    3D 43 7A 79
Fingerprint : MD5 :
    49 13 13 E1 B2 AC 18 9A 31 41 DC 8C B4 D6 06 68
Fingerprint : SHA256 :
    6F 76 78 68 F3 70 F1 53 CE 39 31 D9 05 C5 C5 9F
    F2 B8 EE 21 49 16 1D 90 64 6D AC EB 0C A7 74 17
Signature Algorithm : EC_ecdsa_with_SHA384 (1.2.840.10045.4.3.3)
Value
    30 65 02 30 0A B0 2F 72 39 9E 24 5A 22 FE AC 95
    0D 0C 6D 6C 2F B3 E7 81 F6 C1 36 1B 9A B0 6F 07
    59 2A A1 4C 02 13 7E DD 06 D6 FE 4B E4 03 BC B1
    AC 49 54 1E 02 31 00 90 0E 46 2B 04 37 EE 2C 5F
    1B 9C 69 E5 99 60 84 84 10 71 1A DA 63 88 33 E2
    22 CC E6 1A 4E F4 61 CC 51 F9 EE A0 8E F4 DC B5
    0B B9 72 58 C3 C7 A4
Trust Status : Enabled
```

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 **runmqckm** コマンドを使用することもできます。 **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必要があります。

TLS 1.3 CipherSpecs

V 9.1.4

TLS 1.3 CipherSpecs は、ECDSA 証明書と RSA 証明書の両方をサポートします。

楕円曲線 CipherSpec と NSA Suite B

IBM 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、39 ページの『IBM 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 5. 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 5. 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

416 ページの『CipherSpecs の有効化』
CipherSpec は、**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのどちらかにおいて、**SSLCIPH** パラメーターを使用することにより有効にします。

266 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。

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

19 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。

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 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 6. 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、IBM MQ システムに対するアクセス権限を持っていないはずの IP アド レスから接続要求が出されていて、そのアドレスからの接続を一時的にブロックしておかなければフ ァイアウォールをアップデートできないという場合もあります。 それらの接続試行は、IBM MQ チャネルからではなく、IBM 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、381 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、383 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、386 ページの『クライアント・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、385 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、386 ページの『SSL または TLS 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の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、387 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、383 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

IBM 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、384 ページの『MCAUSER ユーザー ID へのユーザー 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、385 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、383 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、ホスト名、キュー・マネージャー名、またははは SSL/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。 考えられる一致は以下のとおりです。

- 最も具体的な一致は、ワイルドカード文字を使用しない名前です。以下はその例です。
  - チャネル名 A.B.C
  - IP アドレス 192.0.2.6
  - ホスト名 hursley.ibm.com
  - キュー・マネージャー名 192.0.2.6
- 最も総称的な一致は単一のアスタリスク (*) で、これは以下に一致します。
  - すべてのチャネル名
  - すべての IP アドレス
  - すべてのホスト名
  - すべてのキュー・マネージャー名
- ストリングの開始位置にアスタリスクがあるパターンは、ストリングの開始位置に定義値があるパターンより総称的です。
  - チャネルの場合、*.B.C は A.* より総称的です
  - IP アドレスの場合、*.0.2.6 は 192.* より総称的です
  - ホスト名の場合、*.ibm.com は hursley.* より総称的です
  - キュー・マネージャー名の場合、*QUEUEMANAGER は QUEUEMANAGER* より総称的です
- ストリングの特定の位置にアスタリスクがあるパターンは、ストリングの同じ位置に定義値があるパターンより総称的です。ストリングのそれ以降の各位置についても同様です。
  - チャネルの場合、A.*.C は A.B.* より総称的です
  - IP アドレスの場合、192.*.2.6 は 192.0.* より総称的です
  - ホスト名の場合、hursley.*.com は hursley.ibm.* より総称的です
  - キュー・マネージャー名の場合、Q*MANAGER は QUEUE* より総称的です

- ストリングの特定の位置にアスタリスクがあるパターンが複数ある場合には、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。
  - チャネルの場合、A.* は A*.C より総称的です
  - IP アドレスの場合、192.* は 192.*.2.* より総称的です。
  - ホスト名の場合、`hurlsey.*` は `hursley.*.com` より総称的です
  - キュー・マネージャー名の場合、Q* は Q*MGR より総称的です
- さらに IP アドレスに関しては、次のような補足事項があります。
  - 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。
- 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 7. 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=	E 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)	タイトル
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

表 7. サブストリングの優先順位 (続き)		
順序	**識別名のサブストリング**	**名前**
15	UNSTRUCTUREDNAME=	ホスト名
16	UNSTRUCTUREDADDRESS=	IP アドレス
17	DNQ=	識別名修飾子

したがって、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、IBM 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

93 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。

CHLAUTH および *CONNAUTH* の相互作用

チャネル上で単一の会話が行われた場合に、IBM MQ でチャネル認証レコード (CHLAUTH) と接続認証 (CONNAUTH) がどのように相互作用するかについて説明します。

異なるタイプのバインディング

IBM MQ では、以下の 2 つのアプリケーションの接続方法がサポートされます。

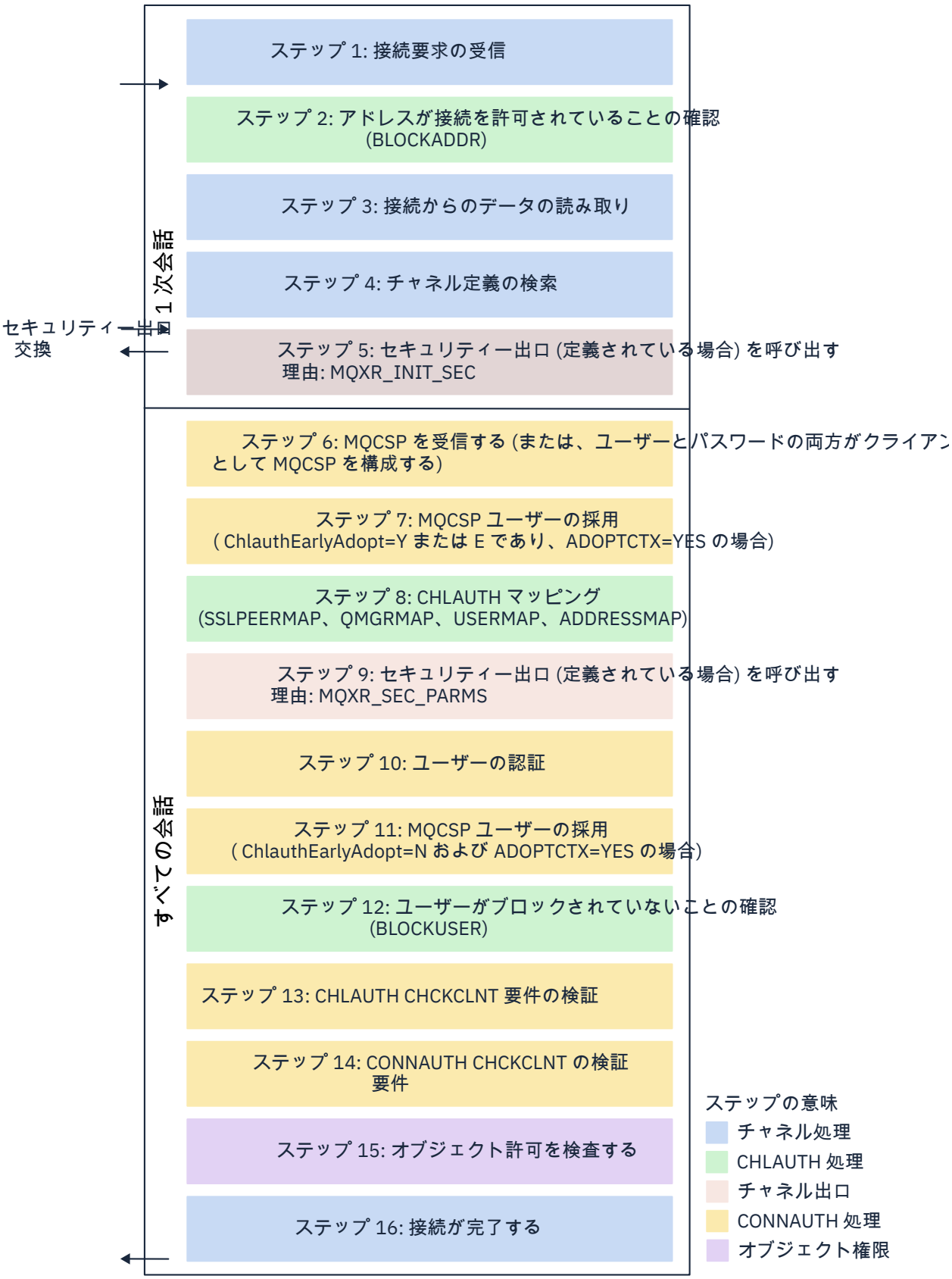
ローカル・バインディング
アプリケーションとキュー・マネージャーが同じオペレーティング・イメージ上にある場合に適用されます。 CHLAUTH は、このタイプのアプリケーション接続には関係しません。

クライアント・バインディング
アプリケーションとキュー・マネージャーがネットワークを使用して通信する場合に適用されます。 アプリケーションおよびキュー・マネージャーは、同じマシン上で実行されていても、異なるマシン上で実行されていてもかまいません。 IBM MQ では、クライアント接続はサーバー接続 (SVRCONN) チャネルの形で処理されます。この場合、CONNAUTH と CHLAUTH の両方を使用できます。

チャネルの受信側のバインディング手順

アプリケーションがキュー・マネージャーに接続すると、チャネルの両側がもう一方の側で何がサポートされているかを理解していることを確認するために、多数の検査が行われます。 チャネルの受信側では、クライアントが接続を許可されていることを確認するために、CHLAUTH および CONNAUTH を含む追加の検査が行われます。このプロセスは結果に影響する可能性があるため、セキュリティー出口も含まれる場合があります。 このチャネル接続フェーズは、バインディング・フェーズ とも呼ばれます。

次の図は、(キュー・マネージャーで) サーバーの終了時に SVRCONN チャネルが通過するステップをリストしています。



ステップ 1: 接続要求の受信
チャネル・イニシエーターまたはリスナーが、ネットワーク上の場所から接続要求を受信します。

ステップ 2: アドレスに接続が許可されているか
データを読み取る前に、IBM MQ は CHLAUTH 規則に照らしてパートナーの IP アドレスを検査し、そのアドレスが *BLOCKADDR* 規則に含まれているかどうかを確認します。 アドレスが見つからない、つまりブロックされていない場合は、次のステップに進みます。

ステップ 3: チャネルからのデータの読み取り
IBM MQ がデータをバッファーに読み込み、送信された情報の処理を開始します。

ステップ 4: チャネル定義の検索
最初のデータ・フローで、IBM MQ は、まず、送信側が開始しようとしているチャネルの名前を送信します。 これにより、受信側キュー・マネージャーは、そのチャネルに指定されているすべての設定を含むチャネル定義を検索できます。

ステップ 5: セキュリティー出口の呼び出し (定義されている場合)
チャネルにセキュリティー出口 (SCYEXIT) が定義されている場合、MQXR_SEC_PARMS に設定された出口理由 (MQCXP.**ExitReason**) で このセキュリティー出口が呼び出されます。

ステップ 6: MQCSP の受信
クライアントからユーザー ID とパスワードが提供される場合は、必要に応じて MQCSP を作成します。

クライアントが、互換モードで実行される Java または JMS アプリケーションである場合、クライアントは MQCSP 構造をキュー・マネージャーに渡しません。 その代わりに、アプリケーションがユーザー ID とパスワードを提供した場合は、ここで MQCSP 構造が作成されます。

ステップ 7: MQCSP ユーザーの採用 (ChlauthEarlyAdopt が *Y* で ADOPTCTX が YES の場合)
クライアントによって表明されたユーザー ID が認証されます。

表明された識別名から短縮ユーザー ID へのマッピングが、LDAP を使用して CONNAUTH によって行われる場合、このステップでそのマッピングが行われます。

認証が成功すると、ユーザー ID がチャネルによって採用され、CHLAUTH マッピング・ステップで使用されます。

注 : IBM MQ 9.0.4 以降、新しいキュー・マネージャーの qm.ini ファイルの channels スタンザに、**ChlauthEarlyAdopt=** *Y* パラメーターが自動的に追加されます。

ステップ 8: CHLAUTH マッピング
マッピング規則 *SSLPEERMAP*、*USERMAP*、*QMGRMAP*、および *ADDRESSMAP* を探すために、CHLAUTH キャッシュが再度検査されます。

着信チャネルに最も正確に一致する規則が使用されます。 この規則に **USERSRC***(CHANNEL)* または *(MAP)* が含まれている場合、チャネルはバインディングに進みます。

CHLAUTH 規則が、**USERSRC***(NOACCESS)* が設定された規則に評価される場合、次に行われるステップ 9 で資格情報が有効なユーザー ID とパスワードによってオーバーライドされる場合を除き、アプリケーションはチャネルへの接続をブロックされます。

ステップ 9: セキュリティー出口の呼び出し (定義されている場合)
チャネルにセキュリティー出口 (SCYEXIT) が定義されている場合、MQXR_SEC_PARMS に設定された出口理由 (MQCXP.**ExitReason**) で このセキュリティー出口が呼び出されます。

MQCSP へのポインターは、MQCXP 構造の **SecurityParms** フィールドに含められます。

MQCSP 構造には、ユーザー ID (MQCSP.**CSPUserIdPtr**) およびパスワード (MQCSP.**CSPPasswordPtr**) へのポインターが含まれます。

出口でユーザー ID とパスワードを変更することもできます。 以下の例は、セキュリティー出口でユーザー ID とパスワードの値を監査ログに出力する方法を示しています。

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

```
        pMQCXP -> SecurityParms -> CSPPasswordLength,
        pMQCXP -> SecurityParms -> CSPPasswordPtr);
```

出口では、チャネルを閉じるように IBM MQ に指示できます。これは、*MQXCC_CLOSE_CHANNEL* を **Exitresponse** フィールドで返すことによって行います。 そのようにしない場合、チャネル処理は続行され、接続認証フェーズに進みます。

注 : 表明されたユーザーがセキュリティー出口によって変更された場合、CHLAUTH マッピング・ルールは新規ユーザーに再適用されません。

ステップ 10: ユーザーの認証

キュー・マネージャーで CONNAUTH が有効化されている場合、認証フェーズが発生します。

このことを確認するには、MQSC コマンド「DISPLAY QMGR CONNAUTH」を発行します。

z/OS 以下の例は、IBM MQ for z/OS 上で実行されているキュー・マネージャーからのコマンド **DISPLAY QMGR CONNAUTH** の出力を示しています。

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

Multi 以下の例は、 IBM MQ for Multiplatforms で実行されているキュー・マネージャーからのコマンド '**DISPLAY QMGR CONNAUTH**' の出力を示しています。

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

CONNAUTH 値は、**AUTHINFO** IBM MQ オブジェクトの名前です。

オペレーティング・システム認証 (**AUTHTYPE**(*IDPWOS*)) は IBM MQ for Multiplatforms と IBM MQ for z/OS の両方で有効であるため、この例ではオペレーティング・システム認証を使用しています。

z/OS 以下の例は、IBM MQ for z/OS で実行されているキュー・マネージャーからの **AUTHTYPE**(*IDPWOS*) の出荷時デフォルト・オブジェクトを示しています。

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

Multi 以下の例は、IBM MQ for Multiplatforms で実行されているキュー・マネージャーからの **AUTHTYPE**(*IDPWOS*) の出荷時デフォルト・オブジェクトを示しています。

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

AUTHINFO TYPE(IDPWOS) には、CHCKCLNT と呼ばれる属性があります。 この値が *REQUIRED* に変更された場合、すべてのクライアント・アプリケーションが有効なユーザー ID とパスワードを提供する必要があります。

ユーザーがステップ 7 で認証された場合は、ステップ 9 でセキュリティー出口によって MQCXP 構造の SecurityParms フィールドのユーザーまたはパスワードが変更された場合を除き、ユーザーは再認証されません。

ステップ 11: MQCSP ユーザーのコンテキストの採用 (ChlauthEarlyAdopt が *N* で ADOPTCTX が YES の場合)

チャネルが MCAUSER またはアプリケーションが提供したユーザー ID のどちらを使用して実行されるかを制御する ADOPTCTX 属性を設定できます。

MQCSP で表明されたユーザー ID、または MQXCP 構造の **SecurityParms** フィールドが正常に認証され、**ADOPTCTX** が *YES* である場合は、ステップ 9 でセキュリティー出口によって MQCXP 構造の **SecurityParms** フィールドのユーザーまたはパスワードが変更された場合を除き、ステップ 7 および 8 から得られたユーザーのコンテキストが、このアプリケーションで使用するコンテキストとして採用されます。

表明されたこのユーザー ID が、IBM MQ リソースを使用する権限があるかどうかの検査対象となるユーザー ID です。

例えば、SVRCONN チャネルで MCAUSER が設定されておらず、クライアントは Linux マシンで「johndoe」を使用して実行されているとします。 アプリケーションでは MQCSP でユーザー「fred」を指定しているため、チャネルは「johndoe」をアクティブな MCAUSER として使用して実行を開始します。 CONNAUTH チェックの後、ユーザー「fred」が採用され、チャネルは「fred」をアクティブな MCAUSER として使用して実行されます。

ステップ 12: ユーザーがブロックされていないことの確認 (BLOCKUSER)

CONNAUTH の検査が成功すると、CHLAUTH キャッシュが再度検査され、アクティブな MCAUSER が *BLOCKUSER* 規則によってブロックされているかどうかが確認されます。 ユーザーがブロックされている場合は、チャネルが終了します。

Step13: CHLAUTH CHCKCLNT 要件の検証

ステップ 8 で選択した CHLAUTH 規則で REQUIRED または REQDADM の CHCKCLNT 値が追加で指定されている場合は、要件を満たすために有効な CONNAUTH ユーザー ID が指定されていることを確認するために、検証が行われます。

- CHCKCLNT (REQUIRED) が設定されている場合、ユーザーはステップ 7 または 10 で認証されている必要があります。 それ以外の場合、接続は拒否されます。
- CHCKCLNT (REQDADM) が設定されている場合、この接続が特権接続であると判別されるには、ステップ 7 または 10 でユーザーが認証されている必要があります。 それ以外の場合、接続は拒否されます。
- CHCKCLNT (ASQMGR) が設定されている場合、このステップはスキップされます。

注 :

1. CHCKCLNT (REQUIRED) または CHCKCLNT (REQDADM) が設定されているが、キュー・マネージャーで CONNAUTH が有効になっていない場合、構成に矛盾があるため、接続は MQRC_SECURITY_ERROR (2063) 戻りコードで失敗します。
2. このステップでは、ユーザーは再認証されません。

ステップ 14: CONNAUTH CHCKCLNT 要件を検証します。

キュー・マネージャーで CONNAUTH が有効化されている場合、認証フェーズが発生します。

着信接続に設定されている要件を判別するために、CONNAUTH CHCKCLNT 値が検査されます。

- CHCKCLNT (NONE) が設定されている場合、このステップはスキップされます。
- CHCKCLNT (OPTIONAL) が設定されている場合、このステップはスキップされます。
- CHCKCLNT (REQUIRED) が設定されている場合、ユーザーはステップ 7 または 10 で認証されている必要があります。 それ以外の場合、接続は拒否されます。

- CHCKCLNT (REQDADM) が設定されている場合、この接続が特権接続であると判別されるには、ステップ 7 または 10 でユーザーが認証されている必要があります。 それ以外の場合、接続は拒否されます。

注 : このステップでは、ユーザーは再認証されません。

Multi **ステップ 15: オブジェクト許可を検査する**
キュー・マネージャーに接続する適切な権限がアクティブな MCAUSER にあることを確認する検査が行われます。

ULW 詳しくは、オブジェクト権限マネージャーを参照してください。

IBM i 詳しくは、153 ページの『オブジェクト権限マネージャー (IBM i)』を参照してください。

ステップ 16: 接続が完了する
前述のステップが正常に完了すると、接続が完了します。

関連概念
CONNAUTH
リソースに対するアクセス権限がユーザーにあるかどうかを、提供されたユーザー ID とパスワードを使用して検査するようにキュー・マネージャーを構成することができます。

関連資料
SET CHLAUTH
ALTER AUTHINFO

CHLAUTH アクセスの問題の解決
チャネル認証レコード (CHLAUTH) を使用している場合に、特定のアクセスの問題を解決するために推奨される方法を示します。

デフォルトの CHLAUTH 規則

CHLAUTH の処理には、3 つのデフォルト規則があります。

- すべての MQ-admin* ユーザーによるすべてのチャネルへのアクセスを禁止
- すべての SYSTEM.* に対するアクセス権限がありません すべてのユーザーによるチャネル
- SYSTEM.ADMIN.SVRCONN チャネルへのアクセスを許可 (非 MQ-admin ユーザー)

最初の 2 つの規則では、すべてのチャネルへのアクセスをブロックします。 3 つ目の規則はより具体的であるため、他の 2 つより優先され、チャネルが SYSTEM.ADMIN.SVRCONN チャネルである場合は、チャネルへのアクセスが許可されます。

一般的な接続エラー

CHLAUTH 規則は、チャネルを開始できるかどうかを決定するために使用され、MCAUSER から別のユーザー ID へのマッピングを許可します。 チャネルを開始できない場合は、一般に以下のエラーが発生します。

- RC 2035 MQRC_NOT_AUTHORIZED
- RC 2059 MQRC_Q_MGR_NOT_AVAILABLE
- AMQ4036 アクセスが許可されていません
- AMQ9776: チャネルがユーザー ID によってブロックされました
- AMQ9777: チャネルがブロックされました。
- MQJE001: MQException が発生しました: 完了コード 2、理由 2035
- MQJE036: キュー・マネージャーが接続を拒否しました

アクセスは厳密にブロックする必要があるため、誰がチャネルにアクセスして開始できるかを制御するための CHLAUTH 規則をさらに追加します。 一時的な措置として、また上記のエラーをトラブルシューティングするために、以下の操作を実行できます。

- 60 ページの『CHLAUTH 規則の無効化』

- 60 ページの『CHLAUTH 規則の変更または削除』

CHLAUTH 規則の無効化

一時的な措置として、また上記のエラーをトラブルシューティングするために、CHLAUTH 規則を無効化できます。 ルールはいつでも再度有効にすることができます。CHLAUTH ルールを無効にすると接続の問題が解決される場合は、これが原因であることが分かります。

CHLAUTH 規則を無効化するには、以下のコマンドを発行します。

```
runmqsc: ALTER QMGR CHLAUTH (DISABLED)
```

CHLAUTH を *WARN* に設定することもできます。この場合はアクセスが許可され、規則の結果がログに記録されます。

CHLAUTH 規則の変更または削除

問題の原因である 1 つ以上の CHLAUTH 規則を削除または変更することもできます。

CHLAUTH 規則を変更するには、ACTION (REPLACE) を指定して SET CHLAUTH コマンドを使用します。 例えば、すべての MQ-admin ユーザーがすべてのチャネルにアクセスできない原因となっているデフォルト規則を、ブロックするのではなく WARN に変更するには、以下のコマンドを発行します。

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

CHLAUTH 規則を削除するには、ACTION (REMOVE) を指定して SET CHLAUTH コマンドを使用します。 例えば、すべての MQ-admin ユーザーがすべてのチャネルにアクセスできない原因となっているデフォルト規則を削除するには、以下のコマンドを発行します。

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

MATCH (RUNCHECK) を使用したアクセスのテスト

runmqsc で CHLAUTH 規則の MATCH (*RUNCHECK*) オプションを使用して、CHLAUTH 規則の結果をテストできます。 **MATCH** (*RUNCHECK*) オプションは、特定のインバウンド・チャネルがこのキュー・マネージャーに接続してきた場合に、実行時にそのチャネルと突き合わせるレコードを返します。 以下を指定してください。

- チャネル名
- ADDRESS 属性
- SSLPEER 属性 (インバウンド・チャネルが SSL または TLS を使用する場合のみ)
- QMNAME (インバウンド・チャネルがキュー・マネージャー・チャネルである場合)
- CLNTUSER 属性 (インバウンド・チャネルがクライアント・チャネルである場合)

以下の例では、デフォルト規則が有効な場合に、どの CHLAUTH 規則によって、MQ-admin ユーザー johndoe による CHAN1 という名前のチャネルへのアクセスが許可されるかを調べます。

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

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

ユーザー johndoe に対してチャネルは実行されず、*MQADMIN ユーザーに対する BLOCKUSER 規則が原因でこのユーザーはブロックされます。

以下の例では、デフォルト規則が有効な場合に、どの CHLAUTH 規則によって、非 MQ-admin ユーザー alice による CHAN1 という名前のチャネルへのアクセスが許可されるかを調べます。

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

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

ユーザー alice に対してチャネルが実行され、チャネルは alice を MCAUSER として受け入れます。MCAUSER は、IBM MQ オブジェクト権限を検査するために使用されるユーザー ID です。

関連資料
SET CHLAUTH
DISPLAY CHLAUTH

ユーザーの新規 *CHLAUTH* 規則の作成
ユーザーの一般的なシナリオのいくつかと、それらを実現するための CHLAUTH 規則の例を示します。

このトピックには、次のシナリオがあります。

- 61 ページの『特定の MQ-admin ユーザーに対するアクセス制御』
- 62 ページの『特定のユーザーおよび IBM MQ クライアント・アプリケーションのアクセス制御』
- 62 ページの『ユーザーの証明書識別名 (DN) を使用した特定のユーザーのアクセス制御』
- 63 ページの『特定ユーザーの mqm ユーザーへのマッピング』

特定の MQ-admin ユーザーに対するアクセス制御

このシナリオでは、管理の観点専用に、IBM MQ Explorer からの接続に使用するサーバー接続チャネルをセットアップします。 この用途のための特定のチャネルがあり、接続を許可する IP アドレスが 1 つ以上定義済みです。接続が指定された IP アドレスからのものでない場合、'mqm' ID のアクセスはブロックされます。

IBM MQ Explorer および MQ-admin ユーザー用に、ADMIN.CHAN という名前の SVRCONN チャネルを作成します。

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

テストのために、MQ-admin グループに含まれるユーザーとそうでないユーザーを 1 人ずつ定義していることを確認してください。 このシナリオでは、mqadm は MQ-admin グループに含まれており、alice は含まれていません。

デフォルトの CHLAUTH 規則が設定されています。 特定のユーザーに、特定の IP アドレスから MQ-admin として ADMIN.CHAN にアクセスすることを許可する 3 つのルールを追加します。

- すべてのアドレスからの NOACCESS を設定する
- ユーザー nobody のみをブロックするために、このチャネルに対して BLOCKUSER を設定する。これにより、*MQADMIN BLOCKUSER がオーバーライドされる
- 特定のアドレスのサブネットでユーザー mqadm にアクセスを許可し、mqadm ユーザー権限にマップする

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

この時点で、ユーザー mqadm は、指定された IP アドレス範囲から ADMIN.CHAN チャネルにアクセスして開始できます。

任意の時点で MATCH (RUNCHECK) を実行して、これらの各コマンドの結果を確認できます。

```
runmqsc:
DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('mqadm') ADDRESS
('192.168.1.138')
AMQ8878: Display channel authentication record details.
CHLAUTH(ADMIN.CHAN) TYPE(USERMAP)
ADDRESS(192.168.1.*) CLNTUSER(mqadm)
MCAUSER(mqadm)

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

この時点では、CHLAUTH レコードを持つユーザーのみが ADMIN.CHAN を使用したアクセスを許可されます。

特定のユーザーおよび IBM MQ クライアント・アプリケーションのアクセス制御

このシナリオでは、(setmqaut を使用して) 正しい IBM MQ 権限を提供するために、特定のユーザーに対して IBM MQ 権限を設定する必要がある場合は、デフォルトの CHLAUTH 規則で十分です。

このシナリオでは、MQ-admin ユーザーではないユーザー mqapp1 に対して権限が設定されます。 特定のアプリケーションと特定のユーザーによって使用される SVRCONN チャネル APP1.CHAN を作成します。

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

デフォルトの CHLAUTH 規則が設定されている場合、ユーザー mqapp1 は APP1.CHAN チャネルを開始できます。

IBM MQ クライアント・アプリケーションからのユーザー ID が IBM MQ オブジェクト権限の検査に使用されます。 この場合、「mqapp1」ユーザーが IBM MQ クライアント・アプリケーションを実行していると仮定し、このユーザー ID が IBM MQ オブジェクト権限の検査に使用されます。 したがって、mqapp1 がアプリケーションで必要な IBM MQ オブジェクトに対するアクセス権限を持っている場合、問題はありません。 そうでない場合は、権限エラーになります。

mqapp1 ユーザー ID に対して特定の CHLAUTH 規則を作成することでセキュリティーをさらに強化できますが、デフォルト規則では、このチャネルには MQ-admin グループのいずれのメンバーもアクセスできません。

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

ユーザーの証明書識別名 (DN) を使用した特定のユーザーのアクセス制御

このシナリオでは、キュー・マネージャーに渡される証明書をユーザーが持っている必要があります。 DN は CHLAUTH 規則の SSLPEER 設定と突き合わされます。SSLPEER ではワイルドカード文字を使用できます。

一致した場合、IBM MQ オブジェクト権限を検査するために、別の MCAUSER にユーザーをマップすることもできます。 MCAUSER のマッピングによって、IBM MQ オブジェクト権限マネージャー (OAM) で管理する必要があるユーザーの数を最小化できます。

証明書を使用している TLS チャネルがあり、次のようにするための規則が必要です。

- 特定のチャネルについてすべてのユーザーをブロックする

- IBM MQ OAM アクセスにユーザーのクライアントを使用する、特定の SSLPEER を持つユーザーのみを許可する

```
.
# block all users on any IP address.
SET CHLAUTH('SSL1.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*')
USERSRC(NOACCESS) DESCR(''block all'') WARN(NO) ACTION(ADD)
.
# override - no MQM admin rule (allow mqm group /mqm admin users to
connect.
SET CHLAUTH('SSL1.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody')
DESCR('override no mqm admin rule') WARN(NO) ACTION(ADD)
.
# allow particular SSLPEER, use client id coming in from channel
SET CHLAUTH('SSL1.SVRCONN') TYPE(SSLPEERMAP)
SSLPEER('CN=JOHNDOE,O=IBM,C=US') USERSRC(CHANNEL) ACTION(ADD)
```

チャネルに接続するクライアント・ユーザー ID が IBM MQ オブジェクトの IBM MQ OAM 権限に使用されるため、このユーザー ID には適切な IBM MQ 権限が必要です。

必要に応じて、以下を使用して別の IBM MQ ユーザー ID にマップできます。

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

これを USERSRC(CHANNEL) の代わりに使用します。

特定ユーザーの mqm ユーザーへのマッピング

これは、61 ページの『特定の MQ-admin ユーザーに対するアクセス制御』の内容に対する追加または変更です。

IBM MQ OAM でセットアップされた IBM MQ オブジェクト権限を持つ mqm ユーザーまたは MQ-admin ユーザー ID に特定のユーザーをマップするには、以下の CHLAUTH 規則を追加します。

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

これにより、特定のチャネル ADMIN.CHAN について、johndoe ユーザーが許可され、mqm ユーザーにマップされます。

関連概念

59 ページの『CHLAUTH アクセスの問題の解決』
チャネル認証レコード (CHLAUTH) を使用している場合に、特定のアクセスの問題を解決するために推奨される方法を示します。

63 ページの『チャネルの新規 CHLAUTH 規則の作成』
チャネルの一般的なシナリオのいくつかと、そうしたシナリオを実現するための CHLAUTH 規則の例を示します。独自の CHLAUTH 規則を作成する時に役立ちます。

関連資料

SET CHLAUTH

DISPLAY CHLAUTH

チャネルの新規 *CHLAUTH* 規則の作成
チャネルの一般的なシナリオのいくつかと、そうしたシナリオを実現するための CHLAUTH 規則の例を示します。独自の CHLAUTH 規則を作成する時に役立ちます。

このトピックには、次のシナリオがあります。

- 64 ページの『特定の IP アドレス範囲からの特定のチャネルへのアクセスのみを許可する』
- 64 ページの『特定のチャネルに対し、すべてのユーザーをブロックするが、特定のユーザーにのみ接続を許可する』
- 64 ページの『受信側チャネルおよび送信側チャネルに対する CHLAUTH の使用』

特定の IP アドレス範囲からの特定のチャネルへのアクセスのみを許可する

このシナリオでは、以下のようにすることが必要です。

- あらゆる場所からのチャネルへのアクセスを禁止する
- 特定の IP アドレスまたはアドレス範囲からのアクセスを許可する

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

これにより、指定された特定の IP アドレス範囲から接続が行われた場合にのみ、APP2.CHAN チャネルの開始が許可されます。

MCAUSER として接続しているユーザーが mqapp2 にマップされ、その結果として IBM MQ OAM 権限を取得します。

特定のチャネルに対し、すべてのユーザーをブロックするが、特定のユーザーにのみ接続を許可する

このシナリオでは、チャネル MY.SVRCONN へのアクセスに対してデフォルトの CHLAUTH 規則が設定されています。

以下を追加する必要があります。

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

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

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

このコードの最初の部分は、すべてのユーザーの MY.SVRCONN への接続をブロックし、接続が特定のユーザー ID johndoe から行われた場合にのみ、MY.SVRCONN チャネルの開始を許可します。

チャネルに接続しているユーザー johndoe が、IBM MQ オブジェクトの IBM MQ OAM 権限に使用されます。 したがって、このユーザー ID には適切な IBM MQ 権限が必要です。

必要に応じて、以下を使用して別の IBM MQ ユーザー ID にマップできます。

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

これを USERSRC(CHANNEL) の代わりに使用します。

受信側チャネルおよび送信側チャネルに対する CHLAUTH の使用

CHLAUTH 規則を使用して受信側チャネルと送信側チャネルのセキュリティーを強化し、受信側チャネルへのアクセスを制限できます。 CHLAUTH 規則に対して追加または変更を行う場合、更新された CHLAUTH 規則はチャネルの開始時にのみ適用されることに注意してください。そのため、チャネルが既に実行中の場合は、CHLAUTH の更新を適用するために、チャネルを停止してから再開する必要があります。

CHLAUTH 規則は任意のチャネルで使用できますが、いくつかの制約事項があります。 例えば、USERMAP 規則は SVRCONN チャネルにのみ適用されます。

この例では、特定の IP アドレスからの接続にのみ TO.MYSVR1 チャネルの開始が許可されます。

```
# First you could lock down the channel by disallowing all
```

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

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

この例では、特定のキュー・マネージャーからの接続のみが許可されます。

```
# Lock down all access:
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')

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

関連概念

59 ページの『CHLAUTH アクセスの問題の解決』
チャネル認証レコード (CHLAUTH) を使用している場合に、特定のアクセスの問題を解決するために推奨される方法を示します。

61 ページの『ユーザーの新規 CHLAUTH 規則の作成』
ユーザーの一般的なシナリオのいくつかと、それらを実現するための CHLAUTH 規則の例を示します。

関連資料

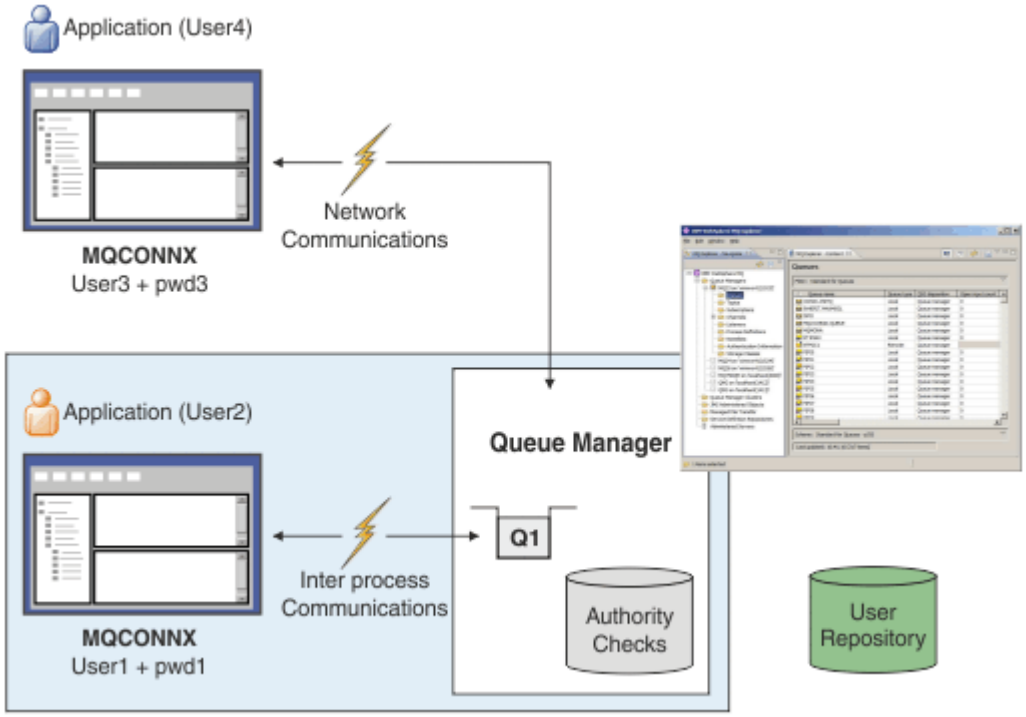
SET CHLAUTH

DISPLAY CHLAUTH

接続認証

接続認証は、以下のようなさまざまな方法で行えます。

- アプリケーションでユーザー ID とパスワードを提供できます。 アプリケーションをクライアントとして使用することもできますし、アプリケーションでローカル・バインディングを使用することもできます。
- 提供されたユーザー ID とパスワードに基づいて動作するようにキュー・マネージャーを構成できます。
- ユーザー ID とパスワードの組み合わせが有効かどうかを、リポジトリーを使用して判別できます。



この図では、2 つのアプリケーションが 1 つのキュー・マネージャーに対して接続を行っています。一方のアプリケーションはクライアントであり、もう一方はローカル・バインディングを使用しています。 アプリケーションはキュー・マネージャーに接続するためにさまざまな API を使用する可能性がありますが、どれもユーザー ID とパスワードを提供する機能があります。 アプリケーションが実行されているユーザー ID、図の `User2` および `User4` (IBM MQ に提示される通常のオペレーティング・システム・ユーザー ID) は、アプリケーション `User1` および `User3` によって提供されるユーザー ID とは異なる場合があります。

キュー・マネージャーは構成コマンドを受け取り (この図では IBM MQ Explorer を使用)、リソースのオープンを管理し、それらのリソースに対するアクセス権限を検査します。 IBM MQ には多くの異なるリソースがあり、アプリケーションがそれらにアクセスするためには権限を必要とする可能性があります。 この図は出力用キューのオープンを示していますが、他のリソースにも同じ原理が当てはまります。

ユーザー ID とパスワードの検査に使用されるリポジトリーの詳細については、ユーザー・リポジトリーを参照してください。

関連概念

66 ページの『接続認証: 構成』
リソースに対するアクセス権限がユーザーにあるかどうかを、提供されたユーザー ID とパスワードを使用して検査するようにキュー・マネージャーを構成することができます。

70 ページの『接続認証: アプリケーションの変更』

71 ページの『接続認証: ユーザー・リポジトリー』
使用している各キュー・マネージャーに対して、ユーザー ID とパスワードの認証のために、異なるタイプの認証情報オブジェクトを選択できます。

接続認証: 構成

リソースに対するアクセス権限がユーザーにあるかどうかを、提供されたユーザー ID とパスワードを使用して検査するようにキュー・マネージャーを構成することができます。

キュー・マネージャーでの接続認証をオンにする

キュー・マネージャー・オブジェクトでは、**CONNAUTH** 属性を認証情報 (AUTHINFO) オブジェクトの名前に設定できます。 このオブジェクトには、以下の 2 つのタイプ (AUTHTYPE 属性) のいずれかを指定できます。

IDPWOS
キュー・マネージャーがローカル・オペレーティング・システムを使用してユーザー ID とパスワードを認証するように指示します。

IDPWLDAP
キュー・マネージャーが LDAP サーバーを使用してユーザー ID とパスワードを認証するように指示します。

注 : 他のタイプの認証情報オブジェクトを **CONNAUTH** フィールドで使用することはできません。

IDPWOS と IDPWLDAP はいくつかの属性が似ており、ここではそれらについて説明します。 その他の属性は後で取り上げます。

ローカル接続を検査するには、AUTHINFO 属性 **CHCKLOCL** (ローカル接続検査) を使用します。 クライアント接続を検査するには、AUTHINFO 属性 **CHCKCLNT** (クライアント接続検査) を使用します。 キュー・マネージャーが変更を認識するためには、構成をリフレッシュする必要があります。

```
ALTER QMGR CONNAUTH(USE.PW)
DEFINE AUTHINFO(USE.PW) +
AUTHTYPE(IDPWOS) +
FAILDLAY(10) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED)
REFRESH SECURITY TYPE(CONNAUTH)
```

CONNAUTH 内の USE.PW は、AUTHINFO 定義と一致するストリングです。

CHCKLOCL と **CHCKCLNT** に値を設定するときには、以下に示す値セットから選択します。これによって検査の厳密性を変えることができます。

NONE
検査をオフにします。

OPTIONAL
アプリケーションからユーザー ID とパスワードが提供された場合、それらが有効なペアであることを確認します。ただし、それらの提供は必須ではありません。 このオプションは、例えばマイグレーションの際に役立つ場合があります。

重要 : OPTIONAL は、より厳密な CHLAUTH 規則を使用するために設定できる最小の値です。

NONE を選択し、クライアント接続が CHCKCLNT REQUIRED (または z/OS 以外のプラットフォームでは REQDADM) を持つ CHLAUTH レコードと一致する場合、接続は失敗します。 z/OS 以外のプラットフォームではメッセージ AMQ9793 を受け取り、 z/OS ではメッセージ CSQX793E を受け取ります。

REQUIRED
すべてのアプリケーションが有効なユーザー ID とパスワードを提供する必要があります。 以下の注も参照してください。

REQDADM
特権ユーザーは有効なユーザー ID とパスワードを指定しなければなりませんが、非特権ユーザーは OPTIONAL 設定と同じように扱われます。 以下の注も参照してください。 z/OS (この設定は z/OS システムでは使用できません。)

注 :

CHCKLOCL を REQUIRED または REQDADM に設定すると、**runmqsc** を使用してキュー・マネージャーをローカルに管理することができなくなります (エラー AMQ8135: 権限がありません)。 ただし、ユーザーが **runmqsc** コマンド・ラインで -u UserId パラメーターを指定した場合は例外です。 これを設定すると、**runmqsc** からコンソールにユーザーのパスワードを入力するようプロンプトが出されます。

同様に、ローカル・システムで IBM MQ エクスプローラーを実行しているユーザーがキュー・マネージャーに接続しようとすると、エラー AMQ4036 が表示されます。 ユーザー名とパスワードを指定するには、ローカル・キュー・マネージャー・オブジェクトを右クリックして、「**接続の詳細**」 >「**プロパティー ...**」 を選択します。 表示されます。 「**ユーザー ID**」セクションで、使用するユーザー名とパスワードを入力してから「**OK**」をクリックします。

同様の考慮事項は、**CHCKCLNT** を使用したリモート接続にも当てはまります。

CONNAUTH は、移行されたキュー・マネージャーの場合は空ですが、新しいキュー・マネージャーの場合は *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* に設定されます。 その前の **AUTHINFO** 定義では、デフォルトで **CHCKCLNT** が *REQDADM* に設定されます。

したがって、接続するためには、特権ユーザー ID を使用して既存のクライアントの正しいオペレーティング・システム・パスワードを入力する必要があります。

警告 : クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを経由してプレーン・テキストで送信される場合があります。 クライアント・アプリケーションのパスワードが適切に 保護されるようにするには、28 ページの『MQCSP パスワード保護』を参照してください。

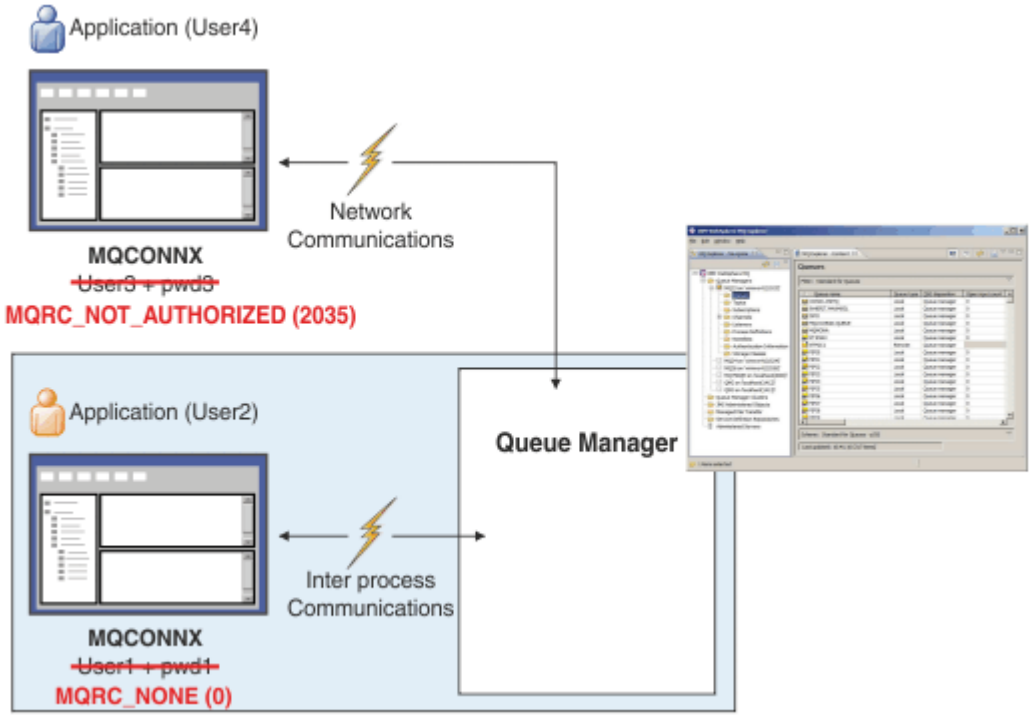
構成の細分度

ユーザー ID とパスワードの検査をオンにするために使用される **CHCKLOCL** と **CHCKCLNT** に加え、**CHCKCLNT** を使用したより具体的な構成が可能になるようにするための CHLAUTH 規則の機能拡張があります。

例えば、全体的な **CHCKCLNT** 値を OPTIONAL に設定した後、CHLAUTH 規則の **CHCKCLNT** を REQUIRED または REQDADM に設定して、特定のチャネルに対してより厳密になるようにアップグレードすることができます。 デフォルトでは、CHLAUTH 規則は CHCKCLNT(ASQMGR) の設定で実行されるので、この細分度を使用する必要はありません。 以下に例を示します。

```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)
SET CHLAUTH('*') TYPE(ADDRESSMAP) +
ADDRESS('*') USERSRC(CHANNEL) +
CHCKCLNT(REQUIRED)
SET CHLAUTH('*') TYPE(SSLPEERMAP) +
SSLPEER('CN=*') USERSRC(CHANNEL)
```

エラー通知



アプリケーションがユーザー ID とパスワードを提供する必要があるときに提供しなかった場合、またはそれらの提供がオプションである場合でも組み合わせが間違っていた場合は、エラーが記録されます。

注 : CHCKLOCL または **CHCKCLNT** で NONE オプションを使用してパスワード検査をオフにすると、無効なパスワードは検出されません。

失敗した認証は **FAILDLAY** 属性で指定された秒数だけ保持された後に、アプリケーションにエラーが返されます。 これにより、アプリケーションが接続を繰り返し試みるのをある程度防ぐことができます。

エラーは、以下のようにいくつかの方法で記録されます。

アプリケーション
アプリケーションに、標準の IBM MQ セキュリティー・エラー RC2035 - MQRC_NOT_AUTHORIZED が返されます。

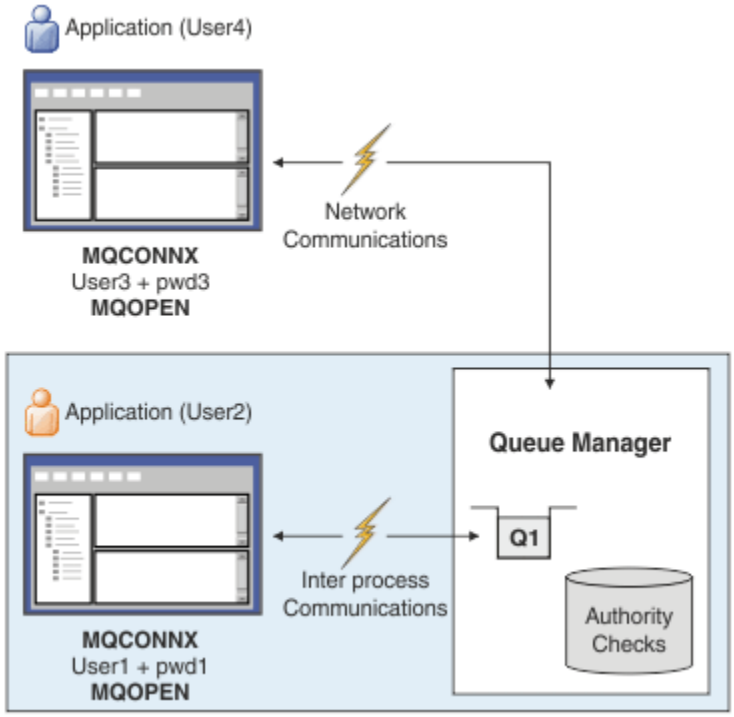
管理者
IBM MQ 管理者にはイベントがエラー・ログで報告されるので、例えば、アプリケーションが拒否された理由が、接続権限がなかったためではなく、ユーザー ID とパスワードの検査に失敗したためであることが分かります。

モニター・ツール
次のイベント・メッセージを SYSTEM.ADMIN.QMGR.EVENT キューに送信して権限イベントをオンにすれば、モニター・ツールにも失敗を通知できます。

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

この「権限がありません」イベントはタイプ 1 接続イベントであり、他のタイプ 1 イベントと同じフィールドがありますが、提供された MQCSP ユーザー ID のフィールドが追加されています。 パスワードはイベント・メッセージには示されません。 つまり、イベント・メッセージには 2 つのユーザー ID があります。つまり、アプリケーションが実行されている ID と、アプリケーションがユーザー ID とパスワードの検査を行うために提示した ID です。

許可との関係



あるアプリケーションを実行しているユーザー ID と、アプリケーションが出力用にキューを開いた際にそのアプリケーションがパスワードと共に提示したユーザー ID が同じではない場合に、アプリケーションがユーザー ID とパスワードを指定することを必須とするようにキュー・マネージャーを構成できます。 例:

```
ALTER QMGR CONNAUTH(USE.PWD)
DEFINE AUTHINFO(USE.PWD) +
```

```
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)
```

ユーザー ID とパスワードの処理は、認証情報オブジェクトの **ADOPTCTX** 属性によって制御されます。

ADOPTCTX(YES)
アプリケーションの許可検査はすべて、パスワードで認証したユーザー ID で行われます。この場合、そのコンテキストが残りの接続存続期間のアプリケーション・コンテキストとして採用されます。

重要 : ADOPTCTX(YES) および OS ユーザー ID を使用する場合は、採用するユーザー ID がユーザー ID の最大長を超えていないことを確認する必要があります。 詳しくは、 81 ページの『ユーザー ID』 を参照してください。

ADOPTCTX(NO)
アプリケーションは接続時に認証のためにユーザー ID とパスワードを提供しますが、その後の許可検査には、アプリケーションを実行しているユーザー ID を使用し続けます。 このオプションは、マイグレーション時に、またはチャネル認証レコードなどの他のメカニズムを使用して 「メッセージ・チャネル・エージェント・ユーザー ID (MCAUSER)」を割り当てることを計画している場合に見つけることができます。

重要 :

認証情報オブジェクトで **ADOPTCTX(YES)** パラメーターを使用する際、`qm.ini` ファイルのチャネル・スタンザで **ChlauthEarlyAdopt** パラメーターを設定しなければ、別のセキュリティー・コンテキストを採用できません。

例えば、デフォルトの認証情報オブジェクトが **ADOPTCTX(YES)** に設定され、ユーザー `fred` がログインしているとします。 次の 2 つの CHLAUTH 規則が構成されています。

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

次のコマンドを、ユーザー bob のセキュリティー・コンテキストを採用してコマンドを認証する目的で発行します。

```
runmqsc -c -u bob QMGR
```

実際には、キュー・マネージャーは bob ではなく `fred` のセキュリティー・コンテキストを使用するため、接続に失敗します。

ChlauthEarlyAdopt について詳しくは、 channels スタンザの属性を参照してください。

関連概念
65 ページの『接続認証』

70 ページの『接続認証: アプリケーションの変更』

71 ページの『接続認証: ユーザー・リポジトリー』
使用している各キュー・マネージャーに対して、ユーザー ID とパスワードの認証のために、異なるタイプの認証情報オブジェクトを選択できます。

接続認証: アプリケーションの変更

アプリケーションは MQCONNX の呼び出し時に、接続セキュリティー・パラメーター (MQCSP) 構造体の中でユーザー ID とパスワードを指定できます。 そのユーザー ID とパスワードは、キュー・マネージャーとともに提供されるオブジェクト権限マネージャー (OAM) (z/OS システムではキュー・マネージャーとともに提供される許可サービス・コンポーネント) に検査のために渡されます。 ユーザー独自のカスタム・インターフェースを作成する必要はありません。

アプリケーションがクライアントとして実行されている場合、ユーザー ID とパスワードは、クライアント・サイドおよびサーバー・サイドのセキュリティー出口にも処理のために渡されます。 これらは、チャネル・インスタンスのメッセージ・チャネル・エージェント・ユーザー ID (MCAUSER) 属性の設定にも使

用できます。 セキュリティー出口は、出口理由 MQXR_SEC_PARMS で呼び出されてこの処理が行われます。 クライアント・サイドのセキュリティー出口と接続前出口は、キュー・マネージャーに送られる前の MQCONN に変更を加えることができます。

警告 : クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを経由してプレーン・テキストで送信される場合があります。 クライアント・アプリケーションのパスワードが適切に 保護されるようにするには、28 ページの『MQCSP パスワード保護』を参照してください。

XAOPEN ストリングを使用してユーザー ID とパスワードを指定することにより、アプリケーション・コードが変更されないようにすることができます。

注 :

IBM WebSphere MQ 6.0 以降、セキュリティー出口で MQCSP を設定できるようになりました。 したがって、これ以降のレベルのクライアントはアップグレードする必要はありません。

ただし、 IBM MQ 8.0 より前のバージョンの IBM MQ では、MQCSP は、アプリケーションによって提供されたユーザー ID とパスワードに制限を設けませんでした。 IBM MQ が提供するフィーチャーでこれらの値を使用する場合は、それらのフィーチャーの使用に適用される制限がありますが、ユーザー自身の出口に渡すだけであれば、それらの制限は適用されません。

関連概念

65 ページの『接続認証』

66 ページの『接続認証: 構成』
リソースに対するアクセス権限がユーザーにあるかどうかを、提供されたユーザー ID とパスワードを使用して検査するようにキュー・マネージャーを構成することができます。

71 ページの『接続認証: ユーザー・リポジトリー』
使用している各キュー・マネージャーに対して、ユーザー ID とパスワードの認証のために、異なるタイプの認証情報オブジェクトを選択できます。

接続認証*:* ユーザー・リポジトリー

使用している各キュー・マネージャーに対して、ユーザー ID とパスワードの認証のために、異なるタイプの認証情報オブジェクトを選択できます。

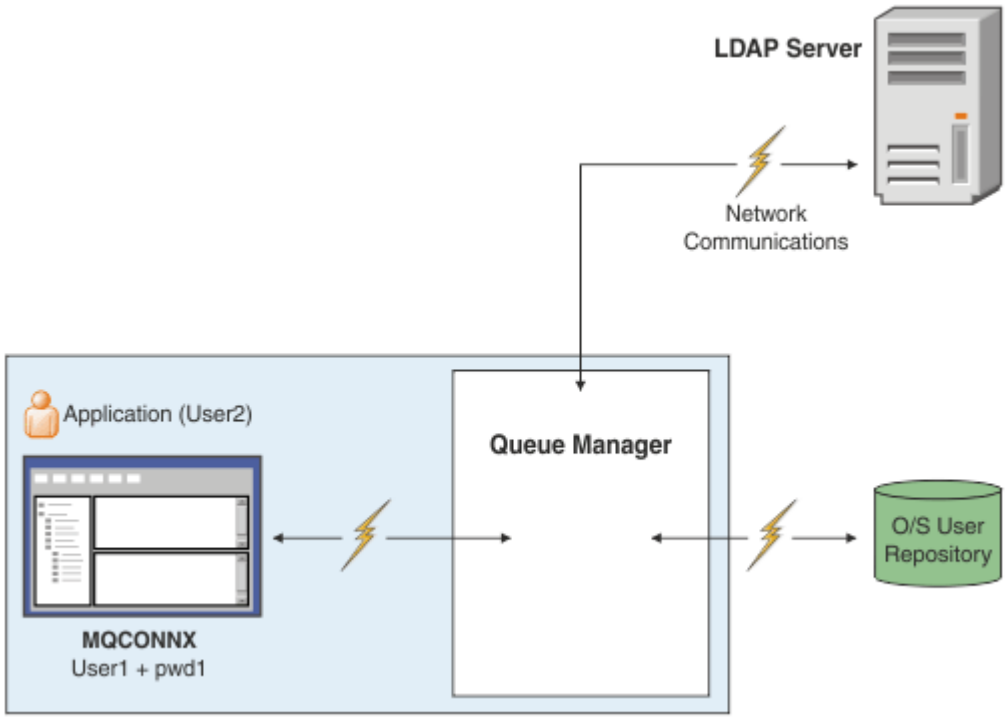


図 7. 認証情報オブジェクトのタイプ

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

認証情報オブジェクトには、図に示した 2 つの種類があります。

- IDPWOS は、キュー・マネージャーがローカル・オペレーティング・システムを使用してユーザー ID とパスワードを認証するように指示するために使用します。 ローカル・オペレーティング・システムの使用を選択すると、前述したように、設定する必要があるのは共通属性になります。
- IDPWLDAP は、キュー・マネージャーが LDAP サーバーを使用してユーザー ID とパスワードを認証することを指示するために使用します。 LDAP サーバーの使用を選択する場合は、このトピックの追加情報が提供されます。

各キュー・マネージャーで使用するために選択できる認証情報オブジェクトは 1 つのタイプのみです。これは、キュー・マネージャーの **CONNAUTH** 属性に該当する認証情報オブジェクトの名前を指定することによって選択します。

認証での LDAP サーバーの使用

CONNAME フィールドに、キュー・マネージャーの LDAP サーバーのアドレスを設定します。 コンマ区切りリストを使用して、追加の LDAP サーバーのアドレスを指定できます。これは、LDAP サーバー自体にこの機能が提供されていない場合の予備として役立ちます。

必須の LDAP サーバー ID とパスワードを **LDAPUSER** フィールドと **LDAPPWD** フィールドに設定します。これにより、キュー・マネージャーは LDAP サーバーにアクセスして、ユーザー・レコードに関する情報を見つけることができるようになります。

LDAP サーバーへのセキュア接続

チャネルの場合と異なり、LDAP サーバーとの通信に TLS の使用を有効にするための **SSLCIPH** パラメーターは用意されていません。 この場合、IBM MQ は LDAP サーバーへのクライアントとして機能し、LDAP サーバーで多くの構成を行えるようになります。 IBM MQ のいくつかの既存のパラメーターは、接続が機能する方法を構成するために使用されます。

SECCOMM フィールドの設定により、LDAP サーバーへの接続に TLS を使用するかどうかを制御します。

この属性に加えて、キュー・マネージャー属性 **SSLFIPS** と **SUITEB** によって、選択される暗号仕様のセットが制限されます。 LDAP サーバーに対してキュー・マネージャーを識別するために使用される証明書は、キュー・マネージャー証明書 (`ibmwebspheremq` *`qmgr-name`* または **CERTLABL** 属性の値) です。 詳細については、デジタル証明書ラベルを参照してください。

LDAP ユーザー・リポジトリー

LDAP ユーザー・リポジトリーを使用する場合、キュー・マネージャーに LDAP サーバーの位置を通知すること以外に、キュー・マネージャーで必要となる追加の構成がいくつかあります。

LDAP サーバーで定義されるユーザー ID は、一意に識別される階層構造になっています。 そのため、アプリケーションはキュー・マネージャーに接続して、そのユーザー ID を完全修飾の階層型ユーザー ID として提示できます。

ただし、アプリケーションが提示しなければならない情報を簡単にするため、階層の第 1 部分をすべての ID で共通と見なし、それをアプリケーションによって提供される短縮された ID の前に自動的に追加するように、キュー・マネージャーを構成できます。 キュー・マネージャーは、その後、LDAP サーバーに完全な ID を提示できます。

LDAP が ID を検索する LDAP 階層内の初期ポイントに BASEDNU を設定します。 BASEDNU を設定する際、LDAP 階層内で ID を検索するときに結果が 1 つだけ返されるようにする必要があります。

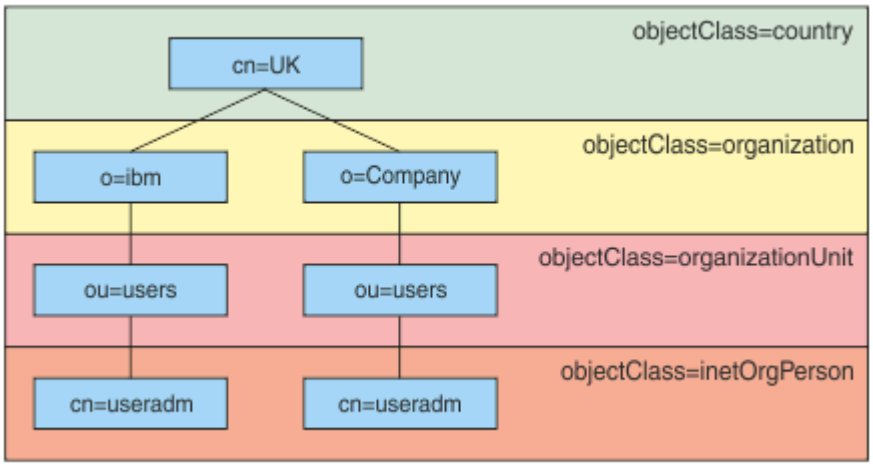


図 *8. LDAP* 階層の例

例えば、73 ページの図 8 では、BASEDNU を "ou=users,o=ibm,c=UK" または ",o=ibm,c=UK" に設定できます。 ただし、"cn=useradm" を含む識別名は "o=ibm" ブランチと "o=Company" ブランチの両方に存在するため、BASEDNU を "c=UK" に設定できません。 パフォーマンスおよびセキュリティー上の理由で、必要なユーザー ID すべてを参照できる LDAP 階層の最も高いポイントを使用します。 この例では、"ou=users,o=ibm,c=UK" がそれに該当します。

アプリケーションによっては、キュー・マネージャーに LDAP 属性名 (CN= など) を付けずにユーザー ID を送信する場合があります。 LDAP 属性名に USRFIELD を設定すると、アプリケーションから受信したユーザー ID に接頭部としてこの値が追加されます。 この機能は、オペレーティング・システムのユーザー ID から LDAP ユーザー ID に移行する際に役立つ場合があります。アプリケーションではこれら両方を同じストリングで提示できるため、アプリケーションを変更せずに済みます。

LDAP サーバーに提示される完全なユーザー ID は次のようになります。

```
USRFIELD = ID_from_application BASEDNU
```

関連概念

65 ページの『接続認証』

66 ページの『接続認証: 構成』
リソースに対するアクセス権限がユーザーにあるかどうかを、提供されたユーザー ID とパスワードを使用して検査するようにキュー・マネージャーを構成することができます。

70 ページの『接続認証: アプリケーションの変更』

ユーザー *ID* とパスワードを挿入するためのクライアント・サイドのセキュリティー出口 *(mqccred)*

ユーザー ID とパスワードを送信するために必要なクライアント・アプリケーションがあるが、まだソースを変更できない場合、IBM MQ 8.0 に付属の **mqccred** というセキュリティー出口を使用できます。
mqccred によって、クライアント・アプリケーションの代わりに `.ini` ファイルからユーザー ID とパスワードが提供されます。 このユーザー ID とパスワードはキュー・マネージャーに送信され、そこで認証が行われます (そのように構成した場合)。

概要

mqccred は、ご使用のクライアント・アプリケーションと同じマシン上で実行されるセキュリティー出口です。 これを使用すると、ユーザー ID とパスワードの情報がクライアント・アプリケーション自体では提供されない場合に、そのアプリケーションの代わりに提供できるようになります。 ユーザー ID とパスワードの情報は、接続セキュリティー・パラメーター (MQCSP) という構造で提供され、接続認証が構成されていればキュー・マネージャーによって認証されます。

ユーザー ID とパスワードの情報は、クライアント・マシンにある `.ini` ファイルから取り出されます。 このファイルにあるパスワードは、**runmqccred** コマンドを使用して難読化することによって、また、クライアント・アプリケーション (およびその出口) を実行しているユーザー ID のみが読み取り可能になるように `.ini` ファイルのファイル・アクセス権を設定することによって、保護されています。

ロケーション

mqccred は以下の場所にインストールされています。

Windows プラットフォーム
installation_directory\Tools\c\Samples\mqccred\ ディレクトリー内

UNIX プラットフォーム
installation_directory/samp/mqccred ディレクトリー内

注 : この出口には以下の特長があります。

1. 純粋にセキュリティー・チャネル出口として動作します。チャネルに対して定義されているそのような唯一の出口である必要があります。
2. 通常、クライアント・チャネル定義テーブル (CCDT) によって指定されますが、Java クライアントが出口を JNDI オブジェクトに直接指定したり、手動で MQCD 構造を構築するアプリケーション用に出口が構成されたりする場合もあります。
3. **mqccred** プログラムと **mqccred_r** プログラムを `var/mqm/exits` ディレクトリーにコピーする必要があります。

   例えば、64 ビットの UNIX プラットフォーム・マシンでは、以下のコマンドを実行します。

   ```
   cp installation_directory/samp/mqccred/lib64/* /var/mqm/exits
   ```

   詳細については、mqccred のテスト方法を示すステップごとの例を参照してください。
4. 以前のバージョンの IBM MQ (IBM WebSphere MQ 7.0.1 以降) で実行できます。

ユーザー ID とパスワードのセットアップ

.ini ファイルには、キュー・マネージャーごとのスタンザと、キュー・マネージャーの指定なしのグローバル設定が含まれています。 各スタンザには、キュー・マネージャーの名前、ユーザー ID、およびプレーン・テキストまたは難読化されたパスワードが含まれています。

.ini ファイルは、任意のエディターを使用して手動で編集し、スタンザにプレーン・テキストのパスワード属性を追加する必要があります。 提供されている **runmqccred** プログラムを実行します。このプログラムは、 .ini ファイルを取得して、**Password** 属性を **OPW** 属性 (難読化された形式のパスワード) に置換します。

コマンドおよびそのパラメーターの説明については、runmqccred を参照してください。

mqccred.ini ファイルには、ユーザー ID とパスワードの情報が含まれています。

出口と同じディレクトリーに、自社での開始点となるテンプレート .ini ファイルが用意されています。

デフォルトでは、このファイルは $HOME/.mqs/mqccred.ini で検索されます。 このファイルを他の場所に置く場合は、以下のように環境変数 *MQCCRED* を使用してその場所を指します。

```
MQCCRED=C:\mydir\mqccred.ini
```

MQCCRED を使用する場合、この変数に構成ファイルの絶対パス名 (.ini ファイル・タイプを含む) を設定する必要があります。 このファイルにはパスワード (難読化されている場合でも) が含まれているため、許可されていないユーザーが読み取ることができないように、オペレーティング・システムの特権を使用してファイルを保護することをお勧めします。 適切なファイル・アクセス権がないと、出口は正常に実行されません。

アプリケーションが既に MQCSP 構造を提供している場合、通常、出口はそれを順守し、.ini ファイルからの情報を挿入しません。 ただし、スタンザの **Force** 属性を使用して、これを指定変更することはできます。

Force を値 *TRUE* に設定すると、アプリケーション提供のユーザー ID とパスワードは除去され、ini ファイル内のユーザー ID とパスワードに置き換えられます。

また、**Force** 属性をファイルのグローバル・セクションで設定することで、そのファイルのデフォルト値を設定することができます。

Force のデフォルト値は *FALSE* です。

すべてのキュー・マネージャー、または個々のキュー・マネージャーに対して、ユーザー ID とパスワードを指定できます。 以下に、mqccred.ini ファイルの例を示します。

```
# comments are permitted
AllQueueManagers:
User=abc
OPW=%^&aervrgtsr

QueueManager:
Name=QMA
User=user1
OPW=H&^dbgfh

Force=TRUE

QueueManager:
Name=QMB
User=user2
password=passw0rd
```

注:

1. 個々のキュー・マネージャー定義は、グローバル設定より優先されます。
2. 属性は大/小文字を区別しません。

制約事項

この出口が使用中である場合、アプリケーションを実行しているユーザーのローカル・ユーザー ID は、クライアントからサーバーに渡されません。 使用可能な ID 情報は、ini ファイルの内容からの情報のみです。

したがって、**ADOPTCTX(YES)** を使用するか、いずれかの使用可能なメカニズム (例えば 47 ページの『チャネル認証レコード』) によってインバウンド接続要求を適切なユーザー ID にマップするように、キュー・マネージャーを構成する必要があります。

重要 : 新規パスワードを追加するか、古いパスワードを更新する場合、**runmqccred** コマンドはプレーン・テキストのパスワードを処理するだけで、難読化されたパスワードは処理しません。

デバッグ

この出口は、標準 IBM MQ トレースが使用可能になるとそこに書き込みを行います。

構成上の問題をデバッグできるように、出口は直接標準出力に書き込むこともできます。

通常は、チャネルに関して、チャネル・セキュリティー出口データ (**SCYDATA**) 構成は必要ありません。 ただし、以下を指定することはできます。

エラー
構成ファイルを見つけられなかった場合などの、エラー状態に関する情報のみ表示します。

DEBUG
これらのエラー状態と、追加のトレース・ステートメントの一部を表示します。

NOCHECKS
ファイル・アクセス権に関する制約や、保護されていないパスワードを `.ini` ファイルに含めてはならないというより詳細な制約を迂回します。

これらのエレメントの 1 つ以上をコンマで区切って、順不同で **SCYDATA** フィールドに入れることができます。 例えば、`SCYDATA=(NOCHECKS,DEBUG)`です。

これらの項目は大/小文字を区別し、大文字で入力する必要があることに注意してください。

mqccred の使用

ファイルをセットアップすると、以下のように `SCYEXIT('mqccred(ChlExit)')` 属性を含めるようにクライアント接続チャネル定義を更新することによって、チャネル出口を呼び出すことができます。

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

関連資料
SCYDATA
SCYEXIT
runmqccred

Java クライアントを使用した接続認証

接続認証は、キュー・マネージャーを、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するように構成できる、IBM MQ のフィーチャーです。 アプリケーションがクライアント・バインディングを使用する Java アプリケーションである場合、接続認証は互換モードまたは MQCSP 認証モードで実行できます。

互換モード

IBM MQ 8.0 より前には、Java クライアントは、クライアント接続チャネルを介してユーザー ID とパスワードをサーバー接続チャネルに送信し、MQCD 構造体の **RemoteUserIdentifier** フィールドおよび

RemotePassword フィールドでそれらをセキュリティー出口に提供できました。 互換モードには、この動作が残っています。

このモードを接続認証と組み合わせて使用して、以前同じジョブを実行するために使用されていたセキュリティー出口から移行できます。

互換モードを使用している場合は、ADOPTCTX(YES) を使用するか、TLS 証明書に基づく CHLAUTH ルールなどの別の方法で、実行中の MCAUSER を設定する必要があります。これは、互換モードでは、クライアント・サイドのユーザー ID がキュー・マネージャーに送信されないためです。

互換モードの操作は、個々の接続ベースまたはグローバルで有効にすることができます。

- IBM MQ classes for Java では、 **com.ibm.mq.MQQueueManager** コンストラクターに渡されるプロパティー・ハッシュ・テーブルのプロパティー *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* を `false` に設定します。
- IBM MQ classes for JMS では、接続を作成する前に、適切な接続ファクトリーで、プロパティー *JmsConstants.USER_AUTHENTICATION_MQCSP* を `false` に設定します。
- グローバルに、アプリケーションの開始時にコマンド行で Java システム・プロパティー *-Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication= N* を指定します。以下に例を示します。

```
java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
```

互換モードはデフォルトの設定です。

MQCSP 認証モード

このモードでは、クライアント・サイドのユーザー ID は、認証されるユーザー ID とパスワードと同様に送信されるので、ADOPTCTX(NO) を使用できます。 ユーザー ID とパスワードは、MQCXP 構造で提供される MQCSP 構造のサーバー接続セキュリティー出口で使用できます。

この操作モードは、個々の接続ベースまたはグローバルで有効にすることができます。

- IBM MQ classes for Java では、 **com.ibm.mq.MQQueueManager** コンストラクターに渡されるプロパティー・ハッシュ・テーブルのプロパティー *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* を `true` に設定します。
- IBM MQ classes for JMS では、接続を作成する前に、適切な接続ファクトリーで、プロパティー *JmsConstants.USER_AUTHENTICATION_MQCSP* を `true` に設定します。
- グローバルに、システム・プロパティー *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* を (例えば `-Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=Y` をコマンド行に追加することによって) true を示す値に設定します。

IBM MQ Explorer での認証モードの選択

IBM MQ Explorer は Java アプリケーションであるため、互換モードと MQCSP 認証モードの 2 つのモードを同様に使用できます。

V 9.1.0 IBM MQ 9.1.0 以降では、MQCSP 認証モードがデフォルトです。 IBM MQ 9.1 より前では、互換モードがデフォルトです。

ユーザー ID を指定するパネルに、互換モードを有効化または無効化するチェック・ボックスがあります。

- V 9.1.0 IBM MQ 9.1.0 から、デフォルトでは、このチェック・ボックスは選択されていません。 互換モードを使用するには、このチェック・ボックスを選択します。
- IBM MQ 9.1.0 より前では、デフォルトで、このチェック・ボックスは有効になっています。 MQCSP 認証を使用するには、チェック・ボックスをクリアします。

関連概念

65 ページの『接続認証』

70 ページの『接続認証: アプリケーションの変更』

71 ページの『接続認証: ユーザー・リポジトリー』
使用している各キュー・マネージャーに対して、ユーザー ID とパスワードの認証のために、異なるタイプの認証情報オブジェクトを選択できます。

IBM MQ でのメッセージ・セキュリティー

IBM MQ インフラストラクチャーでのメッセージ・セキュリティーは、Advanced Message Security によって提供されます。

Advanced Message Security (AMS) は、IBM 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
559 ページの『Advanced Message Security』
Advanced Message Security (AMS) は、IBM 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 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。

セキュリティー要件の計画

このトピック集では、IBM 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 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。

IBM MQ は、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。

IBM MQ のインストールを計画する時に、セキュリティーに関する事柄をいくつか検討する必要があります。

- Multi マルチプラットフォーム では、これらの局面を無視して何も対処しない場合、IBM MQ を使用できません。
- z/OS z/OS では、これらの局面を無視した場合の影響は、IBM MQ リソースが無保護になることです。 つまり、すべてのユーザーがすべての IBM MQ リソースにアクセスして変更できるようになります。

IBM MQ を管理する権限

IBM MQ 管理者には、次の権限が必要です。

- IBM MQ を管理するためのコマンドを実行する権限
- IBM MQ Explorer を使用する権限
- IBM i IBM i 管理パネルおよびコマンドを使用する権限
- z/OS z/OS で、操作と制御パネルを使用する権限
- z/OS IBM MQ ユーティリティー・プログラム CSQUTIL を z/OS で使用する
- z/OS z/OS で、キュー・マネージャーのデータ・セットにアクセスする権限

詳細については、次の章を参照してください。

- ULW 399 ページの『UNIX, Linux, and Windows 上の IBM MQ を管理する権限』
- IBM i 83 ページの『IBM i 上の IBM MQ を管理する権限』
- z/OS 84 ページの『z/OS 上の IBM MQ を管理する権限』

IBM 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 IBM 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 IBM 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 IBM 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。

詳しくは、86 ページの『アプリケーションで IBM MQ を使用するための権限』 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 IBM 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。

詳しくは、107 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、IBM 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。

- 120 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』
- 121 ページの『IBM MQ Publish/Subscribe のセキュリティー』
- 122 ページの『IBM 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 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。 TLS を使用する IBM 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 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。

- 383 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』
- 384 ページの『MCAUSER ユーザー ID へのユーザー ID のマッピング』
- 385 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』
- 387 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。 図は、サーバーに接続された IBM MQ MQI client 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。

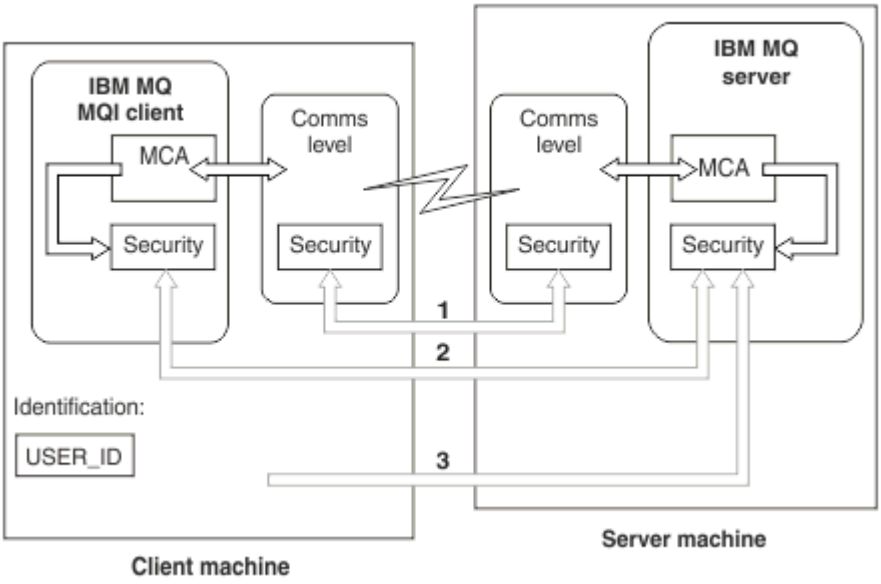


図 9. クライアント/サーバー間接続のセキュリティー

1. 通信レベル

   矢印 1 を参照。 セキュリティーを通信レベルで実装するには、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: 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 TLS 識別名を使用して制御することができます。 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、47 ページの『チャネル認証レコード』 にあります。

3. 接続認証

   矢印 3 を参照。 クライアントは ID とパスワードを送信します。 詳しくは、66 ページの『接続認証: 構成』 を参照してください。

4. 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。

5. 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 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。

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 MQ MQI client 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。

IBM MQ 8.0 以降、パスワードは、MQCSP 構造に組み込んで送信することができます。

警告 : クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを経由してプレーン・テキストで送信される場合があります。 クライアント・アプリケーションのパスワードが適切に 保護されるようにするには、28 ページの『MQCSP パスワード保護』を参照してください。

ユーザー *ID*

クライアント・アプリケーションのユーザー ID を作成するときに、ユーザー ID は許容最大長を超えてはなりません。 また、予約済みユーザー ID である UNKNOWN と NOBODY は使用できません。 クライアントが接続するサーバーが IBM MQ for Windows サーバーである場合は、アットマーク (@) の使用をエスケープする必要があります。許可されるユーザー ID の長さは、サーバに使用されるプラットフォームによって異なります。

- z/OS Linux UNIX z/OS および UNIX and Linux では、ユーザー ID の最大長は 12 文字です。
- IBM i IBM i では、ユーザー ID の最大長は 10 文字です。
- Windows Windows では、IBM MQ MQI client と IBM MQ サーバーの両方が 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へのアクセス権限がサーバーにある場合、ユーザー ID の最大長は 20 文字です。 ただし、IBM MQ サーバーが Windows サーバーではない場合、ユーザー ID は 12 文字に切り捨てられます。
- MQCSP 構造を使用して資格情報を渡す場合は、ユーザー ID の最大長は 1024 文字です。 MQCSP 構造ユーザー ID を使用して、IBM MQ が許可に使用するユーザー ID の最大長を回避することはできません。 MQCSP 構造の詳細については、331 ページの『MQCSP 構造を使用したユーザーの識別および認証』を参照してください。

UNIX and Linux システムでは、デフォルトではユーザー ID が認証に使用され、グループは許可に使用されます。 ただし、ユーザー ID に対して許可するようにこれらのシステムを構成することができます。 詳しくは、348 ページの『UNIX and Linux での OAM ユーザーに基づく許可』を参照してください。 Windows システムは、認証と許可の両方にユーザー ID を使用し、許可にグループを使用することができます。

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。

制限付きユーザー ID

ユーザー ID UNKNOWN およびグループ NOBODY は、IBM MQ に対して特別な意味を持ちます。 UNKNOWN というオペレーティング・システムまたは NOBODY というグループでユーザー ID を作成すると、意図しない結果になる可能性があります。

IBM MQ for Windows サーバーへの接続時のユーザー ID

Windows

IBM 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。

許可の計画

管理権限を持つユーザーや、IBM 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 MQ MQI client 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 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 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
- アプリケーションで IBM MQ を使用するための権限

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 MQ を管理する権限

IBM 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 MQ 管理者には、次の権限が必要です。

- IBM MQ を管理するためのコマンドを実行する権限。
- Windows Linux IBM MQ Explorer を使用する。
- z/OS z/OS で、操作と制御パネルを使用する権限。
- z/OS z/OS では、 IBM MQ ユーティリティー・プログラム CSQUTIL を使用します。
- z/OS z/OS で、キュー・マネージャーのデータ・セットにアクセスする権限。

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。

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

IBM 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 MQ リソースにアクセスして、IBM 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 システムおよび Windows システムで IBM 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、IBM 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 でのルート・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 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に IBM MQ リソースへの全アクセス権限があります。

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 IBM 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 システムでは、管理者グループのメンバーも、すべての IBM MQ リソースに対するアクセス権限を持ちます。

管理者は、**runmqsc** 制御コマンドを使用して、IBM 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。

IBM MQ Explorer 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、IBM MQ Explorer 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 MQ Explorer 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、86 ページの『アプリケーションで IBM MQ を使用するための権限』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。
- z/OS IBM MQ for z/OS 上のコマンド・サーバーによって処理される MQSC コマンドについては、84 ページの『z/OS でのコマンド・セキュリティーとコマンド・リソース・セキュリティー』を参照してください。

UNIX および Windows システム上の IBM MQ を管理するために必要な権限について詳しくは、関連情報を参照してください。

IBM i *IBM i 上の IBM MQ を管理する権限*

IBM i で IBM MQ 管理者になるには、 *QMQMADM* グループのメンバーでなければなりません。 このグループには、UNIX システムおよび Windows システムの mqm グループと同様のプロパティーがあります。 特に、IBM MQ for IBM i のインストール時に QMQMADM グループが作成され、その QMQMADM グループのメンバーには、システム上の IBM MQ のすべてのリソースに対するアクセス権限が付与されます。 *ALLOBJ 権限がある場合は、すべての IBM MQ リソースにもアクセスできます。

管理者は、IBM MQ を管理する CL コマンドを使用できます。それらのコマンドの 1 つに GRTMQMAUT がありますが、これは他のユーザーに権限を付与するために使用されるものです。 別のコマンド STRMQMMQSC は、管理者がローカル・キュー・マネージャーに対して MQSC コマンドを発行するためのものです。

IBM MQ for IBM i によって提供される CL コマンドは 2 つのグループに分かれます。

グループ 1
このカテゴリーのコマンドを発行するユーザーは、QMQMADM グループのメンバーであるか *ALLOBJ 権限を持っていなければなりません。 例えば、このカテゴリーには GRTMQMAUT や STRMQMMQSC が属します。

グループ 2
このカテゴリーのコマンドを発行するユーザーは、QMQMADM グループのメンバーである必要も *ALLOBJ 権限を持っている必要もありません。 代わりに、次の 2 レベルの権限が必要です。

- このコマンドを使用するための IBM i 権限がユーザーに必要です。 この権限を付与するには、GRTOBJAUT コマンドを使用します。
- このコマンドに関連した IBM MQ オブジェクトにアクセスするための IBM MQ 権限がユーザーに必要です。 この権限を付与するには、GRTMQMAUT コマンドを使用します。

このグループのコマンドの例を次に示します。

- CRTMQMQ (MQM キューの作成)
- CHGMQMPRC (MQM プロセスの変更)
- DLTMQMNL (MQM 名前リストの削除)
- DSPMQMAUTI (MQM 認証情報の表示)
- CRTMQMCHL (MQM チャネルの作成)

このグループのコマンドの詳細については、86 ページの『アプリケーションで IBM MQ を使用するための権限』を参照してください。

グループ 1 およびグループ 2 のコマンドの完全なリストについては、154 ページの『IBM i 上の IBM MQ オブジェクトのアクセス権限』を参照してください。

IBM i 上の IBM MQ を管理するために必要な権限について詳しくは、 IBM i の管理 を参照してください。

z/OS *z/OS 上の IBM MQ を管理する権限*

このトピック集では、IBM MQ for z/OS を管理するために必要な権限のさまざまな側面を取り上げます。

z/OS *z/OS 上の権限検査*

IBM MQ for z/OS は、システム許可機能 (SAF) を使用して、権限検査の要求を、z/OS セキュリティー・サーバー・リソース・アクセス制御機能 (RACF)などの外部セキュリティー・マネージャー (ESM) に経路指定します。IBM MQ は、それ自体の権限検査を行いません。

RACF を ESM として使用することを前提としています。 別の ESM を使用する場合は、ご使用の ESM に適した方法で、RACF 用に記載されている情報を解釈する必要があります。

キュー・マネージャーごとに個別に、またはキュー共有グループ内のすべ てのキュー・マネージャーに対して、権限検査をオンにするか、オフにするか を指定できます。このレベルの制御は、サブシステム・セキュリティー と呼ばれます。 特定のキュー・マネージャーに対してサブシステム・セキュリティーをオフにする場合、そのキュー・マネージャーに対する権限検査は行われません。

特定のキュー・マネージャーに対してサブシステム・セキュリティーをオンにする場合は、権限検査を次の 2 つのレベルで実行することがでます。

キュー共用グループ・レベルのセキュリティー
権限検査では、キュー共有グループ内のすべてのキュー・マネージャーによって共有される RACF プロファイルを使用します。 これは、定義し、保持するプロファイルが少なくなり、セキュリティーの管理が簡単になることを意味します。

キュー・マネージャー・レベルのセキュリティー
権限検査では、キュー・マネージャーに固有の RACF プロファイルを使用します。

キュー共有グループ・レベル・セキュリティーとキュー・マネージャー・レベル・セキュリティーを組み合わせて使用できます。 例えば、あるキュー・マネージャーに固有のプロファイルが、そのキュー・マネージャーが属するキュー共有グループのプロファイルを指定変更するように調整できます。

サブシステム・セキュリティー、キュー共有グループ・レベル・セキュリティー、およびキュー・マネージャー・レベル・セキュリティーをオンまたはオフにするには、スイッチ・プロファイル を定義します。スイッチ・プロファイルは、 IBM MQ に対して特別な意味を持つ通常の RACF プロファイルです。

z/OS *z/OS でのコマンド・セキュリティーとコマンド・リソース・セキュリティー*

コマンド・セキュリティーは、コマンドを実行する権限に関連しており、コマンド・リソース・セキュリティーは、リソース上で操作を実行する権限に関連しています。 コマンド・セキュリティーとコマンド・リソース・セキュリティーは両方とも、RACF クラスを使用して実装されます。

IBM MQ 管理者が MQSC コマンドを発行すると、権限検査が実行されます。 これは、コマンド・セキュリティー と呼ばれます。

コマンド・セキュリティーをインプリメントするには、特定の RACF プロファイルを定義し、必要なレベルでこれらのプロファイルへのアクセス権を、必要なグループとユーザー ID に与える必要があります。 コマンド・セキュリティー用のプロファイルの名前には、MQSC コマンドの名前が含まれます。

一部の MQSC コマンドは、IBM MQ リソースを対象として操作を実行します (例えば、ローカル・キューを作成するための DEFINE QLOCAL コマンド)。 管理者が MQSC コマンドを発行すると、権限検査が実行され、コマンドで指定されたリソースに対して要求された操作を実行できるかどうかが判別されます。 これは、コマンド・リソース・セキュリティー と呼ばれます。

コマンド・リソース・セキュリティーをインプリメントするには、特定の RACF プロファイルを定義し、必要なレベルでこれらのプロファイルへのアクセス権を、必要なグループとユーザー ID に与える必要があります。 コマンド・リソース・セキュリティー用のプロファイルの名前には、IBM MQ リソースの名前およびそのタイプ (QUEUE、PROCESS、NAMELIST、TOPIC、AUTHINFO、または CHANNEL) が含まれています。

コマンド・セキュリティーとコマンド・リソース・セキュリティーとは、独立した別個のものです。 例えば、管理者が次のコマンドを発行するとします。

```
DEFINE QLOCAL(MOON.EUROPA)
```

この場合、次の権限検査が実行されます。

- DEFINE QLOCAL コマンドを発行する権限が管理者に与えられているかどうかのコマンド・セキュリティー検査
- MOON.EUROPA と呼ばれるローカル・キューに対して操作を実行する権限が、管理者にあるかどうかのコマンド・リソース・セキュリティー検査

コマンド・セキュリティーとコマンド・リソース・セキュリティーは、スイッチ・プロファイルの定義により、オンまたはオフにすることができます。

z/OS *MQSC* コマンドとシステム・コマンド入力キュー *(z/OS)*

このトピックでは、z/OS のシステム・コマンド入力キューに送信された MQSC コマンドをコマンド・サーバーが処理する方法を取り上げます。

コマンド・セキュリティーとコマンド・リソース・セキュリティーは、コマンド・サーバーが、システム・コマンド入力キューから、MQSC コマンドが入っているメッセージを取り出すときにも使用されます。権限検査に使用されるユーザー ID は、MQSC コマンドが入っているメッセージ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。 *UserIdentifier* フィールドとその設定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。

MQSC コマンドが入っているメッセージは、次の状況のもとでシステム・コマンド入力キューに送信されます。

- 操作と制御パネルは、ターゲット・キュー・マネージャーのシステム・コマンド入力キューに、MQSC コマンドを送信します。 MQSC コマンドは、パネル上で管理者が選択するアクションに対応しています。 各メッセージ内の *UserIdentifier* フィールドは、管理者の TSO ユーザー ID に設定されます。
- IBM MQ ユーティリティー・プログラム CSQUTIL の COMMAND 機能は、入力データ・セット内の MQSC コマンドを、ターゲット・キュー・マネージャーのシステム・コマンド入力キューに送信します。 COPY 機能と EMPTY 機能は、DISPLAY QUEUE コマンドと DISPLAY STGCLASS コマンドを送信します。 各メッセージ内の *UserIdentifier* フィールドは、ジョブ・ユーザー ID に設定されます。
- CSQINPX データ・セット内の MQSC コマンドは、チャネル・イニシエーターが接続されているキュー・マネージャーのシステム・コマンド入力キューに送信されます。 各メッセージ内の *UserIdentifier* フィールドは、チャネル・イニシエーターのアドレス・スペース・ユーザー ID に設定されます。

  CSQINP1 および CSQINP2 データ・セットから MQSC コマンドが発行される場合、権限検査は実行されません。 RACF データ・セット保護を使用して、これらのデータ・セットを更新できる人物を管理できます。
- キュー共有グループ内で、チャネル・イニシエーターは、接続されているキュー・マネージャーのシステム・コマンド入力キューに、START CHANNEL コマンドを送信する場合があります。 共有伝送キューを使用するアウトバウンド・チャネルが、トリガーによって開始されるときに、コマンドが送信されます。 各メッセージ内の *UserIdentifier* フィールドは、チャネル・イニシエーターのアドレス・スペース・ユーザー ID に設定されます。
- アプリケーションは、システム・コマンド入力キューに MQSC コマンドを送信できます。 デフォルトでは、各メッセージ内の *UserIdentifier* フィールドは、アプリケーションに関連付けられているユーザー ID に設定されます。
- UNIX, Linux, and Windows システムでは、 **runmqsc** 制御コマンドを間接モードで使用して、 z/OS 上のキュー・マネージャーのシステム・コマンド入力キューに MQSC コマンドを送信することができます。 各メッセージ内の *UserIdentifier* フィールドは、**runmqsc** コマンドを発行した管理者のユーザー ID に設定されます。

z/OS キュー・マネージャーのデータ・セットへのアクセス権限 *(z/OS)*

IBM MQ for z/OS 管理者には、キュー・マネージャー・データ・セットにアクセスするための権限が必要です。 このトピックでは、どのデータ・セットで RACF 保護が必要になるかを知ることができます。

これらのデータ・セットには、次のものがあります。

- V 9.1.0 キュー・マネージャーの開始タスク・プロシージャーの CSQINP1、CSQINP2、CSQINPT で参照されるデータ・セット。

- キュー・マネージャーのページ・セット、アクティブ・ログ・データ・セット、アーカイブ・ログ・データ・セット、およびブートストラップ・データ・セット (BSDS)
- チャネル・イニシエーターの開始タスク・プロシージャーで CSQXLIB および CSQINPX によって参照されるデータ・セット

無許可のユーザーが、キュー・マネージャーを開始したり、キュー・マネージャーの任意のデータにアクセスしたりすることができないように、データ・セットを保護する必要があります。 データ・セットを保護するには、RACF データ・セット保護を使用してください。

アプリケーションで IBM MQ を使用するための権限

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 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 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、および IBM MQ for z/OS に付属のアプリケーションが含まれます。 IBM MQ for z/OS に付属のアプリケーションには、次のものがあります。

- 操作と制御パネル
- IBM MQ ユーティリティー・プログラム CSQUTIL
- 送達不能キュー・ハンドラー・ユーティリティー CSQUDLQH

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

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 IBM 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、107 ページの『チャネル許可』を参照してください。

z/OS 上で、アプリケーションは、MQSC コマンドを使用して、これらの IBM MQ オブジェクトにアクセスすることもできますが、コマンド・セキュリティーとコマンド・リソース・セキュリティーは、このような状況で、権限検査を提供します。 z/OS 詳細については、84 ページの『z/OS でのコマンド・セキュリティーとコマンド・リソース・セキュリティー』および 85 ページの『MQSC コマンドとシステム・コマンド入力キュー (z/OS)』を参照してください。

IBM i では、グループ 2 の CL コマンドを発行するユーザーに対して、コマンドに関連する IBM MQ オブジェクトにアクセスするための権限が必要な場合があります。 詳しくは、86 ページの『権限検査が実行される場合』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。

IBM i では、ユーザーがこれらの IBM MQ オブジェクトにアクセスするグループ 2 の CL コマンドを発行するときにも権限検査が実行される場合があります。 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 MQ にサインオンする必要はありません。 IBM 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 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。 z/OS では、リモート・キュー・マネージャー名と一致する RACF キュー・プロファイルに対して検査が行われます。 マルチプラットフォーム 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チェックが含まれていることがあります。 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。

IBM 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、89 ページの『メッセージ・コンテキスト』を参照してください。

IBM i **IBM i では、ユーザーが IBM MQ オブジェクトを操作するグループ 2 の CL コマンドを発行するとき。**

このルールには、グループ 2 の CL 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。

コマンドに関連付けられている IBM MQ オブジェクトを操作する権限をユーザーが持っているかどうかを判別するための検査が実行されます。 この検査は、ユーザーが QMQMADM グループのメンバーである場合または *ALLOBJ 権限を持っている場合を除いて、実行されます。 必要な権限は、コマンドがオブジェクトに対して行う操作の種類によって決まります。 例えば、コマンド **CHGMQMQ** (MQM キューの変更) では、コマンドで指定されたキューの属性を変更する権限が必要です。 これに対し、コマンド **DSPMQMQ** (MQM キューの表示) では、コマンドで指定されたキューの属性を表示する権限が必要です。

多くのコマンドは複数のオブジェクトを操作します。 例えば、コマンド **DLTMQMQ** (MQM キューの削除) を発行するには、次の権限が必要です。

- コマンドで指定されたキュー・マネージャーに接続する権限
- コマンドで指定されたキューを削除する権限

一部のコマンドはまったくオブジェクトを操作しません。 この場合、ユーザーは、これらのコマンドのいずれかを発行するために IBM i 権限のみを必要とします。 **STRMQMLSR**、 そのようなコマンドの例として、MQM リスナーを開始します。

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。

ULW IBM i IBM i *IBM i 、 UNIX, Linux, and Windows システムで IBM MQ オブジェクトを処理する権限*

IBM 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。

認証。
IBM 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 は、アプリケーションがキュー・マネージャーに接続するときに、その認証検査を実行しますが、以下の条件が該当します。

- MQCSP 構造体が接続アプリケーションによって提供されている場合、および
- MQCSP 構造体の「*AuthenticationType*」属性には、値 MQCSP_AUTH_USER_ID_AND_PWD が与えられます。
- 構成された AUTHINFO オブジェクトの CHCKLOCL または CHKCCLNT 値が「NONE」ではありません。

OAM の認証ステップでは、オペレーティング・システム・サービスを使用してパスワードを検証します。オペレーティング・システム・サービスは、ユーザー名の誤ったパスワード・テスト試行回数が多すぎないようにするなどの追加検査を実行するように構成されている可能性があります。

新しい許可サービス・コンポーネントを作成する場合、またはベンダーから 1 つ取得する場合は、代替認証メカニズムを使用することができます。

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。

IBM i IBM i 、 UNIX, Linux, and Windows システムでは、 許可サービス 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 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きに、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 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 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 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 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。

IBM i IBM i では、ユーザーが QMQMADM グループのメンバーでもなく、*ALLOBJ 権限も持たない場合は、認証サービスは、これらの IBM MQ オブジェクトまたは認証情報オブジェクトを操作するグループ 2 の CL コマンドをユーザーが発行するときにも権限検査を行います。

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 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 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 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。 IBM i IBM i および 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® および Linux ではこの制限を上げていますが、IBM MQ では引き続きすべての UNIX プラットフォーム上で 12 文字という制限が課されています。 12 文字を超えるユーザー ID を使用すると、IBM 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 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。

警告 : クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを経由してプレーン・テキストで送信される場合があります。 クライアント・アプリケーションのパスワードが適切に保護されるようにするには、IBM MQCSP パスワード保護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
```

このコマンドにより、グループ VOYAGER のメンバーは、名前が文字 MOON. で始まる任意のキューにメッセージを入れることができます。 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。

IBM i IBM i では、管理者は、CL コマンド GRTMQMAUT を使用して権限を付与し、CL コマンド RVKMQMAUT を使用して権限を取り消します。 総称プロファイルも使用できます。例えば、次の CL コマンドがあるとします。

```
GRTMQMAUT MQMNAME(JUPITER) OBJTYPE(*Q) OBJ('MOON.*') USER(VOYAGER) AUT(*PUT)
```

このコマンドは 前記の **setmqaut** コマンドと同じ機能を提供します。このコマンドにより、グループ VOYAGER のメンバーは、名前が文字 MOON. で始まる任意のキューにメッセージを入れることができます。

IBM i CL コマンド DSPMQMAUT は、指定されたオブジェクトに対してユーザーまたはグループが持つ 現在の権限を表示します。 CL コマンド WRKMQMAUT および WRKMQMAUTD は、オブジェクトおよび総称プロファイルに関連した現在の権限も処理できます。

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。

Multi *PCF* を使用しての *OAM* コマンドへのアクセス

IBM i、 UNIX, Linux, and 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 *8. 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法について詳しくは、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。

z/OS *IBM MQ 上で z/OS オブジェクトを処理する権限*

z/OS では、MQI への呼び出しに関連した権限検査が 7 つのカテゴリーに分けられています。 特定の RACF プロファイルを定義して、それらのプロファイルに対する適切なアクセス権を与えなければなりません。 さらに、検査するユーザー ID の数を制御するために、*RESLEVEL* プロファイルを使用します。

MQI への呼び出しに関連した権限検査の 7 つのカテゴリーは、以下のとおりです。

接続のセキュリティー
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実行される権限検査

キュー・セキュリティー
アプリケーションがキューを開くか、永続動的キューを削除するときに実行される権限検査

プロセス・セキュリティー
アプリケーションがプロセス・オブジェクトを開くときに実行される権限検査

名前リスト・セキュリティー
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る権限検査

代替ユーザー・セキュリティー
アプリケーションが、オブジェクトを開くときに代替ユーザー権限を要求する場合に実行される権限検査

コンテキスト・セキュリティー
アプリケーションがキューを開くときに、そのキューに入れるメッセージ内でコンテキスト情報を設定するか渡すことを指定する場合に実行される権限検査

トピック・セキュリティー
アプリケーションがトピックを開くときに実行される権限検査

各カテゴリーの権限検査は、コマンド・セキュリティーとコマンド・リソース・セキュリティーがインプリメントされるのと同じ方法でインプリメントされます。 特定の RACF プロファイルを定義し、必要なレベルでこれらのプロファイルへのアクセス権を、必要なグループとユーザー ID に与える必要があります。 キュー・セキュリティーの場合、アクセスのレベルにより、アプリケーションがキュー上で実行できる操作のタイプが決まります。 コンテキスト・セキュリティーの場合、アクセスのレベルにより、アプリケーションが次の操作を実行できるかどうかが決まります。

- すべてのコンテキスト・フィールドを渡す
- すべてのコンテキスト・フィールドを渡し、identity コンテキスト・フィールドを設定する
- すべてのコンテキスト・フィールドを渡し、設定する

権限検査の各カテゴリーは、スイッチ・プロファイルの定義によってオンまたはオフにすることができます。

接続セキュリティー以外のすべてのカテゴリーは、まとめて *API* リソース・セキュリティー と呼ばれます。

デフォルトでは、バッチ接続を使用したアプリケーションからの MQI 呼び出しの結果、API リソース・セキュリティー検査が実行される場合、1 つのユーザー ID だけが検査されます。 CICS または IMS アプリケーションからの MQI 呼び出しの結果、またはチャネル・イニシエーターからの MQI 呼び出しの結果として検査が実行されると、2 つのユーザー ID が検査されます。

しかし、*RESLEVEL* プロファイル を定義すると、検査されるユーザー ID がゼロか、1 つか、または 2 つかを制御できます。 検査されるユーザー ID の数は、アプリケーションがキュー・マネージャーに接続するときの接続のタイプに関連したユーザー ID、およびそのユーザー ID が RESLEVEL プロファイルに対して持つアクセス・レベルによって決まります。 各タイプの接続に関連したユーザー ID は、次のとおりです。

- バッチ接続用の接続タスクのユーザー ID
- CICS 接続用の CICS アドレス・スペース・ユーザー ID
- IMS 接続用の IMS 領域アドレス・スペース・ユーザー ID
- チャネル・イニシエーター接続用のチャネル・イニシエーター・アドレス・スペース・ユーザー ID

z/OS で IBM MQ オブジェクトを処理する権限について詳しくは、84 ページの『z/OS 上の IBM MQ を管理する権限』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 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 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。

IBM i *IBM MQ for IBM i オブ ジェクトのセキュリティー*

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 MQ for IBM i 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。 この権限は、オブジェクトおよび定義に関して取る処置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もつ。
- 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。

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このリモート・サイトでメッセージを送出する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。

- チャネル定義で、メッセージには受け入れ可能なコンテキスト 権限が入っていなければならず、入っていない場合にはメッセージが廃棄されることを指定する。
- 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 TLS の識別名 (DN)、あるいはリモート・キュー・マネージャー名のいずれかに基づいて MCAUSER 値を設定する。
- 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。
- 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。

IBM MQ for IBM i がセキュリティーを作動さ せるにあたっては、以下のことがあてはまります。

- ユーザーは、IBM i によって識別および承認される。
- アプリケーションによって呼び出されたキュー・マネージャー・サービスは、キュー・マネージャーのユーザー・プロファイルでの認可に基づいて実行されるが、この実行はユーザーのプロセス内で行われる。
- ユーザー・コマンドによって呼び出されたキュー・マネージャー・サービスは、キュー・マネージャーのユーザー・プロファイルでの認可に基づいて実行される。

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

この ID で IBM MQ 管理コマンドを使用する場合は、管理ユーザーが Windows システム上の 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 システムでのユーザー 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には、メッセージ・チャネル・エージェント (MCA) が英字の適切な変換を行う必要があります。

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, and Windows 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 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 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 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 9. 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 97 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 97 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 MQ MQI clients 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。

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 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 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 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 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。

ULW UNIX, Linux, and Windows 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、小文字に変更されます。

IBM i 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。

Linux UNIX 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、IBM 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。 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

98 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。

103 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。

110 ページの『SSL/TLS を使用したチャネルの保護』
IBM MQ の TLS 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、98 ページの図 10 です。

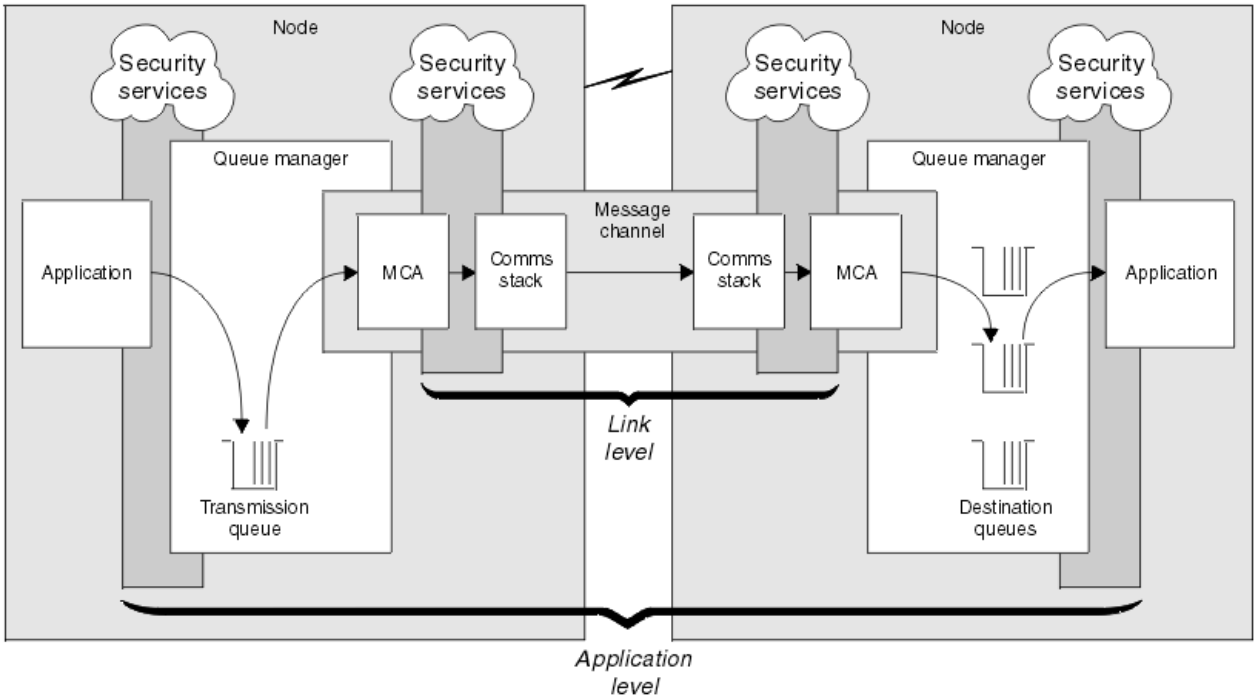


図 *10.* リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティー

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。

V 9.1.4 z/OS z/OS データ・セット暗号化によって、キューに保管されているメッセージをある程度は保護できますが、対象となるのはローカル・キュー・マネージャーにある 保存データのみです。 データ・セット暗号化による IBM MQ for z/OS での保存データの機密性 のセクションを参照してください。 を参照してください。

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、98 ページの図 10 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、IBM 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、IBM 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、98 ページの図 10 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 MQ によって提供されるリンク・レベル・セキュリティー

IBM 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、TLS を使用することです。 IBM MQ での TLS の使い方の詳細については、22 ページの『IBM MQ での TLS セキュリティー・プロトコル』を参照してください。 認証を行うために、IBM 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。 詳細内容は 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

独自のリンク・レベル・セキュリティー・サービスを提供できます。 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。

103 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 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、IBM 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、98 ページの図 10 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*

Advanced Message Security (AMS) は、IBM 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 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 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。 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。

Advanced Message Security は、IBM 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 MQ サーバーとクライアントをどちらもサポートします。
- 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。

詳しくは、559 ページの『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

独自のアプリケーション・レベル・セキュリティー・サービスを提供できます。 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 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。

API 出口および API 交差出口によって、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 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 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、104 ページの図 11 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。

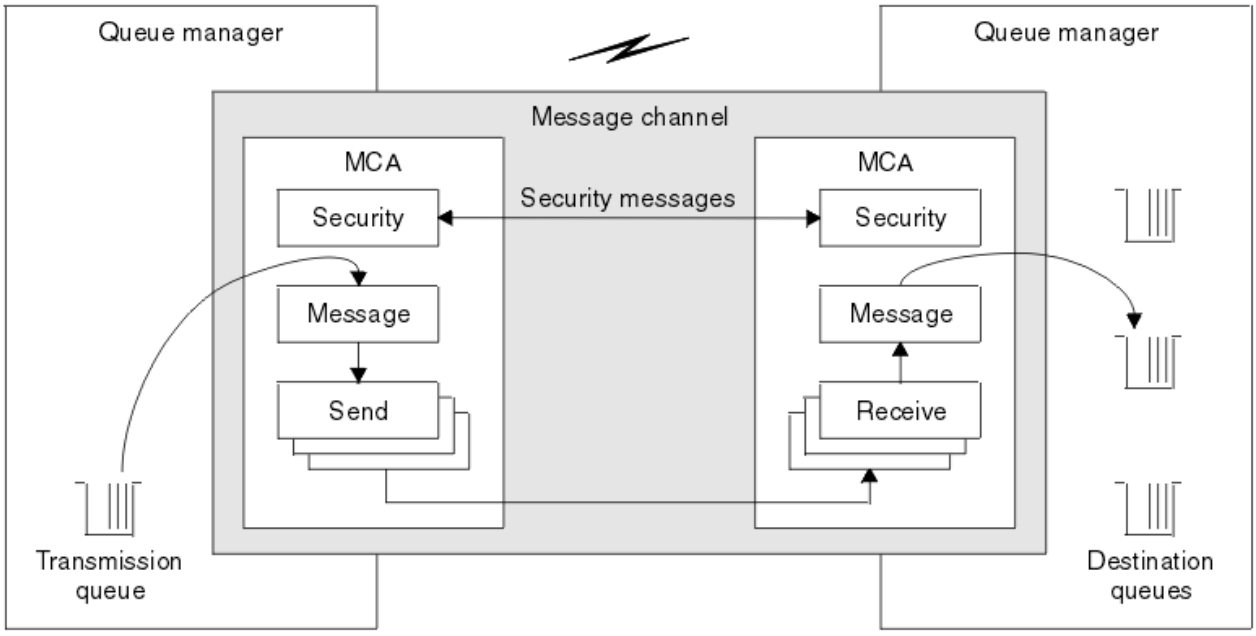


図 *11.* メッセージ・チャネル上のセキュリティー出口、メッセージ出口、送信出口、および受信出口

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、101 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 MQ MQI client・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、101 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 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 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、101 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、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 ができます。 Advanced Message Security (AMS) 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することもできます。

リンク・レベルでは、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

110 ページの『SSL/TLS を使用したチャネルの保護』
IBM MQ の TLS 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 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。

102 ページの『計画 Advanced Message Security』
Advanced Message Security (AMS) は、IBM 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 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。

チャネル出口呼び出しおよびデータ構造体

関連資料

API 出口参照

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 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。 モニター操作については、モニターおよびパフォーマンスのセクションで説明されています。

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 MQ インターフェースを使用して IBM 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 MQ を構成します。

IBM MQ に、制御外で行われた変更
変更によっては、IBM MQ の動作に影響を及ぼす可能性がありますが、IBM 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 MQ の運用制御
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 MQ を装備イベントを発行するように構成できます。

IBM 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 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 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 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 MQ リソースに対するアクセス権を提供する必要があります。 メッセージ・チャネル・エージェント (MCA) はキュー・マネージャー間でメッセージを移動させる基本的な IBM MQ アプリケーションであり、正しく作動するにはさまざまな IBM MQ リソースに対するアクセス権を必要とします。

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。

IBM MQ では、TLS サポートでチャネルを保護できます。

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、IBM 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、特殊権限は必要ありません。
  - 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。
  - 受信側チャネルの MCAUSER ユーザー ID に *dsp* 権限および *ctrlx* 権限を付与する必要はありません。
  - IBM MQ 8.0.0 Fix Pack 4 より前では、IBM MQ 管理特権がないユーザー ID を使用する場合、チャネルが機能するためには、チャネルの **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。

    IBM MQ 8.0.0 Fix Pack 4 以降、チャネルがそれ自体を再同期してシーケンス番号を修正するときの権限検査がなくなりました。

    ただし、RESET CHANNEL コマンドを手動で実行する場合は、すべてのリリースで引き続き **+dsp** および **+ctrlx** が必要です。

    重要 : メッセージ・バッチ確認でチャネルのリセットが必要になる場合、IBM MQ は、チャネルへの照会を実行しようとします。そのためには **+dsp** 権限が必要です。
  - 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。 以下のコマンドは、リモート・システムのユーザー ID に権限を付与するために発行できるコマンド・タイプの例です。

```
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
47 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 MQ チャネル・イニシエーターは IBM 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 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。128 ページの表 12 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、103 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/TLS を使用したチャネルの保護

IBM MQ の TLS 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。

デジタル証明書と鍵リポジトリー

キュー・マネージャーの証明書ラベル属性 (**CERTLABL**) には、大部分のチャネルで使用する個人証明書の名前を設定し、異なる証明書が必要なチャネルにはその証明書ラベルを設定して、例外としてオーバーライドすることをお勧めします。

多くのチャネルで、キュー・マネージャーに設定したデフォルトの証明書とは異なる証明書が必要な場合は、それらのチャネルをいくつかのキュー・マネージャーに分割するか、キュー・マネージャーの前に MQIPT プロキシーを使用して異なる証明書を提示することを検討してください。

すべてのチャネルに対して異なる証明書を使用することも可能ですが、鍵リポジトリーに格納する証明書が多すぎると、TLS チャネルの始動時にパフォーマンスが影響を受ける恐れがあります。 鍵リポジトリーに入れる証明書の数は、およそ 50 個以内にしてください。100 という数は、鍵リポジトリーが大きくなるにつれて GSKit パフォーマンスが急激に落ち込む最大値であると考えてください。

同じキュー・マネージャー上で複数の証明書を許可すると、複数の CA 証明書が同じキュー・マネージャーで使用される可能性が高くなります。 これにより、証明書が別個の認証局によって発行された場合に、証明書サブジェクト識別名の名前空間が競合する可能性が高くなります。

専門的な認証局ではこうした問題は慎重に扱われることが多いですが、社内の認証局では明確な命名規則が存在しないことが多く、複数の CA の間で予期せぬ一致が生じることがあります。

証明書のサブジェクト識別名に加えて、発行者識別名を確認するようにしてください。 これを行うには、チャネル認証 SSLPEERMAP レコードを使用して、サブジェクト DN と発行者 DN がそれぞれ一致するように **SSLPEER** と **SSLCERTI** の両方のフィールド設定します。

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。

ULW Windows、UNIX、および Linux システムの場合は、自己署名証明書を使用できます。 説明は、290 ページの『UNIX, Linux, and Windows での自己署名個人証明書の作成』を参照してください。

IBM i IBM i システムの場合は、ローカル CA で署名された証明書を使用できます。 手順については、275 ページの『IBM i でのサーバー証明書の要求』を参照してください。

z/OS z/OS システムの場合は、自己署名証明書とローカル CA 署名証明書のどちらでも使用できます。 手順については、319 ページの『z/OS での自己署名個人証明書の作成』 または 319 ページの『z/OS での個人証明書の要求』 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、43 ページの『IBM 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。 IBM MQ での証明書妥当性検査ポリシーの詳細については、42 ページの『IBM MQ における証明書妥当性検査ポリシー』を参照してください。

証明書失効検査の計画

異なる認証局からの複数の証明書を許可すると、不要な追加の証明書失効検査が発生する可能性があります。

特に、特定の CA からの失効サーバーの使用を明示的に構成した場合 (例えば、AUTHINFO オブジェクトまたは認証情報レコード (MQAIR) 構造を使用)、別の CA から証明書が提示された場合に失効検査は失敗します。

証明書失効サーバーを明示的に構成しないでください。 その代わりに、それぞれの証明書に証明書拡張の独自の失効サーバー・ロケーション (例えば、CRL 配布ポイントまたは OCSP AuthorityInfoAccess) が含まれる、暗黙的な検査を有効にする必要があります。

詳しくは、OCSPCheckExtensions および CDPCheckExtensions を参照してください。

TLS サポート用のコマンドおよび属性

Transport Layer Security (TLS) 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・セキュリティーが提供されています。 IBM MQ の TLS サポートにより、チャネル定義で特定のチャネルが TLS 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、TLS 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・パラメーターは、TLS をサポートしています。

CERTLABL
使用する個人証明書ラベルを定義します。

SSLCRLNL
SSLCRLNL 属性は、証明書取り消し場所を提供して、拡張 TLS 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 システムの場合、**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
TLS を使用しているチャネルが TLS 接続の確立に失敗した場合に TLS 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 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
UNIX, Linux, and Windows システムの場合、キー・リポジトリーとキュー・マネージャーを関連付けます。 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 TLS 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 TLS 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 TLS をサポートしています。

CERTLABL
使用する個人証明書ラベルを定義します。

SSLCAUTH
IBM MQ が TLS クライアントからの証明書を必要としており、証明書を検証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、TLS_RSA_WITH_AES_128_CBC_SHA。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 **runmqckm** (iKeycmd) コマンド、および UNIX, Linux, and Windows 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

TLS 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。

- 22 ページの『IBM MQ での TLS セキュリティー・プロトコル』

TLS 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

TLS 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

z/OS *IBM MQ for z/OS サーバー接続チャネル*

IBM MQ for z/OS SVRCONN チャネルをセキュアにするには、チャネル認証を実装するか、TLS を使用してセキュリティー出口を追加する必要があります。 SVRCONN チャネルには、デフォルトではセキュリティー出口が定義されていません。

セキュリティー上の問題

SVRCONN チャネルは、 例えば SYSTEM.DEF.SVRCONN のように、定義された初期の状態ではセキュアではありません。 SVRCONN チャネルをセキュアにするには、SET CHLAUTH コマンドを使用してチャネル認証をセットアップするか、セキュリティー出口をインストールして TLS を実装する必要があります。

公開されているサンプル・セキュリティー出口を使用するか、自分でセキュリティー出口を作成するか、あるいはセキュリティー出口を購入しなければなりません。

ユーザー自身で SVRCONN チャネルの セキュリティー出口を記述する場合は、準備されている使用しやすいサンプルから始めることができます。

IBM MQ for z/OS の場合、hlq.SCSQC37S ライブラリーのメンバー CSQ4BCX3 は、C 言語で記述されたセキュリティー出口のサンプルです。 サンプル CSQ4BCX3 は、事前構成されて、hlq.SCSQAUTH ライブラリー内にも含められています。

CSQ4BCX3 サンプル出口を実装するには、コンパイルされたメンバー hlq.SCSQAUTH(CSQ4BCX3) を、CHIN プロシージャー内の CSQXLIB DD に割り振られたロード・ライブラリー にコピーします。 CHIN ではロード・ライブラリーを「プログラム管理」に設定しておく 必要があることにご注意ください。

CSQ4BCX3 がセキュリティー出口に なるように SVRCONN チャネルを変更します。

V 9.1.4 クライアントが SVRCONN チャネルを使用して接続すると、CSQ4BCX3 は MQCD の **RemoteUserIdentifier** と **RemotePassword** のペアを使用して認証します。IBM MQ 9.1.4 以降の場合は、MQCSP の **CSPUserIdPtr** と **CSPPasswordPtr** のペアを使用します。 認証に成功すると、CSQ4BCX3 は **RemoteUserIdentifier** を **MCAUserIdentifier** にコピーし、スレッドの ID コンテキストを変更します。

Long Term Support および IBM MQ 9.1.4 より前の Continuous Delivery の場合、クライアントがその SVRCONN チャネルを使用して接続すると、 CSQ4BCX3 は MQCD の **RemoteUserIdentifier** と

RemotePassword のペアを使用して認証を行います。 認証に成功すると、CSQ4BCX3 は **RemoteUserIdentifier** を **MCAUserIdentifier** にコピーし、スレッドの ID コンテキストを変更します。

IBM MQ Java クライアントを記述する場合、ポップアップを使用してユーザーを照会して、MQEnvironment.userID および MQEnvironment.password を設定することができます。 接続が確立されると、これらの値が渡されます。

機能するセキュリティー出口が用意できたら、次は接続が確立される際にユーザー ID とパスワード、さらにはすべての後続の IBM MQ メッセージの内容が、ネットワーク上を平文で転送されるという問題点について検討する必要があります。 TLS を使用して、この初期接続情報と IBM MQ メッセージの内容を暗号化できます。

例

IBM MQ Explorer SVRCONN チャネル SYSTEM.ADMIN.SVRCONN 以下のステップを実行します。

1. hlq.SCSQAUTH(CSQ4BCX3) を、CHINIT プロシージャー内の CSQXLIB DD に割り振られたロード・ライブラリーにコピーします。
2. ロード・ライブラリーがプログラム管理であることを確認します。
3. セキュリティー出口 CSQ4BCX3 を使用するように SYSTEM ADMIN.SVRCONN を変更します。
4. IBM MQ Explorer で、z/OS キュー・マネージャー名を右クリックして、「**接続詳細**」 > 「**プロパティー**」 > 「**ユーザー ID**」を選択して、z/OS のユーザー ID を入力します。
5. パスワードを入力して z/OS キュー・マネージャーに接続します。

補足情報

出口 CSQ4BCX3 をプログラム管理の環境で機能するようにするには、CHIN アドレス・スペースにロードされたすべてのものが、プログラム管理ライブラリーからロードされている必要があります。 例えば、STEPLIB 内のすべてのライブラリーや、CSQXLIB DD で指定されているすべてのライブラリーなどです。 ロード・ライブラリーをプログラム管理に設定するには、RACF コマンドを実行します。 次の例では、ロード・ライブラリー名は MY.TEST.LOADLIB です。

```
RALTER PROGRAM * ADDMEM('MY.TEST.LOADLIB'//NOPADCHK)
SETROPTS WHEN(PROGRAM)REFRESH
```

CSQ4BCX3 を実装するように SVRCONN チャネルを変更するには、次の IBM MQ コマンドを発行します。

```
ALTER CHANNEL(SYSTEM ADMIN.SVRCONN) CHLTYPE(SVRCONN) SCYEXIT(CSQ4BCX3)
```

上記の例では、使用されている SVRCONN チャネル名は SYSTEM ADMIN.SVRCONN です。

チャネル出口についての詳細は、103 ページの『チャネル出口プログラム』を参照してください。

関連タスク

z/OS におけるチャネル出口プログラムの作成

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、IBM MQ 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。

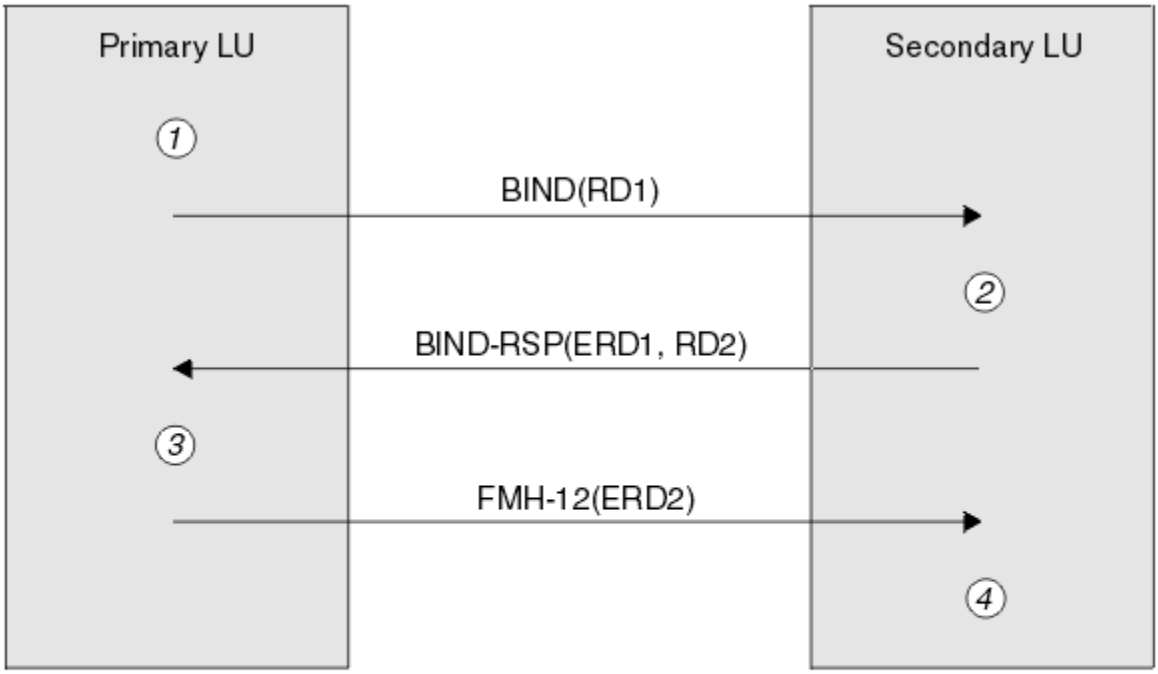
セッション・レベルの認証

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。

116 ページの図 12 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 *12.* 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、116 ページの図 12 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 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。

Windows IBM i UNIX 会話レベル認証のサポート *(IBM i、UNIX、Windows)*

このトピックでは、IBM i、UNIX、Windows での会話レベル認証の動作の概要を取り上げます。

IBM i、UNIX、Windows での会話レベル認証のサポートを 118 ページの図 13 で示します。 図の中の番号は、以下の説明内の番号と対応しています。

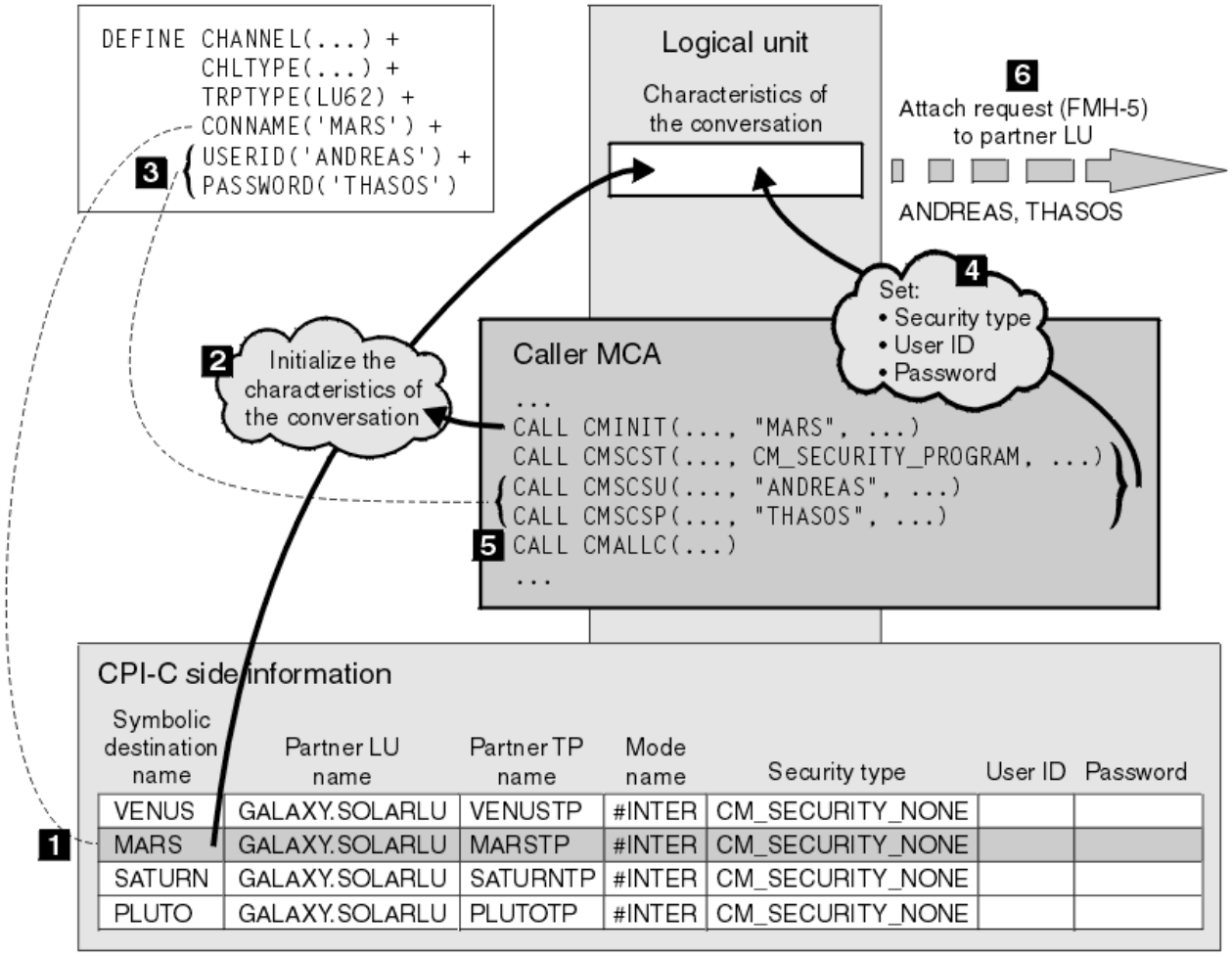


図 *13.* 会話レベルの認証に対する *IBM MQ* のサポート

IBM i 、UNIX、および Windows 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。

会話レベルの認証および *IBM MQ for z/OS*

このトピックでは、z/OS 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 MQ for z/OS では、MCA は CPI-C を使用しません。 代わりに、一部の CPI-C 機能を持つ拡張プログラム間通信機能 (APPC) のインプリメンテーションである、APPC/MVS TP Conversation Callable Services を使用します。 呼び出し側 MCA が会話を割り振る場合、セキュリティー・タイプ SAME が呼び出しで指定されます。 したがって、APPC/MVS LU が、アウトバウンド会話ではなく、インバウンド会話用のみの持続検査をサポートするので、次の 2 つの可能性があります。

- 相手側 LU が APPC/MVS LU を信頼し、既に検証されたユーザー ID を受け入れる場合、APPC/MVS LU は、次のものを含む接続要求を送信する。
  - チャネル・イニシエーター・アドレス・スペースのユーザー ID
  - RACF が使用される場合に、チャネル・イニシエーター・アドレス・スペースのユーザー ID の現行接続グループの名前である、セキュリティー・プロファイル名
  - 既に検証済みの標識
- 相手側 LU が APPC/MVS LU を信頼せず、既に検証されたユーザー ID を受け入れない場合、APPC/MVS LU は、セキュリティー情報が入っていない接続要求を送信する。

IBM MQ for z/OS 上では、DEFINE CHANNEL コマンド上の USERID および PASSWORD パラメーターは、メッセージ・チャネルには使用できず、MQI チャネルのクライアント接続側のみで有効です。 したがって、APPC/MVS LU からの接続要求には、これらのパラメーターによって指定された値が入っていることはありません。

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 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、IBM 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、IBM 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。 z/OS IBM MQ for z/OS の固有の考慮事項については、260 ページの『z/OS のキュー・マネージャー・クラスターのセキュリティー』を参照してください。

関連タスク

458 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 MQ Publish/Subscribe のセキュリティー

IBM 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 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 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 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 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 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 IBM 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 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、95 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 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、381 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 MQ Internet Pass-Thru のセキュリティー

IBM MQ 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 MQ Internet Pass-Thru (MQIPT) は IBM MQ のオプション・コンポーネントで、インターネットを介してリモート・サイト間のメッセージング・ソリューションを実装するために使用できます。

MQIPT 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、IBM 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 IBM 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。 Transport Layer Security (TLS) 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。

IBM MQ システムが MQIPT と通信する場合、MQIPT で SSL プロキシー・モードを使用している場合を除き、以下のように、IBM MQ で使用される CipherSpec が MQIPT で使用される CipherSuite と一致することを確認してください。

- MQIPT が TLS サーバーの役目をし、IBM MQ が TLS クライアントとして接続する場合、IBM MQ が使用する CipherSpec は、関連した MQIPT 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。
- MQIPT が TLS クライアントの役目をし、IBM MQ TLS サーバーに接続する場合、MQIPT CipherSuite は、受信側の IBM 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。

MQIPT から統合 IBM MQ TLS サポートにマイグレーションする場合は、 **mqiptKeyman** または **mqiptKeycmd** を使用して、 MQIPT 鍵リングからデジタル証明書を転送します。

細については、 IBM MQ Internet Pass-Thru を参照してください。

z/OS IBM MQ for z/OS のセキュリティー実装チェックリスト

このトピックでは、IBM MQ の各キュー・マネージャーでセキュリティー実装をテストして定義するための段階的な手順を取り上げます。

RACF は、IBM MQ セキュリティー・クラスの定義を、付属の静的クラス記述子テーブル (CDT) で提供します。 チェックリストを検討して、ご使用のセットアップで必要なクラスを決定します。 それらが、182 ページの『RACF セキュリティー・クラス』の説明に従ってアクティブ化されていることを確認してください。

詳細については、他のセクション (特に 192 ページの『IBM MQ リソースへのアクセスを制御するためのプロファイル』) を参照してください。

セキュリティー検査が必要な場合は、以下のチェックリストを参考にしながらセキュリティー検査を実装できます。

1. RACF の MQADMIN クラス (大文字のプロファイル) または MXADMIN クラス (大/小文字混合のプロファイル) のアクティブ化
   - セキュリティーを実装するのは、キュー共有グループ・レベルですか、キュー・マネージャー・レベルですか、それともその両方を組み合わせますか。

     187 ページの『キュー共用グループ・レベル・セキュリティーまたはキュー・マネージャー・レベル・セキュリティーを制御するためのプロファイル』を参照してください。
2. 接続セキュリティーが必要ですか。
   - **はい**: MQCONN クラスをアクティブ化します。 MQCONN クラス内で、キュー・マネージャー・レベルかキュー共有グループ・レベルのどちらかで、該当する接続プロファイルを定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。

     注 :「MQCONN」 API 要求または CICS または IMS アドレス・スペースのユーザー ID のユーザーのみが、対応する接続プロファイルへのアクセス権を持っている必要があります。
   - **いいえ**: MQADMIN または MXADMIN クラス内で、キュー・マネージャー・レベルかキュー共用グループ・レベルのどちらかで、hlq.NO.CONNECT.CHECKS プロファイルを定義します。
3. コマンドのセキュリティー検査が必要ですか。
   - **はい**: MQCMDS クラスをアクティブ化します。 MQCMDS クラス内で、キュー・マネージャー・レベルかキュー共有グループ・レベルのどちらかで、該当するコマンド・プロファイルを定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。

     キュー共有グループを使用している場合は、キュー・マネージャー自体とチャネル・イニシエーターによって使用されるユーザー ID を組み込む必要がある場合があります。 251 ページの『IBM MQ for z/OS のリソース・セキュリティーのセットアップ』を参照してください。
   - **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、必要なキュー・マネージャーまたはキュー共有グループの hlq.NO.CMD.CHECKS プロファイルを定義します。
4. コマンドで使用するリソースに関するセキュリティーが必要ですか。
   - **はい**: MQADMIN または MXADMIN クラスが活動状態になるようにします。 MQADMIN または MXADMIN クラス内で、キュー・マネージャー・レベルかキュー共有グループ・レベルのどちらかで、リソースを保護するための該当するプロファイルをコマンドで定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。 CSQ6SYSP の CMDUSER パラメーターを、コマンド・セキュリティー検査に使用するデフォルトのユーザー ID に設定します。

     キュー共有グループを使用している場合は、キュー・マネージャー自体とチャネル・イニシエーターによって使用されるユーザー ID を組み込む必要がある場合があります。 251 ページの『IBM MQ for z/OS のリソース・セキュリティーのセットアップ』を参照してください。
   - **いいえ**: MQADMIN クラスまたは MXADMIN クラス内で、必要なキュー・マネージャーまたはキュー共用グループ用に hlq.NO.CMD.RESC.CHECKS プロファイルを定義します。
5. キュー・セキュリティーが必要ですか。
   - **はい**: MQQUEUE クラスまたは MXQUEUE クラスをアクティブ化します。 MQQUEUE または MXQUEUE クラス内で、必要なキュー・マネージャーまたはキュー共有グループ用に該当するキュー・プロファイルを定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。
   - **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、必要なキュー・マネージャーまたはキュー共有グループの hlq.NO.QUEUE.CHECKS プロファイルを定義します。
6. プロセス・セキュリティーが必要ですか。

- **はい**: MQPROC クラスまたは MXPROC クラスをアクティブ化します。 キュー・マネージャー・レベルまたはキュー共用グループ・レベルの適切なプロセス・プロファイルを定義し、それらのプロファイルに対するアクセス権を対象のユーザーまたはグループに与えます。
- **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、該当するキュー・マネージャーまたはキュー共有グループの hlq.NO.PROCESS.CHECKS プロファイルを定義します。

7. 名前リスト・セキュリティーが必要ですか。
   - **はい**: MQNLIST クラスまたは MXNLIST クラスをアクティブ化します。 MQNLIST クラスまたは MXNLIST クラスで、キュー・マネージャー・レベルまたはキュー共有グループ・レベルの適切な名前リスト・プロファイルを定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。
   - **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、必要なキュー・マネージャーまたはキュー共有グループの hlq.NO.NLIST.CHECKS プロファイルを定義します。
8. トピック・セキュリティーが必要ですか。
   - **はい**: MXTOPIC クラスをアクティブ化します。 MXTOPIC クラスでキュー・マネージャー・レベルまたはキュー共有グループ・レベルの適切なトピック・プロファイルを定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。
   - **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、必要なキュー・マネージャーまたはキュー共有グループの hlq.NO.TOPIC.CHECKS プロファイルを定義します。
9. コンテキストの使用に関連する MQOPEN または MQPUT1 オプションの使用の保護が必要なユーザーがいますか。
   - **はい**: MQADMIN または MXADMIN クラスが活動状態になるようにします。 MQADMIN クラスまたは MXADMIN クラス内で、キュー、キュー・マネージャー、またはキュー共有グループ・レベルで、hlq.CONTEXT.queuename プロファイルを定義します。 その後、これらのプロファイルへのアクセス権を対象のユーザーまたはグループに付与します。
   - **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、必要なキュー・マネージャーまたはキュー共有グループの hlq.NO.CONTEXT.CHECKS プロファイルを定義します。
10. 代替ユーザー ID の使用の保護が必要ですか。
    - **はい**: MQADMIN または MXADMIN クラスが活動状態になるようにします。 適切な hlq.ALTERNATE.USER。 必要なキュー・マネージャーまたはキュー共有グループの *alternateuserid* プロファイル。これらのプロファイルへのアクセスを必要なユーザーまたはグループに許可します。
    - **いいえ**: MQADMIN クラスまたは MXADMIN クラスで、必要なキュー・マネージャーまたはキュー共有グループの hlq.NO.ALTERNATE.USER.CHECKS プロファイルを定義します。
11. RESLEVEL に基づくリソース・セキュリティー検査で使用するユーザー ID を調整する必要がありますか。
    - **はい**: MQADMIN または MXADMIN クラスが活動状態になるようにします。 MQADMIN クラスまたは MXADMIN クラス内で、キュー・マネージャー・レベルかキュー共有グループ・レベルのどちらかで、hlq.RESLEVEL プロファイルを定義します。 その後、このプロファイルへのアクセス権を必要なユーザーまたはグループに付与します。
    - **いいえ**: hlq.RESLEVEL に適用できる総称プロファイルが、MQADMIN クラスまたは MXADMIN クラスの中に存在しないようにします。 必要なキュー・マネージャーまたはキュー共用グループ用に hlq.RESLEVEL プロファイルを定義して、他のユーザーまたはグループがそれにアクセスしないようにします。
12. 未使用のユーザー ID を IBM MQ からタイムアウトにする必要がありますか。
    - **はい**: 使用するタイムアウトを値を決めてから、MQSC の ALTER SECURITY コマンドを実行して、TIMEOUT パラメーターと INTERVAL パラメーターを変更します。
    - **いいえ**: MQSC の ALTER SECURITY コマンドを実行して、INTERVAL 値をゼロに設定します。

    注 : キュー・マネージャーの始動の際に MQSC ALTER SECURITY コマンドが自動的に発行されるようにするためには、サブシステムで使用される CSQINP1 初期設定入力データ・セットを更新してください。

13. 分散キューイングを使用しますか。
    - **はい**: チャネル認証レコードを使用します。 詳しくは、47 ページの『チャネル認証レコード』を参照してください。
    - 各チャネルについて該当する MCAUSER 属性値を決めるか、適切なチャネル・セキュリティー出口を提供できます。
14. Transport Layer Security (TLS) を使用しますか。
    - **はい**: 指定された識別名 (DN) を含む TLS 個人証明書を提示する任意のユーザーが特定の MCAUSER を使用するよう指定するには、タイプ SSLPEERMAP 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。
    - TLS インフラストラクチャーの計画を立てます。 z/OS の System SSL フィーチャーをインストールします。 RACF 内で、証明書名フィルター (CNF) (使用する場合) と、ディジタル証明書をセットアップします。 SSL 鍵リングをセットアップします。 SSLKEYR キュー・マネージャー属性が非ブランクになるようにして SSL 鍵リングを指すようにします。 また、確実に、SSLTASKS の値を 2 以上にします。
    - **いいえ**: SSLKEYR をブランク、SSLTASKS をゼロにします。

    TLS について詳しくは、22 ページの『IBM MQ での TLS セキュリティー・プロトコル』を参照してください。
15. クライアントを使用しますか。
    - **はい**: チャネル認証レコードを使用します。
    - 各サーバー接続チャネルについて該当する MCAUSER 属性値を決めるか、適切なチャネル・セキュリティー出口を提供できます (必要な場合)。
16. スイッチ設定を確認します。

    IBM MQ は、キュー・マネージャーが開始した時に、セキュリティー設定を表示するメッセージを発行します。 これらのメッセージを使用して、スイッチが正しく設定されているかどうかを確認してください。
17. クライアント・アプリケーションからパスワードを送信しますか。
    - **はい**: 最適な保護のため、z/OS フィーチャーがインストールされ、Integrated Cryptographic Service Facility (ICSF) が開始されていることを確認します。
    - **いいえ**: ICSF が開始されていないことを報告するエラー・メッセージを無視することができます。

    ICSF の詳細については、260 ページの『Integrated Cryptographic Service Facility (ICSF) の使用』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。

ULW UNIX, Linux, and Windows 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 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 MQ をだれが管理できるか
IBM 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 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 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 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。

ULW UNIX, Linux, and Windows での権限の機能

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。

ULW *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。

127 ページの表 10、127 ページの表 11、128 ページの表 12、および 128 ページの表 13 は、それぞれの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 *10. 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 (128 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 *11. 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 (128 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 **(128 ページの『2』)**	MQZAO_INPUT	適用外	適用外
MQOO_OUTPUT (通常キュー) **(129 ページの『3』)**	MQZAO_OUTPUT	適用外	適用外
MQOO_PASS_IDENTITY_CONTEXT **(129 ページの『4』)**	MQZAO_PASS_IDENTITY_CONTEXT	適用外	検査しない
MQOO_PASS_ALL_CONTEXT **(129 ページの『4』, 129 ページの『5』)**	MQZAO_PASS_ALL_CONTEXT	適用外	検査しない
MQOO_SET_IDENTITY_CONTEXT **(129 ページの『4』, 129 ページの『5』)**	MQZAO_SET_IDENTITY_CONTEXT	適用外	MQZAO_SET_IDENTITY_CONTEXT **(129 ページの『6』)**
MQOO_SET_ALL_CONTEXT **(129 ページの『4』, 129 ページの『7』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(129 ページの『6』)**
MQOO_OUTPUT (伝送キュー) **(129 ページの『8』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(129 ページの『6』)**
MQOO_SET	MQZAO_SET	適用外	検査しない

表 *11. MQOPEN* 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 (128 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_ALTERNATE_ USER_AUTHORITY	**(129 ページの『9』)**	**(129 ページの『9』)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(129 ページの『9』、129 ページの『10』)**

表 *12. 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 (128 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 **(129 ページの『11』)**	適用外	検査しない
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(129 ページの『11』)**	適用外	検査しない
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(129 ページの『11』)**	適用外	MQZAO_SET_ IDENTITY_CONTEXT **(129 ページの『6』)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(129 ページの『11』)**	適用外	MQZAO_SET_ ALL_CONTEXT **(129 ページの『6』)**
(伝送キュー) **(129 ページの『8』)**	MQZAO_SET_ ALL_CONTEXT	適用外	MQZAO_SET_ ALL_CONTEXT **(129 ページの『6』)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(129 ページの『12』)**	適用外	MQZAO_ALTERNATE_ USER_AUTHORITY **(129 ページの『10』)**

表 *13. 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 (128 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 **(129 ページの『13』)**	適用外	適用外
MQCO_DELETE _PURGE	MQZAO_DELETE **(129 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 129 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. 検査は、次の記述が両方とも当てはまる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。

ULW 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

オブジェクト	必要な権限
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
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 (134 ページの『1』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (134 ページの『2』)
トピック	MQZAO_CREATE (134 ページの『2』)
プロセス	MQZAO_CREATE (134 ページの『2』)
キュー・マネージャー	適用外
名前リスト	MQZAO_CREATE (134 ページの『2』)
認証情報	MQZAO_CREATE (134 ページの『2』)
チャネル	MQZAO_CREATE (134 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (134 ページの『2』)
リスナー	MQZAO_CREATE (134 ページの『2』)
サービス	MQZAO_CREATE (134 ページの『2』)
コミュニケーション情報	MQZAO_CREATE (134 ページの『2』)

DEFINE 「オブジェクト」 REPLACE (134 ページの『1』 134 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

オブジェクト	必要な権限
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
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
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

コマンド	オブジェクト	必要な権限
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
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 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

コマンド	オブジェクト	必要な権限
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

ULW *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

Copy *object* (without replace) (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 **(140 ページの『2』)**

オブジェクトのコピー (置き換えあり)(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

オブジェクト	必要な権限
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

Create *object* (without replace) (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)**
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 (置き換えあり)(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

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
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
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

オブジェクト	必要な権限
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
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
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 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	「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	「mqm」グループ・メンバーシップが必要

コマンド	オブジェクト	必要な権限
Suspend Queue Manager Cluster	「mqm」グループ・メンバーシップが必要	「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	「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。

AIX 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。

このタスクについて

AIX では、SMITTY を使用して、グループの作成、グループへのユーザーの追加、グループ内のユーザーのリストの表示、グループからのユーザーの削除を行えます。

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. グループを作成するには、以下のステップを実行します。
   a) 「**Add a Group (グループの追加)**」を選択して Enter キーを押します。
   b) 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。
   c) Enter キーを押してグループを作成します。
4. ユーザーをグループに追加するには、以下のステップを実行します。
   a) 「**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。
   b) 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。
   c) 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。

d) Enter キーを押してグループにその名前を追加します。

5. グループ内のユーザーを表示するには、以下の手順を実行します。
   a)「**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。
   b) 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。
6. グループからユーザーを削除するには、以下のステップを実行します。
   a)「**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。
   b) 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。
   c) グループから除去するユーザーの名前を削除します。
   d) Enter キーを押してグループからその名前を除去します。

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。

このタスクについて

Linux では、グループ情報は /etc/group ファイル内に保持されます。 コマンドを使用して、グループの作成、グループへのユーザーの追加、グループ内のユーザーのリストの表示、グループからのユーザーの削除を行えます。

手順

1. 新規グループを作成するには、**groupadd** コマンドを使用します。
   次のコマンドを入力します。

   ```
   groupadd -g group-ID group-name
   ```

   group-ID はグループの数値 ID、*group-name* はグループの名前です。
2. 補助グループにメンバーを追加するには、 **usermod** コマンドを使用して、そのユーザーが現在メンバーになっている補助グループと、そのユーザーがメンバーになる補助グループをリストします。
   例えば、ユーザーがすでに groupa というグループのメンバーで、groupb のメンバーにもなろうとしている場合、以下のコマンドを使用します。

   ```
   usermod -G groupa,groupb user-name
   ```

   user-name は、ユーザー名です。
3. グループのメンバーを表示するには、**getent** コマンドを使用します。
   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。
4. 補助グループからメンバーを除去するには、 **usermod** コマンドを使用して、ユーザーをメンバーのままにする補助グループをリストします。
   例えば、ユーザーの 1 次グループが users であり、そのユーザーがグループ mqm、 groupa 、および groupb のメンバーでもある場合、 mqm グループからユーザーを削除するには、次のコマンドを使用します。

   ```
   usermod -G groupa,groupb user-name
   ```

   user-name は、ユーザー名です。

Solaris Solaris でのグループの作成と管理

Solaris 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。

このタスクについて

Solaris では、グループ情報は /etc/group ファイル内に保持されます。 コマンドを使用して、グループの作成、グループへのユーザーの追加、グループ内のユーザーのリストの表示、グループからのユーザーの削除を行えます。

手順

1. 新規グループを作成するには、**groupadd** コマンドを使用します。
   次のコマンドを入力します。

   ```
   groupadd -g group-ID group-name
   ```

   group-ID はグループの数値 ID、*group-name* はグループの名前です。
2. 補助グループにメンバーを追加するには、 **usermod** コマンドを使用して、そのユーザーが現在メンバーになっている補助グループと、そのユーザーがメンバーになる補助グループをリストします。
   例えば、ユーザーがすでに groupa というグループのメンバーで、groupb のメンバーにもなろうとしている場合、以下のコマンドを使用します。

   ```
   usermod -G groupa,groupb user-name
   ```

   user-name は、ユーザー名です。
3. グループのメンバーを確認するには、/etc/group ファイルでそのグループの項目を調べます。
4. 補助グループからメンバーを除去するには、 **usermod** コマンドを使用して、ユーザーをメンバーのままにする補助グループをリストします。
   例えば、ユーザーの 1 次グループが users であり、そのユーザーがグループ mqm、 groupa 、および groupb のメンバーでもある場合、 mqm グループからユーザーを削除するには、次のコマンドを使用します。

   ```
   usermod -G groupa,groupb user-name
   ```

   user-name は、ユーザー名です。

Windows Windows でのグループの作成と管理

Windows の場合、「コンピュータの管理」機能を使用してワークステーションやメンバー・サーバー・マシンのグループを管理できます。

このタスクについて

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、Windows の「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行った変更は、ユーザーが再ログインするまで有効にならない場合があります。

Windows *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 *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 *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 *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。

Windows 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 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 MQ を実行する場合に、これらの違いが IBM 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。

Windows *IBM MQ Windows* サービスのローカル・ユーザー・アカウントとドメイン・ユーザー・アカウント

IBM MQ は実行中に、許可ユーザーのみがキュー・マネージャーまたはキューにアクセスできることを検査する必要があります。 そのためには、そのようなアクセスを試みるユーザーの情報を IBM MQ で照会するための特別なユーザー・アカウントが必要です。

Prepare IBM MQ Wizard での特別なユーザー・アカウントの構成

Prepare IBM MQ Wizard は、Windows サービスが、それを使用する必要のあるプロセス間で共有されるように、特別なユーザー・アカウントを作成します (IBM MQ 準備ウィザードでの IBM MQ の構成を参照)。

Windows サービスは、IBM MQ インストール済み環境のクライアント・プロセス間で共有されます。 インストールごとに 1 つのサービスが作成されます。 各サービスの名前は MQ_*InstallationName* で、表示名は IBM MQ(*InstallationName*)です。

各サービスは対話式および非対話式のログオン・セッション間で共有する必要があるため、それらを特別なユーザー・アカウントの下で起動する必要があります。 すべてのサービスに対して 1 つの特別なユーザー・アカウントを使用することも、個別の特別なユーザー・アカウントを作成することもできます。 それぞれの特別なユーザー・アカウントには、「サービスとしてログオン」するユーザー権限必要です。詳しくは、146 ページの表 14 を参照してください。 ユーザー ID にサービスを実行する権限がない場合、サービスは開始されず、Windows システム・イベント・ログにエラーが返されます。 多くの場合、Prepare IBM MQ Wizard を実行し、それによりユーザー ID が正しくセットアップされます。 ただし、ユーザー ID を手動で構成した場合は、問題が発生し、解決が必要になる可能性があります。

IBM MQ をインストールして Prepare IBM MQ Wizard を初めて実行すると、「サービスとしてログオン」を含め、必要な設定と権限を持つ MUSR_MQADMIN というサービスのローカル・ユーザー・アカウントが作成されます。

以降のインストールでは、Prepare IBM MQ Wizard によって、名前が MUSR_MQADMIN*x* のユーザー・アカウントが作成されます。ここで *x* は、まだ存在しないユーザー ID を示す、次に使用可能な番号です。 MUSR_MQADMIN*x* 用のパスワードは、アカウントの作成時にランダムに生成され、サービス用のログオン環境を構成するために使用されます。 生成されたパスワードは有効期限切れになりません。

一定期間が過ぎるとアカウントのパスワードの変更を必要とするアカウント・ポリシーがシステム上でセットアップされていても、この IBM MQ アカウントはその影響を受けません。

このパスワードは、この一回限りの処理以外では使用されず、Windows オペレーティング・システムによって、レジストリーの安全な部分に保管されます。

IBM MQ と Active Directory の併用

Active Directory ディレクトリー・サービスを使用しているドメイン・コントローラー上にユーザー・アカウントが定義されている一部のネットワーク構成では、IBM MQ を実行しているローカル・ユーザー・アカウントが、その他のドメイン・ユーザー・アカウントのグループ・メンバーシップを照会するために必要な権限を持っていないことがあります。 IBM MQ をインストールする時に、Prepare IBM MQ Wizard は、テストを実行し、ネットワーク構成について尋ねることによって、この点を確認します。

IBM MQ が実行されているローカル・ユーザー・アカウントに必要な権限がない場合、Prepare IBM MQ Wizard は、特定のユーザー権限を持つドメイン・ユーザー・アカウントのアカウント詳細を求めるプロンプトを表示します。 Windows ドメイン・アカウントの作成とセットアップの方法については、IBM MQ 用の Windows 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。 ドメイン・ユーザー・アカウントが必要なユーザー権限については、146 ページの表 14 を参照してください。

ドメイン・ユーザー・アカウントの有効なアカウント詳細を Prepare IBM MQ Wizard に入力すると、ウィザードは IBM MQ Windows サービスを、新規アカウントの下で実行するように構成します。 アカウント詳細は、レジストリーのセキュア部分に保持され、ユーザーが読み取ることはできません。

サービスが稼働すると IBM MQ Windows サービスも起動し、サービスが終了するまで稼働し続けます。 Windows サービスの起動後にサーバーにログオンする IBM MQ 管理者は、IBM MQ Explorer を使用してサーバー上のキュー・マネージャーを管理できます。 これによって、IBM MQ Explorer と既存の Windows サービス・プロセスを接続できます。 ただし、これら 2 つの処理の実行には、次に示すように、それぞれ異なるレベルの許可が必要です。

- 起動プロセス: 起動許可

- IBM MQ 管理者: アクセス権

IBM MQ Windows サービスに必要なユーザー権限

以下の表に、 IBM MQ インストール済み環境の Windows サービスが実行されるローカル・ユーザー・アカウントおよびドメイン・ユーザー・アカウントに必要なユーザー権限をリストします。

表 *14. IBM MQ Windows* サービスに必要なユーザー権限

アクセス権	**説明**
バッチ・ジョブとしてログオン	IBM MQ Windows サービスは、このユーザー・アカウントの下で実行できる。
サービスとしてログオン	ユーザーは、IBM MQ Windows サービスを設定し、構成済みアカウントを使用してログオンできる。
システムをシャットダウン	サービスのリカバリーが失敗したときにシステムをシャットダウンするように構成されている場合、IBM MQ Windows サービスはサーバーを再始動できる。
割り当て量の増加	オペレーティング・システムの `CreateProcessAsUser` 呼び出しに必要。
オペレーティング・システムの一部として動作する	オペレーティング・システムの `LogonUser` 呼び出しに必要。
全探索検査の迂回	オペレーティング・システムの `LogonUser` 呼び出しに必要。
処理レベル・トークンの置換	オペレーティング・システムの `LogonUser` 呼び出しに必要。

注 : ASP および IIS アプリケーションを実行する環境では、「プログラムのデバッグ」権限が必要になる可能性があります。

ご使用のドメイン・ユーザー・アカウントには、これらの Windows ユーザー権限が、ローカル・セキュリティー・ポリシー・アプリケーションにリストされるのと同様に有効なユーザー権限として設定されている必要があります。 そうでない場合は、ローカル・セキュリティー・ポリシー・アプリケーションをサーバー上でローカルに使用するか、またはドメイン・セキュリティー・アプリケーション・ドメイン全体を使用して、ユーザー権限を設定します。

Windows *Windows* サーバーのセキュリティー権限

Windows Server では、ローカル・ユーザーとドメイン・ユーザーのどちらがインストールを実行するかによって、 IBM MQ のインストールの動作が異なります。

ローカル・ユーザーが IBM MQ をインストールする場合、Prepare IBM MQ Wizard は、IBM 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。 Prepare IBM MQ Wizard は、ネットワーク構成についてユーザーに質問し、Windows 2000 以降で実行されているドメイン・コントローラー上に他のユーザー・アカウントが定義されているかどうかを判別します。 ある場合、IBM 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。 Prepare IBM MQ Wizard は、「IBM MQ 準備ウィザードによる IBM MQ の構成」で説明されているように、このユーザーのアカウントの詳細を求めるプロンプトをユーザーに出します。

ドメイン・ユーザーが IBM MQ をインストールする場合、Prepare IBM MQ Wizard は、 IBM 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、Prepare IBM MQ Wizard は常に、使用する IBM 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ユーザへプロンプトで表示します。

IBM 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、 Prepare IBM MQ Wizard を使用して構成されるまで、IBM MQ は正しく動作できません。 Prepare IBM MQ Wizard 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。

詳しくは、IBM MQ でのドメイン・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 *IBM MQ* サービスと関連したユーザー名の変更

IBM MQ サービスに関連したユーザー名を変更できます。そのためには、Prepare IBM MQ Wizard を使用して新しいアカウントを作成し、詳細情報を入力します。

このタスクについて

IBM MQ をインストールして、Prepare IBM MQ Wizard を初めて実行すると、MUSR_MQADMIN という名前でサービス用のローカル・ユーザー・アカウントが作成されます。 以降のインストールでは、Prepare IBM MQ Wizard によって、名前が MUSR_MQADMIN*x* のユーザー・アカウントが作成されます。ここで *x* は、まだ存在しないユーザー ID を示す、次に使用可能な番号です。

IBM MQ サービスと関連したユーザー名を MUSR_MQADMIN または MUSR_MQADMIN*x* からその他の名前に変更する必要がある場合があります。 例えば、キュー・マネージャーが Db2® と関連がある場合、8 文字を超えるユーザー名は受諾されないので、この変更を行う必要があります。

手順

1. 新規ユーザー・アカウントを作成します (例えば **NEW_NAME**)。
2. Prepare IBM MQ Wizard を使用して、新規ユーザー・アカウントの詳細を入力します。

関連タスク

IBM MQ 準備ウィザードを使用した IBM MQ の構成

Windows *IBM MQ Windows* サービスのローカル・ユーザー・アカウントのパスワードの変更

「コンピューターの管理」パネルを使用して、IBM MQ Windows サービスのローカル・ユーザー・アカウントのパスワードを変更できます。

このタスクについて

IBM MQ Windows サービスのローカル・ユーザー・アカウントのパスワードを変更するには、以下のステップを実行します。

手順

1. サービスを実行しているユーザーを識別します。
2. 「コンピュータの管理」パネルから、IBM MQ のサービスを停止します。
3. 個人のパスワードを変更する場合と同じようにして、必要なパスワードを変更します。
4. 「コンピュータの管理」パネルから、IBM MQ サービスのプロパティーに移動します。
5. 「**ログオン**」ページを選択します。
6. 指定したアカウント名が、パスワードが変更されたユーザーと一致していることを確認します。
7. 「**パスワード**」フィールドおよび「**確認パスワード**」フィールドにパスワードを入力し、「**OK**」をクリックします。

Windows ドメイン・ユーザー・アカウントの下で実行されているインストール済み環境の *IBM MQ Windows* サービスのパスワードの変更

Prepare IBM MQ Wizard を使用してドメイン・ユーザー・アカウントのアカウント詳細を入力する代わりに、「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 MQ サービスの「**ログオン**」 の詳細を変更することができます。

このタスクについて

インストール済み環境の IBM MQ Windows サービスがドメイン・ユーザー・アカウントの下で実行されている場合、以下のようにアカウントのパスワードを変更することができます。

手順

1. ドメイン・コントローラー上でドメイン・アカウントのパスワードを変更します。 パスワードを変更するには、ドメイン管理者に問い合わせる必要があります。
2. IBM MQ サービスの 「**ログオン**」 ページを変更するには、以下の手順を実行します。
   a) サービスを実行しているユーザーを識別します。
   b)「コンピュータの管理」パネルから、IBM MQ のサービスを停止します。
   c) 個人のパスワードを変更する場合と同じようにして、必要なパスワードを変更します。
   d)「コンピュータの管理」パネルから、IBM MQ サービスのプロパティーに移動します。
   e)「**ログオン**」ページを選択します。
   f) 指定したアカウント名が、パスワードが変更されたユーザーと一致していることを確認します。
   g)「**パスワード**」フィールドおよび「**確認パスワード**」フィールドにパスワードを入力し、「**OK**」をクリックします。

   ユーザー・インターフェース・アプリケーションから発行される MQSC コマンドや、システムの始動、シャットダウン、またはサービスのリカバリー時に自動的に実行される MQSC コマンドはすべて、IBM MQ Windows サービスを実行しているユーザー・アカウントで実行されます。 したがって、このユーザー・アカウントは、IBM MQ 管理権限を持っていなければなりません。 デフォルトでは、このユーザー・アカウントは、サーバー上のローカル mqm グループに追加されます。 このメンバーシップが削除されると、IBM MQ Windows サービスは機能しなくなります。 ユーザー権限の詳細については、146 ページの『IBM MQ Windows サービスに必要なユーザー権限』を参照してください。

   IBM MQ Windows サービスを実行するユーザー・アカウントでセキュリティー上の問題が発生した場合、システムのイベント・ログにエラー・メッセージと説明が書き込まれます。

関連タスク

IBM MQ 準備ウィザードを使用した IBM MQ の構成

Windows *Windows* サーバーをドメイン・コントローラーへプロモートする際の考慮事項

Windows サーバーをドメイン・コントローラーにプロモートする場合、ユーザーやグループの権限に関連するセキュリティー設定が適切かどうかを考慮する必要があります。 サーバーとドメイン・コントローラーの間で Windows マシンの状態を変更する場合、 IBM MQ はローカルに定義された mqm グループを使用するため、 IBM MQ の操作に影響を与える可能性があることを考慮する必要があります。

ドメイン・ユーザーとグループに関連したセキュリティー設定

IBM 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、IBM 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 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、セキュリティー設定の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。 ドメイン・コントローラーがセットアップされていて、ローカル・アカウントにドメイン・ユーザー・アカウントのグループ・メンバーシップを照会する権限がある場合、インストール・プロセス中に IBM 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。 ただし、ドメイン・コントローラーがセットアップされていて、ローカル・アカウントにドメイン・ユーザー・アカウントのグループ・メンバーシップを照会する権限がない場合は、ドメインで定義されているユーザーにキュー・マネージャーやキューへのアクセス権限があるかどうかの検査を IBM MQ が実行できないので、アクセスが失敗します。 このようにしてセットアップしたドメイン・コントローラー

で Windows を使用する場合は、必要な権限を持った特別なドメイン・ユーザー・アカウントを使用する必要があります。

この場合、以下の点についての知識が必要です。

- 対象バージョンの Windows の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 MQ Windows サービスを構成する方法。

詳細については、Configuring user accounts for IBM MQ のユーザー・アカウントの構成を参照してください。

IBM MQ からローカル mqm グループへのアクセス

Windows サーバーをドメイン・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する際に、IBM MQ はローカルの mqm グループへのアクセスを失います。

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。 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.
```

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 IBM 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 IBM 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 *Windows でのネストされたグループに関する制限*

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、IBM 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 混合」の機能レベルとなっています。(Windows サーバ 2008 と Windows サーバ 2012 は Windows 2003 ドメインモデルに準拠しています)。 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、IBM MQ によって使用される ID に関する制限があります。 IBM 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 IBM 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。

Windows リモート環境から *IBM MQ* を使用するためのユーザー権限

IBM MQ へのリモート接続でキュー・マネージャーを作成したり開始したりするには、「グローバル・オブジェクトの作成」ユーザー・アクセス権限が必要です。

このタスクについて

注 : 管理者には、デフォルトで「グローバル・オブジェクトの作成」ユーザー・アクセスがあります。 このため管理者は、ユーザー権限を変更することなく、リモート側から接続されているキュー・マネージャーの作成および開始を行うことができます。

Terminal Services または Remote Desktop Connection のいずれかを使用して Windows マシンに接続している時に、キュー・マネージャーの作成、開始、削除で問題が発生する場合は、「グローバル・オブジェクトの作成」ユーザー・アクセス権限がないことが原因になっている可能性があります。

「グローバル・オブジェクトの作成」 ユーザー・アクセスは、グローバル・ネームスペースにオブジェクトを作成することを許可されたユーザーを制限します。 アプリケーションでグローバル・オブジェクトを作成するには、 グローバル・ネームスペースでアプリケーションが実行されているか、 またはアプリケーションを実行中のユーザーに、「グローバル・オブジェクトの作成」 ユーザー・アクセスが適用されている必要があります。

Terminal Services または Remote Desktop Connection を使用する Windows マシンにリモート側から接続する場合、アプリケーションは自身のローカル・ネームスペースで稼働します。 IBM MQ Explorer または **crtmqm** か **dltmqm** コマンドを使ってキュー・マネージャーを作成または削除しようとする場合、または **strmqm** コマンドでキュー・マネージャーを開始しようとする場合は、権限エラーになります。 これによって、IBM MQ FDC がプローブ ID XY132002 で作成されます。

IBM MQ Explorer を使用して、 または **amqmdain qmgr start** コマンドを使用してキュー・マネージャーを開始すると、正しく開始できます。 これは、これらのコマンドが直接キュー・マネージャーを始動しないためです。 これらのコマンドは、代わりにキュー・マネージャーを開始する要求をグローバル・ネームスペースで稼働中の別のプロセスに送信します。

ターミナル・サービスを使用している状態で、IBM MQ を管理するためのさまざまな手法が機能しない場合は、「グローバル・オブジェクトの作成」ユーザー権限を設定してみてください。

手順

1. 以下のようにして、「管理ツール」パネルを開きます。

   Windows Server 2008 および Windows Server 2012
   「**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**」でこのパネルにアクセスします。

   Windows 8.1
   「**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。
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. 「**ローカル ポリシー**」を展開します。
4. 「**ユーザー権利の割り当て**」をクリックします。
5. 「グローバル・オブジェクトの作成」ポリシーに新規ユーザーまたはグループを追加します。

Windows *SSPI チャネル出口プログラム (Windows)*

IBM 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、 IBM WebSphere MQ 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 IBM MQ MQI client・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、 IBM WebSphere MQ 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 IBM MQ MQI client・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 セキュリティー・テンプレート・ファイルの適用 *(Windows)*

テンプレートを適用すると、IBM 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、IBM 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、IBM 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、IBM MQ をインストールする前にマシンを構成してください。

IBM 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、IBM 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。

Windows *IBM 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 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。

このタスクについて

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 MQ に接続する場合、問題が発生する可能性があります。

IBM 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。 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 MQ がプロセスを変更して、IBM MQ 管理者に SYNCHRONIZE 権限を付与します。 ただし、IBM MQ プロセスを実行するアカウントでは、要求されたアクセスを許可する前に、追加の許可を必要とする場合があります。

IBM 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。

手順

1. 「ローカル セキュリティ ポリシー」ツールを始動して、「**セキュリティの設定**」->「**ローカル ポリシー**」->「**ユーザー権限の割り当て**」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 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。

IBM i でのセキュリティーのセットアップ

IBM i では、IBM MQ のオブジェクト権限マネージャー (OAM) と IBM i のオブジェクト・レベル・セキュリティーによってセキュリティーを実装します。

IBM MQ オブジェクトへのアクセス権限を決定する際に検討する必要のあるセキュリティーに関する考慮事項。

自社内のユーザーに権限を設定する際には、次の点を考慮する必要があります。

1. IBM MQ for IBM i コマンドに関する権限の認可と取り消しは、IBM i の GRTOBJAUT コマンドおよび RVKOBJAUT コマンドを使用して行ってください。

   QMQM ライブラリーでは、特定の非コマンド (*cmd) オブジェクトの ***PUBLIC** 権限は ***USE** に設定されます。 これらのオブジェクトの権限を変更したり、権限リストを使用して権限を付与したりしないでください。 誤った権限が付与されると、IBM MQ の機能が失われてしまう場合があります。
2. IBM MQ for IBM i のインストール時に、次の特殊ユーザー・プロファイルが作成されます。

   QMQM
   主に、内部製品専用機能に使用します。 ただし、MQCNO_FASTPATH_BINDINGS を使用するトラステッド・アプリケーションの実行には使用できます。 MQCONNX 呼び出しを使用した、キュー・マネージャーへの接続を参照してください。

   QMQMADM
   IBM MQ の管理者用のグループ・プロファイルとして使用します。 このグループ・プロファイルで、CL コマンドおよび IBM MQ リソースへのアクセス権限が与えられます。

   IBM MQ のコマンドを呼び出すプログラムをサブミットするために SBMJOB を使用する場合、USER が明示的に QMQMADM に設定されていてはなりません。 その場合、QMQM か、またはグループとして QMQMADM が指定されている別のユーザー・プロファイルに USER を設定してください。
3. チャネル・コマンドをリモート・キュー・マネージャーに送信する場合は、ユーザー・プロファイルが、ターゲット・システム上のグループ QMQMADM のメンバーになっていることを確かめます。 PCF および MQSC チャネル・コマンドについては、 IBM MQ for IBM i CL コマンドを参照してください。
4. ユーザーに関連付けられたグループ集合は、OAM によってグループの許可が計算されるとキャッシュされます。

グループ集合がキャッシュされた後、ユーザーのグループ・メンバーシップに行われる変更は、キュー・マネージャーを再始動するか、RFRMQMAUT を実行してセキュリティーをリフレッシュするまで認識されません。

5. 特に重要なコマンドを使用する権限を持つユーザーの数を制限してください。 特に重要なコマンドには次のようなものがあります。
   - メッセージ・キュー・マネージャーの作成 (CRTMQM)
   - メッセージ・キュー・マネージャーの削除 (DLTMQM)
   - メッセージ・キュー・マネージャーの開始 (STRMQM)
   - メッセージ・キュー・マネージャーの終了 (ENDMQM)
   - コマンド・サーバーの開始 (STRMQMCSVR)
   - コマンド・サーバーの終了 (ENDMQMCSVR)
6. チャネル定義には、セキュリティー出口プログラムの指定が含まれています。 チャネルの作成と変更には、特別の考慮が必要です。 セキュリティー出口の詳細については、104 ページの『セキュリティー出口の概要』を参照してください。
7. チャネル出口プログラムおよびトリガー・モニター・プログラムは置き換え可能です。 この種の置き換えのセキュリティーは、プログラマーの責任です。

IBM i オブジェクト権限マネージャー (IBM i)

オブジェクト権限マネージャー (OAM) は、キューやプロセス定義などの IBM MQ オブジェクトを操作するためのユーザーの許可を管理します。 また、OAM は、特定のオブジェクトへのアクセス権限を特定のグループのユーザーに与えたり、取り消したりするためのコマンド・インターフェースを提供します。 あるリソースへのアクセスを認める決定は OAM が行い、キュー・マネージャーはその決定に従います。 OAM が決定できない場合は、キュー・マネージャーは該当のリソースへのアクセスを妨げます。

OAM により、以下を制御することができます。

- MQI を介した IBM MQ 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、要求された操作に関する許可を、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。

  特に、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。

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。 また、一部のグループには、あるキューの読み取りおよび書き込みの権限は与えるが、そのキューの変更または削除の権限は与えないということもできます。

IBM MQ for IBM i のコマンドおよび IBM MQ for IBM i オブジェクトに対する操作の実行

IBM i IBM i 上の IBM MQ 権限

IBM MQ のオブジェクトにアクセスするには、コマンドを発行したり、参照されるオブジェクトにアクセスしたりするための権限が必要になります。 管理者は、IBM MQ のすべてのリソースにアクセスできます。

IBM MQ オブジェクトへのアクセスは、次の権限により制御されます。

1. IBM MQ コマンドの発行
2. コマンドにより参照される IBM MQ オブジェクトへのアクセス

IBM MQ for IBM i のすべての CL コマンドは出荷時に QMQM を所有者として提供され、管理プロファイル (QMQMADM) は *PUBLIC アクセス権限が *EXCLUDE に設定された *USE 権限を持ちます。

注 : QSRDUPER プログラムは、IBM MQ for IBM i ライセンス・プログラム・インストーラーによって、QSYS 内のコマンド (*CMD) オブジェクトを複製するために使用されます。 IBM i V5R4 以降では QSRDUPER プログラムが変更され、デフォルトの動作で、元のコマンドの複製ではなくプロキシー・コマンドが作成さ

れるようになりました。 プロキシー・コマンドは属性 PRX を持ち、コマンド実行を別のコマンドにリダイレクトします。 コピーされるコマンドと同じ名前のプロキシー・コマンドがライブラリー QSYS に存在する場合、プロキシー・コマンドに対する専用権限は製品ライブラリーのコマンドには付与されません。 QSYS 内のプロキシー・コマンドのプロンプト送出または実行を試行すると、製品ライブラリー内のターゲット・コマンドの権限が検査されます。 このため、*CMD オブジェクトに対する権限の変更は、製品ライブラリー (QMQM) 内で行う必要があり、QSYS では権限を変更する必要はありません。 以下に例を示します。

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

製品の CL コマンドの一部において権限構造が変更されました。それらの変更を加えるのに必要となる、IBM MQ オブジェクトに対する OAM 権限があれば、それらのコマンドをパブリックに使用できるようになりました。

IBM i で IBM MQ 管理者になるには、 *QMQMADM* グループのメンバーでなければなりません。 このグループのプロパティーは、UNIX、Linux システムおよび Windows システムにおける mqm グループのプロパティーと類似しています。 特に、IBM MQ for IBM i のインストール時に QMQMADM グループが作成され、その QMQMADM グループのメンバーには、システム上の IBM MQ のすべてのリソースに対するアクセス権限が付与されます。 *ALLOBJ 権限がある場合は、すべての IBM MQ リソースにもアクセスできます。

管理者は、IBM MQ を管理する CL コマンドを使用できます。 それらのコマンドの 1 つに GRTMQMAUT がありますが、これは他のユーザーに権限を付与するために使用されるものです。 別のコマンド STRMQMMQSC は、管理者がローカル・キュー・マネージャーに対して MQSC コマンドを発行するためのものです。

関連概念

83 ページの『IBM i 上の IBM MQ を管理する権限』

IBM i *IBM i 上の IBM MQ オブジェクトのアクセス権限*

IBM MQ CL コマンドの実行に必要なアクセス権限。

IBM MQ for IBM i では、この製品の CL コマンドを次の 2 つのグループに分類しています。

グループ 1

これらのコマンドを処理するには、ユーザーが QMQMADM ユーザー・グループに含まれているか、*ALLOBJ 権限を持っている必要があります。 これらの権限のいずれかを持つユーザーは、他の権限を必要とすることなく、すべてのカテゴリーのすべてのコマンドを処理できます。

注 : これらの権限は、他のすべての OAM 権限を指定変更します。

これらのコマンドは次のようにグループ分けすることができます。

- コマンド・サーバー・コマンド
  - ENDMQMCSVR、IBM MQ コマンド・サーバーの終了
  - STRMQMCSVR、IBM MQ コマンド・サーバーの開始
- 送達不能キュー・ハンドラー・コマンド
  - STRMQMDLQ、IBM MQ 送達不能キュー・ハンドラーの開始
- リスナー・コマンド
  - ENDMQMLSR、IBM MQ リスナーの終了
  - STRMQMLSR、非オブジェクト・リスナーの開始
- メディア回復コマンド
  - RCDMQMIMG、 IBM MQ オブジェクト・イメージの記録
  - RCRMQMOBJ、IBM MQ オブジェクトの再作成
  - WRKMQMTRN、IBM MQ トランザクションの処理
- キュー・マネージャー・コマンド
  - CRTMQM、メッセージ・キュー・マネージャーの作成

- DLTMQM、メッセージ・キュー・マネージャーの削除
- ENDMQM、メッセージ・キュー・マネージャーの終了
- STRMQM、メッセージ・キュー・マネージャーの開始

- Security Commands
  - GRTMQMAUT、IBM MQ オブジェクト権限の認可
  - RVKMQMAUT、IBM MQ オブジェクト権限の取り消し
- トレース・コマンド
  - TRCMQM、IBM MQ ジョブのトレース
- トランザクション・コマンド
  - RSVMQMTRN、IBM MQ トランザクションの解決
- トリガー・モニター・コマンド
  - STRMQMTRM、トリガー・モニターの開始
- IBM MQSC コマンド
  - RUNMQSC、IBM MQSC コマンドの実行
  - STRMQMMQSC、IBM MQSC コマンドの開始

グループ 2

その他のコマンドで、次の 2 レベルの権限が必要です。

1. コマンドを実行するための IBM i 権限。 IBM MQ 管理者は、**GRTOBJAUT** コマンドを使用してこれを設定し、ユーザーまたはユーザー・グループの *PUBLIC(*EXCLUDE) 制限を指定変更します。

   以下に例を示します。

   ```
   GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
   ```

2. ステップ 1 で正しい IBM i 権限を付与された、コマンドに関連付けられた IBM MQ オブジェクトを操作するための IBM MQ 権限。

   この権限は、必要なアクションに適した OAM 権限を持つユーザーによって制御され、IBM MQ 管理者が **GRTMQMAUT** コマンドを使用して設定します。

   以下に例を示します。

   ```
   GRTMQMAUT *connect authority to the queue manager + *admchg authority to
           the queue
   ```

これらのコマンドは次のようにグループ分けすることができます。

- チャネル・コマンド
  - CHGMQMCHL、IBM MQ チャネルの変更

    キュー・マネージャーに対する *connect 権限、およびチャネルに対する *admchg 権限が必要です。
  - CPYMQMCHL、IBM MQ チャネルのコピー

    キュー・マネージャーに対する *connect および *admcrt 権限、 コピーされるデフォルトのチャネル・タイプに対する *admdsp 権限、およびチャネル・オブジェクト・クラスに対する *admcrt 権限が必要です。

    例えば、Sender チャネルをコピーするには、SYSTEM.DEF.SENDER チャネルに対する *admdsp 権限が必要です。
  - CRTMQMCHL、IBM MQ チャネルの作成

キュー・マネージャーに対する *connect および *admcrt 権限、 作成されるデフォルトのチャネル・タイプに対する *admdsp 権限、チャネル・オブジェクト・クラスに対する *admcrt 権限が必要です。

例えば、Sender チャネルを作成するには、SYSTEM.DEF.SENDER チャネルに対する *admdsp 権限が必要です。

- DLTMQMCHL、IBM MQ チャネルの削除

  キュー・マネージャーに対する *connect 権限、および チャネルに対する *admdlt 権限が必要です。

- RSVMQMCHL、IBM MQ チャネルの解決

  キュー・マネージャーに対する *connect 権限、およびチャネルに対する *ctrlx 権限が必要です。

- 表示コマンド

  DSP コマンドを処理するには、ユーザーにキュー・マネージャーに対する *connect および *admdsp 権限を、次にリストする特定のオプションを指定して付与する必要があります。

  - DSPMQM、メッセージ・キュー・マネージャーの表示
  - DSPMQMAUT、IBM MQ オブジェクト権限の表示
  - DSPMQMAUTI、IBM MQ 認証情報の表示 - 認証情報オブジェクトに対する *admdsp
  - DSPMQMCHL、IBM MQ チャネルの表示 - チャネルに対する *admdsp
  - DSPMQMCSVR、IBM MQ コマンド・サーバーの表示
  - DSPMQMNL、IBM MQ 名前リストの表示 - 名前リストに対する *admdsp
  - DSPMQMOBJN、IBM MQ オブジェクト名の表示
  - DSPMQMPRC、IBM MQ プロセスの表示 - プロセスに対する *admdsp
  - DSPMQMQ、IBM MQ キューの表示 - キューに対する *admdsp
  - DSPMQMTOP、IBM MQ トピックの表示 - トピックに対する *admdsp

- コマンドの処理

  WRK コマンドを処理し、オプション・パネルを表示するには、ユーザーにキュー・マネージャーに対する *connect および *admdsp 権限を、次にリストする特定のオプションを指定して付与しなければなりません。

  - WRKMQM、メッセージ・キュー・マネージャーの処理
  - WRKMQMAUT、IBM MQ オブジェクト権限の処理
  - WRKMQMAUTD、IBM MQ オブジェクト権限データの処理
  - WRKMQMAUTI、IBM MQ 認証情報の処理
    - 「IBM MQ 認証情報オブジェクトの変更」コマンドの場合は *admchg。
    - 「IBM MQ 認証情報オブジェクトの作成およびコピー」コマンドの場合は *admcrt。
    - 「IBM MQ 認証情報オブジェクトの削除」コマンドの場合は *admdlt。
    - IBM MQ 認証情報オブジェクトの表示コマンドについては、 *admdsp を参照してください。
  - WRKMQMCHL、IBM MQ チャネルの処理

    以下の権限が必要です。

    - IBM MQ チャネルの変更コマンドの場合は *admchg。
    - *admclr (Clear IBM MQ Channel コマンドの場合)。
    - *admcrt (Create および Copy IBM MQ Channel コマンドの場合)。
    - IBM MQ チャネルの削除コマンドの場合は *admdlt。
    - *admdsp (Display IBM MQ Channel コマンドの場合)。
    - IBM MQ チャネル開始コマンドの場合は *ctrl。

- IBM MQ チャネル終了コマンドの場合は *ctrl 。
- *ctrl (Ping IBM MQ チャネル・コマンドの場合)。
- IBM MQ チャネルのリセット・コマンドの場合は *ctrlx 。
- 「IBM MQ チャネルの解決」コマンドの *ctrlx 。

– WRKMQMCHST、IBM MQ チャネル状況の処理

チャネルに対する *admdsp 権限が必要です。

– WRKMQMCL、IBM MQ クラスターの処理
– WRKMQMCLQ、IBM MQ クラスター・キューの処理
– WRKMQMCLQM、IBM MQ クラスター・キュー・マネージャーの処理
– WRKMQMLSR、IBM MQ リスナーの処理
– WRKMQMMSG、IBM MQ メッセージの処理

これには、キューに対する *browse 権限が必要です。

– WRKMQMNL、IBM MQ 名前リストの処理

以下の権限が必要です。

- *admchg (Change IBM MQ Namelist コマンドの場合)。
- 「IBM MQ 名前リストの作成およびコピー」コマンドの場合は *admcrt 。
- IBM MQ 名前リストの削除コマンドの場合は *admdlt 。
- Display IBM MQ Namelist コマンドの *admdsp 。

– WRKMQMPRC、IBM MQ プロセスの処理

以下の権限が必要です。

- IBM MQ プロセスの変更コマンドの場合は *admchg 。
- Create and Copy IBM MQ Process コマンドの場合は *admcrt 。
- IBM MQ プロセスの削除コマンドの場合は *admdlt 。
- Display IBM MQ Process コマンドの場合は *admdsp 。

– WRKMQMQ、IBM MQ キューの処理

以下の権限が必要です。

- IBM MQ キューの変更コマンドの場合は *admchg 。
- IBM MQ キューのクリア・コマンドの場合は *admclr 。
- *admcrt (Create and Copy IBM MQ Queue コマンドの場合)
- IBM MQ キューの削除コマンドの場合は *admdlt 。
- IBM MQ キューの表示コマンドの場合は *admdsp 。

– WRKMQMQSTS、IBM MQ キュー状況の処理
– WRKMQMTOP、IBM MQ トピックの処理

以下の権限が必要です。

- IBM MQ トピックの変更コマンドの場合は *admchg 。
- IBM MQ トピックの作成およびコピー・コマンドについては、 *admcrt
- IBM MQ トピックの削除コマンドについては、 *admdlt を参照してください。
- *admdsp (Display IBM MQ Topic コマンドの場合)。

– WRKMQMSUB、IBM MQ サブスクリプションの処理

• その他のチャネル・コマンド

チャネル・コマンドを処理するには、次にリストする特定の権限をユーザーに付与しなければなりません。

- ENDMQMCHL、IBM MQ チャネルの終了

  キュー・マネージャーに対する *connect 権限 、およびチャネルに関連付けられた送信キューに対する *allmqi 権限 が必要です。

- ENDMQMLSR、IBM MQ リスナーの終了

  キュー・マネージャーに対する *connect 権限、および名前付きのリスナー・オブジェクトに対する *ctrl 権限が必要です。

- PNGMQMCHL、IBM MQ チャネルの ping

  キュー・マネージャーに対する *connect 権限と *inq 権限、およびチャネル・オブジェクトに対する *ctrl 権限が必要です。

- RSTMQMCHL、IBM MQ チャネルのリセット

  キュー・マネージャーに対する *connect 権限が必要です。

- STRMQMCHL、IBM MQ チャネルの開始

  キュー・マネージャーに対する *connect 権限、およびチャネル・オブジェクトに対する *ctrl 権限が必要です。

- STRMQMCHLI、IBM MQ チャネル・イニシエーターの開始

  キュー・マネージャーには *connect および *inq 権限が、チャネルの伝送キューに関連した開始キューには *allmqi 権限が必要です。

- STRMQMLSR、IBM MQ リスナーの開始

  キュー・マネージャーに対する *connect 権限、および名前付きのリスナー・オブジェクトに対する *ctrl 権限が必要です。

- その他のコマンド

以下のコマンドを処理するには、次にリストする特定の権限をユーザーに付与しなければなりません。

- CCTMQM、メッセージ・キュー・マネージャーへの接続

  IBM MQ オブジェクト権限を必要としません。

- CHGMQM、メッセージ・キュー・マネージャーの変更

  これには、キュー・マネージャーに対する *connect 権限と *admchg 権限が必要です。

- CHGMQMAUTI、IBM MQ 認証情報の変更

  キュー・マネージャーに対する *connect 権限、および認証情報オブジェクトに対する *admchg および *admdsp 権限が必要です。

- CHGMQMNL、IBM MQ 名前リストの変更

  キュー・マネージャーに対する *connect 権限、および 名前リストに対する *admchg 権限が必要です。

- CHGMQMPRC、IBM MQ プロセスの変更

  キュー・マネージャーに対する *connect 権限、および プロセスに対する *admchg 権限が必要です。

- CHGMQMQ、IBM MQ キューの変更

  キュー・マネージャーに対する *connect 権限、および キューに対する *admchg 権限が必要です。

- CLRMQMQ、IBM MQ キューの消去

キュー・マネージャーに対する *connect 権限、および キューに対する *admclr 権限が必要です。

- CPYMQMAUTI、IBM MQ 認証情報のコピー

  キュー・マネージャーに対する *connect 権限、認証情報オブジェクトに対する *admdsp 権限、および認証情報オブジェクト・クラスに対する *admcrt 権限が必要です。

- CPYMQMNL、IBM MQ 名前リストのコピー

  これには、キュー・マネージャーに対する *connect 権限と *admcrt 権限が必要です。

- CPYMQMPRC、IBM MQ プロセスのコピー

  これには、キュー・マネージャーに対する *connect 権限と *admcrt 権限が必要です。

- CPYMQMQ、IBM MQ キューのコピー

  これには、キュー・マネージャーに対する *connect 権限と *admcrt 権限が必要です。

- CRTMQMAUTI、IBM MQ 認証情報の作成

  キュー・マネージャーに対する *connect 権限、認証情報オブジェクトに対する *admdsp 権限、および認証情報オブジェクト・クラスに対する *admcrt 権限が必要です。

- CRTMQMNL、IBM MQ 名前リストの作成

  そのためには、キュー・マネージャーに対する *connect 権限と *admcrt 権限、およびデフォルトの名前リストに対する *admdsp 権限が必要です。

- CRTMQMPRC、IBM MQ プロセスの作成

  これには、キュー・マネージャーに対する *connect 権限と *admcrt 権限、およびデフォルト・プロセスに対する *admdsp 権限が必要です。

- CRTMQMQ、IBM MQ キューの作成

  これには、キュー・マネージャーに対する *connect 権限と *admcrt 権限、およびデフォルト・キューに対する *admdsp 権限が必要です。

- CVTMQMDTA、IBM MQ データ・タイプ・コマンドの変換

  IBM MQ オブジェクト権限を必要としません。

- DLTMQMAUTI、IBM MQ 認証情報の削除

  キュー・マネージャーに対する *connect 権限、および認証情報オブジェクトに対する *ctrlx 権限が必要です。

- DLTMQMNL、IBM MQ 名前リストの削除

  キュー・マネージャーに対する *connect 権限、および 名前リストに対する *admdlt 権限が必要です。

- DLTMQMPRC、IBM MQ プロセスの削除

  キュー・マネージャーに対する *connect 権限、および プロセスに対する *admdlt 権限が必要です。

- DLTMQMQ、IBM MQ キューの削除

  キュー・マネージャーに対する *connect 権限、および キューに対する *admdlt 権限が必要です。

- DSCMQM、メッセージ・キュー・マネージャーからの切断

  IBM MQ オブジェクト権限を必要としません。

- RFRMQMAUT、セキュリティーのリフレッシュ

  キュー・マネージャーに対する *connect 権限が必要です。

- RFRMQMCL、クラスターのリフレッシュ

  キュー・マネージャーに対する *connect 権限が必要です。

– RSMMQMCLQM、クラスター・キュー・マネージャーの再開

  キュー・マネージャーに対する *connect 権限が必要です。
– RSTMQMCL、クラスターのリセット

  キュー・マネージャーに対する *connect 権限が必要です。
– SPDMQMCLQM、クラスター・キュー・マネージャーの中断

  キュー・マネージャーに対する *connect 権限が必要です。

IBM i *IBM i でのアクセス許可*

この情報は、アクセス許可に関係するさまざまなコマンドについて理解するために使用します。

GRTMQMAUT および RVKMQMAUT コマンド上の AUT キーワードによって定義される許可は、次のように類別できます。

- MQI 呼び出しに関する許可
- 許可に関する管理コマンド
- Context authorizations
- 一般許可、すなわち、MQI 呼び出しまたはコマンドに関するもの、あるいはその両方に関するもの

次の表は、MQI 呼び出し、コンテキスト呼び出し、MQSC および PCF コマンド、および一般操作の、AUT パラメーターを使用するさまざまな権限をリストしています。

表 *15. MQI* 呼び出しについての許可

AUT	**説明**
*ALTUSR	MQOPEN および MQPUT1 呼び出しに対して、他のユーザーの権限を使用できる。
*BROWSE	BROWSE オプションを指定した MQGET 呼び出しを発行して、キューからメッセージを取り出す。
*CONNECT	MQCONN 呼び出しを発行して、指定のキュー・マネージャーにアプリケーションを接続する。
*GET	MQGET 呼び出しを発行して、キューからメッセージを取り出す。
*INQ	MQINQ 呼び出しを発行して、特定のキューの照会を行う。
*PUB	トピックを開き、MQPUT 呼び出しを使用してメッセージをパブリッシュする。
*PUT	MQPUT 呼び出しを発行して、特定のキューにメッセージを書き込む。
*RESUME	MQSUB 呼び出しを使用して、サブスクリプションを再開する。
*SET	MQSET 呼び出しを発行して、MQI からキューに属性を設定する。 複数のオプションを適用するようにキューをオープンする場合は、各オプションについての許可を持っている必要があります。
*SUB	MQSUB 呼び出しを使用して、トピックへのサブスクリプションを作成、変更、または再開する。

表 *16.* コンテキスト呼び出しについての許可

AUT	**説明**
*PASSALL	すべてのコンテキストを指定のキューに渡す。 すべてのコンテキスト・フィールドが元の要求からコピーされます。
*PASSID	アイデンティティー・コンテキストを指定のキューに渡す。 アイデンティティー・コンテキストは、要求のアイデンティティー・コンテキストと同じです。

表 16. コンテキスト呼び出しについての許可 (続き)	
AUT	**説明**
*SETALL	すべてのコンテキストを指定のキューに設定する。 これは特別なシステム・ユーティリティーによって使用されます。
*SETID	アイデンティティー・コンテキストを指定のキューに設定する。 これは特別なシステム・ユーティリティーによって使用されます。

表 17. *MQSC* および *PCF* 呼び出しについての許可	
AUT	**説明**
*ADMCHG	指定のオブジェクトの属性を変更する。
*ADMCLR	指定のオブジェクトをクリアする (PCF の「オブジェクトのクリア」コマンドのみ)。
*ADMCRT	指定のタイプのオブジェクトを作成する。
*ADMDLT	指定のオブジェクトを削除する。
*ADMDSP	指定のオブジェクトの属性を表示する。

表 18. 一般操作についての許可	
AUT	**説明**
*ALL	オブジェクトに適用可能なすべての操作を使用する。 all 権限は、オブジェクト・タイプに該当する権限 alladm、allmqi、および system の和集合と同等です。
*ALLADM	オブジェクトに適用可能なすべての管理操作を実行する。
*ALLMQI	オブジェクトに適用可能なすべての MQI 呼び出しを使用する。
*CTRL	チャネル、リスナー、およびサービスの開始とシャットダウンの制御
*CTRLX	シーケンス番号をリセットし、未確定チャネルを解決する。

IBM i *IBM i でのアクセス許可コマンドの使用*

この情報は、アクセス許可コマンドについて学習したり、コマンドの例を使用したりするのに使用します。

GRTMQMAUT コマンドの使用

必要な許可を持っている場合は、GRTMQMAUT コマンドを使用すると、特定のオブジェクトにアクセスする認可をユーザー・プロファイルまたはユーザー・グループに与えることができます。 次の例は、GRTMQMAUT コマンドを使用する方法を示しています。

1.
```
GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
          AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
```

   この例のそれぞれの指定の意味は次のとおりです。

   - RED.LOCAL.QUEUE は、オブジェクト名です。
   - *LCLQ (ローカル・キュー) は、オブジェクト・タイプです。
   - GROUPA は、許可が変更される、システム上のユーザー・プロファイルの名前です。 このプロファイルは、他のユーザーの管理者用のグループ・プロファイルとして使用できます。
   - *BROWSE と *PUT は、特定のキューに与えられる許可です。

     *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。

*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。

- saturn.queue.manager は、キュー・マネージャー名です。

2. 次のコマンドは、ユーザー JACK と JILL に、デフォルトのキュー・マネージャーについての、すべてのプロセス定義に対して適用可能なすべての許可を与えます。

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```

3. 次のコマンドは、ユーザー GEORGE に、キュー・マネージャー TRENT 上のキュー ORDERS にメッセージを書き込む権限を与えます。

```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

RVKMQMAUT コマンドの使用

必要な許可を持っている場合は、RVKMQMAUT コマンドを使用すると、特定のオブジェクトにアクセスするために以前に与えた許可を、ユーザー・プロファイルまたはユーザー・グループから除去することができます。 次の例は、RVKMQMAUT コマンドを使用する方法を示しています。

1.
```
RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
AUT(*PUT) MQMNAME('saturn.queue.manager')
```

前の例で与えた、指定したキューにメッセージを書き込む権限は、GROUPA については取り消されます。

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```

キュー・マネージャー PAYROLLQM によって所有され、PAY という文字で始まる名前のすべてのキューからメッセージを入手する権限は、システムのすべてのユーザーから取り消されます。ただし、ユーザーやユーザーが属するグループに対して個別に許可が与えられている場合は取り消されません。

DSPMQMAUT コマンドの使用

MQM 権限の表示 (DSPMQMAUT) コマンドは、指定されたオブジェクトおよびユーザーについて、ユーザーがオブジェクトについて持っている許可のリストを表示します。 次の例は、このコマンドの使い方を示しています。

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

RFRMQMAUT コマンドの使用

「MQM セキュリティーのリフレッシュ (RFRMQMAUT)」コマンドでは、キュー・マネージャーを停止して再開する必要なしに、変更内容をオペレーティング・システム・レベルで反映して、OAM の許可グループ情報をただちに更新できます。 次の例は、このコマンドの使い方を示しています。

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i 許可指定表 (IBM i)

この情報は、キュー・オブジェクト、プロセス・オブジェクト、およびキュー・マネージャー・オブジェクトに対する特定の API 呼び出し、およびそれらの呼び出しの特定のオプションを使用するためにどんな許可が必要か調べるのに使用します。

163 ページの表 19 以降の許可指定表には、許可の機能と、適用される制限が正確に定義されています。 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、次のデータを指定する 1 組のテーブルとして情報を示します。

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、プロセス定義、キュー・マネージャー、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。

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 **GRTMQMAUT** および **RVKMQMAUT** 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、キーワード MQZAO_SET_ALL_CONTEXT はキーワード *SETALL に、というふうに対応します。 これらの定数は、製品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 authorizations

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、 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。名前解決の過程で最初に検出された定義について検査が行われます。この定義は、キュー・マネージャー別名定義が直接オープンされる場合以外はキュー・マネージャー別名ではない定義です。つまり、オブジェクト記述子の *ObjectName* フィールドに現れた名前について検査が行われます。 特定のオブジェクトをオープンするには必ず権限が必要です。さらに、キューに依存しない権限 (キュー・マネージャー・オブジェクトに関する許可を介して取得する) が必要なこともあります。

163 ページの表 19、164 ページの表 20、164 ページの表 21、および 165 ページの表 22 は、それぞれの呼び出しに必要な許可を要約しています。

注 : 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ついては、これらの表に記載されていません。 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。

表 *19. 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 (165 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 オプション	適用外	適用外	MQZAO_CONNECT

表 *20. 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 (165 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 **(165 ページの『2』)**	MQZAO_INQUIRE **(165 ページの『2』)**	MQZAO_INQUIRE **(165 ページの『2』)**
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 **(165 ページの『3』)**	MQZAO_INPUT	適用外	適用外
MQOO_OUTPUT (通常キュー) **(165 ページの『4』)**	MQZAO_OUTPUT	適用外	適用外
MQOO_PASS_ IDENTITY_CONTEXT **(165 ページの『5』)**	MQZAO_PASS_ IDENTITY_CONTEXT	適用外	検査しない
MQOO_PASS_ALL_ CONTEXT **(165 ページの『5』, 165 ページの『6』)**	MQZAO_PASS_ ALL_CONTEXT	適用外	検査しない
MQOO_SET_ IDENTITY_CONTEXT **(165 ページの『5』, 165 ページの『6』)**	MQZAO_SET_ IDENTITY_CONTEXT	適用外	MQZAO_SET_ IDENTITY_CONTEXT **(165 ページの『7』)**
MQOO_SET_ ALL_CONTEXT **(165 ページの『5』, 165 ページの『8』)**	MQZAO_SET_ ALL_CONTEXT	適用外	MQZAO_SET_ ALL_CONTEXT **(165 ページの『7』)**
MQOO_OUTPUT (伝送キュー) **(165 ページの『9』)**	MQZAO_SET_ ALL_CONTEXT	適用外	MQZAO_SET_ ALL_CONTEXT **(165 ページの『7』)**
MQOO_SET	MQZAO_SET	適用外	検査しない
MQOO_ALTERNATE_ USER_AUTHORITY	**(165 ページの『10』)**	**(165 ページの『10』)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(165 ページの『10』、165 ページの『11』)**

表 *21. 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 (165 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 **(166 ページの『12』)**	適用外	検査しない
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(166 ページの『12』)**	適用外	検査しない
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(166 ページの『12』)**	適用外	MQZAO_SET_ IDENTITY_CONTEXT **(165 ページの『7』)**

表 *21. 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 (165 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 **(166 ページの『12』)**	適用外	MQZAO_SET_ ALL_CONTEXT **(165 ページの『7』)**
(伝送キュー) **(165 ページの『9』)**	MQZAO_SET_ ALL_CONTEXT	適用外	MQZAO_SET_ ALL_CONTEXT **(165 ページの『7』)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(166 ページの『13』)**	適用外	MQZAO_ALTERNATE_ USER_AUTHORITY **(165 ページの『11』)**

表 *22. 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 (165 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 **(166 ページの『14』)**	適用外	適用外
MQCO_DELETE _PURGE	MQZAO_DELETE **(166 ページの『14』)**	適用外	適用外

表の注:

1. モデル・キューがオープンされる場合は、次のようになります。
   - 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. オープンされるオブジェクトのタイプに応じて、キュー、プロセス、名前リスト、またはキュー・マネージャー・オブジェクトのいずれかが検査されます。
3. MQOO_INPUT_* も指定する必要があります。 このオプションは、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。
4. この検査は、注 165 ページの『9』 に示した場合以外は、すべての場合の出力において実行されます。
5. MQOO_OUTPUT も指定する必要があります。
6. このオプションは、MQOO_PASS_IDENTITY_CONTEXT も暗黙的に指定されます。
7. 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。
8. 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。
9. 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。
10. MQOO_INQUIRE (あらゆるオブジェクト・タイプの場合)、または (キューの場合) MQOO_BROWSE、MQOO_INPUT_*、MQOO_OUTPUT、または MQOO_SET 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。
11. この許可では、任意の *AlternateUserId* を指定できます。

12. 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。
13. 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、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。
14. 検査は、次の記述が両方とも当てはまる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 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。

全体の注:

1. 特殊許可 MQZAO_ALL_MQI には、オブジェクト・タイプに関係する次の許可がすべて含まれます。
    - MQZAO_CONNECT
    - MQZAO_INQUIRE
    - MQZAO_SET
    - MQZAO_BROWSE
    - MQZAO_INPUT
    - MQZAO_OUTPUT
    - MQZAO_PASS_IDENTITY_CONTEXT
    - MQZAO_PASS_ALL_CONTEXT
    - MQZAO_SET_IDENTITY_CONTEXT
    - MQZAO_SET_ALL_CONTEXT
    - MQZAO_ALTERNATE_USER_AUTHORITY
2. MQZAO_DELETE (注 166 ページの『14』 を参照) および MQZAO_DISPLAY は、管理許可として分類されます。 したがって、MQZAO_ALL_MQI には含まれません。
3. 「検査しない」は、許可検査が行われないことを意味します。
4. 「適用外」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。 例えば、プロセス・オブジェクトに MQPUT 呼び出しを発行できません。

IBM i *IBM i でのエスケープ PCF の中の MQSC コマンドに関する許可*

これらの許可を付与されたユーザーは、管理コマンドをエスケープ PCF メッセージとして発行できます。こうした方法を使用して、プログラムは、管理コマンドを管理ユーザーに代わって実行させるためにメッセージとしてキュー・マネージャーに送ることができます。

このセクションには、エスケープ PCF に含まれる各 MQSC コマンドに必要な権限についての要約が示されています。

「適用外」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。

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 DISPLAY 権限
- エスケープ PCF コマンドのテキスト内の MQSC コマンドを実行する権限

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

オブジェクト	必要な権限
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

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

DEFINE *object* NOREPLACE (170 ページの『1』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (170 ページの『2』)
トピック	MQZAO_CREATE (170 ページの『2』)
プロセス	MQZAO_CREATE (170 ページの『2』)
キュー・マネージャー	適用外
名前リスト	MQZAO_CREATE (170 ページの『2』)
認証情報	MQZAO_CREATE (170 ページの『2』)
チャネル	MQZAO_CREATE (170 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (170 ページの『2』)
リスナー	MQZAO_CREATE (170 ページの『2』)
サービス	MQZAO_CREATE (170 ページの『2』)

DEFINE 「オブジェクト」 REPLACE (170 ページの『1』 170 ページの『3』)

オブジェクト	必要な権限
キュー	MQZAO_CHANGE

オブジェクト	必要な権限
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

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

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	
サービス	

PING CHANNEL

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	適用外
サービス	適用外

RESET CHANNEL

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_EXTENDED
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

RESOLVE CHANNEL

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_EXTENDED
クライアント接続チャネル	適用外
リスナー	適用外

オブジェクト	必要な権限
サービス	適用外

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

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

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。 GRTMQMAUT 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対して作成権限が与えられます。
3. このオプションは、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。

IBM i *IBM i での PCF コマンドについての許可*

これらの許可を付与されたユーザーは、管理コマンドを PCF コマンドとして発行できます。 こうした方法を使用して、プログラムは、管理コマンドを管理ユーザーに代わって実行させるためにメッセージとしてキュー・マネージャーに送ることができます。

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、権限の検査がこの操作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 DISPLAY 権限

特殊権限 MQZAO_ALL_ADMIN には、以下の権限が含まれます。

- MQZAO_CHANGE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY
- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

MQZAO_CREATE は、特定のオブジェクトまたはオブジェクト・タイプに固有ではないため、これには含まれません。

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

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

オブジェクト	必要な権限
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

Copy *object* (置き換えなし) (176 ページの『1』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (176 ページの『2』)
トピック	MQZAO_CREATE (176 ページの『2』)
プロセス	MQZAO_CREATE (176 ページの『2』)
キュー・マネージャー	適用外
NamelistMQZAO_CREATE	MQZAO_CREATE (176 ページの『2』)
認証情報	MQZAO_CREATE (176 ページの『2』)
チャネル	MQZAO_CREATE (176 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (176 ページの『2』)
リスナー	MQZAO_CREATE (176 ページの『2』)
サービス	MQZAO_CREATE (176 ページの『2』)

「オブジェクト」 のコピー (置換を伴う) (176 ページの『1』 176 ページの『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

Create *object* (置き換えなし) (176 ページの『3』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (176 ページの『2』)
トピック	MQZAO_CREATE (176 ページの『2』)
プロセス	MQZAO_CREATE (176 ページの『2』)
キュー・マネージャー	適用外

オブジェクト	必要な権限
名前リスト	MQZAO_CREATE (176 ページの『2』)
認証情報	MQZAO_CREATE (176 ページの『2』)
チャネル	MQZAO_CREATE (176 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (176 ページの『2』)
リスナー	MQZAO_CHANGE
サービス	MQZAO_CHANGE

「オブジェクト」 の作成 (置換を伴う) (176 ページの『3』 176 ページの『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

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	MQZAO_DELETE
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

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

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

Ping Channel

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	適用外
サービス	適用外

Reset Channel

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_EXTENDED
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

Reset Queue Statistics

オブジェクト	必要な権限
キュー	MQZAO_DISPLAY および MQZAO_CHANGE
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	適用外
クライアント接続チャネル	適用外
リスナー	
サービス	

Resolve Channel

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_EXTENDED
クライアント接続チャネル	適用外
リスナー	適用外

オブジェクト	必要な権限
サービス	適用外

Start Channel

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	適用外
サービス	適用外

Stop Channel

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	適用外
サービス	適用外

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。 GRTMQMAUT 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対し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. このオプションは、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。

IBM i IBM i における総称 OAM プロファイル

オブジェクト権限マネージャー (OAM) 総称プロファイルを使用すると、各オブジェクトが個別に作成されるたびに別個の **GRTMQMAUT** コマンドを発行するのではなく、多くのオブジェクトに対する権限を一度にユ

ーザーに対して設定することができます。 **GRTMQMAUT** コマンドで総称プロファイルを使用すると、今後作成される、そのプロファイルに適したオブジェクトすべてに総称権限を設定できます。

このセクションの後半では、総称プロファイルの使用法をより詳しく説明します。

- 177 ページの『ワイルドカード文字の使用』
- 177 ページの『プロファイルの優先順位』

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。このため、ABC.?EF と指定すると、プロファイルに付与した許可が、ABC.DEF、ABC.CEF、ABC.BEF などの名前で作成されたオブジェクトすべてに適用されます。

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 はオブジェクト AB.CD、AB.ED、および AB.FD に該当します。

*****
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 に一致 します。 (この方法で使用する * は、必ず修飾子 1 つを示すため、ABC.JKL には一致**しない**ことに注意してください。)
- 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 に該当します。

******
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、キーワード OBJTYPE (*PRC) を使用してプロセスを識別する場合、** をプロファイル名として使用して、すべてのプロセスに対する許可を変更します。
- 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。

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。

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

最初のものは、プロファイル AB. * と一致する名前を持つプリンシパル FRED のすべてのキューに対する書き込み権限を与えます。 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・マッチングの規則に従って、いずれかの GRTMQMAUT 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。 違いが検出される場所では、常に非総称文字のほうが総称文字よりも限定的です。 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. **

IBM i IBM i でのインストール済み許可サービスの指定

使用する許可サービス・コンポーネントを指定することができます。

パラメーター **Service Component name** を **GRTMQMAUT** および **RVKMQMAUT** に使用すると、インストール済み許可サービス・コンポーネントの名前を指定できます。

開始パネルで **F24** を選択し、いずれかのコマンドの次のパネルで **F9=All parameters** を使用すると、インストール済み許可コンポーネント (*DFT) または必要な許可サービス・コンポーネント (キュー・マネージャーの qm.ini ファイルのサービス・スタンザに指定されている) を指定できます。

DSPMQMAUT も、この追加パラメーターがあります。 このパラメーターを使用すると、すべてのインストール済み許可コンポーネント (*DFT)、または指定された許可サービス・コンポーネント名に、指定されたオブジェクト名、オブジェクト・タイプ、およびユーザーが含まれるかどうかを検索します。

IBM i IBM i における権限プロファイルを使用した処理と使用しない処理

この情報は、権限プロファイルを使用した処理方法と、権限プロファイルを使用しない処理方法について学習するのに使用します。

178 ページの『権限プロファイルを使った処理』の説明どおり、権限プロファイルを使用する方法と、次に説明する、権限プロファイルを使用しない方法があります。

権限プロファイルなしで処理するには、**GRTMQMAUT** の Authority パラメーターとして *NONE を使用することにより、権限なしのプロファイルを作成します。 これにより、既存のプロファイルはすべて変更されなくなります。

RVKMQMAUT で、Authority パラメーターとして *REMOVE を使用して、既存の権限プロファイルを除去します。

権限プロファイルを使った処理

権限プロファイルの作成に関連するコマンドは 2 つあります。

- **WRKMQMAUT**
- **WRKMQMAUTD**

これらのコマンドへは、コマンド行から直接アクセスする方法と、WRKMQM パネルから次のようにしてアクセスする方法があります。

1. キュー・マネージャー名を入力し、Enter キーを押して **WRKMQM** 結果パネルにアクセスする。
2. このパネルで F23=More options を選択します。

オプション 24 は、 **WRKMQMAUT** コマンド の結果パネルを選択し、オプション 25 は、SSL バインディング層で使用される **WRKMQMAUTI** コマンドを選択します。

WRKMQMAUT

このコマンドを使用すると、権限キューに保持されている権限データを処理することができます。

注: このコマンドを実行するには、ユーザーがキュー・マネージャーに *connect および *admdsp 権限を持っている必要があります。 ただし、プロファイルの作成または削除には、QMQMADM 権限が必要です。

画面に情報を出力する場合、権限プロファイル名とそのタイプを示したリストが表示されます。 出力を印刷する場合、すべての権限データ、登録済みユーザー、およびそのユーザーが持つ権限を示した詳細リストが出力されます。

このパネルでオブジェクト名またはプロファイル名を入力して ENTER を押すと、 **WRKMQMAUT** の結果パネルが表示されます。

4=Delete を選択すると、新しいパネルが表示されます。このパネルから、指定した総称権限プロファイル名に登録されているすべてのユーザー名を削除することを確認できます。 このオプションは、すべてのユーザーに対してオプション *REMOVE と共に **RVKMQMAUT** を実行し、総称プロファイル名に**のみ**適用されます。

12=Work with profile を選択すると、 179 ページの『WRKMQMAUTD』 で説明されているように、 **WRKMQMAUTD** コマンド結果パネルに移動します。

WRKMQMAUTD

このコマンドを使用すると、特定の権限プロファイル名およびオブジェクト・タイプで登録されたすべてのユーザーを表示できます。 このコマンドを実行するには、ユーザーがキュー・マネージャーに *connect および *admdsp 権限を持っている必要があります。 ただし、プロファイルの付与、実行、作成、または削除には、QMQMADM 権限が必要です。

開始入力パネルから F24=More keys を選択し、次にオプション F9=All Parameters を選択すると、 **GRTMQMAUT** および **RVKMQMAUT** についてサービス・コンポーネント名が表示されます。

注: F11=Display Object Authorizations キーは、以下のタイプの権限を切り替えます。

- オブジェクト許可
- Context authorizations
- MQI authorizations

表示されるオプションは次のとおりです。

2=Grant
現行の権限に追加処理を行うための **GRTMQMAUT** パネルに進みます。

3=Revoke
現行の定義から一部を除去するための **RVKMQMAUT** パネルに進みます。

4=Delete
指定されたユーザーに対する権限データを削除するためのパネルに進みます。 **RVKMQMAUT** をオプション *REMOVE と共に実行します。

5=Display
既存の **DSPMQMAUT** コマンドに進みます。

F6=Create
プロファイル権限レコードを作成するための **GRTMQMAUT** パネルに進みます。

IBM i オブジェクト権限マネージャーのガイドライン (IBM i)

オブジェクト権限マネージャー (OAM) を使用するための追加のヒント

機密操作へのアクセスの制限

一部の操作は重要度が高いので、その実行は特権ユーザーに限ります。 例:

- 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・キューの作成とコピー

キュー・マネージャー・ディレクトリー

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、製品専用です。 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。

キュー

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)。

別名キューまたはリモート・キューの場合、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。 ユーザー・プロファイルに、別名キューへのアクセスを許可し、その解決先のローカル・キューへのアクセスは認めないという場合もあります。

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、一部のユーザーが別名を作成して通常のアクセス管理を逃れる事態が生じます。

代替ユーザー権限

代替ユーザー権限は、あるユーザー・プロファイルが IBM MQ オブジェクトにアクセスしているときに別のユーザー・プロファイルの権限を使用できるかどうかを制御するものです。 この手法は、サーバーがプログラムから要求を受け取り、その要求に必要な権限が確実にプログラムに付与されているようにする上で重要です。 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。

以下に例を示します。

- ユーザー・プロファイル PAYSERV のもとで実行中のサーバーが、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・プロファイル USER1 によってキューに置かれます。
- 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・プロファイル (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・プロファイル (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・プロファイルとして 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・プロファイルは、オブジェクト記述子の *AlternateUserId* フィールドに指定します。

注 : 代替ユーザー・プロファイルは、どの IBM MQ オブジェクトでも使用できます。 代替ユーザー・プロファイルを使用しても、別のリソース管理プログラムが使用するユーザー・プロファイルには影響しません。

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。

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。

コンテキスト・オプションの詳細については、メッセージ・コンテキストを参照してください。

リモート・セキュリティーに関する考慮事項

リモート・セキュリティーについては、以下を考慮します。

書き込む権限
複数のキュー・マネージャーにまたがるセキュリティーについては、チャネルが別のキュー・マネージャーから送られたメッセージを受け取ったときに使用する書き込み権限を指定することができます。

このパラメーターは、RCVR、RQSTR、または CLUSRCVR チャネル・タイプの場合のみ有効です。 チャネル属性 PUTAUT は次のように指定します。

DEF
デフォルト・ユーザー・プロファイル。 これは、メッセージ・チャネル・エージェントを実行するための QMQM ユーザー・プロファイルです。

CTX
メッセージ・コンテキスト内のユーザー・プロファイル。

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。 特別の権限は必要ありません。 しかし、メッセージを伝送キューに直接書き込むには、特殊な許可が必要です。

チャネル出口
チャネル出口は、追加されたセキュリティーに使用されます。

チャネル認証レコード
チャネル・レベルで接続システムに付与されたアクセス権限に対してさらに正確な制御を実行するために使用します。

リモート・セキュリティーの詳細については、「107 ページの『チャネル許可』」を参照してください。

SSL/TLS を使用したチャネルの保護

Transport Layer Security (TLS) 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・セキュリティーが提供されています。 IBM MQ の TLS サポートにより、チャネル定義で特定のチャネルが TLS セキュリティーを使用することを指定できます。 また、使用したい暗号化アルゴリズムなど、望ましいセキュリティーを詳しく指定することもできます。

IBM MQ の TLS サポートでは、キュー・マネージャー認証情報オブジェクト、さまざまな CL コマンドと MQSC コマンド、および必要な TLS サポートを詳細に定義するキュー・マネージャー・パラメーターとチャネル・パラメーターが使用されます。

次の CL コマンドは、TLS をサポートします。

WRKMQMAUTI
認証情報オブジェクトの属性を処理します。

CHGMQMAUTI
認証情報オブジェクトの属性を変更します。

CRTMQMAUTI
認証情報オブジェクトを作成します。

CPYMQMAUTI
既存の認証情報オブジェクトをコピーして、認証情報オブジェクトを作成します。

DLTMQMAUTI
認証情報オブジェクトを削除します。

DSPMQMAUTI
特定の認証情報オブジェクトの属性を表示します。

TLS 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。

- TLS を使用したチャネルの保護

TLS 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

z/OS でのセキュリティーのセットアップ

z/OS に固有のセキュリティーに関する考慮事項。

IBM MQ for z/OS のセキュリティーは、RACF または同等の外部セキュリティー・マネージャー (ESM) を使用して制御されます。

以下の手順は、RACF の使用を前提としています。

関連資料
セキュリティー・シナリオ: z/OS で 2 つのキュー・マネージャーを使用する場合
セキュリティー・シナリオ: z/OS でキュー共有グループを使用する場合

z/OS RACF セキュリティー・クラス

RACF クラスは、 IBM MQ セキュリティー検査に必要なプロファイルを保持するために使用されます。 多くの場合、メンバー・クラスには、それに相当するグループ・クラスがあります。 汎用プロファイルを受け入れるには、それらのクラスをアクティブ化して有効にしなければなりません。

RACF の各クラスには、検査手順のいずれかの時点で使用するプロファイルを 1 つ以上格納します (182 ページの表 23 を参照してください)。

表 *23. RACF* で使用される *IBM MQ* クラス

メンバー・クラス	グループ・クラス	目次
MQADMIN	GMQADMIN	プロファイル: 主に管理タイプ機能のプロファイルを格納するために使用します。 以下に例を示します。 • IBM MQ のセキュリティー・スイッチのためのプロファイル • RESLEVEL セキュリティー・プロファイル • 代替ユーザー・セキュリティーのためのプロファイル • コンテキスト・セキュリティー・プロファイル • コマンド・リソース・セキュリティーのためのプロファイル
MXADMIN	GMXADMIN	プロファイル: 主に管理タイプ機能のプロファイルを格納するために使用します。 以下に例を示します。 • IBM MQ のセキュリティー・スイッチのためのプロファイル • RESLEVEL セキュリティー・プロファイル • 代替ユーザー・セキュリティーのためのプロファイル • コンテキスト・セキュリティー・プロファイル • コマンド・リソース・セキュリティーのためのプロファイル このクラスには、大文字と大/小文字混合の両方の RACF プロファイルを格納できます。
MQCONN		接続セキュリティーのために使用するプロファイル
MQCMDS		コマンド・セキュリティーのために使用するプロファイル
MQQUEUE	GMQQUEUE	キュー・リソース・セキュリティーで使用するプロファイル
MXQUEUE	GMXQUEUE	キュー・リソース・セキュリティーで使用するプロファイル (大/小文字混合と大文字)
MQPROC	GMQPROC	プロセス・リソース・セキュリティーで使用するプロファイル
MXPROC	GMXPROC	プロセス・リソース・セキュリティーで使用するプロファイル (大/小文字混合と大文字)

表 23. *RACF* で使用される *IBM MQ* クラス (続き)

メンバー・クラス	グループ・クラス	目次
MQNLIST	GMQNLIST	名前リスト・リソース・セキュリティーで使用するプロファイル
MXNLIST	GMXNLIST	名前リスト・リソース・セキュリティーで使用するプロファイル (大/小文字混合と大文字)
MXTOPIC	GMXTOPIC	トピック・セキュリティーで使用される、大/小文字混合および大文字プロファイル

一部のクラスには、関連するグループ・クラス があります。グループ・クラスを使用すれば、同じようなアクセス要件のリソースをグループとしてまとめることができます。 メンバー・クラスとグループ・クラスの違い、およびメンバー・クラスまたはグループ・クラスをいつ使用するかについて詳しくは、「z/OS Security Server RACF セキュリティー管理者のガイド」を参照してください。

セキュリティー検査を実行するには、まずクラスをアクティブ化する必要があります。 IBM MQ のすべてのクラスをアクティブ化する場合は、以下の RACF コマンドを使用できます。

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                  MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

さらに、それらのクラスで汎用プロファイルを格納できるようにセットアップすることも必要です。 そのためには、RACF コマンド SETROPTS を使用します。例えば、以下のようなコマンドです。

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                 MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

z/OS RACF プロファイル

IBM MQ によって使用されるすべての RACF プロファイルには、キュー・マネージャー名またはキュー共有グループ名のいずれかである接頭部が含まれています。 ワイルドカードとして % 記号を使用する場合には、注意が必要です。

IBM MQ によって使用されるすべての RACF プロファイルには、接頭部が含まれています。 キュー共有グループ・レベルのセキュリティーであれば、その接頭部はキュー共有グループ名になります。 キュー・マネージャー・レベルのセキュリティーであれば、その接頭部はキュー・マネージャー名になります。 キュー・マネージャー・レベルとキュー共有グループ・レベルのセキュリティーを組み合わせて使用する場合は、両方のタイプの接頭部が付いたプロファイルを使用することになります。 (キュー共有グループ・レベルとキュー・マネージャー・レベルのセキュリティーについては、「IBM MQ for z/OS 概念: セキュリティー」を参照してください。)

例えば、キュー共有グループ QSG1 に含まれている QUEUE_FOR_SUBSCRIBER_LIST というキューをキュー共有グループ・レベルで保護する場合は、該当するプロファイルを以下のような名前で RACF に定義します。

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

キュー・マネージャー STCD に属している QUEUE_FOR_LOST_CARD_LIST というキューをキュー・マネージャー・レベルで保護する場合は、該当するプロファイルを以下のような名前で RACF に定義します。

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

したがって、別々のキュー・マネージャーとキュー共有グループで同じ RACF データベースを共有しながら、別々のセキュリティー・オプションを設定することが可能になります。

想定外のユーザー・アクセスを防止するために、プロファイルで汎用キュー・マネージャー名を使用しないでください。

IBM MQ では、オブジェクト名で % 記号を使用できるようになっています。 一方、RACF では、% 文字を 1 文字のワイルドカードとして使用します。 したがって、オブジェクト名を定義するときに % 文字を名前の中で使用する場合は、対応するプロファイルを定義するときにこの点を考慮に入れる必要があります。

例えば、キュー・マネージャー CRDP に属している CREDIT_CARD_%_RATE_INQUIRY というキューの場合は、プロファイルを以下のような名前で RACF に定義します。

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

このキューを CRDP.** などの汎用プロファイルで保護することはできません。

IBM MQ では、大/小文字混合のオブジェクト名を使用できるようになっています。 そのようなオブジェクトを保護するには、以下のいずれかのプロファイルを定義します。

1. 該当する大/小文字混合の RACF クラスに含まれている大/小文字混合のプロファイル
2. 該当する大文字の RACF クラスに含まれている汎用プロファイル

大/小文字混合プロファイルと大/小文字混合の RACF クラスを使用する場合は、265 ページの『z/OS キュー・マネージャーの大/小文字混合のセキュリティーへの移行』で説明されている手順を実行してください。

IBM MQ によって値が提供されるときに、プロファイルまたはプロファイルの一部が大文字だけになる場合もあります。 次のとおりです。

- スイッチ・プロファイル。
- すべての高位修飾子 (HLQ) (サブシステム ID やキュー共有グループ ID など)。
- SYSTEM オブジェクトのプロファイル。
- デフォルト・オブジェクトのプロファイル。
- **MQCMDS** クラス (したがって、すべてのコマンド・プロファイルは大文字だけになります)。
- **MQCONN** クラス (したがって、すべての接続プロファイルは大文字だけになります)。
- **RESLEVEL** プロファイル。
- コマンド・リソース・プロファイルの 'object' 修飾 (hlq.QUEUE.queuename など)。 リソース名だけが大/小文字混合になります。
- 動的キュー・プロファイル hlq.CSQOREXX.*、hlq.CSQUTIL.*、CSQXCMD.*。
- hlq.CONTEXT.resourcename の 'CONTEXT' 部分。
- hlq.ALTERNATE.USER.userid の 'ALTERNATE.USER' 部分。

例えば、キュー・マネージャー QM01 に属している PAYROLL.Dept1 というキューがあるとします。その場合のクラスとプロファイルは、以下のようになります。

- 大/小文字混合のプロファイル: IBM MQ RACF クラス MXQUEUE でプロファイルを定義できます。

  ```
  RDEFINE MXQUEUE MQ01.PAYROLL.Dept1
  ```

- 大文字のプロファイル: IBM MQ RACF クラス MQQUEUE でプロファイルを定義できます。

  ```
  RDEFINE MQQUEUE MQ01.PAYROLL.*
  ```

大/小文字混合のプロファイルを使用する最初の例のほうが、リソースへのアクセス権限を与える操作をきめ細かく制御できます。

z/OS スイッチ・プロファイル

IBM MQ によって実行されるセキュリティー検査を制御するために、スイッチ・プロファイル を使用します。 スイッチ・プロファイルは、 IBM MQ に対して特別な意味を持つ通常の RACF プロファイルです。 スイッチ・プロファイルに含まれているアクセス・リストは、IBM MQ によって使用されません。

IBM MQ は、テーブル 「サブシステム・レベル・セキュリティーのスイッチ・プロファイル」、「キュー共用グループまたはキュー・マネージャー・レベル・セキュリティーのスイッチ・プロファイル」 および 「リソース検査用のスイッチ・プロファイル」 に表示されるスイッチ・タイプごとに内部スイッチを維持します。 スイッチ・プロファイルは、キュー共有グループ・レベルでも、キュー・マネージャー・レベルでも、その両方の組み合わせでも管理できます。 キュー共有グループで 1 セットのセキュリティー・スイッチ・プロファイルを使用すれば、キュー共有グループに含まれているすべてのキュー・マネージャーのセキュリティーを制御できます。

セキュリティー・スイッチをオンに設定すると、そのスイッチに関連したセキュリティー検査が実行されます。 セキュリティー・スイッチをオフに設定すると、そのスイッチに関連したセキュリティー検査が迂回されます。 すべてのセキュリティー・スイッチをオンに設定するのが、デフォルトの動作です。

z/OS スイッチとクラス

キュー・マネージャーを開始したり、セキュリティーをリフレッシュしたりすると、IBM MQ が各種の RACF クラスの状態に基づいてスイッチを設定します。

キュー・マネージャーを開始すると (あるいは IBM MQ の REFRESH SECURITY コマンドによって MQADMIN クラスまたは MXADMIN クラスをリフレッシュすると)、IBM MQ はまず、RACF とそれぞれの場合に該当するクラスの状況を確認します。

- MQADMIN クラス (大文字のプロファイルを使用している場合)
- MXADMIN クラス (大/小文字混合のプロファイルを使用している場合)

以下のいずれかの条件が真になっていると、サブシステム・セキュリティーがオフに設定されます。

- RACF が非アクティブになっているか、インストールされていません。
- MQADMIN クラスまたは MXADMIN クラスが定義されていません (これらのクラスは、クラス記述子テーブル (CDT) に含まれているので、RACF で常に定義されています)。
- MQADMIN クラスまたは MXADMIN クラスがアクティブ化されていません。

RACF と MQADMIN クラスまたは MXADMIN クラスの両方がアクティブになっていれば、IBM MQ は、MQADMIN クラスまたは MXADMIN クラスをチェックして、いずれかのスイッチ・プロファイルが定義されているかどうかを確認します。 最初にチェックするのは、186 ページの『サブシステム・セキュリティーを制御するためのプロファイル』で取り上げられているプロファイルです。 サブシステム・セキュリティーが必要なければ、IBM MQ は、サブシステム・セキュリティーの内部スイッチをオフに設定し、それ以上の検査を実行しません。

各プロファイルによって、対応する IBM MQ のスイッチがオンに設定されるかオフに設定されるかが決まります。

- スイッチがオフになれば、そのタイプのセキュリティーが非アクティブになります。
- いずれかの IBM MQ スイッチがオンに設定されている場合、 IBM MQ は、 IBM MQ スイッチに対応するセキュリティーのタイプに関連付けられた RACF クラスの状況を検査します。 そのクラスがインストールされていないか、アクティブになっていない場合は、IBM MQ スイッチがオフに設定されます。 例えば、MQPROC クラスまたは MXPROC クラスがアクティブになっていなければ、プロセス・セキュリティー検査は実行されません。 このクラスがアクティブになっていないということは、この RACF データベースを使用するすべてのキュー・マネージャーとキュー共有グループで NO.PROCESS.CHECKS プロファイルを定義するのと同じ意味になります。

z/OS スイッチのしくみ

セキュリティー・スイッチをオフに設定するには、NO.* を定義します。 プロファイルを切り替えます。 NO.* をオーバーライドすることができます。 YES.* を定義することによって、キュー共有グループ・レベルで設定されるプロファイル キュー・マネージャーのプロファイル。

セキュリティー・スイッチをオフに設定するには、NO.* を定義する必要があります。 プロファイルを切り替えます。 NO.* の存在。 プロファイルは、特定のキュー・マネージャーでキュー共有グループ・レベルの設定をオーバーライドすることを選択しない限り、そのタイプのリソースに対してセキュリティー検査が実行 **されない** ことを意味します。 これについては、186 ページの『キュー共用グループ・レベルの設定のオーバーライド』で説明されています。

キュー・マネージャーがキュー共有グループのメンバーでない場合は、キュー共有グループ・レベルのプロファイルやオーバーライド・プロファイルを定義する必要はありません。 ただし、後日キュー・マネージャーをキュー共有グループに加える場合は、それらのプロファイルを忘れずに定義する必要があります。

各 NO.* IBM MQ が検出するスイッチ・プロファイルは、そのタイプのリソースの検査をオフにします。 スイッチ・プロファイルは、キュー・マネージャーの開始時にアクティブ化されます。 対象のキュー・マネージャーの実行中にスイッチ・プロファイルを変更した場合は、IBM MQ の REFRESH SECURITY コマンドを実行すると、IBM MQ がその変更を認識できるようになります。

スイッチ・プロファイルは、どんな場合でも MQADMIN クラスまたは MXADMIN クラスで定義する必要があります。 GMQADMIN クラスや GMXADMIN クラスで定義しないでください。「サブシステム・レベル・セキュリティーのためのスイッチ・プロファイル」および「リソース検査のためのスイッチ・プロファイル」の表では、有効なスイッチ・プロファイルと、それらによって制御されるセキュリティー・タイプが示されています。

キュー共用グループ・レベルの設定のオーバーライド

キュー共用グループ・レベルのセキュリティー設定を、そのグループのメンバーである特定のキュー・マネージャーに関してオーバーライドできます。 グループ内の他のキュー・マネージャーで実行されない個々のキュー・マネージャーに対してキュー・マネージャー検査を実行する場合は、(qmgr-name.YES. *) を使用します。 スイッチ・プロファイル。

逆に、キュー共有グループ内の特定のキュー・マネージャーに対して特定の検査を実行しない場合は、(qmgr-name.NO. *) を定義します。 キュー・マネージャー上のその特定のリソース・タイプのプロファイルを定義します。キュー共有グループのプロファイルは定義しません。 (IBM MQ は、キュー・マネージャー・レベルのプロファイルが見つからない場合にのみ、キュー共有グループ・レベルのプロファイルを検査します。)

z/OS サブシステム・セキュリティーを制御するためのプロファイル

IBM MQ は、サブシステム、キュー・マネージャー、キュー共有グループでサブシステム・セキュリティー検査が必要かどうかを確認します。

IBM MQ によって実行される最初のセキュリティー検査に基づいて、IBM MQ サブシステム全体でセキュリティー検査が必要かどうかが判断されます。 サブシステム・セキュリティーは不要であると指定すれば、それ以上の検査は実行されません。

サブシステム・セキュリティーが必要かどうかの判断材料として、以下のスイッチ・プロファイルがチェックされます。 チェックの順序をまとめたのが、187 ページの図 14 です。

表 *24.* サブシステム・レベル・セキュリティーのためのスイッチ・プロファイル

スイッチ・プロファイル名	**制御するリソースまたは検査のタイプ**
qmgr-name.NO.SUBSYS.SECURITY	このキュー・マネージャーのサブシステム・セキュリティー
qsg-name.NO.SUBSYS.SECURITY	このキュー共有グループのサブシステム・セキュリティー
qmgr-name.YES.SUBSYS.SECURITY	このキュー・マネージャーのサブシステム・セキュリティーのオーバーライド

キュー・マネージャーがキュー共有グループのメンバーでなければ、IBM MQ は、qmgr-name.NO.SUBSYS.SECURITY スイッチ・プロファイルのみをチェックします。

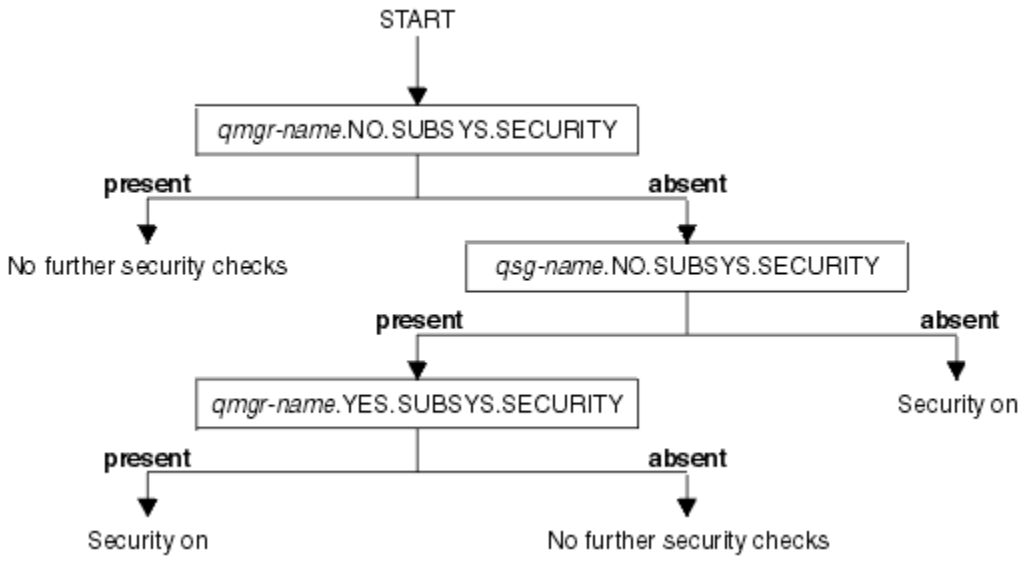


図 *14.* サブシステム・セキュリティーの確認

z/OS キュー共用グループ・レベル・セキュリティーまたはキュー・マネージャー・レベル・セキュリティーを制御するためのプロファイル

サブシステム・セキュリティー検査が必要であれば、IBM MQ は、キュー共有グループ・レベルまたはキュー・マネージャー・レベルでセキュリティー検査が必要かどうかを確認します。

IBM MQ は、セキュリティー検査が必要であると判断すると、キュー共有グループ・レベルとキュー・マネージャー・レベルのいずれかまたは両方で検査が必要かどうかを確認します。 キュー・マネージャーがキュー共有グループのメンバーでなければ、それらの検査は実行されません。

必要なレベルの判断材料として、以下のスイッチ・プロファイルがチェックされます。 チェックの順序をまとめたのが、188 ページの図 15 と 188 ページの図 16 です。

表 *25.* キュー共有グループ・レベル・セキュリティーまたはキュー・マネージャー・レベル・セキュリティーのためのスイッチ・プロファイル

スイッチ・プロファイル名	制御するリソースまたは検査のタイプ
qmgr-name.NO.QMGR.CHECKS	このキュー・マネージャーのキュー・マネージャー・レベル検査なし
qsg-name.NO.QMGR.CHECKS	このキュー共用グループについてのキュー・マネージャー・レベルの検査 (検査なし)
qmgr-name.YES.QMGR.CHECKS	このキュー・マネージャーについてのキュー・マネージャー・レベルの検査 (オーバーライド)
qmgr-name.NO.QSG.CHECKS	このキュー・マネージャーについてのキュー共用グループ・レベルの検査 (検査なし)
qsg-name.NO.QSG.CHECKS	このキュー共有グループのキュー共有グループ・レベル検査なし
qmgr-name.YES.QSG.CHECKS	このキュー・マネージャーのキュー共有グループ・レベル検査のオーバーライド

サブシステム・セキュリティーがアクティブになっている状態で、キュー共有グループ・レベルとキュー・マネージャー・レベルの両方のセキュリティー・スイッチをオフにすることはできません。 そのようにしても、IBM MQ が両方のレベルでセキュリティー検査をオンに設定します。

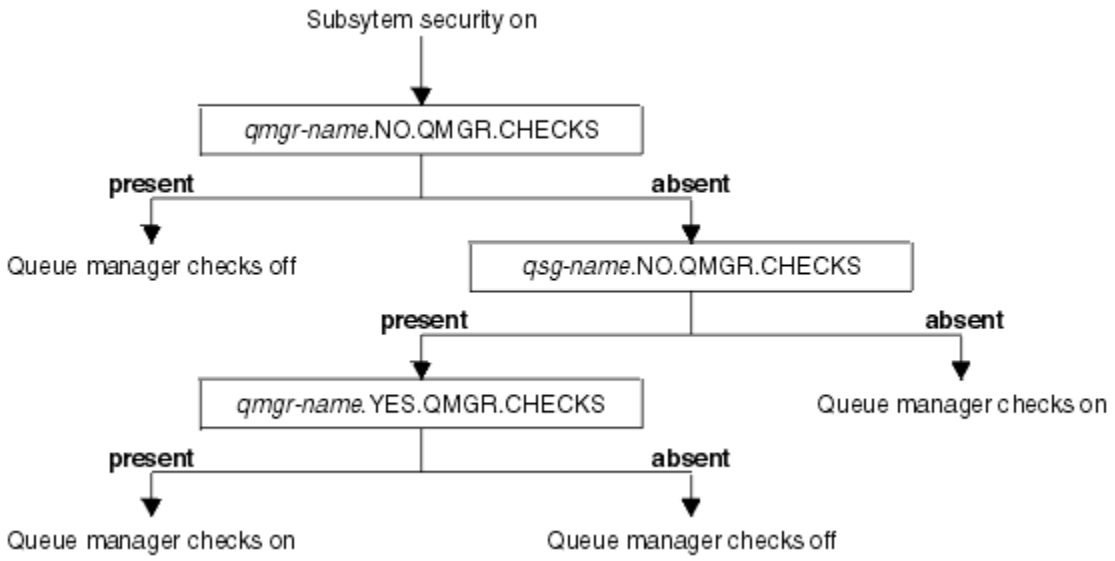


図 *15.* キュー・マネージャー・レベルのセキュリティーの必要性を調べる検査

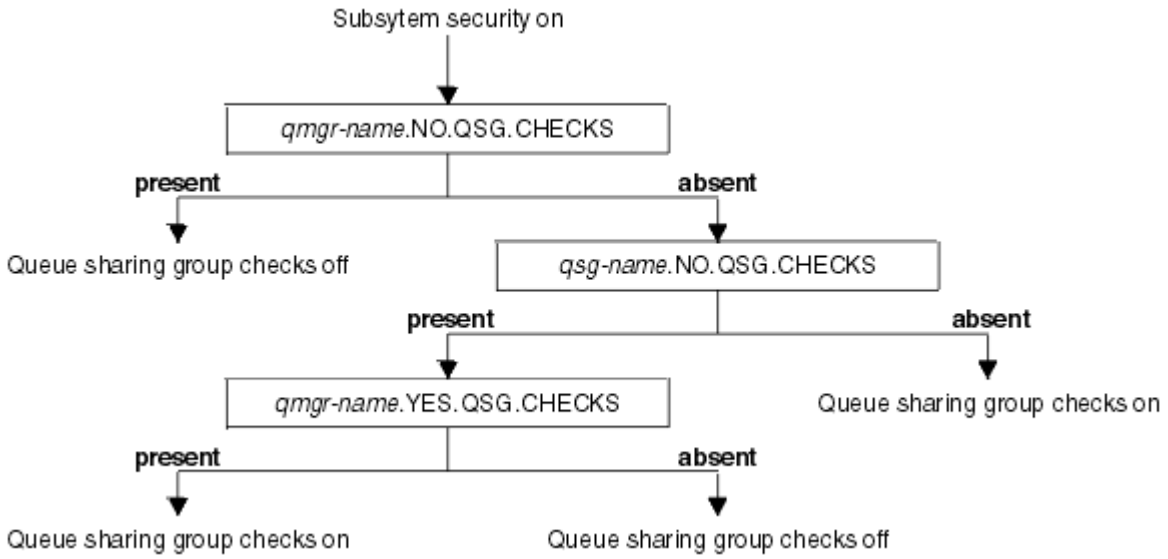


図 *16.* キュー共有グループ・レベル・セキュリティーの確認

z/OS セキュリティー・スイッチの有効な組み合わせ

スイッチの組み合わせについては、特定の組み合わせだけが有効です。 無効なスイッチ設定の組み合わせを使用すると、メッセージ CSQH026I が生成され、キュー共有グループ・レベルとキュー・マネージャー・レベルの両方でセキュリティー検査がオンに設定されます。

セキュリティー・レベルのタイプごとに有効なスイッチ設定の組み合わせをまとめたのが、188 ページの表 26、189 ページの表 27、189 ページの表 28、190 ページの表 29 です。

表 *26.* キュー・マネージャー・レベル・セキュリティーのセキュリティー・スイッチの有効な組み合わせ

組み合わせ
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS

表 *26.* キュー・マネージャー・レベル・セキュリティーのセキュリティー・スイッチの有効な組み合わせ *(*続き*)*

組み合わせ
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

表 *27.* キュー共有グループ・レベル・セキュリティーのセキュリティー・スイッチの有効な組み合わせ

組み合わせ
qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

表 *28.* キュー・マネージャー・レベル・セキュリティーとキュー共有グループ・レベル・セキュリティーのセキュリティー・スイッチの有効な組み合わせ

組み合わせ
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS QSG.* プロファイルは定義しない
QMGR.* プロファイルは定義しない qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
どちらのスイッチのプロファイルも定義しない

表 29. 両方のレベルの検査のスイッチが**オン**になるセキュリティー・スイッチのその他の有効な組み合わせ
組み合わせ
qmgr-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

z/OS リソース・レベルの検査

リソースへのアクセスを制御するために、いくつかのスイッチ・プロファイルを使用します。 キュー・マネージャーまたはキュー共有グループで検査の実行を停止するプロファイルもあります。 それらのプロファイルをオーバーライドする場合は、特定のキュー・マネージャーで検査を有効にするプロファイルを使用します。

190 ページの表 30 は、IBM MQ リソースへのアクセスを制御するために使用されるスイッチ・プロファイルを示しています。

キュー・マネージャーがキュー共有グループの一部であり、キュー・マネージャーとキュー共有グループの両方のセキュリティーがアクティブになっている場合は、YES.* を使用できます。 プロファイルを切り替えて、キュー共有グループ・レベルのプロファイルをオーバーライドし、特定のキュー・マネージャーのセキュリティーを具体的にオンにします。

キュー・マネージャーとキュー共有グループの両方に適用されるプロファイルもあります。 その種のプロファイルには、*hlq* というストリングの接頭部が付いています。必要に応じて、キュー共有グループまたはキュー・マネージャーの名前に置き換えてください。 プロファイル名に *qmgr-name* という接頭部が付いているのは、キュー・マネージャーのオーバーライド・プロファイルです。その接頭部をキュー・マネージャーの名前に置き換える必要があります。

表 30. リソース検査のためのスイッチ・プロファイル

制御される資源検査のタイプ	スイッチ・プロファイル名	特定のキュー・マネージャーのオーバーライド・プロファイル
接続のセキュリティー	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
キュー・セキュリティー	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
プロセス・セキュリティー	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
名前リスト・セキュリティー	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
コンテキスト・セキュリティー	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
代替ユーザー・セキュリティー	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS

表 30. リソース検査のためのスイッチ・プロファイル (続き)		
制御される資源検査のタイプ	**スイッチ・プロファイル名**	**特定のキュー・マネージャーのオーバーライド・プロファイル**
コマンドのセキュリティー	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
コマンド資源のセキュリティー	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
トピック・セキュリティー	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS
注 : hlq.NO などの汎用スイッチ・プロファイル。* * IBM MQ によって無視される		

例えば、キュー共有グループ QSG3 のメンバーであるキュー・マネージャー QM01 でプロセス・セキュリティー検査を実行する一方で、そのグループの他のキュー・マネージャーではプロセス・セキュリティー検査を実行しない場合は、以下のスイッチ・プロファイルを定義します。

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

キュー共有グループに含まれている QM02 以外のすべてのキュー・マネージャーでキュー・セキュリティー検査を実行する場合は、以下のスイッチ・プロファイルを定義します。

```
QM02.NO.QUEUE.CHECKS
```

(プロファイルが定義されていなければ、検査は自動的に有効になるので、キュー共有グループのプロファイルを定義する必要はありません。)

z/OS スイッチ定義の例

それぞれの IBM MQ サブシステムでは、セキュリティー要件が異なります。別々のセキュリティー要件を実装するには、別々のスイッチ・プロファイルを使用します。

4 つの IBM MQ サブシステムが定義されています。

- MQP1 (実動システム)
- MQP2 (実動システム)
- MQD1 (開発システム)
- MQT1 (テスト・システム)

4 つのキュー・マネージャーはすべて、キュー共用グループ QS01 のメンバーです。 すべての IBM MQ RACF クラスが定義され、アクティブ化されています。

それぞれのサブシステムには、別々のセキュリティー要件があります。

- どちらの実動システムでも、IBM MQ の全セキュリティー検査をキュー共有グループ・レベルでアクティブにする必要があります。

  そのためには、以下のプロファイルを指定します。

  ```
  RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
  ```

  これで、キュー共有グループに含まれているすべてのキュー・マネージャーでキュー共有グループ・レベルの検査が設定されます。 実動システムでは、すべての検査が必要なので、実動キュー・マネージャーで他のスイッチ・プロファイルを定義する必要はありません。
- テスト・キュー・マネージャー MQT1 でも全セキュリティー検査が必要です。 ただし、この点については後で変更する可能性があるので、キュー・マネージャー・レベルでセキュリティーを定義しておけば、

キュー共有グループの他のメンバーに影響を与えずに、このキュー・マネージャーのセキュリティー設定を変更できるようになります。

そのためには、MQT1 で NO.QSG.CHECKS プロファイルを以下のように定義します。

```
RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
```

- 開発キュー・マネージャー MQD1 には、キュー共有グループの他のメンバーとは異なるセキュリティー要件があります。 つまり、接続セキュリティーとキュー・セキュリティーだけをアクティブにする必要があります。

  そのためには、このキュー・マネージャーで MQD1.YES.QMGR.CHECKS プロファイルを定義してから、検査の必要のないリソースのセキュリティー検査のスイッチをオフにするために以下のプロファイルを定義します。

```
RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
```

キュー・マネージャーがアクティブになっている場合は、DISPLAY SECURITY MQSC コマンドを実行して、現在のセキュリティー設定を表示できます。

また、MQADMIN クラスで、該当するスイッチ・プロファイルを定義または削除すれば、キュー・マネージャーの実行中にスイッチ設定を変更することができます。 スイッチ設定の変更をアクティブにするには、MQADMIN クラスで REFRESH SECURITY コマンドを実行する必要があります。

DISPLAY SECURITY コマンドと REFRESH SECURITY コマンドを使用するための詳細については、246 ページの『z/OS でキュー・マネージャーのセキュリティーをリフレッシュする操作』を参照してください。

z/OS IBM MQ リソースへのアクセスを制御するためのプロファイル

定義されている可能性があるスイッチ・プロファイルに加えて、 IBM MQ リソースへのアクセスを制御するために RACF プロファイルを定義する必要があります。 この一連のトピックには、さまざまなタイプの IBM MQ リソースの RACF プロファイルに関する情報が含まれています。

特定のセキュリティー検査に対応するリソース・プロファイルを定義していない場合に、ユーザーがその検査の実行を伴う要求を送信すると、IBM MQ によってアクセスが拒否されます。 非アクティブ化したセキュリティー・スイッチに関連したセキュリティー・タイプについては、プロファイルを定義する必要はありません。

z/OS 接続セキュリティーのためのプロファイル

接続セキュリティーがアクティブになっている場合は、MQCONN クラスでプロファイルを定義し、それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えることによって、それらのグループまたはユーザー ID が IBM MQ に接続できるようにする必要があります。

接続を可能にするには、対象のプロファイルに対する RACF READ アクセス権をユーザーに与えなければなりません。 (キュー・マネージャー・レベルのプロファイルがなくても、キュー・マネージャーがキュー共有グループのメンバーになっていて、キュー共有グループ・レベルのプロファイルに基づく検査が実行されるようにセキュリティーがセットアップされていれば、その検査が実行されます。)

キュー・マネージャー名で修飾されている接続プロファイルでは、特定のキュー・マネージャーに対するアクセスが制御されます。そのプロファイルへのアクセス権が与えられているユーザーは、そのキュー・マネージャーに接続できます。 キュー共有グループ名で修飾されている接続プロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーに対するその接続タイプのアクセスが制御されます。 例えば、QS01.BATCH へのアクセス権が与えられているユーザーは、キュー共有グループ QS01 に含まれているキュー・マネージャーのうち、キュー・マネージャー・レベルのプロファイルが定義されていないすべてのキュー・マネージャーに対するバッチ接続を使用できます。

注 :

1. さまざまなセキュリティー要求で検査されるユーザー ID については、234 ページの『セキュリティー検査のためのユーザー ID (z/OS)』を参照してください。
2. 接続時には、リソース・レベル・セキュリティー (RESLEVEL) 検査も実行されます。 詳細については、228 ページの『RESLEVEL セキュリティー・プロファイル』を参照してください。

IBM MQ のセキュリティーで認識される接続のタイプを以下にまとめます。

- バッチ接続とバッチ・タイプの接続。例えば、以下のような接続があります。
  - z/OS バッチ・ジョブ
  - TSO アプリケーション
  - USS サインオン
  - Db2 ストアード・プロシージャー
- CICS 接続
- 制御およびアプリケーション・プロセス領域からの IMS 接続
- IBM MQ チャネル・イニシエーター

z/OS バッチ接続のための接続セキュリティー・プロファイル

バッチ・タイプの接続をチェックするためのプロファイルは、キュー・マネージャー名またはキュー共有グループ名の後に *BATCH* という語が付いた形式になっています。 接続元のアドレス・スペースに関連したユーザー ID に、接続プロファイルに対する READ アクセス権を与えてください。

バッチ接続とバッチ・タイプの接続をチェックするためのプロファイルは、以下のような形式になっています。

```
hlq.BATCH
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。 キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用している場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。 どちらのプロファイルも見つからなければ、接続要求は失敗します。

バッチ接続とバッチ・タイプの接続の要求では、接続元のアドレス・スペースに関連したユーザー ID に、接続プロファイルに対するアクセス権を与える必要があります。 たとえば、次の RACF コマンドでは、CONNTQM1 グループ内のユーザーが、キュー・マネージャー TQM1 に接続することを許可されます。これらのユーザー ID には、任意のバッチまたはバッチ・タイプ接続を使用することが許可されます。

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

z/OS ローカルでバインドされたアプリケーションでの ***CHCKLOCL*** の使用

CHCKLOCL は、BATCH 接続を介して作成された接続のみに適用され、CICS や IMS で作成された接続には適用されません。 チャネル・イニシエーターを介して作成された接続は、**CHCKCLNT** によって制御されます。

概要

z/OS キュー・マネージャーを、ローカルでバインドされたアプリケーションの一部 (すべてではない) に対してユーザー ID とパスワードの検査を必須にするように構成するには、さらにいくつかの構成を行う必要があります。

これは、**CHCKLOCL** (*REQUIRED*) を構成すると、MQCONN API 呼び出しを使用するレガシー・バッチ・アプリケーションがキュー・マネージャーに接続できなくなるためです。

z/OS の場合のみ、アドレス・スペースの接続セキュリティーに基づくより詳細なメカニズムを使用して、具体的に定義したユーザー ID において、グローバル CHCKLOCL(REQUIRED) 構成を、

CHCKLOCL(OPTIONAL) にダウングレードすることができます。 使用されるメカニズムについて、例とともに以下のテキストで説明します。

CHCKLOCL (*REQUIRED*) の細分性を、単なる EVERYONE より詳細に設定するには、MQCONN クラス内の hlq.batch 接続プロファイルへの接続中のアドレス・スペースに関連付けられたユーザー ID のアクセス・レベルを変更するのと同じ方法で、**CHCKLOCL** を変更します。

アドレス・スペース・ユーザー ID に READ アクセス権 (接続するために最低限必要な権限) しかない場合、**CHCKLOCL** 構成は、記述されているとおりに適用されます。

アドレス・スペース・ユーザー ID に UPDATE アクセス権 (またはそれ以上) がある場合は、**CHCKLOCL** 構成は *OPTIONAL* モードで動作します。 つまり、ユーザー ID とパスワードを指定する必要はありませんが、指定する場合は、ユーザー ID とパスワードは有効なペアでなければなりません。

z/OS キュー・マネージャー用に既に構成済みの接続セキュリティー

ご使用の z/OS キュー・マネージャー用に構成された接続セキュリティーがあって、**CHCKLOCL** (*REQUIRED*) を WAS のローカルでバインドされたアプリケーションのみに適用する場合、以下のステップを実行します。

1. 構成の際に **CHCKLOCL** (*OPTIONAL*) から開始します。 これは、指定されたユーザー ID とパスワードの妥当性を検査するが、必須ではないことを意味します。
2. 以下のコマンドを実行して、接続セキュリティー・プロファイルへのアクセス権を持つすべてのユーザーをリストします。

   ```
   RLIST MQCONN MQ23.BATCH AUTHUSER
   ```

   このコマンドを実行すると、以下のように表示されます。

   ```
   CLASS    NAME
   -----    ----
   MQCONN   MQ23.BATCH

   USER     ACCESS  ACCESS COUNT
   ----     ------  ------ -----
   JOHNDOE  READ    000009
   JDOE1    READ    000003
   WASUSER  READ    000000
   ```

3. リストされた、READ アクセス権を持つ各ユーザー ID について、以下のようにアクセス権を変更します。

   ```
   UPDATE:- PERMIT MQ23.BATCH CLASS(MQCONN) ID(JOHNDOE) ACCESS(UPDATE)
   ```

4. IBM MQ 構成を **CHCKLOCL** (*REQUIRED*) に更新します。

   MQ23.BATCH への UPDATE アクセス権と現行の設定の組み合わせは、**CHCKLOCL** (*OPTIONAL*) を使用していることを意味します。

5. ここで、**CHCKLOCL** (*REQUIRED*) 動作を 1 つの特定のユーザー ID (例えば WASUSER) に適用して、その領域から着信するすべての接続がユーザー ID とパスワードを提供するようにします。

   これを行うには、以下のコマンドを実行して上記で行った変更と逆の操作を行います。

   ```
   PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
   ```

z/OS キュー・マネージャー用に構成されていない接続セキュリティー

この場合、以下のようにする必要があります。

1. 以下のコマンドを発行して、MQCONN クラス内の hlq.BATCH の接続プロファイルを作成します。

```
RDEFINE MQCONN MQ23.BATCH UACC(NONE)
```

2. キュー・マネージャーへのバッチ接続を作成するすべてのユーザー ID に権限を与えて、このプロファイルへの UPDATE アクセス権を持つようにします。 こうすることで、接続時にユーザー ID とパスワードについての **CHCKLOCL** (*REQUIRED*) 要件を迂回します。

   これを行うには、以下のコマンドを実行します。

```
PERMIT MQ23.BATCH CLASS(MQCONN)ID(JOHNDOE) ACCESS(UPDATE)
```

   これらには、以下のようなユーザー ID があります。

   a. CSQUTIL、ISPF パネル、およびその他のローカルでバインドされたツールに使用されます。

   b. キュー・マネージャーへの接続のようなバッチに関連付けられます。 例えば、 Advanced Message Security、IBM Integration Bus、Db2 ストアード・プロシージャー、USS ユーザーと TSO ユーザー、および Java アプリケーションなどがあります。

3. 以下のコマンドを実行して、キュー・マネージャー用のスイッチ・プロファイルを削除します。

```
hlq.NO.CONNECT.CHECKS
```

4. ここで、**CHCKLOCL** (*REQUIRED*) 動作を 1 つの特定のユーザー ID (例えば WASUSER) に適用して、その領域から着信するすべての接続がユーザー ID とパスワードを提供するようにします。

   これを行うには、以下のコマンドを実行して上記で行った変更と逆の操作を行います。

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

z/OS *CICS* 接続のための接続セキュリティー・プロファイル

CICS 接続をチェックするためのプロファイルは、キュー・マネージャー名またはキュー共有グループ名の後に *CICS* という語が付いた形式になっています。 CICS のアドレス・スペースに関連したユーザー ID に、接続プロファイルに対する READ アクセス権を与えてください。

CICS からの接続をチェックするためのプロファイルは、以下のような形式になっています。

```
hlq.CICS
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。 キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用している場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。 どちらのプロファイルも見つからなければ、接続要求は失敗します。

CICS による接続要求では、CICS のアドレス・スペースのユーザー ID に、接続プロファイルに対するアクセス権を与える必要があります。

例えば、次の RACF コマンド では、CICS アドレス・スペースの ユーザー ID KCBCICS が、キュー・マネージャー TQM1 に接続することを許可されます。

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

z/OS *IMS* 接続のための接続セキュリティー・プロファイル

IMS 接続をチェックするためのプロファイルは、キュー・マネージャー名またはキュー共有グループ名の後に *IMS* という語が付いた形式になっています。 IMS の制御領域と従属領域のユーザー ID に、接続プロファイルに対する READ アクセス権を与えてください。

IMS からの接続をチェックするためのプロファイルは、以下のような形式になっています。

```
hlq.IMS
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。 キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用している場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。 どちらのプロファイルも見つからなければ、接続要求は失敗します。

IMS による接続要求では、IMS の制御領域と従属領域のユーザー ID に、接続プロファイルに対するアクセス権を与えてください。

例えば、以下の要件を満たした RACF コマンドを次に示します。

- IMS 領域のユーザー ID IMSREG がキュー・マネージャー TQM1 に接続できるようにします。
- BMPGRP グループに含まれているユーザーが BMP ジョブを実行できるようにします。

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

z/OS チャネル・イニシエーターのための接続セキュリティー・プロファイル

チャネル・イニシエーターからの接続をチェックするためのプロファイルは、キュー・マネージャー名またはキュー共有グループ名の後に *CHIN* という語が付いた形式になっています。 チャネル・イニシエーターの開始済みタスクのアドレス・スペースで使用するユーザー ID に、接続プロファイルに対する READ アクセス権を与えてください。

チャネル・イニシエーターからの接続を検査するためのプロファイルの形 式は、次のとおりです。

```
hlq.CHIN
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。 キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用している場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。 どちらのプロファイルも見つからなければ、接続要求は失敗します。

チャネル・イニシエーターによる接続要求では、チャネル・イニシエーターの開始済みタスクのアドレス・スペースで使用するユーザー ID に、接続プロファイルに対するアクセス権を定義してください。

たとえば、次の RACF コマンドでは、ユーザー ID DQCTRL を使用して実行されているチャネル・イニシエーター・アドレス・スペースが、キュー・マネージャー TQM1 に接続することを許可されます。

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

z/OS キュー・セキュリティーのためのプロファイル

キュー・セキュリティーがアクティブになっている場合は、該当するクラスでプロファイルを定義し、それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与える必要があります。キュー・セキュリティー・プロファイルの名前は、キュー・マネージャーまたはキュー共有グループの名前と、開くキューの名前に基づいています。

キュー・セキュリティーがアクティブになっている場合は、以下のようにする必要があります。

- 大文字のプロファイルを使用する場合は、**MQQUEUE** クラスまたは **GMQQUEUE** クラスでプロファイルを定義します。
- 大/小文字混合のプロファイルを使用する場合は、**MXQUEUE** クラスまたは **GMXQUEUE** クラスでプロファイルを定義します。
- それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えることによって、それらのグループまたはユーザー ID が、キューを使用する IBM MQ API 要求を実行できるようにします。

キュー・セキュリティーのためのプロファイルは、以下のような形式になっています。

```
hlq.queuename
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。queuename は、開くキューの名前 (MQOPEN 呼び出しまたは MQPUT1 呼び出しのオブジェクト記述子で指定する名前) です。

キュー・マネージャー名の接頭部が付いているプロファイルは、そのキュー・マネージャーにおける 1 つのキューへのアクセスを制御します。 キュー共有グループ名の接頭部が付いているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーで、そのキュー名の 1 つ以上のキューに対するアクセスが制御されるか、そのグループに含まれているいずれかのキュー・マネージャーから 1 つの共有キューに対するアクセスが制御されます。 そのアクセスを個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでそのキューに関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

共有キューを使用している場合は、キュー共有グループ・レベルのセキュリティーを使用することをお勧めします。

キュー名が別名キューまたはモデル・キューの名前になっている場合のキュー・セキュリティーの動作 z/OS の詳細については、199 ページの『別名キューに関する考慮事項』と 200 ページの『モデル・キューに関する考慮事項』 を参照してください。

キューを開くために必要な RACF アクセス権は、MQOPEN オプションまたは MQPUT1 オプションの指定内容によって異なります。 MQOO_* オプションと MQPMO_* オプションを複数コーディングした場合は、必要な最高レベルの RACF 権限でキュー・セキュリティー検査が実行されます。

表 *31. MQOPEN* 呼び出しまたは *MQPUT1* 呼び出しを使用する場合にキュー・セキュリティーで必要なアクセス権のレベル

MQOPEN オプションまたは MQPUT1 オプション	**hlq.queuename で必要な RACF アクセス権のレベル**
MQOO_BROWSE	READ
MQOO_INQUIRE	READ
MQOO_BIND_*	UPDATE
MQOO_INPUT_*	UPDATE
MQOO_OUTPUT または MQPUT1	UPDATE

表 *31. MQOPEN* 呼び出しまたは *MQPUT1* 呼び出しを使用する場合にキュー・セキュリティーで必要なアクセス権のレベル *(*続き*)*

MQOPEN オプションまたは MQPUT1 オプション	**hlq.queuename で必要な RACF アクセス権のレベル**
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	UPDATE
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	UPDATE
MQOO_SAVE_ALL_CONTEXT	UPDATE
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	UPDATE
MQOO_SET	ALTER

例えば、IBM MQ キュー・マネージャー QM77 では、RACF グループ PAYGRP 内のすべてのユーザー ID に、「PAY.」で始まる名前を持つすべてのキューへのメッセージを取得または書き込みするためのアクセス権限が付与されます。 そのためには、以下の RACF コマンドを使用します。

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

さらに、PAYGRP グループのすべてのユーザー ID には、PAY の命名規則に合致していないキューにメッセージを書き込むためのアクセス権も必要です。 以下に例を示します。

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

そのためには、以下のようにして、それらのキューのプロファイルを GMQQUEUE クラスで定義し、そのクラスに対するアクセス権を与えます。

```
RDEFINE GMQQUEUE PAYROLL.EXTRAS UACC(NONE)
        ADDMEM(QM77.REQUEST_QUEUE_FOR_PAYROLL,
               QM77.SALARY.INCREASE.SERVER,
               QM77.REPLIES.FROM.SALARY.MODEL)
PERMIT PAYROLL.EXTRAS CLASS(GMQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

注 :

1. アプリケーションがキュー・セキュリティー・プロファイルに対して持っている RACF アクセス権のレベルを変更した場合に、その変更が有効になるのは、そのキューで取得する新しいオブジェクト・ハンドル (つまり、新しい MQOPEN) の場合に限られます。 変更の時点で既に存在していたハンドルは、キューに対する既存のアクセス権をそのまま保持します。 キューに対する既存のアクセス権のレベルではなく変更後のアクセス権のレベルをアプリケーションで使用する必要がある場合は、その変更が必要なオブジェクト・ハンドルごとに、キューをいったん閉じてから再び開く必要があります。
2. 例では、キュー・マネージャー名 QM77 は、キュー共用グループの名前にすることもできます。

キューを開くときのオプションやアクティブになっているセキュリティーのタイプによっては、キューを開いた時点で他のタイプのセキュリティー検査も実行される可能性があります。 z/OS 214 ページの『コンテキスト・セキュリティーのためのプロファイル』 および 212 ページの『代替ユーザー・セキュリティーのためのプロファイル』も参照してください。 キューを開くときのオプションと、キュー・セキュリティー、コンテキスト・セキュリティー、代替ユーザー・セキュリティーがすべてアクティブになっているときに必要なセキュリティー権限をまとめた表については、205 ページの表 36 を参照してください。

パブリッシュ/サブスクライブを使用している場合、以下を検討する必要があります。 MQSUB 要求の処理時には、その要求元のユーザー ID がターゲットの IBM MQ キューにメッセージを書き込むために必要なアクセス権と、IBM MQ トピックにサブスクライブするために必要なアクセス権を持っているかどうかを確認するためのセキュリティー検査が実行されます。

表 *32. MQSUB* 呼び出しを使用する場合にキュー・セキュリティーで必要なアクセス権のレベル

MQSUB オプション	**hlq.queuename で必要な RACF アクセス権のレベル**
MQSO_ALTER、MQSO_CREATE、MQSO_RESUME	UPDATE

注 :

1. hlq.queuename は、パブリケーションの宛先キューです。 これが管理対象キューの場合は、その管理対象キューで使用する適切なモデル・キューへのアクセス権と、作成される動的キューへのアクセス権が必要になります。
2. サブスクリプションを実行するユーザーと、宛先キューからパブリケーションを取得するユーザーを区別する場合は、MQSUB API 呼び出しで指定する宛先キューでこのような技法を使用できます。

z/OS 別名キューに関する考慮事項

別名キューに対して MQOPEN 呼び出しまたは MQPUT1 呼び出しを実行する場合は、その呼び出しのオブジェクト記述子 (MQOD) で指定するキュー名に基づくリソース検査が IBM MQ によって実行されます。 ターゲット・キュー名に対するアクセス権がユーザーにあるかどうかを確認するための検査は行われません。

例えば、PAYROLL.REQUEST という別名キューがターゲット・キュー PAY.REQUEST に解決されるとします。 キュー・セキュリティーがアクティブになっている場合でも、必要なのは PAYROLL.REQUEST キューにアクセスする権限だけです。 PAY.REQUEST キューにアクセスする権限があるかどうかを確認するための検査は行われません。

z/OS 別名キューを使用して *MQGET* 要求と *MQPUT* 要求を区別する方法

MQPUT 呼び出しだけ、または **MQGET** 呼び出しだけを実行できるようにキューに対するアクセス権を制限する必要がある場合は、1 つのアクセス権のレベルで実行できる MQI 呼び出しの範囲が決まっているので、問題が発生します。 そのような場合は、そのキューに解決される別名を 2 つ定義することによって、キューを保護できます。つまり、アプリケーションがそのキューからメッセージを取得できるようにするための別名と、アプリケーションがそのキューにメッセージを書き込めるようにするための別名です。

IBM MQ に対してキューを定義するためのテキストの例を以下に示します。

```
DEFINE QLOCAL(MUST_USE_ALIAS_TO_ACCESS) GET(ENABLED)
       PUT(ENABLED)

DEFINE QALIAS(USE_THIS_ONE_FOR_GETS) GET(ENABLED)
       PUT(DISABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)

DEFINE QALIAS(USE_THIS_ONE_FOR_PUTS) GET(DISABLED)
       PUT(ENABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)
```

さらに、以下のような RACF 定義も作成できます。

```
RDEFINE MQQUEUE hlq.MUST_USE_ALIAS_TO_ACCESS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_GETS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_PUTS UACC(NONE)
```

次に、hlq.MUST_USE_ALIAS_TO_ACCESS キューに対するアクセス権をどのユーザーにも与えないようにして、別名に対するアクセス権を対象のユーザーまたはグループに与えます。 そのためには、以下の RACF コマンドを使用できます。

```
PERMIT hlq.USE_THIS_ONE_FOR_GETS CLASS(MQQUEUE)
       ID(GETUSER,GETGRP) ACCESS(UPDATE)
PERMIT hlq.USE_THIS_ONE_FOR_PUTS CLASS(MQQUEUE)
       ID(PUTUSER,PUTGRP) ACCESS(UPDATE)
```

この結果、ユーザー ID GETUSER とグループ GETGRP に含まれているユーザー ID には、別名キュー USE_THIS_ONE_FOR_GETS によって MUST_USE_ALIAS_TO_ACCESS からメッセージを取得する権限だけが与えられ、ユーザー ID PUTUSER とグループ PUTGRP に含まれているユーザー ID には、別名キュー USE_THIS_ONE_FOR_PUTS によってメッセージを書き込む権限だけが与えられます。

注 :

1. このような技法を使用する場合は、アプリケーション開発者にその旨を知らせて、アプリケーション開発者がそれに応じてプログラムを設計できるようにする必要があります。
2. サブスクリプションを作成するユーザーと、宛先キューからパブリケーションを「取得」するユーザーを区別する場合は、MQSUB API 要求で指定する宛先キューに対して、このような手法を使用できます。

z/OS モデル・キューに関する考慮事項

モデル・キューを開くには、モデル・キュー自体とその解決先の動的キューの両方を開く権限が必要です。動的キュー (IBM MQ ユーティリティーによって使用される動的キューを含む) の総称 RACF プロファイルを定義します。

モデル・キューを開くと、IBM MQ セキュリティーは、以下の 2 つのキュー・セキュリティー検査を実行します。

1. モデル・キューにアクセスする権限があるかどうかの検査
2. モデル・キューの解決先の動的キューにアクセスする権限があるかどうかの検査

動的キュー名の末尾にアスタリスク (*) 文字を組み込むと、その * は、IBM MQ によって生成される文字ストリングに置き換えられ、固有の名前の動的キューが作成されます。 ただし、権限検査ではその生成ストリングを含んだ名前全体が使用されるので、それらのキューについては、汎用プロファイルを定義する必要があります。

例えば、 MQOPEN 呼び出しは、 CREDIT.CHECK.REPLY.MODEL および CREDIT.REPLY.* キュー・マネージャー (またはキュー共有グループ) MQSP。

そのためには、以下の RACF コマンドを実行して、必要なキュー・プロファイルを定義しなければなりません。

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

さらに、対応する RACF PERMIT コマンドを実行して、ユーザーがそれらのプロファイルにアクセスできるようにすることも必要です。

MQOPEN で作成される典型的な動的キュー名は、CREDIT.REPLY.A346EF00367849A0 のような形式になります。 最後の修飾子の正確な値は予測不能です。その種のキュー名で汎用プロファイルを使用しなければならないのは、そのためです。

動的キューにメッセージを書き込む IBM MQ ユーティリティーがいくつかあります。 以下の動的キュー名に対応するプロファイルを定義し、対象のユーザー ID に RACF UPDATE アクセス権を与える必要があります (正確なユーザー ID については、234 ページの『セキュリティー検査のためのユーザー ID (z/OS)』を参照してください)。

```
SYSTEM.CSQUTIL.*  (used by CSQUTIL)
SYSTEM.CSQOREXX.* (used by the operations and control panels)
SYSTEM.CSQXCMD.*  (used by the channel initiator when processing CSQINPX)
CSQ4SAMP.*        (used by the IBM MQ supplied samples)
```

アプリケーション・プログラミングの COPY メンバーでデフォルトで使用される動的キュー名の使用を制御するためのプロファイルを定義することも検討できます。 IBM MQ に用意されているコピーブックには、デフォルトの *DynamicQName* として CSQ.* が含まれています。 そのキュー名に基づいて、該当する RACF プロファイルを設定できます。

注 : アプリケーション・プログラマーが動的キュー名として 1 つの * だけを指定することを認めないでください。 その場合は、hlq. * * を定義する必要があります。 MQQUEUE クラスのプロファイルを作成するには、広範囲にわたるアクセス権限を付与する必要があります。 そのプロファイルは、より具体的な RACF プロファイルがない他の非動的キューでも使用される可能性があります。 そうなれば、それらのキューにアクセスしてほしくないユーザーに、それらのキューに対するアクセス権が与えられる結果になってしまいます。

z/OS 永続動的キューを閉じるときのオプション

アプリケーションが、他のアプリケーションによって作成された永続動的キューをオープンした後、MQCLOSE オプションを使用してそのキューを削除しようとすると、そのときに追加のセキュリティー検査が行われます。

表 *33.* 永続動的キューを閉じるときの各オプションで必要なアクセス権のレベル

MQCLOSE オプション	**hlq.queuename で必要な RACF アクセス権のレベル**
MQCO_DELETE	ALTER
MQCO_DELETE_PURGE	ALTER

z/OS セキュリティーとリモート・キュー

メッセージをリモート・キューに書き込むときに、ローカル・キュー・マネージャーによって実装されるキュー・セキュリティーは、そのリモート・キューを開くときのリモート・キューの指定方法によって異なります。

以下のルールが適用されます。

1. リモート・キューが IBM MQ DEFINE QREMOTE コマンドを使用してローカル・キュー・マネージャーで定義されている場合、検査されるキューはリモート・キューの名前です。 例えば、リモート・キューがキュー・マネージャー MQS1 で以下のように定義されているとします。

   ```
   DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
          RNAME(CREDIT.SCORING.REQUEST)
          RQMNAME(BNK7)
          XMITQ(BANK1.TO.BANK7)
   ```

   この場合は、BANK7.CREDIT.REFERENCE に対応するプロファイルを MQQUEUE クラスで定義する必要があります。

2. 要求の *ObjectQMgrName* がローカル・キュー・マネージャーに解決されない場合は、解決後の (リモート) キュー・マネージャー名に基づいてセキュリティー検査が実行されます。ただし、クラスター・キューの場合は例外であり、その場合は、そのクラスター・キュー名に基づいて検査が実行されます。

   例えば、伝送キュー BANK1.TO.BANK7 がキュー・マネージャー MQS1 で定義されているとします。 その後、*ObjectName* として BANK1.INTERBANK.TRANSFERS を指定し、*ObjectQMgrName* として BANK1.TO.BANK7 を指定して、MQS1 で MQPUT1 要求を実行します。 この場合、その要求を実行するユーザーには、BANK1.TO.BANK7 に対するアクセス権が必要です。

3. キューに対する MQPUT 要求を実行するときに、ローカル・キュー・マネージャーの別名を *ObjectQMgrName* として指定した場合は、キュー名だけがセキュリティー検査の対象になり、キュー・マネージャー名は対象になりません。

リモート・キュー・マネージャーに到達したメッセージは、追加のセキュリティー処理の対象になることがあります。 詳しくは、93 ページの『リモート・メッセージングのセキュリティー』を参照してください。

z/OS 送達不能キュー・セキュリティー

送達不能キューについては、特別な考慮事項があります。送達不能キューには、多くのユーザーがメッセージを書き込む必要がある一方で、メッセージを取り出すためのアクセス権には厳重な制限が必要だからです。 そのためには、送達不能キューと別名キューに別々の RACF 権限を適用します。

未配布メッセージは、送達不能キューという特別なキューに書き込むことができます。 機密データが最終的にそのキューに書き込まれる可能性がある場合は、無許可ユーザーがそのデータを取得できないようにするために、セキュリティーの影響を検討しなければなりません。

以下のそれぞれが、送達不能キューにメッセージを書き込める必要があり ます。

- アプリケーション・プログラム。
- チャネル・イニシエーターのアドレス・スペースと MCA のユーザー ID。 (RESLEVEL プロファイルがない場合や、チャネル・ユーザー ID が検査の対象になるように RESLEVEL プロファイルが定義されている場合は、チャネル・ユーザー ID にも、送達不能キューにメッセージを書き込む権限が必要になります。)
- CKTI、CICS 提供の CICS タスク・イニシエーター。
- CSQQTRMN (IBM MQ に用意されている IMS トリガー・モニター)。

送達不能キューのメッセージを処理する「特別」なアプリケーションだけが送達不能キューからメッセージを取得できるようにする必要があります。 ただし、アプリケーションに MQPUT の送達不能キューに対する RACF UPDATE 権限を付与すると、 MQGET 呼び出しを使用してキューからメッセージを自動的に取り出すことができるため、問題が発生します。 送達不能キューで取得操作を無効にすることはできません。そのようにすれば、「特別」なアプリケーションもメッセージを取得できなくなるからです。

この問題の 1 つの解決策は、送達不能キューに対する 2 つのレベルのアクセス権をセットアップすることです。 つまり、CKTI、メッセージ・チャネル・エージェント・トランザクションまたはチャネル・イニシエーターのアドレス・スペース、「特別」なアプリケーションには、送達不能キューに対する直接アクセス権を与え、その他のアプリケーションには、別名キュー経由のアクセス権だけを与える、ということです。送達不能キューにメッセージを書き込むことはできても、メッセージを取得することはできないアプリケーションのために、その別名を定義します。

その流れをまとめると、以下のようになります。

1. 属性として PUT(ENABLED) と GET(ENABLED) を指定して実際の送達不能キューを定義します (サンプルの thlqual.SCSQPROC(CSQ4INYG) を参照してください)。
2. 送達不能キューに対する RACF UPDATE 権限を以下のユーザー ID に与えます。
   - CKTI、MCA またはチャネル・イニシエーターのアドレス・スペースを実行するためのユーザー ID。
   - 「特別」な送達不能キュー処理アプリケーションに関連したユーザー ID。
3. 実際の送達不能キューに解決される別名キューを定義します。ただし、その別名キューでは、属性として PUT(ENABLED) と GET(DISABLED) を指定します。 その別名キューの名前は、送達不能キュー名と同じ語幹に「.PUT」という文字を追加した名前にします。 例えば、送達不能キュー名が hlq.DEAD.QUEUE であれば、別名キューの名前は hlq.DEAD.QUEUE.PUT になります。
4. 送達不能キューにメッセージを書き込むときに、アプリケーションは別名キューを使用します。 そのアプリケーションでは、以下の処理を実行する必要があります。
   - 実際の送達不能キューの名前を取得します。 そのためには、MQOPEN を使用してキュー・マネージャー・オブジェクトを開いてから、MQINQ を実行して送達不能キュー名を取得します。
   - その名前に「.PUT」という文字を追加して、別名キューの名前を作成します。この場合は、hlq.DEAD.QUEUE.PUT になります。
   - 別名キュー hlq.DEAD.QUEUE.PUT を開きます。

- 別名キューに対して MQPUT を実行することによって、実際の送達不能キューにメッセージを書き込みます。

5. アプリケーションに関連したユーザー ID に、別名に対する RACF UPDATE 権限を与えます。ただし、実際の送達不能キューに対するアクセス権は与えません (権限 NONE)。 つまり、以下のようになります。
   - アプリケーションが別名キューを使用して送達不能キューにメッセージを書き込むことは可能です。
   - 別名キューでは取得操作が無効になっているので、アプリケーションが別名キューを使用して送達不能キューからメッセージを取得することはできません。

   アプリケーションが送達不能キューからメッセージを取得できないのは、そのアプリケーションに正しい RACF 権限が設定されているからでもあります。

203 ページの表 34 は、このソリューションのさまざまな参加者に必要な RACF 権限を要約したものです。

表 *34.* 送達不能キューとその別名に関する *RACF* 権限

関連するユーザー ID	**実際の送達不能キュー (hlq.DEAD.QUEUE)**	**別名送達不能キュー (hlq.DEAD.QUEUE.PUT)**
MCA またはチャネル・イニシエーターのアドレス・スペースと CKTI	UPDATE	NONE
送達不能キューを処理する「特別」なアプリケーション	UPDATE	NONE
ユーザー作成アプリケーションのユーザー ID	NONE	UPDATE

この手法を使用すると、アプリケーションは、送達不能キューの最大メッセージ長 (MAXMSGL) を確認できなくなります。 別名キューから MAXMSGL 属性を取得することは不可能だからです。 したがって、アプリケーションは、最大メッセージ長が 100 MB であるという前提で処理を進めることになります (100 MB というのは、IBM MQ for z/OS でサポートされている最大サイズです)。 実際の送達不能キューでも、MAXMSGL 属性を 100 MB で定義する必要があります。

注 : 通常、ユーザー作成のアプリケーション・プログラムでは、代替ユーザー権限を使って送達不能キューにメッセージを書き込むことはありません。 そうすれば、送達不能キューに対するアクセス権を持つユーザー ID の数が減ることになります。

z/OS システム・キュー・セキュリティー

特定のユーザー ID が特定のシステム・キューにアクセスできるようにするために、RACF アクセス権をセットアップする必要があります。

いくつかのシステム・キューは、IBM MQ の補助的な部分からアクセスされます。

- CSQUTIL ユーティリティー
- メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL)
- 操作と制御パネル
- チャネル・イニシエーターのアドレス・スペース (キューに入れられたパブリッシュ/サブスクライブ・デーモンなど)
- V 9.1.0 mqweb サーバー。MQ Console および REST API によって使用される。

これらを実行するためのユーザー ID には、システム・キューにアクセスするための RACF アクセス権を与える必要があります (204 ページの表 35 を参照してください)。

表 35. *IBM MQ* が必要とする *SYSTEM* キューへのアクセス

SYSTEM キュー	**CSQUTIL**	**CSQ0UTIL**	**mqweb サーバー**	**操作と制御パネル**	**分散キューイングのためのチャネル・イニシエーター**
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	UPDATE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	UPDATE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.CONTROL.QUEUE	-	-	-	-	ALTER
SYSTEM.BROKER.DEFAULT.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	-	-	-	-	UPDATE
SYSTEM.CHANNEL.INITQ	-	-	-	-	UPDATE
SYSTEM.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	-	-	ALTER
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-	-	-	-	UPDATE
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-	-	-	-	ALTER
SYSTEM.COMMAND.INPUT	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.COMMAND.REPLY.*	-	-	-	-	UPDATE
SYSTEM.COMMAND.REPLY.MODEL	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.CSQOREXX.*	-	-	-	UPDATE	-
SYSTEM.CSQUTIL.*	UPDATE	-	-	-	-
SYSTEM.CSQXCMD.*	-	-	-	-	UPDATE
SYSTEM.HIERARCHY.STATE	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.CONTROL	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.PUBS	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.FANREQ	-	-	-	-	UPDATE
SYSTEM.PROTECTION.ERROR.QUEUE	-	-	-	-	UPDATE
SYSTEM.PROTECTION.POLICY.QUEUE	-	UPDATE 204 ページの『1』	-	-	READ
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	-	-	UPDATE
SYSTEM.REST.REPLY.QUEUE	-	-	UPDATE	-	-
SYSTEM.BLUEMIX.REGISTRATION.QUEUE	-	-	-	-	UPDATE

注 :

1. Advanced Message Security アドレス・スペース・ユーザーには、このキューに対する読み取りアクセス権限も必要です。

z/OS *API* リソース・セキュリティーのアクセス権に関する早見表

MQOPEN、**MQPUT1**、**MQSUB**、**MQCLOSE** の各オプションと、それぞれのリソース・セキュリティー・タイプで必要なアクセス権についてまとめます。

表 *36. MQOPEN*、*MQPUT1*、*MQSUB*、および *MQCLOSE* の各オプションと、必要なセキュリティー権限. **(1)** のような注記が付いている箇所については、この表の後にある注を参照してください。

	必要最小限の RACF アクセス権のレベル			
RACF クラス:	**MXTOPIC**	**MQQUEUE または MXQUEUE(1)**	**MQADMIN または MXADMIN**	**MQADMIN または MXADMIN**
RACF プロファイル:	**(15 または 16)**	**(2)**	**(3)**	**(4)**
MQOPEN オプション				
MQOO_INQUIRE		READ **(5)**	検査しない	検査しない
MQOO_BROWSE		READ	検査しない	検査しない
MQOO_INPUT_*		UPDATE	検査しない	検査しない
MQOO_SAVE_ALL_CONTEXT **(6)**		UPDATE	検査しない	検査しない
MQOO_OUTPUT (USAGE=NORMAL) **(7)**		UPDATE	検査しない	検査しない
MQOO_PASS_IDENTITY_CONTEXT **(8)**		UPDATE	READ	検査しない
MQOO_PASS_ALL_CONTEXT **(8)** **(9)**		UPDATE	READ	検査しない
MQOO_SET_IDENTITY_CONTEXT **(8)** **(9)**		UPDATE	UPDATE	検査しない
MQOO_SET_ALL_CONTEXT **(8)** **(10)**		UPDATE	CONTROL	検査しない
MQOO_OUTPUT (USAGE (XMITQ) **(11)**		UPDATE	CONTROL	検査しない
MQOO_OUTPUT (トピック・オブジェクト)	UPDATE **(16)**			
MQOO_OUTPUT (トピック・オブジェクトの別名キュー)	UPDATE **(16)**	UPDATE		
MQOO_SET		ALTER	検査しない	検査しない
MQOO_ALTERNATE_USER_AUTHORITY		**(12)**	**(12)**	UPDATE
MQPUT1 オプション				
通常キューへの書き込み **(7)**		UPDATE	検査しない	検査しない
MQPMO_PASS_IDENTITY_CONTEXT		UPDATE	READ	検査しない
MQPMO_PASS_ALL_CONTEXT		UPDATE	READ	検査しない
MQPMO_SET_IDENTITY_CONTEXT		UPDATE	UPDATE	検査しない
MQPMO_SET_ALL_CONTEXT		UPDATE	CONTROL	検査しない
MQOO_OUTPUT 伝送キューへの書き込み **(11)**		UPDATE	CONTROL	検査しない
MQOO_OUTPUT (トピック・オブジェクト)	UPDATE **(16)**			
MQOO_OUTPUT (トピック・オブジェクトの別名キュー)	UPDATE **(16)**	UPDATE		
MQPMO_ALTERNATE_USER_AUTHORITY		**(13)**	**(13)**	UPDATE

表 36. *MQOPEN*、*MQPUT1*、*MQSUB*、および *MQCLOSE* の各オプションと、必要なセキュリティー権限. **(1)** のような注記が付いている箇所については、この表の後にある注を参照してください。 *(続き)*

	必要最小限の RACF アクセス権のレベル			
RACF クラス:	**MXTOPIC**	**MQQUEUE または MXQUEUE(1)**	**MQADMIN または MXADMIN**	**MQADMIN または MXADMIN**
RACF プロファイル:	**(15 または 16)**	**(2)**	**(3)**	**(4)**
MQCLOSE オプション				
MQCO_DELETE **(14)**		ALTER	検査しない	検査しない
MQCO_DELETE_PURGE **(14)**		ALTER	検査しない	検査しない
MQCO_REMOVE_SUB	ALTER **(15)**			
MQSUB オプション				
MQSO_CREATE	ALTER **(15)**	**(17)**	**(18)**	
MQSO_ALTER	ALTER **(15)**	**(17)**	**(18)**	
MQSO_RESUME	READ **(15)**	**(17)**	検査しない	
MQSO_ALTERNATE_USER_AUTHORITY				UPDATE
MQSO_SET_IDENTITY_CONTEXT			**(18)**	

注:

1. このオプションは、キューだけに制限されているわけではありません。 名前リストの場合は、MQNLIST クラスまたは MXNLIST クラスを使用し、プロセスの場合は、MQPROC クラスまたは MXPROC クラスを使用します。
2. RACF プロファイル hlq.resourcename を使用します。
3. RACF プロファイル hlq.CONTEXT.queuename を使用します。
4. RACF プロファイル: hlq.ALTERNATE.USER。 `alternateuserid`

   `alternateuserid` は、オブジェクト記述子の *AlternateUserId* フィールドで指定するユーザー ID です。 この検査では、*AlternateUserId* フィールドの 12 文字までが使用されますが、他の検査では、ユーザー ID の最初の 8 文字だけが使用されます。
5. 照会のためにキュー・マネージャーを開く場合は、検査は行われません。
6. 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、別名キューで有効です。
7. この検査は、**Usage** キュー属性が MQUS_NORMAL になっているローカル・キューまたはモデル・キューと、別名キューまたはリモート・キュー (接続先のキュー・マネージャーで定義されているキュー) で実行されます。 接続先のキュー・マネージャーの名前ではなく *ObjectQMgrName* を明示的に指定して開いたリモート・キューの場合は、*ObjectQMgrName* と同じ名前のキュー (つまり、**Usage** キュー属性が MQUS_TRANSMISSION になっているローカル・キュー) に基づいて検査が実行されます。
8. MQOO_OUTPUT も指定する必要があります。
9. MQOO_PASS_IDENTITY_CONTEXT もこのオプションによって暗黙指定されます。
10. MQOO_PASS_IDENTITY_CONTEXT、MQOO_PASS_ALL_CONTEXT、MQOO_SET_IDENTITY_CONTEXT もこのオプションによって暗黙指定されます。
11. この検査は、**Usage** キュー属性が MQUS_TRANSMISSION になっていて、出力のために直接開くローカル・キューまたはモデル・キューで実行されます。 リモート・キューを開く場合は、該当しません。
12. MQOO_INQUIRE、MQOO_BROWSE、MQOO_INPUT_*、MQOO_OUTPUT、MQOO_SET のいずれか 1 つも指定する必要があります。 実行される検査は、他のオプションを指定した場合と同じです。

13. 実行される検査は、他のオプションを指定した場合と同じです。
14. 直接開いた (つまり、モデル・キュー経由で開いたのではない) 永続動的キューの場合にのみ該当します。 一時動的キューの削除に関するセキュリティーは必要ありません。
15. RACF プロファイル hlq.SUBSCRIBE.topicname を使用します。
16. RACF プロファイル hlq.PUBLISH.topicname を使用します。
17. MQSUB 要求でパブリケーションの送信先の宛先キューを指定した場合は、そのキューでセキュリティー検査が実行され、そのキューに対する書き込み権限があるかどうかが確認されます。
18. MQSO_CREATE オプションまたは MQSO_ALTER オプションを指定した MQSUB 要求で MQSD 構造のいずれかの ID コンテキスト・フィールドを設定しようとする場合は、MQSO_SET_IDENTITY_CONTEXT オプションも指定する必要があります。さらに、宛先キューのコンテキスト・プロファイルに対する適切な権限も必要になります。

z/OS トピック・セキュリティーのためのプロファイル

トピック・セキュリティーがアクティブになっている場合は、該当するクラスでプロファイルを定義し、それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与える必要があります。

トピック・ツリー内のトピック・セキュリティーの概念については、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。

トピック・セキュリティーがアクティブになっている場合は、以下の操作を実行する必要があります。

- プロファイルを **MXTOPIC** または **GMXTOPIC** クラスで定義します。
- それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えることによって、それらのグループまたはユーザー ID が、トピックを使用する IBM MQ API 要求 を実行できるようにします。

トピック・セキュリティーのためのプロファイルは、以下のような形式になっています。

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

説明:

- hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。
- topicname は、MQSUB 呼び出しでサブスクライブするトピックまたは MQOPEN 呼び出しでパブリッシュするトピックに関連するトピック・ツリー内のトピック管理ノードの名前です。

キュー・マネージャー名の接頭部が付いているプロファイルでは、そのキュー・マネージャーで 1 つのトピックに対するアクセスが制御されます。 キュー共有グループ名が接頭部として付けられているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーで、そのトピック名の 1 つ以上のトピックに対するアクセスが制御されます。 そのアクセスを個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでそのトピックに関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

サブスクライブ

トピックにサブスクライブするには、サブスクライブしようとしているトピックと、パブリケーションの宛先キューの両方にアクセスする必要があります。

MQSUB 要求を実行すると、以下のセキュリティー検査が行われます。

- そのトピックにサブスクライブするための適切なレベルのアクセス権限があるかどうか、および宛先キュー (指定されている場合) が出力用にオープンされているかどうか

- その宛先キューに対する適切なレベルのアクセス権限があるかどうか。

表 *37.* サブスクライブするトピックのセキュリティーで必要なアクセス権のレベル

MQSUB オプション	**MXTOPIC クラスの hlq.SUBSCRIBE.topicname プロファイルに必要な RACF アクセス権**
MQSO_CREATE と MQSO_ALTER	ALTER
MQSO_RESUME	READ

表 *38.* 非管理対象宛先キューを使用してサブスクライブするために必要な追加権限

MQSUB オプション	**MQADMIN クラスまたは MXADMIN クラスの hlq.CONTEXT.queuename プロファイルに必要な RACF アクセス権**
MQSO_CREATE、MQSO_ALTER、MQSO_RESUME	UPDATE
	MQQUEUE または MXQUEUE クラスの hlq.queuename プロファイルに必要な RACF アクセス権
MQSO_CREATE と MQSO_ALTER	UPDATE
	MQADMIN クラスまたは MXADMIN クラスの hlq.ALTERNATE.USER.alternateuserid プロファイルに必要な RACF アクセス権
MQSO_ALTERNATE_USER_AUTHORITY	UPDATE

サブスクリプションの管理対象キューに関する考慮事項

トピックにサブスクライブする権限があ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。ただし、管理対象キューの作成時には、セキュリティー検査は行われません。ユーザーがこの宛先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を判別するためのセキュリティー検査も行われません。

管理対象キューをクローズしたり、削除したりすることはできません。

使用されるモデル・キューは、SYSTEM.DURABLE.MODEL.QUEUE と SYSTEM.NDURABLE.MODEL.QUEUE です。

これらのモデル・キューから作成される管理対象キューはそれぞれ、SYSTEM.MANAGED.DURABLE.A346EF00367849A0、SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0 という形式になります (最後の修飾子は予測不能です)。

これらのキューへのアクセス権を、どのユーザーにも与えないでください。 キューは、いずれの権限も付与されていない SYSTEM.MANAGED.DURABLE.* および SYSTEM.MANAGED.NDURABLE.* の形式の 総称プロファイルを使用して保護できます。

これらのキューからメッセージを取得する場合は、MQSUB 要求で返されるハンドルを使用します。

MQCO_REMOVE_SUB オプションを指定してサブスクリプションの MQCLOSE 呼び出しを明示的に実行するときに、その閉じようとしているサブスクリプションが、そのハンドルで作成したサブスクリプションでない場合は、その閉じる操作を実行する時点で、その操作に関する正しい権限があ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。

表 *39.* サブスクライブ操作でサブスクリプションを閉じるときにトピック・セキュリティーのプロファイルで必要になるアクセス権のレベル	
MQCLOSE オプション	**MXTOPIC クラスの hlq.SUBSCRIBE.topicname プロファイルに必要な RACF アクセス権**
MQCO_REMOVE_SUB	ALTER

パブリッシュ

トピックに関するメッセージをパブリッシュする場合は、そのトピックに対するアクセス権が必要です。 さらに、別名キューを使用する場合は、その別名キューに対するアクセス権も必要になります。

表 *40.* パブリッシュするトピック・セキュリティーに必要なアクセス・レベル	
MQOPEN オプションまたは MQPUT1 オプション	**MXTOPIC クラスの hlq.PUBLISH.topicname プロファイルに必要な RACF アクセス権**
MQOO_OUTPUT または MQPUT1	UPDATE

表 *41.* トピックに解決される別名キューを開くために必要なアクセス・レベル	
MQOPEN オプションまたは MQPUT1 オプション	**別名キューの MQQUEUE または MXQUEUE クラスの hlq.queuename プロファイルに必要な RACF アクセス権**
MQOO_OUTPUT または MQPUT1	UPDATE

トピック名に解決される別名キューをパブリッシュ操作のために開くときのトピック・セキュリティーの動作の詳細については、209 ページの『パブリッシュ操作でトピックに解決される別名キューを使用する場合の考慮事項』を参照してください。

PUT または GET に関する制限のために宛先キューで別名キューを使用することを検討している場合は、199 ページの『別名キューに関する考慮事項』を参照してください。

アプリケーションがトピック・セキュリティー・プロファイルに対して持っている RACF アクセス権のレベルを変更した場合、そのトピックについて取得する新しいオブジェクト・ハンドル (つまり、新しい MQSUB または MQOPEN) に対してのみ変更が有効になります。 変更の時点で既に存在していたハンドルは、トピックに対する既存のアクセス権をそのまま保持します。 さらに、既存のサブスクライバーも、自身の既存のサブスクリプションに対するアクセス権をそのまま保持します。

パブリッシュ操作でトピックに解決される別名キューを使用する場合の考慮事項

トピックに解決される別名キューに対して MQOPEN または MQPUT1 呼び出しを 発行すると、IBM MQ は以下の 2 つのリソース検査を行います。

- MQOPEN 呼び出しまたは MQPUT1 呼び出しのオブジェクト記述子 (MQOD) で指定されている別名キュー名に関する検査
- 別名キューの解決先のトピックに関する検査

この動作は、別名キューが他のキューに解決される場合の動作とは異なります。 パブリッシュ操作を実行するには、両方のプロファイルに対する正しいアクセス権が必要です。

システム・トピック・セキュリティー

以下のシステム・トピックは、チャネル・イニシエーター・アドレス・スペースによってアクセスされます。

これを実行するユーザー ID には、 210 ページの表 42 に示すように、これらのキューに対する RACF アクセス権を付与する必要があります。

表 *42. SYSTEM* トピックで必要なアクセス権		
SYSTEM トピック	**プロファイル**	**分散キューイングのためのチャネル・イニシエーター**
SYSTEM.BROKER.ADMIN.STREAM	hlq.PUBLISH.topicname	UPDATE
SYSTEM.BROKER.ADMIN.STREAM	hlq.SUBSCRIBE.topicname	ALTER

z/OS プロセスのためのプロファイル

プロセス・セキュリティーがアクティブになっている場合は、該当するクラスでプロファイルを定義し、それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与える必要があります。

プロセス・セキュリティーがアクティブになっている場合は、以下のようにする必要があります。

- 大文字のプロファイルを使用する場合は、**MQPROC** クラスまたは **GMQPROC** クラスでプロファイルを定義します。
- 大/小文字混合のプロファイルを使用する場合は、**MXPROC** クラスまたは **GMXPROC** クラスでプロファイルを定義します。
- それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えることによって、それらのグループまたはユーザー ID が、プロセスを使用する IBM MQ API 要求を実行できるようにします。

プロセスのためのプロファイルは、以下のような形式になっています。

```
hlq.processname
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。processname は、開くプロセスの名前です。

キュー・マネージャー名の接頭部が付いているプロファイルは、そのキュー・マネージャーにおける 1 つのプロセス定義へのアクセスを制御します。 キュー共有グループ名の接頭部が付いているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーで、その名前の 1 つ以上のプロセス定義に対するアクセスが制御されます。 そのアクセスを個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでそのプロセス定義に関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

プロセスを開くために必要なアクセス権を以下の表にまとめます。

表 *43.* プロセス・セキュリティーで必要なアクセス権のレベル	
MQOPEN オプション	**hlq.processname で必要な RACF アクセス権のレベル**
MQOO_INQUIRE	READ

例えば、キュー・マネージャー MQS9 では、RACF グループ INQVPRC が文字 V で始まるすべてのプロセスを照会 (MQINQ) できなければなりません。 RACF の定義は、以下のようになります。

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

プロセス定義オブジェクトを開くときに指定するオプションによっては、代替ユーザー・セキュリティーもアクティブになる可能性があります。

z/OS 名前リストのためのプロファイル

名前リスト・セキュリティーがアクティブになっている場合は、該当するクラスでプロファイルを定義し、それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えます。

名前リスト・セキュリティーがアクティブになっている場合は、以下のようにする必要があります。

- 大文字のプロファイルを使用する場合は、**MQNLIST** クラスまたは **GMQNLIST** クラスでプロファイルを定義します。
- 大/小文字混合のプロファイルを使用する場合は、**MXNLIST** クラスまたは **GMXNLIST** クラスでプロファイルを定義します。
- それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えます。

名前リストのためのプロファイルは、以下のような形式になっています。

```
hlq.namelistname
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。namelistname は、開く名前リストの名前です。

キュー・マネージャー名の接頭部が付いているプロファイルでは、そのキュー・マネージャーで 1 つの名前リストに対するアクセスが制御されます。 キュー共有グループ名の接頭部が付いているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーで、その名前の 1 つ以上の名前リストに対するアクセスが制御されます。 そのアクセスを個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでその名前リストに関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

名前リストを開くために必要なアクセス権を以下の表にまとめます。

表 *44.* 名前リスト・セキュリティーで必要なアクセス権のレベル

MQOPEN オプション	**hlq.namelistname で必要な RACF アクセス権のレベル**
MQOO_INQUIRE	READ

例えば、キュー・マネージャー (またはキュー共有グループ) PQM3 で、RACF グループ DEPT571 には、以下の名前リストで照会 (MQINQ) を実行する権限が必要だとします。

- "「DEPT571」" で始まるすべての名前リスト。
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

そのための RACF 定義は、以下のようになります。

```
RDEFINE MQNLIST PQM3.DEPT571.** UACC(NONE)
PERMIT PQM3.DEPT571.** CLASS(MQNLIST) ID(DEPT571) ACCESS(READ)

RDEFINE GMQNLIST NLISTS.FOR.DEPT571 UACC(NONE)
        ADDMEM(PQM3.PRINTER/DESTINATIONS/DEPT571,
               PQM3.AGENCY/REQUEST/QUEUES,
               PQM3.WAREHOUSE.BROADCAST)
PERMIT NLISTS.FOR.DEPT571 CLASS(GMQNLIST) ID(DEPT571) ACCESS(READ)
```

名前リスト・オブジェクトを開くときに指定するオプションによっては、代替ユーザー・セキュリティーがアクティブになる可能性もあります。

システム名前リスト・セキュリティー

いくつかのシステム名前リストは、IBM MQ の補助的な部分からアクセスされます。

- CSQUTIL ユーティリティー
- 操作と制御パネル
- チャネル・イニシエーターのアドレス・スペース (キューに入れられたパブリッシュ/サブスクライブ・デーモンなど)

これらを実行するユーザー ID には、212 ページの表 45 に示すように、これらの名前リストに対する RACF アクセス権を付与する必要があります。

表 *45. IBM MQ* によって必要とされる *SYSTEM* 名前リストへのアクセス

SYSTEM 名前リスト	**CSQUTIL**	**操作と制御パネル**	**分散キューイングのためのチャネル・イニシエーター**
SYSTEM.QPUBSUB.QUEUE.NAMELIST	-	-	READ
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	-	-	READ

z/OS 代替ユーザー・セキュリティーのためのプロファイル

代替ユーザー・セキュリティーがアクティブになっている場合は、該当するクラスでプロファイルを定義し、それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与える必要があります。

AlternateUserId について詳しくは、 AlternateUserID (MQCHAR12)を参照してください。

代替ユーザー・セキュリティーがアクティブになっている場合は、以下のようにする必要があります。

- 大文字のプロファイルを使用する場合は、MQADMIN クラスまたは GMQADMIN クラスでプロファイルを定義します。
- 大/小文字混合のプロファイルを使用する場合は、MXADMIN クラスまたは GMXADMIN クラスでプロファイルを定義します。

それらのプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与えることによって、それらのグループまたはユーザー ID がオブジェクトを開くときに ALTERNATE_USER_AUTHORITY オプションを使用できるようにします。

代替ユーザー・セキュリティーのためのプロファイルは、サブシステム・レベルまたはキュー共有グループ・レベルで指定できます。そのプロファイルは、以下のような形式になっています。

```
hlq.ALTERNATE.USER.alternateuserid
```

`hlq` は、`qmgr-name` (キュー・マネージャー名) または `qsg-name` (キュー共有グループ名) です。
`alternateuserid` は、オブジェクト記述子の *AlternateUserId* フィールドの値です。

キュー・マネージャー名の接頭部が付いているプロファイルでは、そのキュー・マネージャーで代替ユーザー ID の使用が制御されます。 キュー共有グループ名の接頭部が付いているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーで代替ユーザー ID の使用が制御されます。 キュー共有グループに含まれているどのキュー・マネージャーでも、正しいアクセス権を持っているユーザーは、その代替ユーザー ID を使用できます。 そのアクセスを個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでその代替ユーザー ID に関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

代替ユーザー・オプションを指定するときに必要なアクセス権を以下の表にまとめます。

表 *46.* 代替ユーザー・セキュリティーで必要なアクセス権のレベル

MQOPEN オプション、MQSUB オプション、MQPUT1 オプション	**必要な RACF アクセス権のレベル**
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	UPDATE

代替ユーザー・セキュリティー検査のほかに、キュー・セキュリティー、プロセス・セキュリティー、名前リスト・セキュリティー、コンテキスト・セキュリティーに関する他のセキュリティー検査も実行できます。 代替ユーザー ID が指定されている場合に、その代替ユーザー ID が使用されるのは、キュー、プロセス定義、名前リスト・リソースのセキュリティー検査に限られます。 代替ユーザー・セキュリティー検査とコンテキスト・セキュリティー検査では、検査を要求しているユーザー ID が使用されます。 ユーザー ID の処理方法の詳細については、234 ページの『セキュリティー検査のためのユーザー ID (z/OS)』を参照してください。 キューを開くときのオプションと、キュー・セキュリティー、コンテキスト・セキュリティー、代替ユーザー・セキュリティーがすべてアクティブになっているときに必要なセキュリティー検査をまとめた表については、205 ページの表 36 を参照してください。

代替ユーザー・プロファイルによって、代替ユーザー ID として指定されているユーザー ID に関連したリソースに対するアクセス権が要求側のユーザー ID に与えられます。 例えば、キュー・マネージャー QMPY でユーザー ID PAYSERV によって実行する給与計算サーバーで、PS で始まるすべてのスタッフのユーザー ID からの要求を処理するとします。 その給与計算サーバーの処理を要求側ユーザーのユーザー ID で実行する場合は、代替ユーザー権限が使用されることになります。 要求側のプログラムは、MQPMO_DEFAULT_CONTEXT メッセージ書き込みオプションを使用してメッセージを生成するので、その給与計算サーバーは、代替ユーザー ID としてどのユーザー ID を指定したらよいのかを判別できます。 代替ユーザー ID をどこから取得するのかについての詳細は、234 ページの『セキュリティー検査のためのユーザー ID (z/OS)』 を参照してください。

サーバー・プログラムが PS という文字で始まる代替ユーザー ID を指定できるようにするための RACF 定義の例を以下に示します。

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

注 :

1. オブジェクト記述子とサブスクリプション記述子の *AlternateUserId* フィールドの長さは 12 バイトです。 プロファイル検査では 12 バイトがすべて使用されますが、IBM MQ でユーザー ID として使用されるのは、最初の 8 バイトだけです。 ユーザー ID の切り捨てが望ましくない場合は、要求側のアプリケーション・プログラムによって、8 バイトを超える代替ユーザー ID を適切な形式に変換する必要があります。
2. MQOO_ALTERNATE_USER_AUTHORITY、MQSO_ALTERNATE_USER_AUTHORITY、MQPMO_ALTERNATE_USER_AUTHORITY のいずれかを指定した場合に、オブジェクト記述子の

AlternateUserId フィールドを設定しなければ、ブランクのユーザー ID が使用されます。 代替ユーザー・セキュリティー検査の目的では、 *AlternateUserId* 修飾子に使用されるユーザー ID は -BLANK-です。例えば、 `RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-`です。

このプロファイルにアクセスする権限がユーザーにあれば、ブランクのユーザー ID でその他のすべての検査が実行されます。 ブランク・ユーザー ID の詳細については、243 ページの『ブランク・ユーザー ID と UACC レベル』を参照してください。

汎用代替ユーザー・プロファイルを使用できるようなユーザー ID の命名規則があれば、代替ユーザー ID の管理が容易になります。 ない場合は、RACF の RACVARS 機能を使用できます。 RACVARS の使用方法について詳しくは、「*z/OS SecureWay Security Server RACF* セキュリティー管理者のガイド」を参照してください。

代替ユーザー権限で開いたキューにメッセージを書き込むときに、メッセージのコンテキストがキュー・マネージャーによって生成される場合は、MQMD_USER_IDENTIFIER フィールドが代替ユーザー ID に設定されます。

z/OS コンテキスト・セキュリティーのためのプロファイル

IBM MQ では、特定のメッセージの固有のコンテキスト情報へのアクセスを制御するためのプロファイルを使用します。 コンテキストは、メッセージ記述子 (MQMD) に組み込まれます。

コンテキスト・セキュリティーのためのプロファイルの使用

コンテキスト・セキュリティーが活動状態である場合、 以下を行う必要があります。

- 大文字のプロファイルを使用する場合は、**MQADMIN** クラスでプロファイルを定義します。
- 大/小文字混合のプロファイルを使用する場合は、**MXADMIN** クラスでプロファイルを定義します。

プロファイルの名前は `hlq.CONTEXT.queuename` または `hlq.CONTEXT.topicname` です。ここで、

hlq
`qmgr-name` (キュー・マネージャー名) または `qsg-name` (キュー共有グループ名) のいずれかです。

queuename
コンテキスト・プロファイルを定義するキューの完全な名前または汎用プロファイルのいずれかです。

topicname
コンテキスト・プロファイルを定義するトピックのフルネーム、または総称プロファイルのいずれかにすることができます。

キュー・マネージャー名の接頭部が付いていて、キュー名またはトピック名として ** が指定されているプロファイルでは、そのキュー・マネージャーに属するすべてのキューおよびトピックに対するコンテキスト・セキュリティーを制御できます。 個々のキューまたはトピックでこれをオーバーライドするには、そのキューまたはトピックでコンテキストの特定のプロファイルを定義します。

キュー共有グループ名の接頭部が付いていて、キュー名またはトピック名として ** が指定されているプロファイルでは、キュー共有グループ内のキュー・マネージャーに属するすべてのキューおよびトピックのコンテキストを制御できます。 この動作を個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでコンテキストに関するキュー・マネージャー・レベルのプロファイルを定義し、キュー・マネージャー名の接頭部が付いているプロファイルを指定します。 また、キュー名またはトピック名の接尾部が付いたプロファイルを指定することによって、個々のキューまたはトピックでオーバーライドすることもできます。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

このプロファイルに対するアクセス権を対象のグループまたはユーザー ID に与える必要があります。 キューを開くときのコンテキスト・オプションの指定内容に応じて必要になるアクセス権のレベルを以下の表にまとめます。

表 47. コンテキスト・セキュリティーで必要なアクセス権のレベル

MQOPEN オプションまたは MQPUT1 オプション	**hlq.CONTEXT.queuename または hlq.CONTEXT.topicname に必要な RACF アクセス権のレベル**
MQPMO_NO_CONTEXT	コンテキスト・セキュリティー検査なし
MQPMO_DEFAULT_CONTEXT	コンテキスト・セキュリティー検査なし
MQOO_SAVE_ALL_CONTEXT	コンテキスト・セキュリティー検査なし
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	READ
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	READ
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	CONTROL
MQOO_OUTPUT または MQPUT1(USAGE(XMITQ))	CONTROL
MQSUB オプション	
MQSO_SET_IDENTITY_CONTEXT**(注 2)**	UPDATE

注:

1. 分散キューイングで使用されるユーザー ID には、宛先キューにメッセージを書き込むために、`hlq.CONTEXT.queuename` に対する CONTROL アクセス権が必要になります。 使用されるユーザー ID については、238 ページの『チャネル・イニシエーターで使用されるユーザー ID』を参照してください。
2. MQSO_CREATE オプションまたは MQSO_ALTER オプションを指定した MQSUB 要求で MQSD 構造のいずれかの ID コンテキスト・フィールドを設定しようとする場合は、MQSO_SET_IDENTITY_CONTEXT オプションを指定する必要があります。 さらに、宛先キューのコンテキスト・プロファイルに対する適切な権限も必要になります。

システム・コマンド入力キューにコマンドを書き込む場合は、デフォルト・コンテキストのメッセージ書き込みオプションを使用してコマンドに正しいユーザー ID を関連付けます。

例えば、メッセージをキューからオフロードしてキューに再ロードする操作では、IBM MQ に用意されているユーティリティー・プログラム CSQUTIL を使用できます。 CSQUTIL ユーティリティーは、オフロードしたメッセージをキューに復元するときに、MQOO_SET_ALL_CONTEXT オプションを使用してメッセージを元の状態に戻します。 キューを開くときに指定するこのオプションで必要なキュー・セキュリティーのほかに、コンテキスト権限も必要になります。 例えば、キュー・マネージャー MQS1 でグループ BACKGRP にその権限が必要であれば、以下のようにしてその権限を定義します。

```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

指定するオプションや実行するセキュリティーのタイプによっては、キューを開くときに他のタイプのセキュリティー検査も実行される可能性があります。 例えば、キュー・セキュリティー (197 ページの『キュー・セキュリティーのためのプロファイル』を参照) や代替ユーザー・セキュリティー (212 ページの『代替ユーザー・セキュリティーのためのプロファイル』を参照) などが考えられます。 キューを開くときのオプションと、キュー・セキュリティー、コンテキスト・セキュリティー、代替ユーザー・セキュリティーがすべてアクティブになっているときに必要なセキュリティー検査をまとめた表については、205 ページの表 36 を参照してください。

システム・キュー・コンテキスト・セキュリティー

多くのシステム・キューは、IBM MQ の補助的な部分 (チャネル・イニシエーターのアドレス・スペース V 9.1.0 や、IBM MQ Console と REST API で使用する mqweb サーバーなど) からアクセスされます。

これらを実行するユーザー ID には、それらのキューにアクセスするための RACF アクセス権を与える必要があります (216 ページの表 48 を参照)。

表 *48.* コンテキスト操作で必要な *SYSTEM* キューに対するアクセス権

SYSTEM キュー	**分散キューイングのためのチャネル・イニシエーター**	**mqweb サーバー**
SYSTEM.ADMIN.COMMAND.QUEUE	-	CONTROL
SYSTEM.BROKER.CONTROL.QUEUE	CONTROL	-
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	CONTROL	-
SYSTEM.CHANNEL.SYNCQ	CONTROL	-
SYSTEM.CLUSTER.COMMAND.QUEUE	CONTROL	-
SYSTEM.CLUSTER.TRANSMIT.QUEUE	CONTROL	-

z/OS コマンド・セキュリティーのためのプロファイル

コマンドのセキュリティー検査を有効にするには、MQCMDS クラスにプロファイルを追加します。 プロファイル名は、MQSC コマンドに基づいていますが、制御の対象になるのは MQSC コマンドと PCF コマンドの両方です。 プロファイルの適用対象は、キュー・マネージャーまたはキュー共有グループのいずれかになります。

コマンドのセキュリティー検査を実行する場合 (つまり、コマンド・セキュリティー・スイッチ・プロファイル hlq.NO.CMD.CHECKS を定義していない場合) は、MQCMDS クラスにプロファイルを追加する必要があります。

同じセキュリティー・プロファイルで、MQSC コマンドと PCF コマンドの両方を制御します。 コマンド・セキュリティー検査のための RACF プロファイルの名前は、MQSC コマンド名自体に基づいています。 このプロファイルは、以下のような形式になっています。

```
hlq.verb.pkw
```

hlq は、qmgr-name (キュー・マネージャー名) または qsg-name (キュー共有グループ名) のいずれかです。verb は、コマンド名の verb の部分 (例えば ALTER) です。pkw は、オブジェクト・タイプ (例えば、ローカル・キューの場合は QLOCAL) です。

したがって、サブシステム CSQ1 の ALTER QLOCAL コマンドのプロファイル名は、以下のようになります。

```
CSQ1.ALTER.QLOCAL
```

汎用プロファイルを使用して一連のコマンドを保護することにより、保守するプロファイル数を少なくして、結果的にアクセス・リストを少なくすることができます。 まず、具体性の高いプロファイルで保護されていないすべてのコマンドに適用する汎用プロファイルを作成することを検討してください。 そのプロファイルでは UACC(NONE) を定義し、管理者が含まれている RACF グループだけに ALTER アクセス権を与えます。 次に、すべての DISPLAY コマンドに適用する汎用プロファイルを作成し、そのプロファイルに対するアクセスを幅広く認めるようにします。 これらの両端のグループの間に特定のコマンド・セットへのアクセスを必要とするユーザーのグループを特定することができます。その場合、それらのセット用のプロファイルを作成し、それらのユーザー・クラスを表す RACF グループへのアクセスを付与することができます。 ここで大切なのは、不要なコマンドに対するアクセス権をユーザーに与えない、ということです。つまり、「特権は最小限」という原則を適用し、業務に必要なコマンドに対するアクセス権だけをユーザーに与えるようにします。

キュー・マネージャー名の接頭部が付いているプロファイルでは、そのキュー・マネージャーでコマンドの使用が制御されます。 キュー共有グループ名の接頭部が付いているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーでコマンドの使用が制御されます。 そのアクセスを個々のキュー・マネージャーでオーバーライドする場合は、そのキュー・マネージャーでそのコマンドに関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルをチェックします。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

キュー・マネージャー・レベルでコマンド・プロファイルをセットアップすれば、特定のキュー・マネージャーでユーザーがコマンドを実行する操作を制限できます。 あるいは、キュー共有グループでコマンド verb ごとに 1 つのプロファイルを定義することもできます。その場合は、個々のキュー・マネージャーではなくそのプロファイルに基づいてすべてのセキュリティー検査が実行されます。

サブシステム・セキュリティーとキュー共有グループ・セキュリティーの両方がアクティブになっていて、ローカル・プロファイルが見つからない場合は、コマンド・セキュリティー検査によって、キュー共有グループ・プロファイルに対するアクセス権がユーザーにあるかどうかが確認されます。

CMDSCOPE 属性を使用して、キュー共有グループに含まれている他のキュー・マネージャーにコマンドをルーティングする場合は、コマンドが実行される各キュー・マネージャーでセキュリティー検査が行われますが、コマンドが入力されたキュー・マネージャーで検査が行われるとは限りません。

217 ページの表 49 は、IBM MQ MQSC コマンドごとに、コマンド・セキュリティー検査を実行するために必要なプロファイル、および MQCMDS クラス内の各プロファイルに対応するアクセス・レベルを示しています。

222 ページの表 50 は、各 IBM MQ PCF コマンドについて、コマンド・セキュリティー検査を実行するために必要なプロファイルと、MQCMDS クラス内の各プロファイルに対応するアクセス・レベルを示しています。

表 *49. MQSC* コマンドとプロファイルとアクセス権のレベル

コマンド	**MQCMDS のコマンド・プロファイル**	**MQCMDS のアクセス・レベル**	**MQADMIN または MXADMIN のコマンド・リソース・プロファイル**	**MQADMIN または MXADMIN のアクセス・レベル**
ALTER AUTHINFO	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
ALTER BUFFPOOL	hlq.ALTER.BUFFPOOL	ALTER	検査しない	-
ALTER CFSTRUCT	hlq.ALTER.CFSTRUCT	ALTER	検査しない	-
ALTER CHANNEL	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
ALTER NAMELIST	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
ALTER PROCESS	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
ALTER PSID	hlq.ALTER.PSID	ALTER	検査しない	-
ALTER QALIAS	hlq.ALTER.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
ALTER QLOCAL	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	検査しない	-
ALTER QMODEL	hlq.ALTER.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QREMOTE	hlq.ALTER.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
ALTER SECURITY	hlq.ALTER.SECURITY	ALTER	検査しない	-

表 49. *MQSC* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
ALTER SMDS	hlq.ALTER.SMDS	ALTER	検査しない	-
ALTER STGCLASS	hlq.ALTER.STGCLASS	ALTER	検査しない	-
ALTER SUB	hlq.ALTER.SUB	ALTER	検査しない	-
ALTER TOPIC	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
ALTER TRACE	hlq.ALTER.TRACE	ALTER	検査しない	-
ARCHIVE LOG	hlq.ARCHIVE.LOG	CONTROL	検査しない	-
BACKUP CFSTRUCT	hlq.BACKUP.CFSTRUCT	CONTROL	検査しない	-
CLEAR QLOCAL	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
CLEAR TOPICSTR 222 ページの『3』	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
DEFINE AUTHINFO	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DEFINE BUFFPOOL	hlq.DEFINE.BUFFPOOL	ALTER	検査しない	-
DEFINE CFSTRUCT	hlq.DEFINE.CFSTRUCT	ALTER	検査しない	-
DEFINE CHANNEL	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DEFINE LOG	hlq.DEFINE.LOG	ALTER	検査しない	-
DEFINE MAXSMSGS	hlq.DEFINE.MAXSMSGS	ALTER	検査しない	-
DEFINE NAMELIST	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DEFINE PROCESS	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DEFINE PSID	hlq.DEFINE.PSID	ALTER	検査しない	-
DEFINE QALIAS	hlq.DEFINE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QLOCAL	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL	hlq.DEFINE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QREMOTE	hlq.DEFINE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DEFINE STGCLASS	hlq.DEFINE.STGCLASS	ALTER	検査しない	-
DEFINE SUB	hlq.DEFINE.SUB	ALTER	検査しない	-
DEFINE TOPIC	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DELETE AUTHINFO	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DELETE BUFFPOOL	hlq.DELETE.BUFFPOOL	ALTER	検査しない	-
DELETE CFSTRUCT	hlq.DELETE.CFSTRUCT	ALTER	検査しない	-
DELETE CHANNEL	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DELETE NAMELIST	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER

表 49. *MQSC* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
DELETE PROCESS	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DELETE PSID	hlq.DELETE.PSID	ALTER	検査しない	-
DELETE QALIAS	hlq.DELETE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DELETE QLOCAL	hlq.DELETE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QMODEL	hlq.DELETE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QREMOTE	hlq.DELETE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DELETE STGCLASS	hlq.DELETE.STGCLASS	ALTER	検査しない	-
DELETE SUB	hlq.DELETE.SUB	ALTER	検査しない	-
DELETE TOPIC	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DISPLAY ARCHIVE 221 ページの『1』	hlq.DISPLAY.ARCHIVE	READ	検査しない	-
DISPLAY AUTHINFO	hlq.DISPLAY.AUTHINFO	READ	検査しない	-
DISPLAY CFSTATUS	hlq.DISPLAY.CFSTATUS	READ	検査しない	-
DISPLAY CFSTRUCT	hlq.DISPLAY.CFSTRUCT	READ	検査しない	-
DISPLAY CHANNEL	hlq.DISPLAY.CHANNEL	READ	検査しない	-
DISPLAY CHINIT	hlq.DISPLAY.CHINIT	READ	検査しない	-
DISPLAY CHLAUTH	hlq.DISPLAY.CHLAUTH	READ	検査しない	-
DISPLAY CHSTATUS	hlq.DISPLAY.CHSTATUS	READ	検査しない	-
DISPLAY CLUSQMGR	hlq.DISPLAY.CLUSQMGR	READ	検査しない	-
DISPLAY CMDSERV	hlq.DISPLAY.CMDSERV	READ	検査しない	-
DISPLAY CONN 221 ページの『1』	hlq.DISPLAY.CONN	READ	検査しない	-
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	検査しない	-
DISPLAY LOG 221 ページの『1』	hlq.DISPLAY.LOG	READ	検査しない	-
DISPLAY MAXSMSGS	hlq.DISPLAY.MAXSMSGS	READ	検査しない	-
DISPLAY NAMELIST	hlq.DISPLAY.NAMELIST	READ	検査しない	-
DISPLAY PROCESS	hlq.DISPLAY.PROCESS	READ	検査しない	-
DISPLAY PUBSUB	hlq.DISPLAY.PUBSUB	READ	検査しない	-
DISPLAY QALIAS	hlq.DISPLAY.QALIAS	READ	検査しない	-
DISPLAY QCLUSTER	hlq.DISPLAY.QCLUSTER	READ	検査しない	-
DISPLAY QLOCAL	hlq.DISPLAY.QLOCAL	READ	検査しない	-
DISPLAY QMGR	hlq.DISPLAY.QMGR	READ	検査しない	-

表 49. *MQSC* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
DISPLAY QMODEL	hlq.DISPLAY.QMODEL	READ	検査しない	-
DISPLAY QREMOTE	hlq.DISPLAY.QREMOTE	READ	検査しない	-
DISPLAY QSTATUS	hlq.DISPLAY.QSTATUS	READ	検査しない	-
DISPLAY QUEUE	hlq.DISPLAY.QUEUE	READ	検査しない	-
DISPLAY SBSTATUS	hlq.DISPLAY.SBSTATUS	READ	検査しない	-
DISPLAY SMDS	hlq.DISPLAY.SMDS	READ	検査しない	-
DISPLAY SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	検査しない	-
DISPLAY SUB	hlq.DISPLAY.SUB	READ	検査しない	-
DISPLAY SECURITY	hlq.DISPLAY.SECURITY	READ	検査しない	-
DISPLAY STGCLASS	hlq.DISPLAY.STGCLASS	READ	検査しない	-
DISPLAY SYSTEM 221 ページの『1』	hlq.DISPLAY.SYSTEM	READ	検査しない	-
DISPLAY THREAD	hlq.DISPLAY.THREAD	READ	検査しない	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	検査しない	-
DISPLAY TOPIC	hlq.DISPLAY.TOPIC	READ	検査しない	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	検査しない	-
DISPLAY TRACE	hlq.DISPLAY.TRACE	READ	検査しない	-
DISPLAY USAGE 221 ページの『1』	hlq.DISPLAY.USAGE	READ	検査しない	-
MOVE QLOCAL	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
PING CHANNEL	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RECOVER BSDS	hlq.RECOVER.BSDS	CONTROL	検査しない	-
RECOVER CFSTRUCT	hlq.RECOVER.CFSTRUCT	CONTROL	検査しない	-
REFRESH CLUSTER	hlq.REFRESH.CLUSTER	ALTER	検査しない	-
REFRESH QMGR	hlq.REFRESH.QMGR	ALTER	検査しない	-
REFRESH SECURITY	hlq.REFRESH.SECURITY	ALTER	検査しない	-
RESET CFSTRUCT	hlq.RESET.CFSTRUCT	CONTROL	検査しない	-
RESET CHANNEL	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESET CLUSTER	hlq.RESET.CLUSTER	CONTROL	検査しない	-
RESET QMGR	hlq.RESET.QMGR	CONTROL	検査しない	-
RESET QSTATS	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
RESET SMDS	hlq.RESET.SMDS	CONTROL	検査しない	-

表 49. *MQSC* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
RESET TPIPE	hlq.RESET.TPIPE	CONTROL	検査しない	-
RESOLVE CHANNEL	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESOLVE INDOUBT	hlq.RESOLVE.INDOUBT	CONTROL	検査しない	-
RESUME QMGR	hlq.RESUME.QMGR	CONTROL	検査しない	-
RVERIFY SECURITY	hlq.RVERIFY.SECURITY	ALTER	検査しない	-
SET ARCHIVE	hlq.SET.ARCHIVE	CONTROL	検査しない	-
SET CHLAUTH	hlq.SET.CHLAUTH	CONTROL	検査しない	-
SET LOG	hlq.SET.LOG	CONTROL	検査しない	-
SET SYSTEM	hlq.SET.SYSTEM	CONTROL	検査しない	-
START CHANNEL	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
START CHINIT 222 ページの『4』	hlq.START.CHINIT	CONTROL	検査しない	-
START CMDSERV	hlq.START.CMDSERV	CONTROL	検査しない	-
START LISTENER	hlq.START.LISTENER	CONTROL	検査しない	-
START QMGR	なし 221 ページの『2』	-	-	-
START SMDSCONN	hlq.START.SMDSCONN	CONTROL	検査しない	-
START TRACE	hlq.START.TRACE	CONTROL	検査しない	-
STOP CHANNEL	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
STOP CHINIT	hlq.STOP.CHINIT	CONTROL	検査しない	-
STOP CMDSERV	hlq.STOP.CMDSERV	CONTROL	検査しない	-
STOP LISTENER	hlq.STOP.LISTENER	CONTROL	検査しない	-
STOP QMGR	hlq.STOP.QMGR	CONTROL	検査しない	-
STOP SMDSCONN	hlq.STOP.SMDSCONN	CONTROL	検査しない	-
STOP TRACE	hlq.STOP.TRACE	CONTROL	検査しない	-
SUSPEND QMGR	hlq.SUSPEND.QMGR	CONTROL	検査しない	-

注 :

1. これらのコマンドは、キュー・マネージャーによって内部で実行される場合もあります。そのような場合、権限検査は行われません。
2. IBM MQ は、START QMGR コマンドを実行するユーザーの権限を検査しません。 ただし、RACF、または代替のセキュリティー機能を使用して、START QMGR コマンドの結果として発行される START xxxxMSTR コマンドへのアクセスを制御することができます。 そのためには、RACF オペレーター・コマンド (OPERCMDS) クラスにある MVS.START.STC.xxxxMSTR プロファイルに対するアクセス権を制御します。 この手順の詳細については、「*z/OS SecureWay Security Server RACF* セキュリティー管理者のガイド」を参照してください。 この技法を使用しているときに、無許可ユーザーがキュー・マネージャーを開始しようとすると、理由コード 00F30216 で強制終了になります。

3. **hlq.TOPIC.topic** リソースは、TOPICSTR から派生するトピック・オブジェクトを参照します。 詳細については、461 ページの『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。
4. IBM MQ for z/OS V6 より前のリリースでは、セキュリティー検査は MVS.START.STC.CSQ1CHIN 用でした。 IBM MQ for z/OS V6 以降では、リソース名に、追加の JOBNAME 修飾子が追加されています。 これが原因で、チャネル・イニシエーターを開始するときに問題が発生することがあります。

   問題を解決するには、MVS.START.STC を置き換えます。 MVS.START.STC という名前のリソースのプロファイルを持つ *ssid* CHIN。 *ssid* CHIN **.*** または MVS.START.STC。 *ssid* CHIN。 *ssid* CHIN (*ssid* はキュー・マネージャーのサブシステム ID)。 そのためには、RACF UPDATE 権限が必要です。 詳細については、z/OS 製品資料 の「*Operation planning, MVS Commands, RACF Access Authorities, and Resource Names*」を参照してください。

   *ssid*MSTR に対する START には、JOBNAME= パラメーターは含まれません。 一貫性を保つため、MVS.START.STC.ssidMSTR のプロファイルを MVS.START.STC.ssidMSTR.* に更新することができます。

表 *50. PCF* コマンドとプロファイルとアクセス権のレベル

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
Backup CF Structure	hlq.BACKUP.CFSTRUCT	CONTROL	検査しない	-
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change CF Structure	hlq.ALTER.CFSTRUCT	ALTER	検査しない	-
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Namelist	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Change Process	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Change Queue	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	検査しない	-
Change Security	hlq.ALTER.SECURITY	ALTER	検査しない	-
Change SMDS	hlq.ALTER.SMDS	ALTER	検査しない	-
Change Storage Class	hlq.ALTER.STGCLASS	ALTER	検査しない	-
Change Subscription	hlq.ALTER.SUB	ALTER	検査しない	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Clear Topic String 225 ページの『1』	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
Copy Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Copy CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	検査しない	-
Copy Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Copy Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Copy Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Copy Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Copy Subscription	hlq.DEFINE.SUB	ALTER	検査しない	-
Copy Storage Class	hlq.DEFINE.STGCLASS	ALTER	検査しない	-

表 *50. PCF* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	**MQCMDS** のコマンド・プロファイル	**MQCMDS** のアクセス・レベル	**MQADMIN** または **MXADMIN** のコマンド・リソース・プロファイル	**MQADMIN** または **MXADMIN** のアクセス・レベル
Copy Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Create CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	検査しない	-
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Create Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Create Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Storage Class	hlq.DEFINE.STGCLASS	ALTER	検査しない	-
Create Subscription	hlq.DEFINE.SUB	ALTER	検査しない	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete CF Structure	hlq.DELETE.CFSTRUCT	ALTER	検査しない	-
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Namelist	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Delete Process	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Storage Class	hlq.DELETE.STGCLASS	ALTER	検査しない	-
Delete Subscription	hlq.DELETE.SUB	ALTER	検査しない	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Archive	hlq.DISPLAY.ARCHIVE	READ	検査しない	-
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	検査しない	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	検査しない	-
Inquire CF Structure	hlq.DISPLAY.CFSTRUCT	READ	検査しない	-
Inquire CF Structure Names	hlq.DISPLAY.CFSTRUCT	READ	検査しない	-
Inquire CF Structure Status	hlq.DISPLAY.CFSTATUS	READ	検査しない	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	検査しない	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	検査しない	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	検査しない	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	検査しない	-

表 *50. PCF* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	**MQCMDS** のコマンド・プロファイル	**MQCMDS** のアクセス・レベル	**MQADMIN** または **MXADMIN** のコマンド・リソース・プロファイル	**MQADMIN** または **MXADMIN** のアクセス・レベル
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	検査しない	-
Inquire Cluster Queue Manager	hlq.DISPLAY.CLUSQMGR	READ	検査しない	-
Inquire Connection	hlq.DISPLAY.CONNPCF	READ	検査しない	-
Inquire Group	hlq.DISPLAY.GROUP	READ	検査しない	-
Inquire Log	hlq.DISPLAY.LOG	READ	検査しない	-
Inquire Namelist	hlq.DISPLAY.NAMELIST	READ	検査しない	-
Inquire Namelist Names	hlq.DISPLAY.NAMELIST	READ	検査しない	-
Inquire Process	hlq.DISPLAY.PROCESS	READ	検査しない	-
Inquire Process Names	hlq.DISPLAY.PROCESS	READ	検査しない	-
Inquire Pub/Sub Status	hlq.DISPLAY.PUBSUB	READ	検査しない	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	検査しない	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	検査しない	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	検査しない	-
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	検査しない	-
Inquire Security	hlq.DISPLAY.SECURITY	READ	検査しない	-
Inquire SMDS	hlq.DISPLAY.SMDS	READ	検査しない	-
Inquire SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	検査しない	-
Inquire Storage Class	hlq.DISPLAY.STGCLASS	READ	検査しない	-
Inquire Storage Class Names	hlq.DISPLAY.STGCLASS	READ	検査しない	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	検査しない	-
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	検査しない	-
Inquire System	hlq.DISPLAY.SYSTEM	READ	検査しない	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	検査しない	-
トピック名の照会	hlq.DISPLAY.TOPIC	READ	検査しない	-
トピック状況の照会	hlq.DISPLAY.TPSTATUS	READ	検査しない	-
Inquire Usage	hlq.DISPLAY.USAGE	READ	検査しない	-
Move Queue	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Recover CF Structure	hlq.RECOVER.CFSTRUCT	CONTROL	検査しない	-
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	検査しない	-
キュー・マネージャーのリフレッシュ	hlq.REFRESH.QMGR	ALTER	検査しない	-

表 50. PCF コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
Refresh Security	hlq.REFRESH.SECURITY	ALTER	検査しない	-
Reset CF Structure	hlq.RESET.CFSTRUCT	CONTROL	検査しない	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Reset Cluster	hlq.RESET.CLUSTER	CONTROL	検査しない	-
Reset Queue Manager	hlq.RESET.QMGR	CONTROL	検査しない	-
Reset Queue Statistics	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
Reset SMDS	hlq.RESET.SMDS	CONTROL	検査しない	-
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resume Queue Manager	hlq.RESUME.QMGR	CONTROL	検査しない	-
Resume Queue Manager Cluster	hlq.RESUME.QMGR	CONTROL	検査しない	-
Reverify Security	hlq.RVERIFY.SECURITY	ALTER	検査しない	-
Set Archive	hlq.SET.ARCHIVE	CONTROL	検査しない	-
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	検査しない	-
Set Log	hlq.SET.LOG	CONTROL	検査しない	-
Set System	hlq.SET.SYSTEM	CONTROL	検査しない	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Start Channel Initiator	hlq.START.CHINIT	CONTROL	検査しない	-
Start Channel Listener	hlq.START.LISTENER	CONTROL	検査しない	-
Start SMDS Connection	hlq.START.SMDSCONN	CONTROL	検査しない	-
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel Initiator	hlq.STOP.CHINIT	CONTROL	検査しない	-
Stop Channel Listener	hlq.STOP.LISTENER	CONTROL	検査しない	-
Stop SMDS Connection	hlq.STOP.SMDSCONN	CONTROL	検査しない	-
Suspend Queue Manager	hlq.SUSPEND.QMGR	CONTROL	検査しない	-
Suspend Queue Manager Cluster	hlq.SUSPEND.QMGR	CONTROL	検査しない	-

注:

1. **hlq.TOPIC.topic** リソースは、TOPICSTR から派生するトピック・オブジェクトを参照します。 詳細については、461 ページの『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。

V 9.1.0 IBM MQ Console を使用する場合に必要な IBM MQ PCF プロファイルの詳細については、225 ページの『IBM MQ Console - 必要なコマンドセキュリティー・プロファイル』 を参照してください。

V 9.1.0 z/OS *IBM MQ Console* - 必要なコマンドセキュリティー・プロファイル

MQWebAdmin ロールまたは MQWebAdminRO ロールのユーザーによって IBM MQ Console で実行される操作は、mqweb サーバー開始タスクのユーザー ID のセキュリティー・コンテキストの下で実行されます。

IBM MQ Console を使用する場合は、mqweb サーバー開始タスクのユーザー ID に、特定の PCF コマンドを発行するための権限が必要になります。

226 ページの表 51 は、各 IBM MQ PCF コマンドについて、必要なコマンド・セキュリティー・プロファイルと、IBM MQ Console で必要な MQCMDS クラスの各プロファイルに対応するアクセス・レベルを示しています。

表 *51. IBM MQ Console PCF* コマンドとプロファイルとアクセス権のレベル

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Queue	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	検査しない	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Subscription	hlq.DEFINE.SUB	ALTER	検査しない	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Subscription	hlq.DELETE.SUB	ALTER	検査しない	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	検査しない	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	検査しない	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	検査しない	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	検査しない	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	検査しない	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	検査しない	-
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	検査しない	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	検査しない	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	検査しない	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	検査しない	-

表 51. *IBM MQ Console PCF* コマンドとプロファイルとアクセス権のレベル (続き)

コマンド	MQCMDS のコマンド・プロファイル	MQCMDS のアクセス・レベル	MQADMIN または MXADMIN のコマンド・リソース・プロファイル	MQADMIN または MXADMIN のアクセス・レベル
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	検査しない	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	検査しない	-
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	検査しない	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	検査しない	-
トピック名の照会	hlq.DISPLAY.TOPIC	READ	検査しない	-
トピック状況の照会	hlq.DISPLAY.TPSTATUS	READ	検査しない	-
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	検査しない	-
Refresh Security	hlq.REFRESH.SECURITY	ALTER	検査しない	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	検査しない	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

z/OS コマンド・リソース・セキュリティーのためのプロファイル

コマンド・リソース・セキュリティー・スイッチ・プロファイルを定義していない場合 (つまり、コマンドに関連するリソースのセキュリティー検査を実行しようとしている場合) は、該当するクラスに各リソースのリソース・プロファイルを追加する必要があります。 同じセキュリティー・プロファイルで、MQSC コマンドと PCF コマンドの両方を制御します。

コマンド・リソース・セキュリティー・スイッチ・プロファイル `hlq.NO.CMD.RESC.CHECKS` を定義していない場合 (つまり、コマンドに関連するリソースのセキュリティー検査を実行しようとしている場合) は、以下のようにする必要があります。

- 大文字のプロファイルを使用する場合は、各リソースのリソース・プロファイルを **MQADMIN** クラスに追加します。
- 大/小文字混合のプロファイルを使用する場合は、各リソースのリソース・プロファイルを **MXADMIN** クラスに追加します。

同じセキュリティー・プロファイルで、MQSC コマンドと PCF コマンドの両方を制御します。

コマンド・リソース・セキュリティー検査のためのプロファイルは、以下のような形式になっています。

```
hlq.type.resourcename
```

`hlq` は、`qmgr-name` (キュー・マネージャー名) または `qsg-name` (キュー共有グループ名) のいずれかです。

キュー・マネージャー名の接頭部が付いているプロファイルは、そのキュー・マネージャー上でコマンドに関連付けられた資源へのアクセスを制御します。 キュー共有グループ名の接頭部が付いているプロファイルでは、キュー共有グループに含まれているすべてのキュー・マネージャーでコマンドに関連したリソースに対するアクセスが制御されます。 そのアクセスを個々のキュー・マネージャーでオーバーライドす

る場合は、そのキュー・マネージャーでそのコマンド・リソースに関するキュー・マネージャー・レベルのプロファイルを定義します。

キュー・マネージャーがキュー共有グループのメンバーになっている状態で、キュー・マネージャー・レベルとキュー共有グループ・レベルの両方のセキュリティーを使用する場合、IBM MQ は、キュー・マネージャー名の接頭部が付いているプロファイルを最初に検査します。 見つからない場合は、キュー共有グループ名の接頭部が付いているプロファイルを探します。

例えば、サブシステム CSQ1 のモデル・キュー CREDIT.WORTHY に基づくコマンド・リソース・セキュリティー検査のための RACF プロファイル名は、以下のようになります。

```
CSQ1.QUEUE.CREDIT.WORTHY
```

MQADMIN クラスには、すべてのタイプのコマンド・リソースに対応したプロファイルを格納するので、名前が同じでもタイプが異なるリソースを区別するために、プロファイル名の「type」の部分が必要になります。 プロファイル名の「type」の部分は、CHANNEL、QUEUE、TOPIC、PROCESS、NAMELIST のいずれかになります。 例えば、あるユーザーには、hlq.QUEUE.PAYROLL.ONE を定義する権限はあっても、hlq.PROCESS.PAYROLL.ONE を定義する権限はない、というような設定が可能です。

リソース・タイプがキューで、プロファイルがキュー共有グループ・レベルのプロファイルである場合は、そのプロファイルによって、キュー共有グループに含まれている 1 つ以上のローカル・キューに対するアクセスが制御されるか、そのキュー共有グループに含まれているいずれかのキュー・マネージャーから 1 つの共有キューに対するアクセスが制御されます。

z/OS MQSC コマンド、プロファイル、およびそれらのアクセス・レベル は、 IBM MQ MQSC コマンドごとに、コマンド・セキュリティー検査を実行するために必要なプロファイルと、MQCMDS クラス内の各プロファイルに対応するアクセス・レベルを示しています。

z/OS PCF コマンド、プロファイル、およびそれらのアクセス・レベル は、各 IBM MQ PCF コマンドについて、コマンド・セキュリティー検査の実行に必要なプロファイル、および MQCMDS クラス内の各プロファイルに対応するアクセス・レベルを示しています。

z/OS 別名キューとリモート・キューのコマンド・リソース・セキュリティー検査
別名キューとリモート・キューでは、他のキューに対する間接参照が可能になります。 ここでは、これらのキューのセキュリティー検査を検討する場合の追加の注意点を取り上げます。

別名キュー

別名キューを定義すると、コマンド・リソース・セキュリティー検査は、その別名キューの名前だけについて実行され、別名の解決先のターゲット・キューの名前については実行されません。

別名キューは、ローカル・キューとリモート・キューのどちらにも解決される可能性があります。 ユーザーに特定のローカル・キューまたはリモート・キューへのアクセスを許可したくない場合は、次の両方を行う必要があります。

1. それらのローカル・キューとリモート・キューに対するアクセス権をユーザーに与えないようにします。
2. それらのキューの別名を定義する権限をユーザーに与えないように制限します。 つまり、DEFINE QALIAS コマンドと ALTER QALIAS コマンドを実行する権限を与えない、ということです。

リモート・キュー

リモート・キューを定義すると、コマンド・リソース・セキュリティー検査は、そのリモート・キューの名前だけについて実行されます。 リモート・キューのオブジェクト定義の RNAME 属性または XMITQ 属性で指定されているキューの名前については、検査は行われません。

z/OS RESLEVEL セキュリティー・プロファイル

API リソース・セキュリティーで検査するユーザー ID の数を制御するための特別なプロファイルを MQADMIN クラスまたは MXADMIN クラスで定義できます。 このプロファイルは、RESLEVEL プロファイ

ルと呼ばれます。 このプロファイルが API 資源セキュリティーにどのように影響するかは、IBM MQ にアクセスする方法によって異なります。

アプリケーションが IBM MQ に接続しようとすると、IBM MQ は、その接続に関連するユーザー ID が 1 つのプロファイルに対して持っているアクセス権を検査します。そのプロファイルとは、MQADMIN クラスまたは MXADMIN クラスにある以下のプロファイルです。

```
hlq.RESLEVEL
```

hlq は、ssid (サブシステム ID) または qsg (キュー共有グループ ID) のいずれかです。

各接続タイプに関連するユーザー ID は、以下のとおりです。

- バッチ接続用の接続タスクのユーザー ID
- CICS 接続用の CICS アドレス・スペース・ユーザー ID
- IMS 接続用の IMS 領域アドレス・スペース・ユーザー ID
- チャネル・イニシエーター接続用のチャネル・イニシエーター・アドレス・スペース・ユーザー ID

重要 : RESLEVEL は、非常に強力なオプションです。特定の接続に関して、すべてのリソース・セキュリティー検査が迂回される結果になる可能性もあります。

RESLEVEL プロファイルを定義していない場合は、MQADMIN クラスにある他のプロファイルが hlq.RESLEVEL に合致することがないように注意する必要があります。 例えば、MQADMIN に hlq. * * というプロファイルがあるとします。 hlq.RESLEVEL プロファイルがない場合は、hlq. * * の結果に注意してください。 プロファイル (RESLEVEL 検査に使用されるため)

RESLEVEL プロファイルを定義しないでおく代わりに、hlq.RESLEVEL プロファイルを定義して UACC を NONE に設定してください。 アクセス・リストに入れるユーザーまたはグループは、できるだけ少なくします。 RESLEVEL のアクセスを監査する方法の詳細については、254 ページの『z/OS での監査に関する考慮事項』を参照してください。

キュー・マネージャー・レベルのセキュリティーだけを使用する場合、IBM MQ は、qmgr-name.RESLEVEL プロファイルに基づいて RESLEVEL の検査を実行します。 キュー共有グループ・レベルのセキュリティーだけを使用する場合、IBM MQ は、qsg-name.RESLEVEL プロファイルに基づいて RESLEVEL の検査を実行します。 キュー・マネージャー・レベルとキュー共有グループ・レベルのセキュリティーを組み合わせて使用する場合、IBM MQ はまず、キュー・マネージャー・レベルで RESLEVEL プロファイルがあるかどうかを確認します。 見つからない場合は、キュー共有グループ・レベルで RESLEVEL プロファイルをチェックします。

RESLEVEL プロファイルが見つからない場合、IBM MQ は、CICS 接続または IMS 接続で、ジョブ ID とタスク ID (または代替ユーザー ID) の両方の検査を有効にします。 バッチ接続の場合、IBM MQ は、ジョブ・ユーザー ID (または代替ユーザー ID) の検査を有効にします。 チャネル・イニシエーターの場合は、IBM MQ は、チャネル・ユーザー ID と MCA ユーザー ID (または代替ユーザー ID) の検査を有効にします。

RESLEVEL プロファイルが存在する場合の検査のレベルは、環境によっても、そのプロファイルに関するアクセス権のレベルによっても異なります。

キュー・マネージャーがキュー共有グループのメンバーであり、キュー・マネージャー・レベルでこのプロファイルを定義しない場合は、検査のレベルに影響を与えるキュー共有グループ・レベルで定義されたプロファイルが存在する場合があります。2 つのユーザー ID の検査をアクティブ化するには、キュー・マネージャー名またはキュー共有グループ名の接頭部を付けた RESLEVEL プロファイルを定義して UACC(NONE) を設定し、関連するユーザーにそのプロファイルに対するアクセス権を与えないようにします。

RESLEVEL に対するチャネル・イニシエーターのユーザー ID のアクセス権を検討するときには、チャネル・イニシエーターによって確立される接続が、チャネルで使用される接続でもあることを覚えておく必要があります。 つまり、チャネル・イニシエーターのユーザー ID ですべてのリソース・セキュリティー検査を迂回する設定を定義すれば、実質的にすべてのチャネルでセキュリティー検査を迂回する結果になります。RESLEVEL に対するチャネル・イニシエーターのユーザー ID のアクセス権が NONE 以外の値になっていると、アクセス権のレベルが READ または UPDATE の場合は、1 つのユーザー ID のアクセス権だけが検査され、アクセス権のレベルが CONTROL または ALTER の場合は、どのユーザー ID のアクセス権も検査されな

くなります。 RESLEVEL に対するチャネル・イニシエーターのユーザー ID のアクセス権のレベルとして NONE 以外のレベルを設定する場合は、チャネルで実行されるセキュリティー検査にその設定がどんな影響を及ぼすかを正しく理解しておく必要があります。

RESLEVEL プロファイルの使用については、通常のセキュリティー監査レコードが生成されません。 例えば、ユーザーに UAUDIT を設定しても、MQADMIN に含まれている hlq.RESLEVEL プロファイルに対するアクセスは監査されません。

hlq.RESLEVEL プロファイルで RACF の WARNING オプションを使用しても、RESLEVEL クラスのプロファイルで RACF の警告メッセージが生成されることはありません。

COD などの報告メッセージのセキュリティー検査は、発信元のアプリケーションに関連した RESLEVEL プロファイルによって制御されます。 例えば、バッチ・ジョブのユーザー ID に、RESLEVEL プロファイルに対する CONTROL または ALTER 権限がある場合には、 バッチ・ジョブによって実行されるすべてのリソース検査 (報告メッセージのセキュリティー検査も含む) が迂回されます。

RESLEVEL プロファイルを変更した場合、その変更を有効にするためには、ユーザーはいったん切断して再び接続することが必要です。 (特に、RESLEVEL プロファイルに対する分散キューイング・アドレス・スペースのユーザー ID のアクセス権を変更した場合は、チャネル・イニシエーターをいったん停止して再始動することも必要になります。)

RESLEVEL の監査をオフにする場合は、RESAUDIT システム・パラメーターを使用します。

z/OS *RESLEVEL とバッチ接続*

デフォルトでは、バッチ接続またはバッチ・タイプの接続で IBM MQ リソースにアクセスするユーザーには、その操作でそのリソースにアクセスするための権限が必要です。 セキュリティー検査を迂回する場合は、該当する RESLEVEL 定義をセットアップします。

ユーザーが検査されるか検査されないかの決定は、接続時に使用されるユーザー ID (接続検査で使用されるのと同じユーザー ID) に基づいています。

例えば、信頼できるユーザーがバッチ接続で特定のリソースにアクセスする場合は、API リソース・セキュリティー検査を行わず、信頼できないユーザーが同じリソースにアクセスしようとするときは、通常のセキュリティー検査を行う、というような条件を満たすように RESLEVEL をセットアップできます。 API リソース・セキュリティー検査を迂回するように RESLEVEL 検査をセットアップするのは、ユーザーとそのユーザーによって実行されるプログラムを十分に信頼できる場合に限る必要があります。

バッチ接続で行われる検査を以下の表にまとめます。

表 52. バッチ接続時に *RACF* アクセス権の各レベルで行われる検査

RACF のアクセス・レベル	**検査のレベル**
NONE	リソース検査実行
READ	リソース検査実行
UPDATE	リソース検査実行
CONTROL	検査なし
ALTER	検査なし

z/OS *RESLEVEL とシステム機能*

操作と制御パネルと CSQUTIL に対する RESLEVEL の適用。

操作と制御パネルと CSQUTIL ユーティリティーは、キュー・マネージャーのコマンド・サーバーに対して要求を送信するバッチ・タイプのアプリケーションなので、230 ページの『RESLEVEL とバッチ接続』で取り上げている考慮事項が当てはまります。RESLEVEL を使用して、使用する SYSTEM.COMMAND.INPUT および SYSTEM.COMMAND.REPLY.MODEL キューのセキュリティー検査をバイパスすることができますが、動的キュー SYSTEM.CSQXCMD.* については、これをバイパスすることができません。
SYSTEM.CSQOREXX.*、および SYSTEM.CSQUTIL.*

コマンド・サーバーは、キュー・マネージャーの不可欠な部分なので、接続検査や RESLEVEL 検査が関連付けられていません。 したがって、セキュリティーを確保するために、コマンド・サーバーでは、応答のために使用するキューを開く権限が要求側のアプリケーションのユーザー ID にあることを確認する必要があります。 操作と制御パネルの場合、そのキューは SYSTEM.CSQOREXX.* です。 CSQUTIL の場合は、SYSTEM.CSQUTIL.* です。 ユーザーには、RESLEVEL 権限のほかに、それらのキューを使用する権限も必要です (203 ページの『システム・キュー・セキュリティー』を参照してください)。

コマンド・サーバーを使用する他のアプリケーションの場合は、応答先キューとして指定されているキューが対象になります。 そのようなアプリケーションは、自身のユーザー ID よりも信頼できるユーザー ID を (メッセージ・コンテキストで) コマンド・サーバーに渡すことによって、コマンド・サーバーのチェックをかいくぐり、権限のないキューにメッセージを書き込もうとする可能性があります。 そのような操作を防止するには、CONTEXT プロファイルを使用して、SYSTEM.COMMAND.INPUT に書き込まれるメッセージの ID コンテキストを保護します。

z/OS *RESLEVEL と CICS 接続*

CICS 接続で API リソース・セキュリティー検査が行われる場合、デフォルトでは 2 つのユーザー ID が検査されます。 RESLEVEL プロファイルをセットアップすれば、検査対象のユーザー ID を変更できます。

検査対象になる最初のユーザー ID は、CICS アドレス・スペースのユーザー ID です。 これは、 CICS ジョブのジョブ・カード上のユーザー ID、または z/OS STARTED クラスまたは開始プロシージャー・テーブルによって CICS 開始タスクに割り当てられたユーザー ID です。 (CICS の DFLTUSER ではありません。)

検査対象になる 2 番目のユーザー ID は、CICS トランザクションに関連するユーザー ID です。

このいずれかのユーザー ID に、リソースに対するアクセス権がなければ、要求は完了コード MQRC_NOT_AUTHORIZED で失敗します。 CICS アドレス・スペースのユーザー ID と CICS トランザクションを実行するユーザーのユーザー ID の両方に、リソースに対する正しいレベルのアクセス権が必要です。

実行される検査に対する RESLEVEL の影響

RESLEVEL プロファイルのセットアップ方法によって、リソースへのアクセスが要求されたときに検査するユーザー ID を変更できます。 詳しくは、 231 ページの表 53 を参照してください。

検査されるユーザー ID は、接続時のユーザー ID、つまり CICS アドレス・スペースのユーザー ID によって異なります。 この制御により、あるシステムからの IBM MQ 要求 (例えば、テスト・システム、TESTCICS) に対しては API リソース・セキュリティー検査をバイパスし、別のシステム (例えば、実動システム、PRODCICS) に対しては実装することができます。

注 : CICS アドレス・スペース・ユーザー ID を STARTED クラスの "「trusted」" 属性または RACF 開始プロシージャー・テーブル ICHRIN 03 で設定した場合、キュー・マネージャーの RESLEVEL プロファイルによって確立された CICS アドレス・スペースに対するユーザー ID 検査はすべてオーバーライドされます (つまり、キュー・マネージャーは、CICS アドレス・スペースのセキュリティー検査を実行しません)。 詳細については、「*CICS Transaction Server for z/OS V3.2 RACF Security Guide*」を参照してください。

CICS 接続で行われる検査を以下の表にまとめます。

表 *53. CICS* 接続のさまざまな *RACF* アクセス・レベルで行われる検査

RACF のアクセス・レベル	**検査のレベル**
NONE	IBM MQ によって CICS アドレス・スペースのユーザー ID とトランザクションのユーザー ID が検査されます。
READ	IBM MQ によって CICS アドレス・スペースのユーザー ID だけが検査されます。
UPDATE	トランザクションが RESSEC(YES) を指定して CICS に定義されている場合、IBM MQ は、CICS アドレス・スペースのユーザー ID とトランザクション・ユーザー ID を検査します。

表 53. CICS 接続のさまざまな RACF アクセス・レベルで行われる検査 (続き)	
RACF のアクセス・レベル	**検査のレベル**
UPDATE	トランザクションが RESSEC(NO) を指定して CICS に対して定義されている場合、IBM MQ は、CICS アドレス・スペースのユーザー ID のみを検査します。
CONTROL または ALTER	IBM MQ によってどのユーザー ID も検査されません。

z/OS *RESLEVEL と IMS 接続*

IMS 接続で API リソース・セキュリティー検査が行われる場合、デフォルトでは 2 つのユーザー ID が検査されます。 RESLEVEL プロファイルをセットアップすれば、検査対象のユーザー ID を変更できます。

IMS 接続で API リソース・セキュリティー検査が行われる場合、デフォルトでは 2 つのユーザー ID が検査され、リソースに対するアクセスが認められるかどうかが確認されます。

検査対象になる最初のユーザー ID は、IMS 領域のアドレス・スペースのユーザー ID です。 つまり、ジョブ・カードの USER フィールドで記述されているユーザー ID か、z/OS の STARTED クラスまたは開始済みプロシージャー・テーブル (SPT) で領域に割り当てられているユーザー ID です。

検査対象になる 2 番目のユーザー ID は、従属領域で実行される処理に関連したユーザー ID です。 このユーザー ID は、IMS(tm) 接続に関する 2 番目のユーザー ID の決定方法に示すように、従属領域の種類によって決定されます。

1 番目と 2 番目の IMS ユーザー ID のいずれかに、リソースに対するアクセス権がなければ、要求は完了コード MQRC_NOT_AUTHORIZED で失敗します。

IBM MQ RESLEVEL プロファイルの設定では、IMS トランザクションが IBM 提供の MQ IMS トリガー・モニター・プログラム CSQQTRMN からスケジュールされるユーザー ID を変更することはできません。 そのユーザー ID は、そのトリガー・モニターの PSBNAME であり、デフォルトでは CSQQTRMN になります。

実行される検査に対する RESLEVEL の影響

RESLEVEL プロファイルのセットアップ方法によって、リソースへのアクセスが要求されたときに検査するユーザー ID を変更できます。 以下のような検査が可能です。

- IMS 領域のアドレス・スペースのユーザー ID と 2 番目のユーザー ID または代替ユーザー ID が検査されます。
- IMS 領域のアドレス・スペースのユーザー ID だけが検査されます。
- どのユーザー ID も検査されません。

IMS 接続で行われる検査を以下の表にまとめます。

表 54. IMS 接続のさまざまな RACF アクセス・レベルで行われる検査	
RACF のアクセス・レベル	**検査のレベル**
NONE	IMS アドレス・スペースのユーザー ID と IMS の 2 番目のユーザー ID または代替ユーザー ID が検査されます。
READ	IMS アドレス・スペースのユーザー ID が検査されます。
UPDATE	IMS アドレス・スペースのユーザー ID が検査されます。
CONTROL	検査なし
ALTER	検査なし

z/OS *RESLEVEL とチャネル・イニシエーター接続*

チャネル・イニシエーターによって API リソース・セキュリティー検査が行われる場合、デフォルトでは 2 つのユーザー ID が検査されます。 RESLEVEL プロファイルをセットアップすれば、検査対象のユーザー ID を変更できます。

チャネル・イニシエーターによって API リソース・セキュリティー検査が行われる場合、デフォルトでは 2 つのユーザー ID が検査され、リソースに対するアクセスが認められるかどうかが確認されます。

検査対象のユーザー ID になる可能性があるのは、MCAUSER チャネル属性で指定されているユーザー ID、ネットワークから受け取ったユーザー ID、チャネル・イニシエーター・アドレス・スペースのユーザー ID、またはメッセージ記述子の代替ユーザー ID です。 どのユーザー ID が検査されるかは、使用している通信プロトコルと PUTAUT チャネル属性の設定によって異なります。 詳しくは、238 ページの『チャネル・イニシエーターで使用されるユーザー ID』 を参照してください。

このいずれかのユーザー ID に、リソースに対するアクセス権がなければ、要求は完了コード MQRC_NOT_AUTHORIZED で失敗します。

実行される検査に対する RESLEVEL の影響

RESLEVEL プロファイルのセットアップ方法によって、リソースへのアクセスが要求されたときに検査するユーザー ID とそのユーザー ID の数を変更できます。

チャネル・イニシエーターの接続で行われる検査を以下の表にまとめます (すべてのチャネルはこの接続を使用するので、以下の検査はすべてのチャネルでも行われます)。

表 *55.* チャネル・イニシエーター接続時に *RACF* アクセス権の各レベルで行われる検査

RACF のアクセス・レベル	**検査のレベル**
NONE	2 つのユーザー ID が検査されます。
READ	1 つのユーザー ID が検査されます。
UPDATE	1 つのユーザー ID が検査されます。
CONTROL	検査なし
ALTER	検査なし

注 : 検査されるユーザー ID の定義については、238 ページの『チャネル・イニシエーターで使用されるユーザー ID』 を参照してください。

z/OS *RESLEVEL* とグループ内キューイング

グループ内キューイング・エージェントによって API リソース・セキュリティー検査が行われる場合、デフォルトでは 2 つのユーザー ID が検査され、リソースに対するアクセスが認められるかどうかが確認されます。 RESLEVEL プロファイルをセットアップすれば、検査対象のユーザー ID を変更できます。

検査対象のユーザー ID になる可能性があるのは、受信側キュー・マネージャーの IGQUSER 属性で指定されているユーザー ID、SYSTEM.QSG.TRANSMIT.QUEUE にメッセージを書き込むキュー共有グループに含まれているキュー・マネージャーのユーザー ID、メッセージのメッセージ記述子の *UserIdentifier* フィールドで指定されている代替ユーザー ID です。 詳しくは、242 ページの『グループ内キューイング・エージェントで使用されるユーザー ID』 を参照してください。

グループ内キューイング・エージェントは、キュー・マネージャーの内部タスクなので、明示的な接続要求なしで、キュー・マネージャーのユーザー ID で実行されます。 グループ内キューイング・エージェントが開始されるのは、キュー・マネージャーの初期化時です。 グループ内キューイング・エージェントの初期設定時に、IBM MQ は、キュー・マネージャーに関連付けられたユーザー ID が MQADMIN クラス内の次のようなプロファイルに対して持っているアクセスを検査します。

```
hlq.RESLEVEL
```

この検査は、hlq.NO.SUBSYS.SECURITY スイッチが設定されていない限り、常に実行されます。

RESLEVEL プロファイルがないと、IBM MQ では、2 つのユーザー ID の検査が有効になります。 RESLEVEL プロファイルが存在する場合の検査のレベルは、そのプロファイルに関してキュー・マネージャーのユーザー ID に与えられているアクセス権のレベルによって異なります。 グループ内キューイング・エージェン

トに関して RACF(r) アクセス権の各レベルで行われる検査は、グループ内キューイング・エージェントで行われる検査を示しています。

表 56. グループ内キューイング・エージェントに関して *RACF* アクセス権の各レベルで行われる検査	
RACF のアクセス・レベル	**検査のレベル**
NONE	2 つのユーザー ID が検査されます。
READ	1 つのユーザー ID が検査されます。
UPDATE	1 つのユーザー ID が検査されます。
CONTROL	検査なし
ALTER	検査なし
注: 検査されるユーザー ID の定義については、242 ページの『グループ内キューイング・エージェントで使用されるユーザー ID』を参照してください。	

RESLEVEL プロファイルに関してキュー・マネージャーのユーザー ID に与えられているアクセス権を変更した場合は、その新しいアクセス権を適用するために、グループ内キューイング・エージェントをいったん停止して再始動する必要があります。 グループ内キューイング・エージェントを単独で停止して再始動する方法はないので、そのためには、キュー・マネージャーを停止して再始動しなければなりません。

z/OS *RESLEVEL* と検査されるユーザー *ID*

RESLEVEL プロファイルを設定し、そのプロファイルに対するアクセス権を与える例。

それぞれの MQI 要求で検査されるユーザー ID が RESLEVEL によってどのように決定されるかをまとめたのが、バッチ接続でプロファイル名に基づいて行われるユーザー ID 検査から LU 6.2 および TCP/IP サーバー接続チャネルのためのユーザー ID のプロファイル名の検査です。

例えば、QM66 というキュー・マネージャーに以下のような要件があるとします。

- ユーザー WS21B ではリソース・セキュリティーを免除します。
- CICS によって開始され、アドレス・スペース・ユーザー ID CICSWXN の下で実行するタスク WXNCICS は、RESSEC(YES) で定義されたトランザクションに対してのみ完全なリソース検査を実行します。

この要件に該当する RESLEVEL プロファイルを定義するには、以下の RACF コマンドを実行します。

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

次に、以下のコマンドを使用して、そのプロファイルに対するアクセス権をユーザーに与えます。

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

ユーザー ID がキュー・マネージャー QM66 に接続している状態でこの変更を実行した場合は、その変更を有効にするために、ユーザーがいったん切断して再び接続することが必要です。

ユーザーが接続した時点でサブシステム・セキュリティーがアクティブになっていなかったとしても、そのユーザーの接続中にサブシステム・セキュリティーがアクティブになると、全リソース・セキュリティー検査がそのユーザーに適用されます。 そのユーザーに正しい RESLEVEL 処理が適用されるようにするには、再接続が必要です。

z/OS セキュリティー検査のためのユーザー ID (z/OS)

IBM MQ は、ユーザー、端末、アプリケーション、その他のリソースに関連したユーザー ID に基づいてセキュリティー検査を開始します。 このトピック集では、セキュリティー検査の各タイプで使用されるユーザー ID について説明します。

z/OS 接続セキュリティーのためのユーザー *ID*

接続セキュリティーで使用されるユーザー ID は、接続のタイプによって異なります。

接続タイプ	ユーザー ID の内容
バッチ接続	接続側のタスクのユーザー ID。 以下に例を示します。 • TSO のユーザー ID • USER JCL パラメーターでバッチ・ジョブに割り当てられているユーザー ID • STARTED クラスまたは開始済みプロシージャー・テーブルで開始済みタスクに割り当てられているユーザー ID
CICS 接続	CICS アドレス・スペースのユーザー ID。
IMS 接続	IMS 領域アドレス・スペースのユーザー ID。
チャネル・イニシエーター接続	チャネル・イニシエーター・アドレス・スペースのユーザー ID。

z/OS コマンドとコマンド・リソース・セキュリティーのためのユーザー *ID*

コマンド・セキュリティーとコマンド・リソース・セキュリティーで使用されるユーザー ID は、コマンドをどこから実行するかによって異なります。

実行元	ユーザー ID の内容
CSQINP1、CSQINP2、または CSQINPT	検査は行われません。
システム・コマンド入力キュー	コマンドが含まれているメッセージのメッセージ記述子の *UserIdentifier* にあるユーザー ID。 メッセージに *UserIdentifier* が含まれていない場合は、ブランクのユーザー ID がセキュリティー・マネージャーに渡されます。
コンソール	コンソールにサインオンしたユーザー ID。 コンソールへのサインオンが行われていない場合は、CSQ6SYSP の CMDUSER システム・パラメーターで設定されているデフォルトのユーザー ID が使用されます。 コンソールからコマンドを実行する場合は、コンソールで z/OS の SYS AUTHORITY 属性を設定しておく必要があります。
SDSF/TSO コンソール	TSO またはジョブのユーザー ID。
操作と制御パネル	TSO のユーザー ID。 操作と制御パネルを使用する場合は、選択するアクションに対応するコマンドを実行するための適切な権限が必要になります。 さらに、すべての hlq.DISPLAY に対する READ アクセス権限を持っている必要があります。 パネルはさまざまな DISPLAY コマンドを使用して、表示される情報を収集するため、MQCMDS クラスの オブジェクト ・プロファイル。
MGCRE	UTOKEN で MGCRE を使用する場合は、UTOKEN に格納されているユーザー ID。 UTOKEN なしで MGCRE を実行する場合は、TSO またはジョブのユーザー ID が使用されます。
CSQ0UTIL	ジョブのユーザー ID。
CSQUTIL	ジョブのユーザー ID。
CSQINPX	チャネル・イニシエーター・アドレス・スペースのユーザー ID。

z/OS リソース・セキュリティーのためのユーザー *ID (MQOPEN、MQSUB、MQPUT1)*

ここでは、通常のユーザー ID と代替ユーザー ID の内容を接続タイプごとに取り上げます。 検査する数は、RESLEVEL プロファイルで定義されます。 検査されるユーザー ID は、**MQOPEN**、**MQSUB**、**MQPUT1** の各呼び出しで使用するユーザー ID です。

注 : すべてのユーザー ID フィールドは、受け取ったときの状態でそのまま検査されます。変換は行われません。例えば、"「Bob」"、"「BOB」"、および "「bob」" が含まれている 3 つのユーザー ID フィールドは同等ではありません。

z/OS バッチ接続で検査されるユーザー *ID*

バッチ接続で検査されるユーザー ID は、タスクの実行方法と、代替ユーザー ID が指定されているかどうかによって異なります。

表 *57.* バッチ接続でプロファイル名に基づいて行われるユーザー *ID* 検査

オープン時の代替ユーザー ID の指定	hlq.ALTERNATE.USER.userid プロファイル	hlq.CONTEXT.queuename プロファイル	hlq.resourcename プロファイル
NO	-	JOB	JOB
YES	JOB	JOB	ALT (T)

キー

ALT (T)
代替ユーザー ID。

JOB

- TSO または USS のサインオンのユーザー ID。
- バッチ・ジョブに割り当てられているユーザー ID。
- STARTED クラスまたは開始済みプロシージャー・テーブルで開始済みタスクに割り当てられているユーザー ID。
- 実行側の Db2 ストアード・プロシージャーに関連するユーザー ID。

RESLEVEL が READ に設定されていて、代替ユーザー ID の検査がオフになっている Q1 というキューに対して、バッチ・ジョブで MQPUT1 を実行します。

「バッチ接続のために異なる RACF (r) アクセス・レベルで行われる検査」 および 「バッチ接続のプロファイル名に対するユーザー ID 検査」 は、ジョブ・ユーザー ID がプロファイル hlq.Q1 に対して検査されることを示します。

z/OS *CICS* 接続で検査されるユーザー *ID*

CICS 接続で検査されるユーザー ID は、実行される検査が 1 つなのか 2 つなのか、代替ユーザー ID が指定されているかどうかによって異なります。

表 *58.* プロファイル名に基づく *CICS* タイプ・ユーザー *ID* の検査

オープン時の代替ユーザー ID の指定	hlq.ALTERNATE.USER.userid プロファイル	hlq.CONTEXT.queuename プロファイル	hlq.resourcename プロファイル
なし。*1* 種類の検査	-	ADS	ADS
なし。*2* 種類の検査	-	ADS+TXN	ADS+TXN
あり。*1* 種類の検査	ADS	ADS	ADS
あり。*2* 種類の検査	ADS+TXN	ADS+TXN	ADS+ALT

キー

ALT (T)
代替ユーザー ID

ADS
CICS バッチ・ジョブに関連するユーザー ID か、CICS が開始済みタスクとして実行されている場合は、STARTED クラスまたは開始済みプロシージャー・テーブルで指定されているユーザー ID。

TXN
CICS トランザクションに関連するユーザー ID。 通常は、トランザクションを開始した端末ユーザーのユーザー ID です。 CICS の DFLTUSER、PRESET セキュリティー端末、手動サインオン・ユーザーなどが考えられます。

以下の条件で検査されるユーザー ID を判別してください。

- CICS アドレス・スペース・ユーザー ID の RESLEVEL プロファイルに対する RACF アクセス・レベルは、NONE に設定されます。
- MQOPEN 呼び出しを MQOO_OUTPUT と MQOO_PASS_IDENTITY_CONTEXT のキューに対して実行します。

まず、RESLEVEL プロファイルに対する CICS アドレス・スペースのユーザー ID のアクセス権に基づいて検査される CICS のユーザー ID の数を確認します。 トピック 231 ページの『RESLEVEL と CICS 接続』の 231 ページの表 53 からすると、RESLEVEL プロファイルが NONE に設定されている場合は、2 つのユーザー ID が検査されます。 さらに、236 ページの表 58 からすると、以下の検査が実行されます。

- hlq.ALTERNATE.USER.userid プロファイルに基づく検査は行われません。
- hlq.CONTEXT.queuename プロファイルに基づいて、CICS アドレス・スペースのユーザー ID と CICS トランザクションのユーザー ID の両方が検査されます。
- hlq.resourcename プロファイルに基づいて、CICS アドレス・スペースのユーザー ID と CICS トランザクションのユーザー ID の両方が検査されます。

したがって、この MQOPEN 呼び出しでは、4 つのセキュリティー検査が実行されます。

z/OS *IMS* 接続で検査されるユーザー *ID*
IMS 接続で検査されるユーザー ID は、実行される検査が 1 つなのか 2 つなのか、代替ユーザー ID が指定されているかどうかによって異なります。 2 番目のユーザー ID が検査される場合、そのユーザー ID は、従属領域のタイプと、どのユーザー ID が有効かによって異なります。

表 *59.* プロファイル名に基づく *IMS* タイプ・ユーザー *ID* の検査

オープン時の代替ユーザー ID の指定	hlq.ALTERNATE.USER.userid プロファイル	hlq.CONTEXT.queuename プロファイル	hlq.resourcename プロファイル
なし。*1* 種類の検査	-	REG	REG
なし。*2* 種類の検査	-	REG+SEC	REG+SEC
あり。*1* 種類の検査	REG	REG	REG
あり。*2* 種類の検査	REG+SEC	REG+SEC	REG+ALT

キー

ALT (T)
代替ユーザー ID。

REG
通常は、STARTED クラスまたは開始済みプロシージャー・テーブルで設定されているユーザー ID か、IMS が実行中の場合は、実行するジョブの USER JCL パラメーターで設定されているユーザー ID です。

SEC
2 番目のユーザー ID は、従属領域で実行される処理に関連したユーザー ID です。 その決定方法については、238 ページの表 60 を参照してください。

表 60. IMS 接続に関する 2 番目のユーザー ID の決定方法	
従属領域のタイプ	**2 番目のユーザー ID の決定のための階層**
• BMP はメッセージ駆動型で、GET UNIQUE が正常に実行されます。 • IFP と GET UNIQUE が実行されます。 • MPP。	ユーザーがサインオンしている場合は、IMS トランザクションに関連するユーザー ID。 使用可能な場合は、LTERM 名 PSBNAME。
• BMP はメッセージ駆動型で、GET UNIQUE が正常に実行されません。 • BMP はメッセージ駆動型ではありません。 • IFP と GET UNIQUE は実行されません。	IMS 従属領域のアドレス・スペースに関連するユーザー ID がすべてブランクまたはすべてゼロでなければ、そのユーザー ID。 PSBNAME。

z/OS チャネル・イニシエーターで使用されるユーザー *ID*

このトピック集では、受信側チャネルで使用および検査されるユーザー ID と、サーバー接続チャネルで発行されるクライアント MQI 要求で使用および検査されるユーザー ID について説明します。 TCP/IP および LU6.2 に関する情報が提供されます。

受信側チャネル定義の PUTAUT パラメーターを使用すれば、実行されるセキュリティー検査のタイプを確認できます。 IBM MQ ネットワークで一貫したセキュリティー検査を実施するには、ONLYMCA オプションと ALTMCA オプションを使用します。

DISPLAY CHSTATUS コマンドを使用すれば、MCA で使用されるユーザー ID を確認できます。

z/OS *TCP/IP* を使用する受信側チャネル

検査されるユーザー ID は、チャネルの PUTAUT オプションと、実行される検査が 1 つなのか 2 つなのかによって異なります。

表 61. TCP/IP チャネルでプロファイル名に基づいて検査されるユーザー ID			
受信側チャネルまたは要求側チャネルで指定されている PUTAUT オプション	**hlq.ALTERNATE.USER.userid プロファイル**	**hlq.CONTEXT.queuename プロファイル**	**hlq.resourcename プロファイル**
***DEF*、*1* つの検査**	-	CHL	CHL
***DEF*、*2* つの検査**	-	CHL + MCA	CHL + MCA
***CTX*、*1* つの検査**	CHL	CHL	CHL
***CTX*、*2* つの検査**	CHL + MCA	CHL + MCA	CHL + ALT
***ONLYMCA*、*1* つの検査**	-	MCA	MCA
***ONLYMCA*、*2* つの検査**	-	MCA	MCA
***ALTMCA*。*1* 種類の検査**	MCA	MCA	MCA
***ALTMCA*。*2* 種類の検査**	MCA	MCA	MCA + ALT

キー

MCA (MCA ユーザー ID)
受信側の MCAUSER チャネル属性で指定されているユーザー ID。ブランクの場合は、受信側または要求側のチャネル・イニシエーター・アドレス・スペースのユーザー ID が使用されます。

CHL (チャネル・ユーザー ID)
TCP/IP では、チャネルの通信システムでセキュリティーがサポートされているわけではありません。 Transport Layer Security (TLS) が使用されていて、デジタル証明書がパートナーからフローされている場合は、この証明書に関連付けられたユーザー ID (証明書がインストールされている場合)、または RACF の証明書名フィルター (CNF) を使用して検出される一致するフィルターに関連付けられたユーザー ID が使用されます。 関連するユーザー ID が見つからない場合や、TLS を使用していない場合は、PUTAUT パラメーターとして DEF または CTX が定義されているチャネルで、受信側または要求側のチャネル・イニシエーター・アドレス・スペースのユーザー ID がチャネルのユーザー ID として使用されます。

注 : RACF の証明書名フィルター (CNF) を使用すると、RACF の同じユーザー ID を複数のリモート・ユーザーに割り当てることができます。例えば、同一組織単位内のすべてのユーザーに同じセキュリティー権限を与えるのが自然なので、同じユーザー ID を割り当てることができます。 そうすれば、世界中に散らばっているあらゆるリモート・ユーザーの証明書のコピーをサーバーで保管する必要はなくなり、証明書の管理と配布の作業を大幅に簡略化できます。

チャネルの PUTAUT パラメーターが ONLYMCA または ALTMCA に設定されていると、チャネルのユーザー ID は無視され、受信側または要求側の MCA のユーザー ID が使用されます。 この動作は、TLS を使用する TCP/IP チャネルにも当てはまります。

ALT (代替ユーザー ID)
メッセージのメッセージ記述子のコンテキスト情報 (つまり *UserIdentifier* フィールド) で指定されているユーザー ID。 そのユーザー ID は、ターゲットの宛先キューに対する **MQOPEN** 呼び出しまたは **MQPUT1** 呼び出しが実行される前に、オブジェクト記述子の *AlternateUserID* フィールドに移されます。

z/OS *LU 6.2* を使用する受信側チャネル
検査されるユーザー ID は、チャネルの PUTAUT オプションと、実行される検査が 1 つなのか 2 つなのかによって異なります。

表 *62. LU 6.2* チャネルでプロファイル名に基づいて検査されるユーザー *ID*

受信側チャネルまたは要求側チャネルで指定されている PUTAUT オプション	hlq.ALTERNATE.USER.userid プロファイル	hlq.CONTEXT.queuename プロファイル	hlq.resourcename プロファイル
***DEF*、*1* つの検査**	-	CHL	CHL
***DEF*、*2* つの検査**	-	CHL + MCA	CHL + MCA
***CTX*、*1* つの検査**	CHL	CHL	CHL
***CTX*、*2* つの検査**	CHL + MCA	CHL + MCA	CHL + ALT
***ONLYMCA*、*1* つの検査**	-	MCA	MCA
***ONLYMCA*、*2* つの検査**	-	MCA	MCA
***ALTMCA*。*1* 種類の検査**	MCA	MCA	MCA
***ALTMCA*。*2* 種類の検査**	MCA	MCA	MCA + ALT

キー

MCA (MCA ユーザー ID)
受信側の MCAUSER チャネル属性で指定されているユーザー ID。ブランクの場合は、受信側または要求側のチャネル・イニシエーター・アドレス・スペースのユーザー ID が使用されます。

CHL (チャネル・ユーザー ID)

要求側 - サーバー・チャネル
要求側がチャネルを開始した場合は、ネットワークのユーザー ID (チャネルのユーザー ID) を受け取る機会がありません。

要求側のチャネルで PUTAUT パラメーターが DEF または CTX に設定されている場合は、ネットワークからユーザー ID を受け取らないので、要求側のチャネル・イニシエーター・アドレス・スペースのユーザー ID がチャネルのユーザー ID になります。

PUTAUT パラメーターが ONLYMCA または ALTMCA に設定されていると、チャネルのユーザー ID は無視され、要求側の MCA のユーザー ID が使用されます。

その他のチャネル・タイプ
受信側または要求側のチャネルで PUTAUT パラメーターが DEF または CTX に設定されている場合は、チャネルの開始時に通信システムから受け取るユーザー ID がチャネルのユーザー ID になります。

- 送信側のチャネルのプラットフォームが z/OS の場合は、送信側のチャネル・イニシエーター・アドレス・スペースのユーザー ID をチャネルのユーザー ID として受け取ることになります。
- 送信中のチャネルが異なるプラットフォーム (例えば、AIX) 上にある場合、受け取られるチャネル・ユーザー ID は通常チャネル定義の USERID パラメーターによって提供されます。

受信したユーザー ID がブランクの場合や、ユーザー ID を受信しなかった場合は、ブランクのユーザー ID がチャネルのユーザー ID として使用されます。

ALT (代替ユーザー ID)
メッセージのメッセージ記述子のコンテキスト情報 (つまり *UserIdentifier* フィールド) で指定されているユーザー ID。 そのユーザー ID は、ターゲットの宛先キューに対する MQOPEN 呼び出しまたは MQPUT1 呼び出しが実行される前に、オブジェクト記述子の *AlternateUserID* フィールドに移されます。

z/OS クライアント *MQI* 要求
設定されているユーザー ID と環境変数に応じて、さまざまなユーザー ID が使用される可能性があります。それらのユーザー ID は、さまざまなプロファイルに基づいて検査されます。どのプロファイルに基づくかは、使用されている PUTAUT オプションと、代替ユーザー ID が指定されているかどうかによって決まります。

このセクションでは、TCP/IP と LU 6.2 のサーバー接続チャネルでクライアント MQI 要求を実行したときに検査されるユーザー ID について説明します。 MCA のユーザー ID とチャネルのユーザー ID は、前の各セクションで取り上げた TCP/IP と LU 6.2 のチャネルの場合と同じです。

サーバー接続チャネルでは、MCAUSER 属性がブランクの場合には、クライアントから受け取られたユーザー ID が使用されます。

詳しくは、 95 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。

MQOPEN、**MQSUB**、**MQPUT1** の各クライアント要求では、以下のルールに基づいて、検査で使用されるプロファイルを判別できます。

- 要求が代替ユーザー権限を指定している場合は、 *hlq*.ALTERNATE.USER。 *userid* プロファイル。
- 要求にコンテキスト権限が指定されている場合、 *hlq* に対して検査が行われます。CONTEXT。 *queuename* プロファイル。
- **MQOPEN**、**MQSUB**、**MQPUT1** のすべての要求では、*hlq.resourcename* プロファイルに基づいて検査が行われます。

検査で使用されるプロファイルを判別したら、そのプロファイルに基づいてどのユーザー ID が検査されるのかを以下の表から確認できます。

表 63. LU 6.2 および TCP/IP サーバー接続チャネルのためのユーザー ID のプロファイル名の検査

サーバー接続チャネルで指定されている PUTAUT オプション	オープン時の代替ユーザー ID の指定	hlq.ALTERNATE.USER.userid プロファイル	hlq.CONTEXT.queuename プロファイル	hlq.resourcename プロファイル
***DEF*、*1* つの検査**	いいえ	-	CHL	CHL
***DEF*、*1* つの検査**	Yes	CHL	CHL	CHL
***DEF*、*2* つの検査**	いいえ	-	CHL + MCA	CHL + MCA
***DEF*、*2* つの検査**	Yes	CHL + MCA	CHL + MCA	CHL + ALT
***ONLYMCA*、*1* つの検査**	いいえ	-	MCA	MCA
***ONLYMCA*、*1* つの検査**	Yes	MCA	MCA	MCA
***ONLYMCA*、*2* つの検査**	いいえ	-	MCA	MCA
***ONLYMCA*、*2* つの検査**	Yes	MCA	MCA	MCA + ALT

キー

MCA (MCA ユーザー ID)
サーバー接続の MCAUSER チャネル属性に指定されるユーザー ID。ブランクの場合は、チャネル・イニシエーター・アドレス・スペースのユーザー ID が使用されます。

CHL (チャネル・ユーザー ID)
TCP/IP では、チャネルの通信システムでセキュリティーがサポートされているわけではありません。Transport Layer Security (TLS) が使用されていて、デジタル証明書がパートナーからフローされている場合は、この証明書に関連付けられたユーザー ID (証明書がインストールされている場合)、または RACF の証明書名フィルター (CNF) を使用して検出される一致するフィルターに関連付けられたユーザー ID が使用されます。 関連付けられたユーザー ID が検出されない場合、または TLS が使用されていない場合は、PUTAUT パラメーターを DEF または CTX に設定して定義されたチャネルのチャネル・ユーザー ID として、チャネル・イニシエーター・アドレス・スペースのユーザー ID が使用されます。

注 : RACF の証明書名フィルター (CNF) を使用すると、RACF の同じユーザー ID を複数のリモート・ユーザーに割り当てることができます。例えば、同一組織単位内のすべてのユーザーに同じセキュリティー権限を与えるのが自然なので、同じユーザー ID を割り当てることができます。 そうすれば、世界中に散らばっているあらゆるリモート・ユーザーの証明書のコピーをサーバーで保管する必要はなくなり、証明書の管理と配布の作業を大幅に簡略化できます。

PUTAUT パラメーターがチャネルの ONLYMCA または ALTMCA に設定される場合、チャネル・ユーザー ID は無視され、サーバー接続チャネルの MCA ユーザー ID が使用されます。 この動作は、TLS を使用する TCP/IP チャネルにも当てはまります。

ALT (代替ユーザー ID)
メッセージのメッセージ記述子のコンテキスト情報 (つまり *UserIdentifier* フィールド) で指定されているユーザー ID。 このユーザー ID は、クライアント・アプリケーションのために **MQOPEN**、**MQSUB**、または **MQPUT1** 呼び出しが出される前に、オブジェクト記述子またはサブスクリプション記述子の *AlternateUserID* フィールドに移されます。

z/OS チャネル・イニシエーターの例

RACF のプロファイルに基づいてユーザー ID がどのように検査されるのかを示す例。

ユーザーがキュー・マネージャー QM01 のキューに対して **MQPUT1** 操作を実行します。そのキューは、キュー・マネージャー QM02 の QB というキューに解決されます。 メッセージは、QM01.TO.QM02 という TCP/IP チャネルで送信されます。 RESLEVEL は NONE に設定されています。開く操作は、代替ユーザー ID で実行され、コンテキスト検査が行われます。 受信側チャネル定義では、PUTAUT(CTX) が定義されていて、MCA のユーザー ID が設定されています。 この場合、キュー QB にメッセージを書き込むときに、受信側チャネルではどのユーザー ID が使用されるでしょうか。

「**回答**」 233 ページの表 55 は RESLEVEL が NONE に設定されているため、2 つのユーザー ID が検査されていることを示します。

238 ページの表 61 からすると、PUTAUT が CTX に設定されていて、2 つの検査が行われるので、以下のユーザー ID が検査されます。

- チャネル・イニシエーターのユーザー ID と MCAUSER のユーザー ID が hlq.ALTERNATE.USER.userid プロファイルに基づいて検査されます。
- チャネル・イニシエーターのユーザー ID と MCAUSER のユーザー ID が hlq.CONTEXT.queuename プロファイルに基づいて検査されます。
- チャネル・イニシエーターのユーザー ID とメッセージ記述子 (MQMD) で指定されている代替ユーザー ID が hlq.Q2 プロファイルに基づいて検査されます。

z/OS グループ内キューイング・エージェントで使用されるユーザー *ID*

グループ内キューイング・エージェントが宛先キューを開くときに検査されるユーザー ID は、キュー・マネージャー属性 IGQAUT と IGQUSER の値によって決まります。

指定できるユーザー ID は、次のとおりです。

グループ内キューイング・ユーザー ID (IGQ)
受信側キュー・マネージャーの IGQUSER 属性で指定されているユーザー ID。 その値がブランクに設定されている場合は、受信側キュー・マネージャーのユーザー ID が使用されます。 ただし、受信側キュー・マネージャーには、自身で定義されているすべてのキューにアクセスする権限があるので、受信側キュー・マネージャーのユーザー ID のセキュリティー検査は実行されません。 その場合は、次のようにします。

- 検査されるユーザー ID が 1 つだけで、そのユーザー ID が受信側キュー・マネージャーのユーザー ID であれば、セキュリティー検査は実行されません。 これは、IGQAUT が ONLYIGQ または ALTIGQ に設定されている場合に起こることがあります。
- 検査されるユーザー ID が 2 つで、その一方のユーザー ID が受信側キュー・マネージャーのユーザー ID であれば、もう一方のユーザー ID のセキュリティー検査だけが実行されます。 これは、IGQAUT が DEF、CTX、または ALTIGQ に設定されている場合に発生する可能性があります。
- 検査されるユーザー ID が 2 つで、その両方のユーザー ID が受信側キュー・マネージャーのユーザー ID であれば、セキュリティー検査は実行されません。 これは、IGQAUT が ONLYIGQ に設定されている場合に発生する可能性があります。

送信側キュー・マネージャーのユーザー ID (SND)
SYSTEM.QSG.TRANSMIT.QUEUE にメッセージを書き込むキュー共有グループに含まれているキュー・マネージャーのユーザー ID。

代替ユーザー ID (ALT)
メッセージのメッセージ記述子の *UserIdentifier* フィールドで指定されているユーザー ID。

表 *64.* グループ内キューイングでプロファイル名に基づいて検査されるユーザー *ID*

受信側キュー・マネージャーで指定されている IGQAUT オプション	hlq.ALTERNATE.USER.userid プロファイル	hlq.CONTEXT.queuename プロファイル	hlq.resourcename プロファイル
***DEF*、*1* つの検査**	-	SND	SND

表 64. グループ内キューイングでプロファイル名に基づいて検査されるユーザー ID (続き)			
受信側キュー・マネージャーで指定されている IGQAUT オプション	**hlq.ALTERNATE.USER.userid プロファイル**	**hlq.CONTEXT.queuename プロファイル**	**hlq.resourcename プロファイル**
***DEF*、*2* つの検査**	-	SND +IGQ	SND +IGQ
***CTX*、*1* つの検査**	SND	SND	SND
***CTX*、*2* つの検査**	SND + IGQ	SND +IGQ	SND + ALT
***ONLYIGQ*、*1* つの検査**	-	IGQ	IGQ
***ONLYIGQ*、*2* つの検査**	-	IGQ	IGQ
***ALTIGQ*、*1* つの検査**	-	IGQ	IGQ
***ALTIGQ*、*2* つの検査**	IGQ	IGQ	IGQ + ALT

キー

ALT (T)
代替ユーザー ID。

IGQ
IGQ ユーザー ID。

SND
送信側キュー・マネージャーのユーザー ID。

z/OS ブランク・ユーザー *ID* と *UACC* レベル

ブランク・ユーザー ID が発生すると、RACF の未定義のユーザーがサインオンします。 未定義のユーザーに幅広いアクセス権を与えないようにする必要があります。

ブランク・ユーザー ID が発生する可能性があるのは、ユーザーがコンテキスト・セキュリティーまたは代替ユーザー・セキュリティーを使用してメッセージを操作する場合や、IBM MQ にブランク・ユーザー ID が渡される場合です。 例えば、コンテキストなしでシステム・コマンド入力キューにメッセージが書き込まれるときには、ブランク・ユーザー ID が使用されます。

注 : "⋆ " というユーザー ID (つまり、1 つのアスタリスク文字の後に 7 つのスペースがあるユーザー ID) は、未定義のユーザー ID として扱われます。

IBM MQ がブランク・ユーザー ID を RACF に渡すと、RACF の未定義ユーザーがサインオンされます。 その場合は、すべてのセキュリティー検査で、対象のプロファイルに対する汎用アクセス権 (UACC) が使用されることになります。 アクセス権のレベルの設定によっては、UACC によって未定義のユーザーに幅広いアクセス権が与えられる可能性もあります。

例えば、TSO から以下の RACF コマンドを実行するとします。

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

これにより、z/OS 定義のユーザー ID (アクセス・リストに 入っていないもの) と RACF 未定義ユーザー ID の両方に、 そのキューに対してメッセージの書き込みおよび読み取りが可能となるプロファイルを定義することになります。

ブランク・ユーザー ID の問題からシステムを保護するには、アクセス権のレベルを慎重に計画し、コンテキスト・セキュリティーと代替ユーザー・セキュリティーを使用できるユーザーの数を制限する必要があります。 RACF の未定義ユーザー ID を使用するユーザーが、アクセスすべきでないリソースにアクセスすることを防ぐ必要があります。 その一方で、定義済みのユーザー ID を使用するユーザーには、アクセスを認める必要があります。 そうするには、RACF コマンド PERMIT でアスタリスク (*) のユーザー ID を指定

して、定義済みの全ユーザー ID にリソースへのアクセス権を与えることができます。 こうして、未定義のすべてのユーザー ID (例えば "* ") によるアクセスが拒否されます。 例えば、RACF の未定義のユーザー ID がメッセージの書き込みや取得のためにキューにアクセスすることを防止するには、以下のような RACF コマンドを使用できます。

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

z/OS z/OS ユーザー ID と Multi-Factor Authentication (MFA)

IBM Multi-Factor Authentication for z/OS を使用すると、z/OS セキュリティー管理者は、識別されたユーザーに対して z/OS システムにサインオンするために複数の認証要素 (例えば、パスワードと暗号トークンの両方) を使用するように要求することによって SAF 認証を強化できます。 IBM MFA では、RSA SecureId などの時間ベースのワンタイム・パスワード生成テクノロジーもサポートしています。

ほとんどの場合、 IBM MQ は、 IBM MQ 作業を駆動している CICS またはバッチ・システムにユーザーが「ログオン」した方法を認識しません。サインオンしたユーザー ID 資格情報は、z/OS タスクまたはアドレス・スペースに関連付けられ、 IBM MQ はこれを使用してリソースに対する許可を検査します。 MFA に対して有効になっているユーザー ID は、IBM MQ リソースに対する許可と、CICS および IMS ブリッジで使用されるパスチケットを介した認証に使用できます。

重要 : ただし、 *MQCSP_AUTH_USER_ID_AND_PWD* オプションを指定した MQCONNX API 呼び出しでユーザー ID とパスワードの資格情報を渡す IBM MQ Explorer などのアプリケーションを使用する場合は、特別な考慮事項が適用されます。 IBM MQ には、この API 要求で追加の資格情報を渡す機能はありません。

以下のテキストに制約と回避策を示します。

IBM MQ Explorer

IBM MQ Explorer は、MFA が有効になっているユーザー ID で z/OS システムにログオンする場合には使用できません。2 番目の認証要素を IBM MQ Explorer から z/OS に渡す機能がないからです。

また、ユーザー ID とパスワードの資格情報を再使用するために IBM MQ Explorer によって使用される 2 つの異なるメカニズムがあり、1 回限り使用のパスワードが有効になっている場合は特に注意する必要があります。

1. IBM MQ Explorer には、後でログインするために、難読化された形式のパスワードをローカル・マシンに保管する機能があります。 z/OS キュー・マネージャーへの接続ごとにエクスプローラーでパスワードを要求することによって、この機能を無効にする必要があります。

   そのためには、下記のようにしてください。

   a.「**キュー・マネージャー**」を選択します。

   b. 表示されたリストから、必要なキュー・マネージャーを選択し、そのキュー・マネージャーを右クリックします。

   c. 表示されたメニュー・リストから「**接続詳細**」を選択します。

   d. 次のメニュー・リストから「**プロパティー**」を選択して、「**ユーザー ID**」タブを選択します。

   「**パスワードのプロンプト**」ラジオ・ボタンが選択されていることを確認します。
2. IBM MQ Explorer のさまざまな操作 (キューのメッセージのブラウズ、サブスクリプションのテストなど) によって、最初のログオン時に使用された資格情報を使用して IBM MQ に認証する新規スレッドが開始されます。 パスワード資格情報は再使用できないため、これらの操作は使用できません。

これらの問題に対して、MFA 構成レベルで 2 つの回避策があります。

- アプリケーション ID の MFA 除外を使用して、IBM MQ タスクを MFA 処理から完全に除外します。

  そのためには、以下のコマンドを実行します。

  1. `RDEFINE MFADEF MFABYPASS.USERID.chinuser`

ここで、*chinuser* は、チャネル・イニシエーターのアドレス・スペース・レベルのユーザー ID (STC クラスを介してチャネル・イニシエーターに関連付けられている) です。

2. `PERMIT MFABYPASS.USERID.chinuser CLASS MFADEF ACCESS(READ) ID(explorer user)`

このアプローチの詳細については、「アプリケーション用の IBM MFA のバイパス」を参照してください。

- MFA のアウト・オブ・バンドのサポートを使用します。これは、IBM MFA 1.2 で導入されました。 この方法では、IBM MFA Web サーバーに対して事前認証を行い、ユーザー ID とパスワードに加えて、ポリシーによって決定される追加の認証を指定します。 IBM MFA サーバーでは、IBM MQ Explorer 認証ダイアログで指定するキャッシュ・トークン資格情報が生成されます。 セキュリティー管理者はこの資格情報を適切な期間再生できるので、通常どおり IBM MQ Explorer を使用できます。

このアプローチについて詳しくは、「IBM MFA の概要」を参照してください。

z/OS IBM MQ for z/OS セキュリティー管理

IBM MQ では、ストレージ内のテーブルを使用して、各ユーザーと各ユーザーのアクセス要求に関連した情報を格納します。 そのテーブルを効率的に管理し、IBM MQ から外部セキュリティー・マネージャー (ESM) に送られる要求の数を減らすために、いくつかの制御機能が用意されています。

それらの制御機能は、操作/制御パネルと IBM MQ コマンドの両方で利用できます。

z/OS ユーザー *ID* の再検証

IBM MQ リソースを使用しているユーザーの RACF 定義が変更された場合 (例えば、ユーザーを新しいグループに接続することによって)、次回 IBM MQ リソースにアクセスしようとするときにこのユーザーに再度サインオンするようにキュー・マネージャーに指示することができます。 そのために使用できるのが、IBM MQ の RVERIFY SECURITY コマンドです。

- ユーザー HX0804 は、キュー・マネージャー PRD1 の PAYROLL キューからメッセージを取得したり、そのキューにメッセージを書き込んだりしています。 ところが、HX0804 は、同じキュー・マネージャー (PRD1) のいくつかの PENSION キューにもアクセスしなければならなくなりました。
- データ・セキュリティー管理者は、ユーザー HX0804 を PENSION キューへのアクセスが可能な RACF グループに接続します。
- HX0804 がすぐに (つまり、キュー・マネージャー PRD1 をシャットダウンしたり、HX0804 のタイムアウトを待ったりしないで) PENSION キューにアクセスできるようにするには、 以下の IBM MQ コマンドを使用する必要があります。

```
RVERIFY SECURITY(HX0804)
```

注 : キュー・マネージャーの実行中に、ユーザー ID タイムアウトを長期間 (数日間または数週間) オフにした場合、そのときに取り消されるか削除されているユーザーについて、RVERIFY SECURITY コマンドを必ず実行してください。

z/OS ユーザー *ID* のタイムアウト

非アクティブの状態が一定時間続いたユーザーをキュー・マネージャーからサインオフするように IBM MQ を設定できます。

サブシステム・セキュリティーがアクティブになっている場合、ユーザーが IBM MQ リソースにアクセスすると、キュー・マネージャーは、自身に対してそのユーザーのサインオンを試みます。 つまり、そのユーザーは、ESM からの認証を受けます。 このユーザーは、キュー・マネージャーがシャットダウンされるか、ユーザー ID が タイムアウト (認証の失効) または再検証 (再認証) されるまで、 IBM MQ にサインオンしたままになります。

ユーザーがタイムアウトになると、そのユーザー ID は、キュー・マネージャーの内部でサインオフ され、そのユーザーに関して保存されていたセキュリティー関連情報はすべて破棄されます。 キュー・マネージャーの内部で発生するユーザーのサインオンとサインオフは、アプリケーション・プログラムやユーザーには認識されません。

ユーザーがタイムアウトになる可能性があるのは、あらかじめ定められている時間、IBM MQ リソースを使用しなかった場合です。 その時間は、MQSC の ALTER SECURITY コマンドで設定します。

ALTER SECURITY コマンドでは、2 つの値を指定できます。

TIMEOUT
一定時間使用されていないユーザー ID とその関連リソースが IBM MQ のキュー・マネージャーの内部にとどまることができる時間を分単位で指定します。

INTERVAL
ユーザー ID とその関連リソースの *TIMEOUT* の有効期限が切れているかどうかを確認するための検査の間隔を分単位の時間で指定します。

例えば、*TIMEOUT* 値が 30 で、*INTERVAL* 値が 10 であれば、IBM MQ は、10 分ごとにユーザー ID とその関連リソースをチェックして、30 分間使用されていないユーザー ID やリソースがあるかどうかを確認します。 タイムアウトになったユーザー ID がある場合、このユーザー ID はキュー・マネージャー内でサインオフされます。 タイムアウトになっていないユーザー ID に関連した、タイムアウトになったリソース情報が見つかれば、そのリソース情報は破棄されます。 ユーザー ID がタイムアウトにならないようにする場合は、*INTERVAL* 値をゼロに設定します。 ただし、*INTERVAL* 値をゼロにすると、**REFRESH SECURITY** コマンドまたは **RVERIFY SECURITY** コマンドを実行するまで、ユーザー ID とその関連リソースが格納されているストレージは解放されません。

1 回限りのユーザーがたくさんいる場合、この値を調整することは重要です。 小さな間隔値とタイムアウト値を設定すれば、不要になったリソースが解放されます。

注 : *INTERVAL* または *TIMEOUT* でデフォルト以外の値を使用する場合は、キュー・マネージャーを始動するたびにこのコマンドを再入力する必要があります。 この操作を自動化するには、そのキュー・マネージャーの CSQINP1 データ・セットに **ALTER SECURITY** コマンドを組み込みます。

z/OS *z/OS* でキュー・マネージャーのセキュリティーをリフレッシュする操作

IBM MQ for z/OS は、パフォーマンスを改善するために RACF データをキャッシュに入れます。 特定のセキュリティー・クラスを変更する場合は、そのキャッシュに入れた情報をリフレッシュする必要があります。ただしパフォーマンス上の理由から、セキュリティーをリフレッシュする頻度はできるだけ低くします。さらに、TLS セキュリティー情報だけをリフレッシュする、という選択肢もあります。

キューを初めて (またはセキュリティー・リフレッシュ以降初めて) 開くと、IBM MQ は、RACF 検査を実行し、ユーザーのアクセス権限を取得してその情報をキャッシュに入れます。 キャッシュに入れるデータとしては、セキュリティー検査が実行されたユーザー ID やリソースなどの情報があります。 そのキューを同じユーザーが再び開くときには、キャッシュにデータが入っているので、IBM MQ が RACF 検査を実行する必要はありません。したがって、パフォーマンスが向上します。 セキュリティー・リフレッシュの操作を実行すると、キャッシュに入っていたセキュリティー情報が破棄されるので、IBM MQ は、RACF に対する新しい検査を実行しなければならなくなります。 MQADMIN、MXADMIN、MQPROC、MXPROC、MQQUEUE、MXQUEUE、MQNLIST、MXNLIST、MXTOPIC のいずれかのクラスに含まれている RACF リソース・プロファイルの追加、変更、削除の操作を実行するときには、そのクラスを使用するキュー・マネージャーが自身で保管しているセキュリティー情報をリフレッシュすることが必要になります。 そのためには、以下のコマンドを実行します。

- RACF の SETROPTS RACLIST(classname) REFRESH コマンド。RACF レベルでリフレッシュを実行します。
- IBM MQ の REFRESH SECURITY コマンド。キュー・マネージャーが保持しているセキュリティー情報をリフレッシュします。 このコマンドは、変更されたプロファイルにアクセスするキュー・マネージャーごとに実行する必要があります。 キュー共有グループがある場合は、コマンド・スコープ属性を使用して、そのグループに含まれているすべてのキュー・マネージャーにそのコマンドを送信できます。

  注 : 新しいユーザーを既存のグループに接続した場合は、IBM MQ の RVERIFY SECURITY(userid) コマンドを実行する必要があります。 REFRESH SECURITY(*) コマンドを実行しても、そのユーザーが IBM MQ リソースに次回アクセスしようとした時点で、キュー・マネージャーがそのユーザーを再びサインオンすることはありません。

いずれかの IBM MQ クラスで汎用プロファイルを使用している場合も、いずれかの汎用プロファイルの変更、追加、削除の操作を実行したときに、RACF の通常のリフレッシュ・コマンドを実行する必要があります。 たとえば、SETROPTS GENERIC (classname) REFRESH のように発行します。

ただし、RACF リソース・プロファイルが追加、変更、または削除され、それが適用されるリソースがまだアクセスされていない場合 (つまり、情報がキャッシュされないため)、IBM MQ は REFRESH SECURITY コマンドが発行されずに新しい RACF 情報を使用します。

RACF 監査が (RACF の RALTER AUDIT(access-attempt (audit_access_level)) コマンドなどによって) オンになっていると、データがキャッシュに入ることはありません。つまり、IBM MQ は、検査のたびに RACF データベースを直接参照します。 したがって、変更はすぐに認識され、変更内容にアクセスするために REFRESH SECURITY を実行する必要はありません。 RACF 監査がオンになっているかどうかを確認するには、RACF の RLIST コマンドを使用します。 例えば、以下のようなコマンドを実行できます。

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

そうすれば、以下のような結果を受け取ることになります。

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

この例では、監査がオンに設定されています。 詳細については、「*z/OS Security Server RACF* 監査担当者のガイド」と「*z/OS Security Server RACF* コマンド言語解説書」を参照してください。

セキュリティー情報がキャッシュに入れられ、そのキャッシュに入れられた情報が使用される状況をまとめたのが、247 ページの図 17 です。

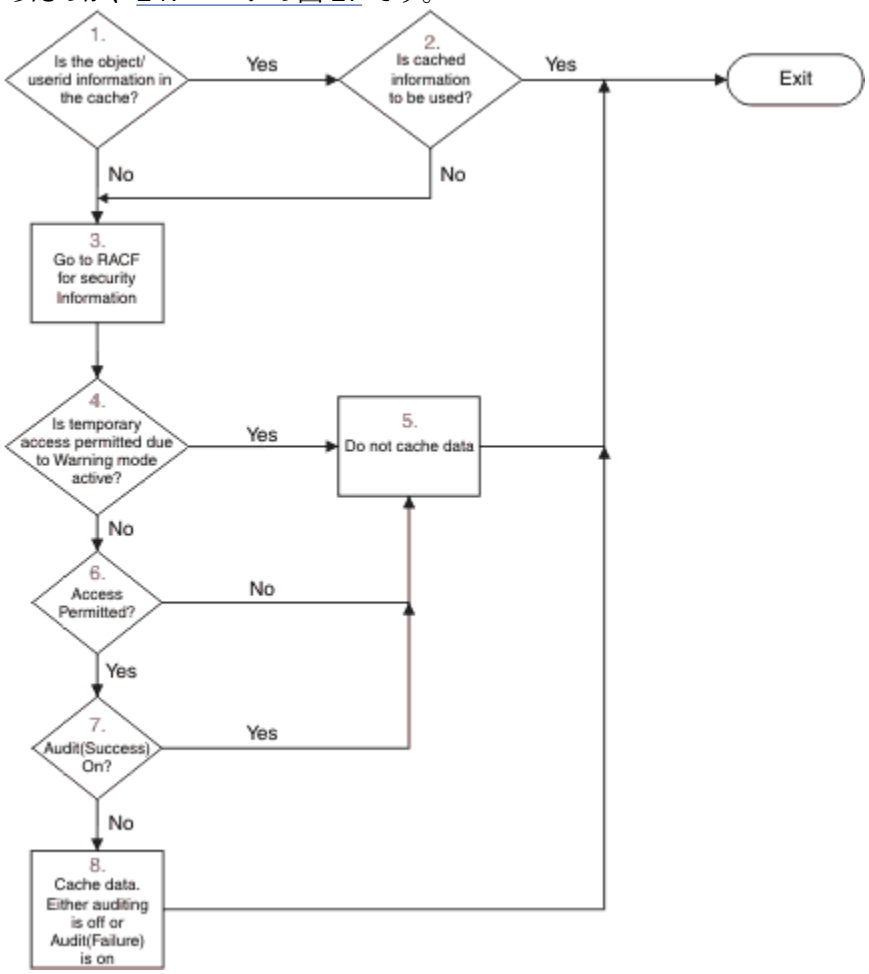


図 *17. IBM MQ* のセキュリティー・キャッシュのロジックの流れ

MQADMIN クラスまたは MXADMIN クラスでスイッチ・プロファイルを追加 したり削除したりしてセキュリティー設定を変更する場合は、以下のいずれか のコマンドを使用して、その変更を動的に取り込みます。

REFRESH SECURITY(*)
REFRESH SECURITY(MQADMIN)
REFRESH SECURITY(MXADMIN)

したがって、キュー・マネージャーを再始動しなくても、新しいセキュリティー・タイプをアクティブにしたり非アクティブにしたりすることが可能です。

パフォーマンス上の理由から、REFRESH SECURITY コマンドの影響を受けるのは、それらのクラスだけになっています。 MQCONN クラスまたは MQCMDS クラスでプロファイルを変更した場合に、REFRESH SECURITY を使用する必要はありません。

注 : RESLEVEL セキュリティー・プロファイルを変更した場合に、MQADMIN クラスまたは MXADMIN クラスをリフレッシュする必要はありません。

パフォーマンス上の理由から、REFRESH SECURITY を使用する頻度は、できるだけ低くします。理想は、オフピーク時に実行することです。 個々のユーザーをアクセス・リストに入れるのではなく、 IBM MQ プロファイルのアクセス・リストに既に存在する RACF グループにユーザーを接続することにより、セキュリティー・リフレッシュの回数を最小限にすることができます。 つまり、リソース・プロファイルではなくユーザーを変更する、という発想です。 さらに、セキュリティー・リフレッシュの代わりに、RVERIFY SECURITY を対象のユーザーに対して実行することもできます。

REFRESH SECURITY の例として、キュー・マネージャー PRMQ の INSURANCE.LIFE で始まるキューに対するアクセスを保護するために新しいプロファイルを定義するとします。 以下の RACF コマンドを使用します。

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

RACF が自身で保管しているセキュリティー情報をリフレッシュするように、例えば、以下のようなコマンドを実行する必要があります。

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

ここで取り上げているのは汎用プロファイルなので、RACF が MQQUEUE に対応する汎用プロファイルをリフレッシュするように、コマンドを実行することも必要です。 以下に例を示します。

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

その後、以下のコマンドを使用して、キュー・プロファイルが変更されたことをキュー・マネージャー PRMQ に通知します。

```
REFRESH SECURITY(MQQUEUE)
```

SSL/TLS セキュリティーのリフレッシュ

TLS 鍵リポジトリーのキャッシュ・ビューをリフレッシュする場合は、TYPE(SSL) オプションを付けて REFRESH SECURITY コマンドを実行します。 そうすれば、チャネル・イニシエーターを再始動しなくても、一部の TLS 設定を更新できます。

z/OS セキュリティー状況の表示

セキュリティー・スイッチや他のセキュリティー制御機能の状況を表示するには、MQSC の DISPLAY SECURITY コマンドを実行します。

DISPLAY SECURITY ALL コマンドの標準的な出力を以下の図に示します。

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMELIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

図 *18. DISPLAY SECURITY* コマンドの標準的な出力

例では、コマンドに応答したキュー・マネージャーで、サブシステム、コマンド、代替ユーザー、プロセス、 名前リスト、およびキュー・セキュリティーがキュー・マネージャー・レベルでは活動状態であるが、キュー共用グループ・レベルでは活動状態でないことを示しています。 接続セキュリティー、コマンド・リソース・セキュリティー、コンテキスト・セキュリティーは、アクティブになっていません。 さらに、この図から分かるように、ユーザー ID をタイムアウトにする機能が活動状態であり、 またキュー・マネージャーはその中で 54 分間使用されなかったユーザー ID を探して、 そのようなユーザー ID を削除する作業を 12 分ごとに行います。

注 : このコマンドを実行すると、現在のセキュリティー状況が表示されます。 RACF に対して定義されているスイッチ・プロファイルの現在の状況や、RACF クラスの状況は、必ずしも反映されません。 例えば、このキュー・マネージャーの最後の再始動や REFRESH SECURITY コマンドの最後の実行の後にスイッチ・プロファイルが変更されている可能性もあります。

z/OS z/OS のセキュリティー・インストール・タスク

IBM MQ をインストールしてカスタマイズしたら、開始済みタスク・プロシージャーを RACF で実行する権限を設定し、さまざまなリソースにアクセスする権限を設定し、RACF 定義をセットアップします。 さらに、システムで TLS を使用するための構成を行うこともできます。

IBM MQ を最初にインストールしてカスタマイズするときに、以下のセキュリティー関連タスクを実行する必要があります。

1. IBM MQ のデータ・セットとシステム・セキュリティーをセットアップします。そのために、以下の作業を行います。
   - キュー・マネージャーの開始済みタスク・プロシージャー xxxxMSTR と分散キューイングの開始済みタスク・プロシージャー xxxxCHIN を RACF で実行するための権限を設定します。
   - キュー・マネージャーのデータ・セットにアクセスする権限を設定します。
   - キュー・マネージャーとユーティリティー・プログラムを使用するユーザー ID に、各種のリソースにアクセスする権限を与えます。
   - カップリング・ファシリティーのリスト構造を使用するキュー・マネージャーにアクセス権を与えます。
   - Db2 を使用するキュー・マネージャーにアクセス権を与えます。
2. IBM MQ セキュリティーの RACF 定義をセットアップします。
3. Transport Layer Security (TLS) を使用する場合は、システムで証明書と鍵を使用するための準備を行います。

z/OS *IBM MQ for z/OS* のデータ・セット・セキュリティーのセットアップ

IBM MQ ユーザーにはさまざまなタイプがあります。 RACF を使用すれば、システム・データ・セットに対するユーザーのアクセスを制御できます。

IBM MQ データ・セットのユーザーとして考えられるのは、以下のエンティティーです。

- キュー・マネージャー自体
- チャネル・イニシエーター
- IBM MQ 管理者 (IBM MQ データ・セットの作成やユーティリティー・プログラムの実行、および同様のタスクを行う必要があります)
- アプリケーション・プログラマー (データ・セットやマクロ、および同様のタスクなど、IBM MQ に用意されているコピーブックを使用する必要があります)
- 以下のいずれか 1 つ以上がかかわるアプリケーション
  - バッチ・ジョブ
  - TSO ユーザー
  - CICS 領域
  - IMS 領域
- データ・セット CSQOUTX と CSQSNAP
- 動的キュー SYSTEM.CSQXCMD.*

IBM MQ データ・セットを使用する可能性があるすべてのユーザーについて、RACF を使用してそのデータ・セットを保護する必要があります。

さらに、すべての「CSQINP」データ・セットに対するアクセスも制御しなければなりません。

z/OS 開始済みタスク・プロシージャーに関する *RACF* 権限

いくつかの IBM MQ データ・セットはキュー・マネージャー専用です。 RACF を使用して IBM MQ データ・セットを保護する場合は、 RACF を使用して、キュー・マネージャー開始タスク・プロシージャー xxxxMSTR、および分散キューイング開始タスク・プロシージャー xxxxCHIN も許可する必要があります。そのためには、STARTED クラスを使用します。 あるいは、開始済みプロシージャー・テーブル (ICHRIN03) を使用することもできます。ただしその場合は、変更を有効にするために、z/OS システムの IPL を実行する必要があります。

詳細については、「*z/OS Security Server RACF* システム・プログラマーのガイド」を参照してください。

識別対象になる RACF のユーザー ID には、開始済みタスク・プロシージャーに含まれているデータ・セットに対するアクセス権が必要です。 例えば、キュー・マネージャーの CSQ1MSTR という開始済みタスク・プロシージャーを RACF のユーザー ID QMGRCSQ1 に関連付けると、ユーザー ID QMGRCSQ1 には、キュー・マネージャー CSQ1 がアクセスする z/OS リソースに対するアクセス権が必要になります。

さらに、キュー・マネージャーのユーザー ID の GROUP フィールドの内容は、そのキュー・マネージャーの STARTED プロファイルの GROUP フィールドの内容と同じでなければなりません。 それぞれの GROUP フィールドの内容が一致しない場合、該当するユーザー ID はシステムに入れません。 このような状況では IBM MQ が未定義ユーザー ID で実行されることになり、結果的にセキュリティー違反のために終了してしまいます。

キュー・マネージャーとチャネル・イニシエーターの開始済みタスク・プロシージャーに関連付けられている RACF ユーザー ID では、TRUSTED 属性を設定してはなりません。

z/OS データ・セットに対するアクセス権限の設定

権限のないユーザーがキュー・マネージャーのインスタンスを実行したり、キュー・マネージャーのデータにアクセスしたりすることを防止するために、IBM MQ のデータ・セットを保護する必要があります。そのためには、z/OS RACF の通常のデータ・セット保護機能を使用します。

251 ページの表 65 は、さまざまなデータ・セットに対してキュー・マネージャー開始タスク・プロシージャーが持つ必要がある RACF アクセスを要約したものです。

表 65. キュー・マネージャーに関連したデータ・セットに対する *RACF* のアクセス権	
RACF アクセス	**データ・セット**
READ	• thlqual.SCSQAUTH および thlqual.SCSQANLx (ここで、x は各国語の言語文字です)。 • キュー・マネージャーの開始タスク・プロシージャーの CSQINP1、CSQINP2、CSQXLIB で参照されるデータ・セット。 • グループ内の他のキュー・マネージャーが所有する SMDS データ・セット。 • グループ内の他のキュー・マネージャーのログ・データ・セット、BSDS データ・セット、およびアーカイブ・ログ・データ・セット。
UPDATE	• すべてのページ・セットとログと BSDS のデータ・セット。 • キュー・マネージャーが所有する SMDS データ・セット
ALTER	• すべてのアーカイブ・ログ・データ・セット。

251 ページの表 66 は、分散キューイング用の開始タスク・プロシージャーが異なるデータ・セットに対して持つ必要がある RACF アクセスを要約したものです。

表 66. 分散キューイングに関連したデータ・セットに対する *RACF* のアクセス権	
RACF アクセス	**データ・セット**
READ	• thlqual.SCSQAUTH、thlqual.SCSQANLx (x は各国語の言語文字)、thlqual.SCSQMVR1。 • LE ライブラリー・データ・セット。 • チャネル・イニシエーター開始タスク・プロシージャーで CSQXLIB および CSQINPX によって参照されるデータ・セット。
UPDATE	• データ・セット CSQOUTX と CSQSNAP

詳しくは、「*z/OS Security Server RACF セキュリティー管理者のガイド*」を参照してください。

V 9.1.4 z/OS データ・セットの暗号化

IBM MQ データ・セットは、データの保護や法規制への準拠といった目的により、z/OS データ・セット暗号化を使用して暗号化できます。

z/OS データ・セット暗号化を使用して、すべてのページ・セット、アクティブ・ログ・データ・セット、アーカイブ・ログ・データ・セット、およびブートストラップ・データ・セット (BSDS) を保護できます。

重要 : IBM MQ for z/OS 9.1.3 以前による z/OS データ・セット暗号化を使用して共有メッセージ・データ・セット (SMDS) を保護することはできません。

データ・セット暗号化による IBM MQ for z/OS での保存データの機密性 のセクションを参照してください。を参照してください。

z/OS *IBM MQ for z/OS のリソース・セキュリティーのセットアップ*

IBM MQ ユーザーにはさまざまなタイプがあります。 RACF を使用すれば、IBM MQ リソースに対するユーザーのアクセスを制御できます。

IBM MQ リソース (キューやチャネルなど) のユーザーとして考えられるのは、以下のエンティティーです。

- キュー・マネージャー自体
- チャネル・イニシエーター
- IBM MQ 管理者 (IBM MQ データ・セットの作成やユーティリティー・プログラムの実行、および同様のタスクを行う必要があります)

- アプリケーション・プログラマー (データ・セットやマクロ、および同様のタスクなど、IBM MQ に用意されているコピーブックを使用する必要があります)
- 以下のいずれか 1 つ以上がかかわるアプリケーション
  - バッチ・ジョブ
  - TSO ユーザー
  - CICS 領域
  - IMS 領域
- データ・セット CSQOUTX と CSQSNAP
- 動的キュー SYSTEM.CSQXCMD.*

IBM MQ リソースを使用する可能性があるすべてのユーザーについて、RACF を使用してそのリソースを保護する必要があります。 特に、チャネル・イニシエーターは、さまざまなリソースにアクセスする必要があるので (258 ページの『z/OS で使用するチャネル・イニシエーターのセキュリティーに関する考慮事項』を参照)、チャネル・イニシエーターを実行するためのユーザー ID には、それらのリソースに対するアクセス権限を与えなければなりません。

キュー共用グループを使用している場合は、キュー・マネージャー内部で色々なコマンドが発行されます。したがって、使用するユーザー ID に、そのようなコマンドを発行する許可を与える必要があります。 つまり、以下のようなコマンドです。

- QSGDISP(GROUP) が設定されているすべてのオブジェクトの DEFINE、ALTER、DELETE
- CHLDISP(SHARED) で使用するすべてのチャネルの START CHANNEL と STOP CHANNEL

z/OS *z/OS システムで TLS を使用するための構成*

このトピックでは、IBM MQ for z/OS で RACF コマンドを使用して Transport Layer Security (TLS) の構成を行う例を取り上げます。

チャネル・セキュリティーで TLS を使用する場合は、システムでいくつかのタスクを実行する必要があります。 (証明書や鍵リポジトリー (鍵リング) に対して RACF コマンドを使用するための詳細については、『z/OS での TLS の取り扱い』を参照してください。)

1. RACF の RACDCERT コマンドを使用して、システムのすべての鍵と証明書を格納するための鍵リングを RACF で作成します。 以下に例を示します。

   ```
   RACDCERT ID(CHINUSER) ADDRING(QM1RING)
   ```

   ID としては、チャネル・イニシエーター・アドレス・スペースのユーザー ID を使用するか、共有鍵リングの場合は、その鍵リングの所有者にするユーザー ID を使用します。
2. RACF の RACDCERT コマンドを使用して、各キュー・マネージャーのデジタル証明書を作成します。

   証明書のラベルは、IBM MQ **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの `ibmWebSphereMQ` にキュー・マネージャーかキュー共有グループの名前を付加した値のどちらかでなければなりません。 詳細は デジタル証明書ラベル を参照してください。 以下の例では `ibmWebSphereMQQM1` です。

   以下に例を示します。

   ```
   RACDCERT ID(USERID) GENCERT
   SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
   WITHLABEL('ibmWebSphereMQQM1')
   ```
3. RACF の RACDCERT コマンドを使用して、RACF で証明書を鍵リングに接続します。 以下に例を示します。

```
RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
CONNECT ID(CHINUSER)
```

さらに、該当する (認証局の) すべての署名者証明書を鍵リングに接続することも必要です。 つまり、このキュー・マネージャーの TLS 証明書のすべての認証局と、このキュー・マネージャーの通信先になるすべての TLS 証明書のすべての認証局が対象になります。 以下に例を示します。

```
RACDCERT ID(CHINUSER)
CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
```

4. それぞれのキュー・マネージャーで、IBM MQ の ALTER QMGR コマンドを使用して、キュー・マネージャーで参照する必要がある鍵リポジトリーを指定します。 例えば、チャネル・イニシエーター・アドレス・スペースが所有している鍵リングの場合は、以下のようにします。

```
ALTER QMGR SSLKEYR(QM1RING)
```

共有鍵リングを使用している場合は、以下のようにします。

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

userid は、共有鍵リングを所有するユーザー ID です。

5. 認証局では、証明書失効リスト (CRL) を使用して、信頼できなくなった証明書を取り消します。 CRL の格納先は、LDAP サーバーです。 LDAP サーバーにあるそのリストにアクセスするには、まず IBM MQ の DEFINE AUTHINFO コマンドを使用して、AUTHTYPE CRLLDAP の AUTHINFO オブジェクトを作成する必要があります。 以下に例を示します。

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

この例では、LDAP サーバーの公開領域に証明書失効リストが格納されているので、LDAPUSER フィールドと LDAPPWD フィールドは必要ありません。

次に、IBM MQ の DEFINE NAMELIST コマンドを使用して、その AUTHINFO オブジェクトを名前リストに組み込みます。 以下に例を示します。

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

最後に、IBM MQ の ALTER QMGR コマンドを使用して、その名前リストを各キュー・マネージャーに関連付けます。 以下に例を示します。

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. IBM MQ の ALTER QMGR コマンドを使用して、TLS 呼び出しを実行できるようにキュー・マネージャーをセットアップします。 つまり、SSL 呼び出しだけを処理するサーバー・サブタスクを定義し、通常のディスパッチャーが SSL 呼び出しの影響を受けずにそのまま通常の処理を続行できるようにします。 そのようなサブタスクを少なくとも 2 つ設定しなければなりません。 以下に例を示します。

```
ALTER QMGR SSLTASKS(8)
```

   この変更を有効にするには、チャネル・イニシエーターを再始動する必要があります。

7. IBM MQ の DEFINE CHANNEL コマンドまたは ALTER CHANNEL コマンドを使用して、各チャネルで使用する暗号仕様を指定します。 以下に例を示します。

```
ALTER CHANNEL(LDAPCHL)
CHLTYPE(SDR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

   チャネルの両端で、同じ暗号仕様を指定する必要があります。

z/OS QSG でのチャネル認証レコードの管理

チャネル認証レコードの適用先は、そのチャネル認証レコードが作成されたキュー・マネージャーです。 キュー共有グループ (QSG) 全体でチャネル認証レコードを共有するわけではありません。 そのため、キュー共有グループに属するすべてのキュー・マネージャーに同じルールを適用する必要がある場合は、すべてのルールの一貫性を保つために、いくつかの管理操作を実行する必要があります。

1. すべての SET CHLAUTH コマンドに必ず CMDSCOPE(*) オプションを追加します。 これにより、コマンドが、キュー共有グループ内の実行中のすべてのキュー・マネージャーに送られます。
2. CMDSCOPE(*) オプションを指定して DISPLAY CHLAUTH コマンドを実行してから、すべてのキュー・マネージャーのレコードが同じであるかどうかを確認するため、応答を分析します。 不整合が見つかった場合は、CMDSCOPE(*) または CMDSCOPE(*qmgr-name*) を使用して同じルールを含む SET CHLAUTH コマンドを実行します。
3. ルールの完全セットが含まれているキュー・マネージャーの CSQINP2 連結に、メンバーを追加します (詳細については、初期化コマンドを参照してください)。 これらは、キュー・マネージャーの初期化プロセスの一部として読み取られます。 SET CHLAUTH コマンドで ACTION(ADD) が使用されると、ルールが存在しない場合にのみ、ルールが追加されます。 ACTION(REPLACE) を使用すると、ルールが既に存在している場合はその既存のルールが置換され、存在しない場合は追加されます。 これにより、キュー共有グループ内のすべてのキュー・マネージャーの CSQINP2 連結に、同じメンバーを配置できます。
4. CSQUTIL ユーティリティー (詳細については IBM MQ (COMMAND) へのコマンドの実行を参照) を使用して、 MAKEDEF または MAKEREP オプションを 指定して 1 つのキュー・マネージャーからルールを抽出します。 その後、CSQUTIL を使用して出力をターゲット・キュー・マネージャーに向けて再生します。

関連概念

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。

z/OS z/OS での監査に関する考慮事項

キュー・マネージャーのセキュリティー監査を実施するために、RACF の通常の監査制御機能を利用できます。 IBM MQ では、独自のセキュリティー統計が収集されません。 唯一の統計は、監査で作成できる統計です。

RACF 監査は、以下の対象に基づいて実施できます。

- ユーザー ID

- リソース・クラス
- プロファイル

詳細については、「*z/OS Security Server RACF* 監査担当者のガイド」を参照してください。

注 : 監査を実施すると、パフォーマンスが低下します。実装する監査が多ければ、それだけパフォーマンスが低下する度合いも大きくなります。 この考慮事項は、RACF の WARNING オプションの使用にも当てはまります。

z/OS *RESLEVEL* の監査

RESAUDIT システム・パラメーターを使用して、RESLEVEL の監査レコードの生成を制御します。 生成されるのは、RACF の GENERAL 監査レコードです。

RESLEVEL の監査レコードを生成する場合は、RESAUDIT システム・パラメーターを YES に設定します。 RESAUDIT パラメーターを NO に設定すると、監査レコードは生成されません。 このパラメーターを設定するための詳細については、『CSQ6SYSP の使用』を参照してください。

RESAUDIT を YES に設定しても、アドレス・スペースのユーザー ID が hlq.RESLEVEL プロファイルに対して持っているアクセス権を確認するための RESLEVEL 検査が実行されるときに、RACF の通常の監査レコードは生成されません。 むしろ、IBM MQ によって要求されるのは、RACF が GENERAL 監査レコード (イベント番号 27) を作成することです。 これらの検査の実行は接続時に限られているので、パフォーマンス・コストは最小で済みます。

RACF 報告書作成プログラム (RACFRW) を使用して、 IBM MQ 一般監査レコードを報告することができます。 RESLEVEL のアクセスに関するレポートを生成するには、以下のような RACFRW コマンドを使用します。

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

Date、 *Time*、および *SYSID* フィールドを除く RACFRW からの報告書の例は、 255 ページの図 19 に示されています。

```
       RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE   4
                                E
                                V  Q
                                E  U
*JOB/USER *STEP/  --TERMINAL--  N  A
   NAME    GROUP     ID    LVL  T  L

 WS21B    MQMGRP IGJZM000   0   27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
    TRUSTED USER                      AUTH=(NONE),REASON=(NONE)
                                 SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                 LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST
PROFILE(QM66.RESLEVEL),
                                 CLASS(MQADMIN), ACCESS EQUATES TO
(CONTROL)',RESULT=SUCCESS,MQADMIN
```

図 *19. RESLEVEL* の一般監査レコードに関する *RACFRW* の出力例

このサンプル出力の LOGSTR データを見ると、TSO ユーザー WS21B には QM66.RESLEVEL に対する CONTROL アクセス権があることがわかります。 つまり、ユーザー WS21B が QM66 のリソースにアクセスするときには、すべてのリソース・セキュリティー検査が迂回されます。

RACFRW を使用するための詳細については、「*z/OS Security Server RACF* 監査担当者のガイド」を参照してください。

z/OS セキュリティーのカスタマイズ

IBM MQ セキュリティーの動作を変更する場合は、SAF 出口 (ICHRFR00) または外部セキュリティー・マネージャーの出口を使用する必要があります。

RACF 出口の詳細については、「*z/OS Security Server RACROUTE* マクロ解説書」の資料を参照してください。

注 : IBM MQ では、ESM に対する呼び出しが最適化されているので、例えば、特定のユーザーが特定のキューを開くたびに RACROUTE 要求が実行されるとは限りません。

z/OS z/OS のセキュリティー違反メッセージ

セキュリティー違反が発生すると、アプリケーション・プログラムの戻りコード MQRC_NOT_AUTHORIZED またはジョブ・ログのメッセージでその違反が通知されます。

戻りコード MQRC_NOT_AUTHORIZED がアプリケーション・プログラムに返される理由をまとめると、以下のようになります。

- ユーザーがキュー・マネージャーに接続する権限を持っていません。 その場合は、バッチ/TSO、CICS、IMS のジョブ・ログに ICH408I メッセージが書き込まれます。
- キュー・マネージャーに対するユーザー・サインオンが失敗しました。 これは、たとえばジョブ・ユーザー ID が無効または不適切であるか、タスク・ユーザー ID または代替ユーザー ID が無効であるためです。 それらのユーザー ID のうち、取り消されたり削除されたりしたために無効になっているユーザー ID が 1 つ以上存在する可能性があります。 この場合、キュー・マネージャーのジョブ・ログに、ICHxxxx メッセージと場合によっては IRRxxxx メッセージが発行され、サインオンが失敗した理由を示します。 以下に例を示します。

```
ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
  LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
```

- 代替ユーザーが要求されたのに、ジョブまたはタスクのユーザー ID に、代替ユーザー ID に対するアクセス権がありませんでした。 この障害の場合は、該当するキュー・マネージャーのジョブ・ログに違反メッセージが書き込まれます。
- コンテキスト・オプションが使用されていたか、出力のために伝送キューを開くときに暗黙指定されていたのに、ジョブのユーザー ID または (該当する場合は) タスクのユーザー ID または代替ユーザー ID に、コンテキスト・オプションに対するアクセス権がありませんでした。 その場合は、該当するキュー・マネージャーのジョブ・ログに違反メッセージが書き込まれます。
- セキュリティーが設定されているキュー・マネージャー・オブジェクト (キューなど) に、権限のないユーザーがアクセスしようとしました。 その場合は、該当するキュー・マネージャーのジョブ・ログにその違反に関する ICH408I メッセージが書き込まれます。 ジョブのユーザー ID または (該当する場合は) タスクのユーザー ID または代替ユーザー ID がこの違反の原因になっている可能性があります。

コマンド・セキュリティーとコマンド・リソース・セキュリティーの違反メッセージも、キュー・マネージャーのジョブ・ログに書き込まれる場合があります。

ICH408I 違反メッセージにユーザー ID ではなくキュー・マネージャーのジョブ名が記述されている場合は、通常、ブランクの代替ユーザー ID が指定されています。 以下に例を示します。

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

MQADMIN profile hlq.ALTERNATE.USER.-BLANK- のアクセス・リストを調べれば、ブランクの代替ユーザー ID を使用する権限を持っているユーザーを確認できます。

ICH408I 違反メッセージは、以下のような場合にも生成される可能性があ ります。

- コンテキストなしでシステム・コマンド入力キューにコマンドが送信された場合。 システム・コマンド入力キューに書き込むユーザー作成プログラムでは、常にコンテキスト・オプションを使用する必要があります。 詳細については 214 ページの『コンテキスト・セキュリティーのためのプロファイル』 を参照してください。
- IBM MQ リソースにアクセスするジョブがそのリソースに関連するユーザー ID を持っていない場合や、IBM MQ アダプターがアダプター環境からユーザー ID を取り出せない場合。

キュー共有グループ・レベルとキュー・マネージャー・レベルの両方のセキュリティーを使用している場合にも、違反メッセージが生成されることがあります。 例えば、キュー・マネージャー・レベルではプロファイルが見つからなかったものの、キュー共有グループ・レベルのプロファイルによってアクセスが認められた、という趣旨のメッセージが書き込まれる場合があります。

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

z/OS アクセスの許可/禁止が正しく行われていない場合の処置

リソースに対するアクセスが正しく制御されていないように思える場合は、「*z/OS Security Server RACF* セキュリティー管理者のガイド」にある手順のほかに、以下のチェックリストを使用できます。

- スイッチ・プロファイルは正しく設定されていますか。
  - RACF はアクティブになっていますか。
  - IBM MQ RACF クラスはインストールされていて、アクティブになっていますか。

    RACF コマンド SETROPTS LIST を使用して確認してください。
  - IBM MQ の DISPLAY SECURITY コマンドを使用して、キュー・マネージャーの現在のスイッチの状況を表示してください。
  - MQADMIN クラスにあるスイッチ・プロファイルを確認してください。

    そのために RACF コマンド SEARCH と RLIST を使用してください。
  - IBM MQ REFRESH SECURITY (MQADMIN) コマンドを発行して、RACF スイッチ・プロファイルを再検査する。
- RACF のリソース・プロファイルが変更されていますか。 例えば、プロファイルの汎用アクセス権やプロファイルのアクセス・リストが変更されていますか。
  - プロファイルは汎用ですか。

    その場合は、RACF コマンド SETROPTS GENERIC(classname) REFRESH を実行してください。
  - このキュー・マネージャーでセキュリティーをリフレッシュしたことがありますか。

    必要であれば、RACF コマンド SETROPTS RACLIST(classname) REFRESH を実行してください。

    必要であれば、IBM MQ の REFRESH SECURITY(*) コマンドを実行してください。
- ユーザーの RACF 定義が変更されていますか。 例えば、ユーザーが新しいグループに接続されたり、ユーザーのアクセス権限が取り消されたりしていますか。
  - IBM MQ の RVERIFY SECURITY(userid) コマンドを実行して、ユーザーを再検証したことがありますか。
- RESLEVEL のためにセキュリティー検査が迂回されていますか。
  - RESLEVEL プロファイルに対する接続側のユーザー ID のアクセス権を確認してください。 RACF の監査レコードを調べて、RESLEVEL の設定を確認してください。
  - チャネルの場合は、チャネル・イニシエーターのユーザー ID が RESLEVEL に対して持っているアクセス権のレベルがすべてのチャネルで継承されるので、すべての検査が迂回されるようなアクセス権のレベル (ALTER など) が設定されていると、すべてのチャネルでセキュリティー検査が迂回されます。

- CICS から実行している場合は、トランザクションの RESSEC 設定を確認してください。
- ユーザーの接続中に RESLEVEL が変更された場合は、新しい RESLEVEL 設定を有効にするために、ユーザーがいったん切断して再接続する必要があります。

- キュー共有グループを使用していますか。
  - キュー共有グループ・レベルとキュー・マネージャー・レベルの両方のセキュリティーを使用している場合は、正しいプロファイルがすべて定義されていることを確認してください。 キュー・マネージャーのプロファイルが定義されていない場合は、プロファイルが見つからないという趣旨のメッセージがログに送信されます。
  - スイッチ設定の無効な組み合わせを使用したことがありますか (その場合は、全セキュリティー検査がオンに設定されます)。
  - キュー共有グループの一部の設定をキュー・マネージャーでオーバーライドするためのセキュリティー・スイッチを定義する必要がありますか。
  - キュー・マネージャー・レベルのプロファイルがキュー共有グループ・レベルのプロファイルよりも優先されていますか。

z/OS z/OS で使用するチャネル・イニシエーターのセキュリティーに関する考慮事項

分散キューイング環境でリソース・セキュリティーを使用する場合は、チャネル・イニシエーターのアドレス・スペースから各種の IBM MQ リソースにアクセスするための適切な権限が必要になります。 Integrated Cryptographic Support Facility (ICSF) を使用して、パスワード保護アルゴリズムをシードできます。

リソース・セキュリティーの使用

分散キューイング環境でリソース・セキュリティーを使用する場合の注意点を以下にまとめます。

システム・キュー
チャネル・イニシエーターのアドレス・スペースには、 203 ページの『システム・キュー・セキュリティー』にリストされているシステム・キュー、およびすべてのユーザー宛先キューと送達不能キューに対する RACF UPDATE アクセス権限が必要です (ただし、 202 ページの『送達不能キュー・セキュリティー』 を参照してください)。

伝送キュー
チャネル・イニシエーターのアドレス・スペースには、すべてのユーザー伝送キューに対する ALTER アクセス権が必要です。

コンテキスト・セキュリティー
チャネルのユーザー ID では (MCA のユーザー ID が指定されている場合はそのユーザー ID でも)、MQADMIN クラスに含まれている hlq.CONTEXT.queuename プロファイルに対する RACF の CONTROL アクセス権が必要です。 RESLEVEL プロファイルによっては、チャネル・ユーザー ID でも、それらのプロファイルに対する CONTROL アクセス権が必要になる場合があります。

すべてのチャネルには、MQADMIN hlq.CONTEXT に対する CONTROL アクセス権限が必要です。 送達不能キュー・プロファイル。 起動側であれ応答側であれ、すべてのチャネルはレポートを生成する可能性があるので、hlq.CONTEXT.reply-q プロファイルに対する CONTROL アクセス権も必要になります。

SENDER、CLUSSDR、SERVER の各チャネルでは、チャネルをウェイクアップして正常に終了させるためのメッセージが伝送キューに書き込まれる場合があるので、hlq.CONTEXT.xmit-queue-name プロファイルに対する CONTROL アクセス権が必要になります。

注 : チャネルのユーザー ID、またはチャネルのユーザー ID の接続先の RACF グループに、hlq.RESLEVEL に対する CONTROL アクセス権または ALTER アクセス権があると、チャネル・イニシエーターまたはそのいずれかのチャネルでリソース検査が行われることはありません。

詳細については、214 ページの『コンテキスト・セキュリティーのためのプロファイル』 232 ページの『RESLEVEL とチャネル・イニシエーター接続』および 234 ページの『セキュリティー検査のためのユーザー ID (z/OS)』を参照してください。

CSQINPX
CSQINPX 入力データ・セットを使用している場合は、チャネル・イニシエーターに、CSQINPX に対する READ アクセス権と、データ・セット CSQOUTX と動的キュー SYSTEM.CSQXCMD.* に対する UPDATE アクセス権も必要になります。

接続のセキュリティー
チャネル・イニシエーターのアドレス・スペースの接続要求では、接続タイプ CHIN が使用されるので、その接続タイプに関する適切なアクセス・セキュリティーを設定する必要があります。196 ページの『チャネル・イニシエーターのための接続セキュリティー・プロファイル』を参照してください。

データ・セット
チャネル・イニシエーターのアドレス・スペースでは、キュー・マネージャーのデータ・セットに対する適切なアクセス権が必要です。250 ページの『データ・セットに対するアクセス権限の設定』を参照してください。

コマンド
分散キューイングのコマンド (DEFINE CHANNEL、START CHINIT、START LISTENER や他のチャネル・コマンドなど) に関する適切なコマンド・セキュリティーを設定する必要があります。217 ページの表 49 を参照してください。

キュー共有グループを使用している場合は、チャネル・イニシエーターが内部でさまざまなコマンドを実行する可能性があるので、チャネル・イニシエーターで使用するユーザー ID には、その種のコマンドを実行するための権限が必要です。 つまり、CHLDISP(SHARED) で使用するすべてのチャネルの START CHANNEL と STOP CHANNEL です。

キュー・マネージャーの PSMODE が使用不可になっていない場合、チャネル・イニシエーターは DISPLAY PUBSUB コマンドに対する読み取り権限を持っている必要があります。

チャネル・セキュリティー
チャネル (特に受信側とサーバー接続) では、適切なセキュリティーをセットアップする必要があります。詳細については、234 ページの『セキュリティー検査のためのユーザー ID (z/OS)』を参照してください。

Transport Layer Security (TLS) プロトコルを使用して、チャネルのセキュリティーを提供することもできます。 IBM MQ での TLS の使用について詳しくは、22 ページの『IBM MQ での TLS セキュリティー・プロトコル』を参照してください。

さらに、サーバー接続セキュリティーの詳細については、95 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。

ユーザー ID
238 ページの『チャネル・イニシエーターで使用されるユーザー ID』と 242 ページの『グループ内キューイング・エージェントで使用されるユーザー ID』で取り上げられているユーザー ID では、以下のアクセス権が必要です。

- 該当する宛先キューと送達不能キューに対する RACF の UPDATE アクセス権
- 受信側でコンテキスト検査が実行される場合は、`hlq.CONTEXT.queuename` プロファイルに対する RACF CONTROL アクセス
- 使用しなければならなくなる可能性がある hlq.ALTERNATE.USER.userid プロファイルに対する適切なアクセス権
- クライアントの場合は、使用するリソースに対する適切な RACF アクセス権

APPC セキュリティー
LU 6.2 伝送プロトコルを使用している場合は、適切な APPC セキュリティーを設定します。 (例えば、APPCLU RACF クラスを使用します。) APPC に対するセキュリティーの設定については、次の資料を参照してください。

- *z/OS V1R2.0 MVS* 計画*: APPC* 管理
- 「*Multiplatform APPC Configuration Guide*」(IBM Redbooks 資料)

アウトバウンド伝送は、"セキュリティー (同一)" APPC オプションを使用します。 その結果、チャネル・イニシエーター・アドレス・スペースのユーザー ID とそのデフォルト・プロファイル (RACF GROUP)

は、ユーザー ID が既に検証されたという標識 (ALREADYV) と一緒にネットワークを流れて受信側に送られることになります。

受信側も z/OS であれば、そのユーザー ID とプロファイルは APPC によって検証され、ユーザー ID は受信側のチャネルに渡されてチャネルのユーザー ID として使用されます。

キュー・マネージャーが APPC を使用して同一または別の z/OS システム上の 別のキュー・マネージャーと通信を行う環境では、次のいずれかであることを確認する必要があります。

- 通信元の LU の VTAM 定義で SETACPT(ALREADYV) が指定されていること
- LU 間の接続に関する RACF の APPCLU プロファイルで CONVSEC(ALREADYV) が指定されていること

セキュリティー設定の変更
チャネルのユーザー ID または MCA のユーザー ID が宛先キューに対して持っている RACF のアクセス権のレベルを変更した場合に、その変更が有効になるのは、宛先キューの新しいオブジェクト・ハンドル (つまり、新しい MQOPEN) の場合に限られます。 MCA がキューを開いたり閉じたりするタイミングは、一定ではありません。チャネルの実行中にそのようなアクセス権が変更されても、MCA は、ユーザー ID の更新後のセキュリティー・アクセス権ではなく既存のセキュリティー・アクセス権を使用して宛先キューにメッセージを書き込む処理を続行できます。 そのような状況を回避するには、更新後のアクセス権のレベルを有効にするために、チャネルをいったん停止して再始動します。

自動リスタート
z/OS の自動リスタート・マネージャー (ARM) を使用してチャネル・イニシエーターを再始動する場合は、XCFAS アドレス・スペースに関連するユーザー ID に、IBM MQ の START CHINIT コマンドを実行する権限を与える必要があります。

Integrated Cryptographic Service Facility (ICSF) の使用

チャネル・イニシエーターでは、TLS が使用されていない場合に、パスワード保護アルゴリズムをシードしてクライアント・チャネルを介して流れるパスワードを難読化する際に、ICSF を使用して乱数を生成できます。 乱数の生成プロセスは、エントロピー と呼ばれます。

z/OS フィーチャーがインストール されており、ICSF が開始されていない場合は、 メッセージ CSQX213E が表示され、 チャネル・イニシエーターではエントロピーに STCK が使用されます。

メッセージ CSQX213E では、パスワード保護アルゴリズムが本来ほど 安全ではないことが警告されます。ただし、プロセスは続行することができます。ランタイムに対するその他の影響はありません。

z/OS フィーチャーが インストールされていない場合、チャネル・イニシエーターでは、自動的に STCK が使用されます。

注 :

1. エントロピーに ICSF を使用すると STCK を使用した場合よりも多くのランダム・シーケンスが生成されます。
2. ICSF を開始する場合は、チャネル・イニシエーターを再始動する必要があります。
3. 特定の CipherSpec には ICSF が必要です。 いずれか 1 つの CipherSpec の使用を試行し、ICSF がインストールされていない場合は、 メッセージ CSQX629E が表示されます。

z/OS のキュー・マネージャー・クラスターのセキュリティー

クラスターのセキュリティーに関する考慮事項は、クラスターに含まれていないキュー・マネージャーやチャネルの場合と同じです。 ただし、チャネル・イニシエーターにアクセス権を与えなければならないシステム・キューと、適切なセキュリティーを設定しなければならないコマンドが少し増えます。

MCA のユーザー ID、チャネル認証レコード、TLS か TLS、およびセキュリティー出口を使用すれば、クラスター・チャネルを (従来型のチャネルと同じ要領で) 認証できます。 クラスター受信側チャネルに関係するチャネル認証レコードまたはセキュリティー出口では、サーバー・キュー・マネージャーのクラスター・キューに対するアクセス権がリモート・キュー・マネージャーにあるかどうかを確認する必要があります。既存のキュー・アクセス・セキュリティーを変更しなくても、IBM MQ のクラスター・サポートを使用し始めることができます。 しかし、他のキュー・マネージャーもクラスターに加わる場合は、

SYSTEM.CLUSTER.COMMAND.QUEUE に書き込むことをそれらのキュー・マネージャーに許可する必要があります。

IBM MQ のクラスター・サポートには、クラスターのメンバーをクライアントの役割だけに制限するためのメカニズムが用意されていません。 したがって、キュー・マネージャーをクラスターに加える場合は、そのキュー・マネージャーを信頼できるかどうかを確認しなければなりません。 クラスターに含まれているキュー・マネージャーは、特定の名前でキューを作成すると、そのキューにメッセージを書き込むアプリケーションの意図とはかかわりなく、そのキューからメッセージを受信できるようになります。

クラスターのメンバーシップを制限する場合は、キュー・マネージャーが受信側チャネルに接続するのを防止する場合と同じ処置を実行します。 クラスターのメンバーシップを制限するには、チャネル認証レコードを使用するか、受信側チャネル上にセキュリティー出口プログラムを作成します。 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することもできます。

注 : アプリケーションが直接 SYSTEM.CLUSTER.TRANSMIT.QUEUE をオープンできるようにすることはお勧めできません。 また、アプリケーションが他の伝送キューを直接オープンできるようにすることもお勧めできません。

リソース・セキュリティーを使用する場合は、258 ページの『z/OS で使用するチャネル・イニシエーターのセキュリティーに関する考慮事項』で取り上げられている考慮事項のほかに以下の考慮事項もあります。

システム・キュー
チャネル・イニシエーターでは、以下のシステム・キューに対する RACF の ALTER アクセス権が必要です。

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE

SYSTEM.CLUSTER.REPOSITORY.QUEUE に対する UPDATE アクセス権も必要です。

クラスタリング用の名前リストに対する READ アクセス権も必要です。

コマンド
クラスター・サポート・コマンド (REFRESH CLUSTER、RESET CLUSTER、SUSPEND QMGR、RESUME QMGR) に関する適切なコマンド・セキュリティーを設定します (217 ページの表 49 を参照)。

z/OS IBM MQ と CICS を併用する場合のセキュリティーに関する考慮事項

IBM MQ 9.0.0 以降でサポートされるすべての CICS バージョンは、CICS 提供バージョンのアダプターおよびブリッジを使用します。

セキュリティーに関する考慮事項の詳細については、以下を参照してください。

- CICS-IBM MQ アダプターのセキュリティー。
- CICS-IBM MQ ブリッジのセキュリティー。

z/OS IBM MQ と IMS を併用する場合のセキュリティーに関する考慮事項

IBM MQ と IMS を併用する場合は、このトピックを参考にしてセキュリティー要件を計画してください。

OPERCMDS クラスの使用

RACF を使用して、OPERCMDS クラスに含まれているリソースを保護する場合は、IBM MQ のキュー・マネージャーのアドレス・スペースに関連するユーザー ID に、接続先の IMS システムに対して MODIFY コマンドを実行する権限を与える必要があります。

IMS ブリッジに関するセキュリティーの考慮事項

IMS ブリッジのセキュリティー要件を決定するときに検討しなければならない 4 つの要素があります。

- IBM MQ から IMS に接続するために必要なセキュリティー権限
- ブリッジを使用して IMS にアクセスするアプリケーションで実行するセキュリティー検査の程度

- それらのアプリケーションに使用権限を与える IMS リソース
- ブリッジが書き込んだり取得したりするメッセージで使用する権限

IMS ブリッジのセキュリティー要件を定義するときに検討しなければならない注意点は、以下のとおりです。

- ブリッジ経由で受け渡しが行われるメッセージは、強力なセキュリティー機能が用意されていないプラットフォームのアプリケーションから発信される可能性があります。
- ブリッジ経由で受け渡しが行われるメッセージは、同じ企業や組織によって制御されているわけではないアプリケーションから発信される可能性があります。

z/OS *IMS* に接続するためのセキュリティーに関する考慮事項

IBM MQ のキュー・マネージャー・アドレス・スペースのユーザー ID に、OTMA グループに対するアクセス権を与えます。

IMS ブリッジは、OTMA クライアントです。 IMS に対する接続は、IBM MQ のキュー・マネージャーのアドレス・スペースのユーザー ID で実行されます。 通常は、開始済みタスク・グループのメンバーとして定義されているユーザー ID です。 そのユーザー ID には、OTMA グループに対するアクセス権を与える必要があります (ただし、/SECURE OTMA 設定が NONE の場合は別です)。

そのためには、FACILITY クラスで以下のプロファイルを定義します。

```
IMSXCF.xcfgname.mqxcfmname
```

xcfgname は XCF グループ名、mqxcfmname は IBM MQ の XCF メンバー名です。

IBM MQ のキュー・マネージャーのユーザー ID に、このプロファイルに対する読み取りアクセス権を与える必要があります。

注 :

1. FACILITY クラスで権限を変更した場合は、RACF コマンド SETROPTS RACLIST(FACILITY) REFRESH を実行してその変更をアクティブ化できます。
2. MQADMIN クラスに hlq.NO.SUBSYS.SECURITY プロファイルが存在すると、ユーザー ID が IMS に渡されず、接続が失敗します (ただし、/SECURE OTMA 設定が NONE の場合は別です)。

z/OS *IMS* ブリッジに関するアプリケーション・アクセス制御

IMS システムごとに、FACILITY クラスで RACF プロファイルを定義します。 IBM MQ のキュー・マネージャーのユーザー ID に適切なレベルのアクセス権を与えます。

IMS ブリッジが接続される IMS> システムごとに、次の RACF> プロファイルを FACILITY クラスに定義して、IMS システムへ渡される各メッセージについてどの程度のセキュリティー検査を行うかを決めることができます。

```
IMSXCF.xcfgname.imsxcfmname
```

xcfgname は XCF グループ名、imsxcfmname は IMS の XCF メンバー名です。 (IMS システムごとに別々のプロファイルを定義する必要があります。)

このプロファイルで IBM MQ キュー・マネージャー・ユーザー ID に対して許可するアクセス・レベルは、IMS ブリッジが IMS に接続したときに IBM MQ に返され、後続のトランザクションで必要なセキュリティー・レベルを示します。 後続のトランザクションでは、IBM MQ が RACF から該当するサービスを要求し、ユーザー ID に正しい権限があれば、メッセージを IMS に渡します。

OTMA では、IMS の /SIGN コマンドがサポートされていませんが、IBM MQ では、各メッセージのアクセス権の検査を設定して、必要なレベルの制御を実装できるようになっています。

次のアクセス・レベル情報を戻すことができます。

NONE または NO PROFILE FOUND
最大レベルのセキュリティーが必要であることを示す値です。この場合は、すべてのトランザクションで認証が必要になります。 MQMD 構造の *UserIdentifier* フィールドで指定されているユーザー ID と MQIIH 構造の *Authenticator* フィールドで指定されているパスワードまたはパスチケットが RACF に登録されているかどうか、その組み合わせが有効かどうかを確認するための検査が行われます。 パスワードまたはパスチケットの UTOKEN が作成され、IMS に渡されます。その UTOKEN はキャッシュに入りません。

注 : MQADMIN クラスに hlq.NO.SUBSYS.SECURITY プロファイルが存在しても、そのプロファイルの定義よりこのレベルのセキュリティーが優先されます。

READ
以下の状況では NONE の場合と同じ認証が実行されることを示す値です。

- 特定のユーザー ID が初めて検出された場合
- 以前に検出されたことがあるユーザー ID でも、パスワードまたはパスチケットの UTOKEN が作成されていないか、キャッシュに入っていない場合

IBM MQ は、必要な場合に UTOKEN を要求して、IMS に渡します。

注 : セキュリティーの再検証の要求が実行された場合は、キャッシュに入っていたすべての情報が失われるので、その後初めて各ユーザー ID が検出された時点で UTOKEN が要求されます。

UPDATE
MQMD 構造の *UserIdentifier* フィールドで指定されているユーザー ID が RACF に登録されているかどうかを確認するための検査が行われます。

UTOKEN が作成され、IMS に渡されます。その UTOKEN はキャッシュに入ります。

CONTROL/ALTER
この IMS システムでは、どのユーザー ID についてもセキュリティー UTOKEN を渡す必要がないことを示す値です。 (通常、このオプションを使用するのは、開発システムとテスト・システムの場合に限られます。)

重要 : MQMD 構造の *UserIdentifier* フィールドで指定されているユーザー ID は、引き続き **CONTROL/ALTER** に渡されます。

注 :

1. このアクセス権は、IBM MQ が IMS に接続した時点で定義され、その接続の期間中存続します。 セキュリティー・レベルを変更する場合は、セキュリティー・プロファイルに対するアクセス権を変更してから、ブリッジをいったん停止して再始動する必要があります (そのためには、OTMA を停止して再始動する、などの方法があります)。
2. FACILITY クラスで権限を変更した場合は、RACF コマンド SETROPTS RACLIST(FACILITY) REFRESH を実行してその変更をアクティブ化できます。
3. パスワードまたはパスチケットを使用することもできますが、IMS ブリッジでは、データが暗号化されないことを覚えておく必要があります。 パスチケットの使用については、265 ページの『IMS ヘッダーでの RACF PassTickets の使用』を参照してください。
4. /SECURE OTMA コマンドを使用して IMS のセキュリティー設定を定義すれば、このような結果の一部を変更することも可能です。
5. キャッシュに入っている UTOKEN の情報を格納しておく期間は、IBM MQ の ALTER SECURITY コマンドの INTERVAL パラメーターと TIMEOUT パラメーターで定義できます。
6. RACF WARNING オプションは、IMSXCF.xcfgname.imsxcfmname プロファイルに対して何の効果もありません。 そのオプションを使用しても、付与されるアクセス権のレベルは変化せず、RACF WARNING メッセージは生成されません。

z/OS *IMS のセキュリティー検査*

ブリッジ経由で受け渡しが行われるメッセージには、セキュリティー情報が含まれています。 どのようなセキュリティー検査が行われるかは、IMS コマンド /SECURE OTMA の設定によって異なります。

ブリッジ経由で受け渡しが行われる各 IBM MQ メッセージには、以下のセキュリティー情報が含まれています。

- MQMD 構造の *UserIdentifier* フィールドに含まれるユーザー ID
- MQIIH 構造の *SecurityScope* フィールドに格納されているセキュリティー・スコープ (MQIIH 構造が存在する場合)
- UTOKEN (IBM MQ サブシステムが、関連の `IMSXCF.xcfgname.imsxcfmname` プロファイルに対して、CONTROL または ALTER アクセス権を持っていない場 合)

どのようなセキュリティー検査が行われるかは、IMS コマンド /SECURE OTMA の設定によって異なります。

/SECURE OTMA NONE
トランザクションのセキュリティー検査は行われません。

/SECURE OTMA CHECK
トランザクション権限またはコマンド権限の検査のために MQMD 構造の *UserIdentifier* フィールドが IMS に渡されます。

IMS の制御領域で ACEE (Accessor Environment Element) が作成されます。

/SECURE OTMA FULL
トランザクション権限またはコマンド権限の検査のために MQMD 構造の *UserIdentifier* フィールドが IMS に渡されます。

IMS の制御領域と IMS の従属領域で ACEE が作成されます。

/SECURE OTMA PROFILE
トランザクション権限またはコマンド権限の検査のために MQMD 構造の *UserIdentifier* フィールドが IMS に渡されます。

MQIIH 構造の *SecurityScope* フィールドに基づいて、IMS の制御領域と従属領域で ACEE で作成されるかどうかが決まります。

注 :

1. TIMS または CIMS クラス、あるいは関連のグループ・クラス GIMS または DIMS の権限を変更する場合、次の IMS コマンドを発行して、その変更を活動化する必要があります。
   - /MODIFY PREPARE RACF
   - /MODIFY COMMIT
2. /SECURE OTMA PROFILE を使用しない場合、MQIIH 構造の *SecurityScope* フィールドに指定された値は無視されます。

z/OS *IMS* ブリッジによるセキュリティー検査

実行する操作に応じて、さまざまな権限を使用します。

ブリッジがメッセージを書き込んだり取得したりするときに使用する権限 は、以下のとおりです。

ブリッジ・キューからメッセージを取得する操作
セキュリティー検査は行われません。

例外または COA レポート・メッセージを書き込む操作
MQMD 構造の *UserIdentifier* フィールドで指定されているユーザー ID の権限を使用します。

応答メッセージを書き込む操作
元のメッセージの MQMD 構造の *UserIdentifier* フィールドで指定されているユーザー ID の権限を使用します。

送達不能キューにメッセージを書き込む操作
セキュリティー検査は行われません。

注 :

1. IBM MQ のクラス・プロファイルを変更した場合は、IBM MQ の REFRESH SECURITY(*) コマンドを実行してその変更をアクティブ化する必要があります。

2. ユーザーの権限を変更した場合は、MQSC の RVERIFY SECURITY コマンドを実行してその変更をアクティブ化する必要があります。

z/OS *IMS ヘッダーでの RACF PassTickets の使用*

IMS ヘッダーでは、パスワードの代わりにパスチケットを使用できます。

IMS ヘッダー (MQIIH) でパスワードの 代わりにパスチケットを使用する場合は、メッセージの送付先になる IMS ブリッジ・キューの STGCLASS 定義の PASSTKTA 属性で、パスチケットを検証するために使用するアプリケーション名を指定します。

PASSTKTA 値をブランクのままにする場合は、パスチケットを生成するための調整が必要になります。 この場合のアプリケーション名は、MVSxxxx という形式でなければなりません (xxxx は、ターゲット・キュー・マネージャーを実行する z/OS システムの SMFID です)。

パスチケットは、ユーザー ID、ターゲット・アプリケーション名、秘密鍵で作成されます。 大文字の英字と数字が含まれている 8 バイトの値です。 使用できるのは 1 回限りであり、有効期間は 20 分です。 パスチケットがローカル RACF システムで生成される場合、RACF は、プロファイルが存在するかどうかを確認するだけで、ユーザーがそのプロファイルに対する権限を持っているかどうかについては確認しません。 パスチケットがリモート・システムで生成される場合、RACF は、ユーザー ID がプロファイルに対して持っているアクセス権を確認します。 パスチケットについて詳しくは、「*z/OS SecureWay Security Server RACF* セキュリティー管理者のガイド」を参照してください。

IMS ヘッダー内のパスチケットは、RACF ではなく、IBM MQ から IMS に与えられます。

z/OS z/OS キュー・マネージャーの大/小文字混合のセキュリティーへの移行

キュー・マネージャーを大/小文字混合のセキュリティーに移行するには、以下の手順を実行します。 使用しているセキュリティー製品のレベルを確認して、新しい IBM MQ 外部セキュリティー・モニター・クラスをアクティブ化します。 **REFRESH SECURITY** コマンドを実行して、大/小文字混合のプロファイルをアクティブ化します。

始める前に

1. すべての IBM MQ 外部セキュリティー・モニター・クラスがアクティブ化されていることを確認します。
2. キュー・マネージャーが開始していることを確認します。

このタスクについて

キュー・マネージャーを大/小文字混合のセキュリティーに変換するには、以下の手順を実行します。

手順

1. すべての既存のプロファイルおよびアクセス・レベルを、大文字のクラスから、同等の大/小文字混合の外部セキュリティー・モニター・クラスにコピーします。
   a) MQADMIN から MXADMIN
   b) MQPROC から MXPROC
   c) MQNLIST から MXNLIST
   d) MQQUEUE から MXQUEUE
2. 次のコマンドを発行して、SCYCASE 属性の値を MIXED に変更します。

```
ALTER QMGR SCYCASE(MIXED)
```

3. 次のコマンドを発行して、既存のセキュリティー・プロファイルをアクティブ化します。

```
REFRESH SECURITY(*) TYPE(CLASSES)
```

4. セキュリティー・プロファイルが正しく機能していることを検査します。

次のタスク

オブジェクト定義を確認し、プロファイルのアクティブ化に必要な **REFRESH SECURITY** を使用して、新規の大/小文字混合プロファイルを適宜作成します。

IBM MQ MQI client・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 MQ MQI client・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。

  IBM MQ 8.0 以降、認証は以下の 2 つのオプションの 1 つにより提供されています。

  – 接続認証フィーチャー。

    接続認証の詳細については、65 ページの『接続認証』を参照してください。

  – TLS 内の相互認証の使用。

    TLS の詳細については、270 ページの『SSL/TLS の取り扱い』を参照してください。
- 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、346 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、384 ページの『MCAUSER ユーザー ID へのユーザー 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、47 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、TLS 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 TLS 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 TLS 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 TLS 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、IBM 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)。 IBM MQ では、そのような接続が必ず拒否されることが保証されておらず、ユーザーは使用している IBM MQ 構成が FIPS 準拠であるかどうかを判別する必要があります。

関連概念

32 ページの『UNIX, Linux, and Windows での連邦情報処理標準 (FIPS)』
Windows システムや UNIX and Linux システム上の SSL/TLS チャネルで暗号化が必要な場合、IBM MQ は IBM Crypto for C (ICC) と呼ばれる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、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

AIX 複数の GSKit V8.0 インストールがある AIX 上での TLS クライアント・アプリケーションの実行

複数の GSKit V8.0 インストールが存在する AIX システムで AIX の 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 が発生することがあります。

複数の GSKit V8.0 インストールが存在する AIX システムでクライアント・アプリケーションを実行すると、TLS の使用時にクライアント接続呼び出しで MQRC_CHANNEL_CONFIG_ERROR が返されることがあります。 /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 ----------------------------------------------------------
```

このエラーは通常、LIBPATH 環境変数または LD_LIBRARY_PATH 環境変数の設定により IBM MQ クライアントが 2 つの異なる GSKit V8.0 インストールからライブラリーの混合セットをロードするときに生じます。 IBM 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 MQ ライブラリーのディレクトリーを組み込んで、IBM 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 (IBM MQ 環境の設定)』を参照してください。

IBM i IBM i 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。 一部のオペレーティング・システムでは、自己署名証明書でテストを実行できます。 ただし、IBM i では、ローカル CA の署名が付いた個人証明書を使用する必要があります。

証明書の作成と管理の詳細については、271 ページの『IBM i での SSL/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 通信または TLS 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、SSL および TLS 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 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。

IBM i では、SSL または TLS クライアントは、正しい IBM 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 の場合は、ibmwebspheremqqm1 です。

- IBM MQ C Client for IBM i の場合、 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)。

IBM 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。 SSL または TLS 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、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。

ULW UNIX、Linux または Windows 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、Linux および Windows システムでは、自己署名証明書でテストを実行できます。

重要 : TLS 対応チャネルを使用して結合させるキュー・マネージャー同士の間で、楕円曲線暗号の署名の付いた証明書と RSA の署名の付いた証明書を混在させることはできません。

TLS 対応チャネルを使用したキュー・マネージャーがすべて RSA の署名の付いた証明書を使用するか、すべて EC の署名の付いた証明書を使用するかのどちらかにしなければなりません。両方を混在させることはできません。

詳しくは、 43 ページの『IBM 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。

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、283 ページの『UNIX, Linux, and Windows での SSL/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 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 では、SSL または TLS クライアントは、正しい IBM 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 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 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、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。

z/OS 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。 z/OS では、自己署名証明書、またはローカル認証局 (CA) の署名の付いた個人証明書で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、315 ページの『z/OS での SSL/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 通信または TLS 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 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。

z/OS の SSL または TLS クライアントは、以下のいずれかの証明書が存在する場合に限って証明書を送信します。

- (共有チャネルの場合のみ) ibmWebSphereMQ の後にキュー共有グループ名を追加した形式のラベルの付いた証明書 (例えば、ibmWebSphereMQQSG1)
- ibmWebSphereMQ の後にキュー・マネージャー名を追加した形式のラベルの付いた証明書 (例えば、ibmWebSphereMQQM1)
- デフォルト証明書 (ibmWebSphereMQ 証明書の場合があります)。

チャネルを共有すると、チャネルはまず、キュー共有グループの証明書を見つけようとします。 キュー共有グループの証明書が見つからない場合は、キュー・マネージャーの証明書を見つけようとします。

z/OS で、IBM 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で使用します。

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。 SSL または TLS 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、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/TLS の取り扱い

これらのトピックでは、IBM MQ での 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。

- 327 ページの『ユーザーの識別および認証』
- 346 ページの『オブジェクトに対するアクセス権限の設定』
- 415 ページの『メッセージの機密性』
- 453 ページの『メッセージのデータ保全性』

- 454 ページの『クラスターのセキュリティーの確保』

IBM i IBM i での SSL/TLS の取り扱い

このトピック集では、IBM MQ for IBM i での Transport Layer Security (TLS) を処理する個別のタスクに関する指示を取り上げます。

IBM i の場合、TLS サポートは、オペレーティング・システムに不可欠の要素として組み込まれています。IBM i でのハードウェア要件とソフトウェア要件で取り上げられている前提条件製品がインストールされていることを確認してください。

IBM i では、DCM (Digital Certificate Manager) ツールを使用して鍵とデジタル証明書を管理します。

DCM へのアクセス

DCM インターフェースにアクセスする手順を取り上げます。

このタスクについて

フレームがサポートされている Web ブラウザーで以下の手順を実行します。

手順

1. `http://`*machine*`.domain:2001` または `https://`*machine*`.domain:2010` にアクセスします。ここで、*machine* は使用しているコンピューターの名前です。
2. 有効なユーザー・プロファイルとパスワードが要求されたら、それぞれの値を入力します。

   新しい証明書ストアを作成するには、ユーザー・プロファイルで *ALLOBJ と *SECADM の特殊権限が設定されている必要があります。 これらの特殊権限がない場合は、個人用証明書の管理、または 許可されているオブジェクトのオブジェクト・シグニチャーの表示のみが可能です。 オブジェクト署名アプリケーションの使用が許可されている場合は、DCM からオブジェクトに署名することもできます。
3. 「Internet Configurations (インターネット構成)」ページで、「**Digital Certificate Manager**」をクリックします。

   「Digital Certificate Manager」ページが表示されます。

IBM i でキュー・マネージャーに証明書を割り当てる操作

DCM を使用して、キュー・マネージャーに証明書を割り当てます。

IBM i の従来のデジタル証明書管理機能を使用して、キュー・マネージャーに証明書を割り当てます。 したがって、キュー・マネージャーがシステム証明書ストアを使用することと、キュー・マネージャーをアプリケーションとして DCM (Digital Certificate Manager) に登録することを指定できます。 そのためには、キュー・マネージャーの **SSLKEYR** 属性の値を *SYSTEM に変更します。

SSLKEYR パラメーターを *SYSTEM に変更すると、 IBM MQ はキュー・マネージャーをサーバー・アプリケーションとして QIBM_WEBSPHERE_MQ_QMGRNAME という固有のアプリケーション・ラベルと Qmgrname (WMQ) の説明付きのラベルで登録します。 *SYSTEM 証明書ストアを使用する場合、チャネルの **CERTLABL** 属性は使用されないことに注意してください。 その後、キュー・マネージャーは DCM (Digital Certificate Manager) でサーバー・アプリケーションとして表示されます。このアプリケーションに対し、システム・ストアで任意のサーバー証明書またはクライアント証明書を割り当てることができます。

キュー・マネージャーはアプリケーションとして登録されるので、CA トラスト・リストの定義などの DCM 拡張機能を実行できます。

SSLKEYR パラメーターが *SYSTEM 以外の値に変更されると、 IBM MQ は、アプリケーションとしてのキュー・マネージャーをデジタル Certificate Manager から登録解除します。 キュー・マネージャーが削除された場合も、DCM から登録解除されます。 十分な *SECADM 権限を備えたユーザーは、手動で DCM のアプリケーションを追加または登録削除できます。

IBM i での鍵リポジトリーのセットアップ

鍵リポジトリーは、接続の両端でセットアップされる必要があります。 デフォルトの証明書ストアを使用するか、独自の証明書ストアを作成することができます。

TLS 接続では、接続の両端に鍵リポジトリー が必要です。 各キュー・マネージャーおよび IBM MQ MQI client には、鍵リポジトリーへのアクセス権が必要です。 ファイル名とパスワードを使用して (つまり、*SYSTEM オプションを使用しないで) 鍵リポジトリーにアクセスする場合は、QMQM ユーザー・プロファイルに次の権限が付与されていることを確認してください。

- 鍵リポジトリーが入っているディレクトリーへの実行権限
- 鍵リポジトリーが入っているファイルの読み取り権限

詳しくは、23 ページの『SSL/TLS 鍵リポジトリー』を参照してください。 *SYSTEM 証明書ストアを使用する場合、 チャネル **CERTLABL** 属性は使用されないことに注意してください。

IBM i では、デジタル証明書は、DCM を使用して管理される証明書ストアに保管されます。 これらのデジタル証明書には、証明書をキュー・マネージャーまたは IBM MQ MQI client に関連付けるラベルがあります。 TLS は、認証のためにその証明書を使用します。

ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの `ibmwebspheremq` にキュー・マネージャーの名前か IBM MQ MQI client ユーザーのログオン ID をすべて小文字で付加した値のどちらかです。 詳細については、デジタル証明書ラベルを参照してください。

キュー・マネージャーまたは IBM MQ MQI client 証明書ストアの名前は、パスと語幹名から構成されます。 デフォルトのパスは `/QIBM/UserData/ICSS/Cert/Server/` であり、デフォルトの語幹名は `Default` です。 IBM i では、デフォルトの証明書ストア `/QIBM/UserData/ICSS/Cert/Server/Default.kdb` は `*SYSTEM` とも呼ばれます。 オプショナルで、独自のパスと語幹名を定義できます。

独自のパスまたはファイル名を定義する場合は、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。

証明書ストア名の指定については、273 ページの『IBM i 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。 証明書ストアの名前は、証明書ストアの作成前または作成後のどちらでも指定できます。

注 : DCM で実行可能な操作は、ユーザー・プロファイルの権限によって制限されます。 例えば、CA 証明書の作成には *ALLOBJ 権限および *SECADM 権限が必要です。

IBM i での証明書ストアの作成
デフォルトの証明書ストアを使用しない場合は、以下の手順で独自の証明書ストアを作成します。

このタスクについて

新しい証明書ストアを作成するのは、IBM i のデフォルトの証明書ストアを使用しない場合に限られます。

IBM i システム証明書ストアを使用することを指定するには、キュー・マネージャーの SSLKEYR 属性の値を *SYSTEM に変更します。 その値は、キュー・マネージャーがシステム証明書ストアを使用することと、キュー・マネージャーが使用可能なアプリケーションとしてデジタル証明書マネージャー (DCM) に登録されていることを示す値です。

手順

1. DCM インターフェースにアクセスします (271 ページの『DCM へのアクセス』を参照)。
2. ナビゲーション・パネルで、「**Create New Certificate Store (証明書ストアの作成)**」をクリックする。
   タスク・フレームに「Create New Certificate Store (証明書ストアの作成)」ページが表示されます。
3. タスク・フレームで、「**Other System Certificate Store (他のシステム証明書ストア)**」を選択して、「**Continue (続行)**」をクリックします。
   タスク・フレームに「Create a Certificate in New Certificate Store (新規証明書ストアでの証明書の作成)」ページが表示されます。
4. 「**No - Do not create a certificate in the certificate store (いいえ。証明書ストアに証明書を作成しません)**」を選択して、「**Continue (続行)**」をクリックします。
   タスク・フレームに「Certificate Store Name and Password (証明書ストア名およびパスワード)」ページが表示されます。

5. 「**証明書ストアのパスとファイル名**」 フィールドに、IFS パスとファイル名を入力します (例: `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`)。
6. **Password** フィールドにパスワードを入力し、**Confirm Password** フィールドにそのパスワードをもう一度入力する。 「**次へ進む**」をクリックします。
   パスワードを書き留めます (大/小文字の区別があります)。そのパスワードは、リポジトリーの鍵を隠すときに必要になります。
7. ブラウザー・ウィンドウをクローズし、DCM を終了する。

次のタスク

DCM を使用して証明書ストアを作成した場合は、必ずパスワードを隠してください (273 ページの『IBM i システムでの証明書ストアのパスワードの隠蔽』 を参照してください)。

関連タスク

278 ページの『IBM i で鍵リポジトリーに証明書をインポートする操作』
証明書をインポートする手順を取り上げます。

IBM i システムでの証明書ストアのパスワードの隠蔽
CL コマンドを使用して、証明書ストアのパスワードを隠します。

以下の指示は、IBM i でキュー・マネージャー用に証明書ストアのパスワードを隠す操作に適用されます。あるいは、 IBM MQ MQI client の場合、*SYSTEM 証明書ストアを使用しない場合 (つまり、MQSSLKEYR 環境が *SYSTEM 以外の値に設定されている場合) は、280 ページの『IBM MQ の IBM i 用 SSL クライアント・ユーティリティー (amqrsslc)』 の 281 ページの『証明書ストア・パスワードを隠しておく操作』 セクションで説明されている手順に従ってください。

キュー・マネージャーの SSLKEYR 属性の値を *SYSTEM に変更することによって、*SYSTEM 証明書ストアを使用するように指定した場合は、このセクションの手順を実行しないでください。

DCM を使用して証明書ストアを作成した場合は、次の証明を使用してパスワードを隠してください。

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

パスワードでは大/小文字を区別します。 パスワードは、272 ページの『IBM i での証明書ストアの作成』 のステップ 6 で入力したとおりに、単一引用符内に入力する必要があります。

注 : デフォルトのシステム証明書ストアを使用していない場合は、パスワードを隠さないと、証明書ストアへのアクセスに必要なパスワードを取得できないので、TLS チャネルを開始しようとしても失敗します。

IBM i 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・マネージャーの証明書ストアの位置を取得する手順を取り上げます。

手順

1. 次の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。

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. コマンドの出力を調べて、証明書ストアのパスと語幹名を見つける。
   例えば、`/QIBM/UserData/ICSS/Cert/Server/Default` です。ここで、`/QIBM/UserData/ICSS/Cert/Server` はパスであり、`Default` は語幹名です。

IBM i 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

CHGMQM または ALTER QMGR を使用して、キュー・マネージャーの証明書ストアの位置を変更します。

手順

CHGMQM コマンドまたは ALTER QMGR MQSC コマンドを使用して、キュー・マネージャーの鍵リポジトリー属性を設定します。

a) CHGMQM: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey')`

b) ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey')`

いずれの場合も、証明書ストアには完全修飾ファイル名 `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb` が含まれます。

次のタスク

キュー・マネージャーの証明書ストアの位置を変更する場合、証明書は、もとの場所から転送されません。証明書ストアを作成した時点で既にインストールされていた CA 証明書では不十分な場合は、新しい証明書ストアに証明書を取り込まなければなりません (278 ページの『IBM i で鍵リポジトリーに証明書をインポートする操作』を参照してください)。 新しい位置についても パスワードを隠しておかなければなりません (273 ページの『IBM i システムでの証明書ストアのパスワードの隠蔽』 を参照)。

IBM i でのテスト用の認証局と証明書の作成

証明書要求に署名するためのローカル CA 証明書を作成する手順と、CA 証明書を作成してインストールする手順を取り上げます。

始める前に

このトピックでは、ローカル認証局 (CA) が存在していないという前提で手順を説明します。 ローカル CA が存在する場合は、275 ページの『IBM i でのサーバー証明書の要求』 に進んでください。

このタスクについて

TLS のインストール時に提供される CA 証明書は、発行側 CA によって署名されます。 IBM i では、システムにおける TLS 通信をテストするためのサーバー証明書に署名できるローカル認証局を生成できます。 ローカル CA 証明書を作成するには、Web ブラウザーで以下の手順を実行します。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』 を参照)。
2. ナビゲーション・パネルで、「**Create a Certificate Authority (認証局の作成)**」をクリックする。
   タスク・フレームに「Create a Certificate Authority (認証局の作成)」ページが表示されます。
3. 「**Certificate store password (証明書ストア・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。
4. 「**認証局 (CA) 名**」フィールドに名前を入力する (例: `TLS Test Certificate Authority`)。
5. 「**共通名**」フィールドと「**組織**」フィールドに適切な値を入力して、国名を選択する。 残りのオプション・フィールドに必要な値を入力する。
6. ローカル CA の有効期間を「**Validity period (有効期間)**」フィールドに 入力する。
   デフォルト値は 1095 日です。
7. 「**次へ進む**」をクリックします。
   CA が作成され、DCM が ローカル CA 用の証明書ストアおよび CA 証明書を作成します。
8. 「**Install certificate (証明書をインストール)**」をクリックする。
   ダウンロード・マネージャーのダイアログ・ボックスが表示されます。
9. CA 証明書を格納する一時ファイルの絶対パス名を入力し、「**Save (保管)**」をクリックする。
10. ダウンロードが完了したら、「**Open (オープン)**」をクリックする。
    「Certificate (証明書)」ウィンドウが表示されます。
11. 「**Install certificate (証明書をインストール)**」をクリックする。
    「Certificate Import (証明書のインポート)」ウィザードが表示されます。
12. 「**Next**」 をクリックします。
13. 「**証明書のタイプに基づいて証明書ストアを自動的に選択**」を選択して、「**次へ**」をクリックする。
14. 「**完了**」をクリックします。

確認ウィンドウが表示されます。

15. 「**OK**」をクリックします。
16. 「証明書」ウィンドウで「**OK**」をクリックする。
17. 「**次へ進む**」をクリックします。

    タスク・フレームに「Certificate Authority Policy (認証局ポリシー)」ページが表示されます。
18. 「**ユーザー証明書の作成を許可**」フィールドで「**はい**」を選択する。
19. 「**Validity period (有効期間)**」フィールドに、ローカル CA によって発行された証明書の有効期間を入力する。

    デフォルト値は 365 日です。
20. 「**次へ進む**」をクリックします。

    タスク・フレームに「Create a Certificate in New Certificate Store (新規証明書ストアでの証明書の作成)」ページが表示されます。
21. アプリケーションが 1 つも選択されていないことを確認する。
22. 「**Continue (続行)**」をクリックし、ローカル CA のセットアップを完了する。

IBM i でのサーバー証明書の要求

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。 新規のサーバー証明書は、デジタル証明書マネージャー (DCM) を使用する認証局から要求することができます。

このタスクについて

Web ブラウザーで以下の手順を実行します。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
2. ナビゲーション・パネルで、「**Select a Certificate Store (証明書ストアの選択)**」をクリックする。

   タスク・フレームに「Select a Certificate Store (証明書ストアの選択)」ページが表示されます。
3. 使用する証明書ストアを選択して、「**Continue (続行)**」をクリックする。
4. オプション: 手順 3 で「***SYSTEM**」を選択した場合は、システム・ストアのパスワードを入力して、「**Continue (続行)**」をクリックする。
5. オプション: 手順 3 で「**Other System Certificate Store (他のシステム証明書ストア)**」を選択した場合は、「**Certificate store path and filename (証明書ストアのパスおよびファイル名)**」フィールドに、証明書ストアを作成したときに設定した IFS パスとファイル名を入力する。 さらに、「**Certificate Store Password (証明書ストア・パスワード)**」フィールドに パスワードを入力する。 次に、「**続行**」をクリックします。
6. ナビゲーション・パネルで、「**Create Certificate (証明書の作成)**」をクリックする。
7. タスク・フレームで 「**Server or client certificate (サーバーまたはクライアント証明書)**」ラジオ・ボタンを 選択し、「**Continue (続行)**」をクリックする。

   タスク・フレームに 「Select a Certificate Authority (CA) (認証局の選択)」ページが表示されます。
8. ワークステーションにローカル CA がある場合は、ローカル CA または商用 CA のいずれかを選択して証明書に署名する。 対象とする CA のラジオ・ボタンを選択し、「**Continue (続行)**」をクリックする。

   タスク・フレームに「Create a Certificate (証明書の作成)」ページが表示されます。
9. オプション: キュー・マネージャーの場合、「**証明書ラベル**」フィールドに証明書ラベルを入力します。

   ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの `ibmwebspheremq` にキュー・マネージャーの名前をすべて小文字で付加した値のどちらかです。 詳細は デジタル証明書ラベル を参照してください。

   例えば、キュー・マネージャー QM1 でデフォルト値を使用するには、`ibmwebspheremqqm1` と入力します。
10. オプション: IBM MQ MQI client の場合、「**Certificate label (証明書ラベル)**」フィールドで、`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値を入力する。

例えば、ibmwebspheremqmyuserID と入力します。

11.「**共通名**」フィールドと「**組織**」フィールドに適切な値を入力して、国名を選択する。 残りのオプション・フィールドに必要な値を入力する。

タスクの結果

証明書への署名に商用 CA を選択した場合は、DCM は PEM (Privacy-Enhanced Mail) 形式で認証要求を作成します。 対象とする CA に要求を転送します。

証明書への署名にローカル CA を選択した場合は、DCM は、証明書が証明書ストアに作成され、使用可能になったことを通知します。

IBM 鍵管理機能のサーバー証明書の要求 *(IBM i)*

ローカル認証局 (CA) の署名の付いた証明書を作成する手順と、商用 CA の署名の付いたサーバー証明書を申請する手順を取り上げます (その証明書は、IBM 鍵管理 (iKeyman) ユーティリティーにインポートします)。

このタスクについて

デジタル証明書マネージャー (DCM) が複数のプラットフォームで IBM MQ の証明書マネージャーとして機能する場合は、ユーザー証明書を使用する必要があります。 他のプラットフォームに配布したり iKeyman ユーティリティーにインポートしたりする個人証明書の場合は、Web ブラウザーで以下の手順を実行します。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
2. **ナビゲーション**・ペインで、「**Create Certificate (証明書の作成)**」をクリックします。

   タスク・フレームに「**Create Certificate (証明書の作成)**」ページが表示されます。
3. 「**Create Certificate (証明書の作成)**」パネルで、「**User certificate (ユーザー証明書)**」ラジオ・ボタンを選択し、「**Continue (続行)**」をクリックします。

   「**Create User Certificate (ユーザー証明書の作成)**」ページが表示されます。
4. 「**Create User Certificate (ユーザー証明書の作成)**」パネルで、「Certificate Information (証明書情報)」の下にある必須フィールド「**Organization name (組織名)**」、「**State (州)**」または「**province (県)**」、「**Country (国)**」または「**region (地域)**」に値を入力します。 オプションとして、「**Organization unit (組織単位)**」、「**Locality (地方)**」または「**city (市)**」の各フィールドに値を入力することもできます。 「**次へ進む**」をクリックします。

   「**Common name (共通名)**」は、iSeries システムにログオンしたときのユーザー ID に自動的に設定されます。
5. 次の「**Create User Certificate (ユーザー証明書の作成)**」パネルで、 「**Install certificate (証明書をインストール)**」をクリックし、「**Continue (続行)**」をクリックします。

   「個人証明書がインストールされました」というメッセージが表示されます。 この証明書のバックアップ・コピーは保持しておいてください。
6. 「**OK**」をクリックします。
7. DCM にアクセスするために使用したインターネット・ブラウザーに応じて、以下の手順を実行します。
   a) Microsoft Edge の場合は、「**ツール**」 **>** 「**インターネットオプション**」 **>** 「**コンテンツ**」**タブ > 「証明書**」**ボタン > 「個人**」**タブ** を選択します。証明書を選択し、「**エクスポート**」をクリックします。
   b) Mozilla Firefox の場合は、「**ツール**」 **>** 「**オプション**」 **>** 「**拡張**」 **>** 「**暗号化**」**タブ > 「証明書の表示**」**ボタン > 「証明書**」**タブ**を選択します。証明書を選択して「**バックアップ**」をクリックします。パスとファイル名を選択して、「**OK**」をクリックします。
8. FTP を使用して、エクスポートした証明書をバイナリー・フォーマットでリモート・システムに転送します。
9. 手順 7 でエクスポートした証明書を鍵データベースの iKeyman ユーティリティーに追加します。
   a) Microsoft Edge で証明書を保存した場合は、『Microsoft .pfx ファイルからのインポート』の手順を使用してください。

b) Mozilla Firefox で証明書を保存した場合は、『UNIX システムまたは Windows システムで鍵リポジトリーに個人証明書をインポートする操作』の手順を使用してください。

インポート時に、個人証明書と署名者証明書のラベル名が IBM MQ が必要とするラベル名に変更されていることを確認してください。 ラベルは、IBM MQ **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの `ibmwebspheremq` にキュー・マネージャーの名前をすべて小文字で付加した値のどちらかでなければなりません。 詳細は デジタル証明書ラベル を参照してください。

IBM i で鍵リポジトリーにサーバー証明書を追加する操作

要求された証明書を鍵リポジトリーに追加する手順を取り上げます。

このタスクについて

CA から新しいサーバー証明書が送信された後、要求の生成に使用した証明書ストア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。

注 :

- サーバー証明書がローカル CA によって署名されている場合は、この手順を行う必要はありません。
- PKCS #12 形式のサーバー証明書を DCM にインポートする場合は、まず事前に、対応する CA 証明書をインポートしておく必要があります。

サーバー証明書をキュー・マネージャーの証明書ストアに受信するには、次の手順を使用します。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
2. ナビゲーション・パネルの「**Manage Certificates (証明書の管理)**」タスク・カテゴリーで、「**Import Certificate (証明書のインポート)**」をクリックする。

   タスク・フレームに 「Import Certificate (証明書のインポート)」ページが表示されます。
3. 証明書タイプに対応するラジオ・ボタンを選択し、「**Continue (続行)**」をクリックする。

   タスク・フレームに、「Import Server or Client Certificate (サーバーまたはクライアント証明書のインポート)」ページ または「Import Certificate Authority (CA) Certificate (認証局証明書のインポート)」ページが 表示されます。
4. 「**Import File (インポート・ファイル)**」フィールドに、インポートする証明書のファイル名を入力し、「**Continue (続行)**」をクリックする。

   DCM がファイルの形式を自動的に判別します。
5. 証明書が「**Server or client (サーバーまたはクライアント)**」証明書の場合は、タスク・フレームにパスワードを入力し、「**Continue (続行)**」をクリックする。

   DCM が、証明書がインポートされたことを通知します。

IBM i で鍵リポジトリーから証明書をエクスポートする操作

証明書をエクスポートすると、公開鍵と秘密鍵の両方がエクスポートされます。 秘密鍵が人手に渡り、セキュリティーが完全に侵害される可能性があるため、この操作は十分に注意して実行する必要があります。

始める前に

ユーザーの証明書を他のユーザーと共有するときには、公開鍵を交換します。 このプロセスについては、**タスク 5 を参照してください**。 「Quick Start Guide for AMS on UNIX」の「Sharing Certificates」。 ここで説明する手順で証明書をエクスポートすると、公開鍵と秘密鍵の両方がエクスポートされます。 秘密鍵が人手に渡り、セキュリティーが完全に侵害される可能性があるため、この操作は十分に注意して実行する必要があります。

このタスクについて

エクスポートする証明書が入っているコンピューターで以下の手順を実行します。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
2. ナビゲーション・パネルで、「**Select a Certificate Store (証明書ストアの選択)**」をクリックする。
   タスク・フレームに「Select a Certificate Store (証明書ストアの選択)」ページが表示されます。
3. 使用する証明書ストアを選択して、「**Continue (続行)**」をクリックする。
4. オプション: 手順 3 で「***SYSTEM**」を選択した場合は、システム・ストアのパスワードを入力して、「**Continue (続行)**」をクリックする。
5. オプション: 手順 3 で「**Other System Certificate Store (他のシステム証明書ストア)**」を選択した場合は、「**Certificate store path and filename (証明書ストアのパスおよびファイル名)**」フィールドに、証明書ストアを作成したときに設定した IFS パスとファイル名を入力し、「**Certificate store password (証明書ストア・パスワード)**」フィールドにパスワードを入力します。 次に、「**続行**」をクリックします。
6. ナビゲーション・パネルの「**Manage Certificates (証明書の管理)**」タスク・カテゴリーで、「**Export Certificate (証明書のエクスポート)**」をクリックする。
   タスク・フレームに「Export a Certificate (証明書のエクスポート)」ページが表示されます。
7. 証明書タイプに対応するラジオ・ボタンを選択し、「**Continue (続行)**」をクリックする。
   タスク・フレームに、「Export Server or Client Certificate (サーバーまたはクライアント証明書のエクスポート)」ページ、または「Export Certificate Authority (CA) Certificate (認証局証明書のエクスポート)」ページが表示されます。
8. エクスポートする証明書を選択する。
9. ラジオ・ボタンを選択し、証明書をファイルにエクスポートするか 別の証明書ストアに直接エクスポートするかを指定する。
10. サーバー証明書またはクライアント証明書をファイルにエクスポートする場合は、以下の情報を指定します。
    - エクスポートする証明書を格納する位置のパスおよびファイル名。
    - 個人用証明書の場合は、エクスポートする証明書およびターゲット・リリースの暗号化に使用するパスワード。 CA 証明書の場合は、パスワードを指定する必要はありません。
11. 証明書を別の証明書ストアに直接エクスポートする場合は、ターゲットの証明書ストアおよびそのパスワードを指定します。
12. 「**次へ進む**」をクリックします。

IBM i で鍵リポジトリーに証明書をインポートする操作

証明書をインポートする手順を取り上げます。

始める前に

PKCS #12 形式の個人用証明書を DCM にインポートする場合は、まず事前に、対応する CA 証明書をインポートしておく必要があります。

このタスクについて

証明書のインポート先となるマシンで、次の手順を実行してください。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
2. ナビゲーション・パネルで、「**Select a Certificate Store (証明書ストアの選択)**」をクリックする。
   タスク・フレームに「Select a Certificate Store (証明書ストアの選択)」ページが表示されます。
3. 使用する証明書ストアを選択して、「**Continue (続行)**」をクリックする。
4. オプション: 手順 3 で「***SYSTEM**」を選択した場合は、システム・ストアのパスワードを入力して、「**Continue (続行)**」をクリックする。

5. オプション: 手順 3 で「**Other System Certificate Store (他のシステム証明書ストア)**」を選択した場合は、「**Certificate store path and filename (証明書ストアのパスおよびファイル名)**」フィールドに、証明書ストアを作成したときに設定した IFS パスとファイル名を入力し、「**Certificate store password (証明書ストア・パスワード)**」フィールドにパスワードを入力します。 次に、「**続行**」をクリックします。
6. ナビゲーション・パネルの「**Manage Certificates (証明書の管理)**」タスク・カテゴリーで、「**Import Certificate (証明書のインポート)**」をクリックする。

   タスク・フレームに「Import Certificate (証明書のインポート)」ページが表示されます。
7. 証明書タイプに対応するラジオ・ボタンを選択し、「**Continue (続行)**」をクリックする。

   タスク・フレームに、「Import Server or Client Certificate (サーバーまたはクライアント証明書のインポート)」ページ、または「Import Certificate Authority (CA) Certificate (認証局証明書のインポート)」ページが表示されます。
8. 「**Import File (インポート・ファイル)**」フィールドに、インポートする証明書のファイル名を入力し、「**Continue (続行)**」をクリックする。

   DCM がファイルの形式を自動的に判別します。
9. 証明書が「**Server or client (サーバーまたはクライアント)**」証明書の場合は、タスク・フレームにパスワードを入力し、「**Continue (続行)**」をクリックする。 DCM が、証明書がインポートされたことを通知します。

IBM i での証明書の削除

個人用証明書を除去するには、次の手順を使用します。

手順

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
2. ナビゲーション・パネルで、「**Select a Certificate Store (証明書ストアの選択)**」をクリックする。

   タスク・フレームに「Select a Certificate Store (証明書ストアの選択)」ページが表示されます。
3. 「**Other System Certificate Store (他のシステム証明書ストア)**」チェック・ボックスを選択し、「**Continue (続行)**」をクリックする。

   「Certificate Store and Password (証明書ストアおよびパスワード)」ページが表示されます。
4. 「**Certificate store path and filename (証明書ストアのパスおよびファイル名)**」フィールドに、証明書ストアを作成したときに設定した IFS パスとファイル名を入力します。
5. 「**Certificate Store Password (証明書ストア・パスワード)**」フィールドに パスワードを入力する。 「**次へ進む**」をクリックします。

   タスク・フレームに「Current Certificate Store (現在の証明書ストア)」ページが表示されます。
6. ナビゲーション・パネルの「**Manage Certificates (証明書の管理)**」タスク・カテゴリーで、「**Delete Certificate (証明書の削除)**」をクリックする。

   タスク・フレームに「Confirm Delete Certificate (証明書の削除の確認)」ページが表示されます。
7. 削除する証明書を選択する。 「**削除**」をクリックします。
8. 証明書を削除するには、「**Yes (はい)**」をクリックする。 これを指定しない場合は「**いいえ**」をクリックします。

   証明書が削除されたら、DCM からそのことが通知されます。

IBM i での片方向認証のための **SYSTEM* 証明書ストアの使用

片方向の認証をセットアップする手順を取り上げます。

始める前に

- キュー・マネージャー、チャネル、伝送キューを作成します。
- サーバーのキュー・マネージャーでサーバーまたはクライアントの証明書を作成します。
- CA 証明書をクライアント・キュー・マネージャーに転送して、鍵リポジトリーにインポートします。
- サーバーとクライアントのキュー・マネージャーでリスナーを開始します。

このタスクについて

IBM i を実行するコンピューターを TLS サーバーとして使用して片方向認証を使用するには、SSL 鍵リポジトリー (SSLKEYR) パラメーターを *SYSTEM に設定します。 そのように設定すると、IBM MQ のキュー・マネージャーがアプリケーションとして登録されます。 その後、キュー・マネージャーに証明書を割り当てることによって、片方向の認証を有効にできます。

また、専用の鍵ストアを使用して、片方向の認証を実装することもできます。それは、クライアント・キュー・マネージャーのダミー証明書を鍵リポジトリーで作成することによって可能です。

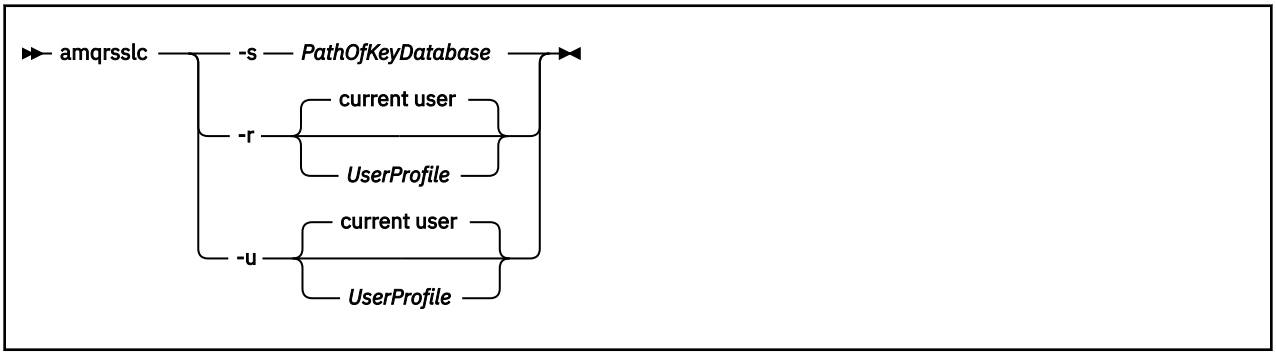
手順

1. サーバーとクライアントのキュー・マネージャーで以下の手順を実行します。
   a) コマンド `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)` を実行して、キュー・マネージャーを変更し、SSLKEYR パラメーターを設定します。
   b) コマンド `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')` を実行して、デフォルトの鍵リポジトリーのパスワードを隠します。
   パスワードは、単一引用符で囲む必要があります。
   c) チャネルを変更して、SSLCIPHER パラメーターに正しい CipherSpec を設定します。
   d) コマンド `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)` を実行して、TLS セキュリティーをリフレッシュします。
2. DCM を使用してサーバー・キュー・マネージャーに証明書を割り当てます。そのためには、以下のようにします。
   a) DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』を参照)。
   b) ナビゲーション・パネルで、「**Select a Certificate Store (証明書ストアの選択)**」をクリックする。
   タスク・フレームに「Select a Certificate Store (証明書ストアの選択)」ページが表示されます。
   c) *SYSTEM 証明書ストアを選択して、「**Continue (続行)**」をクリックします。
   d) 左のパネルで「**Manage Applications (アプリケーションの管理)**」を展開します。
   e)「**View Application (アプリケーションの表示)**」定義を選択して、キュー・マネージャーがアプリケーションとして登録されていることを確認します。
   「SSL (WMQ)」が表に表示されます。
   f)「**Update Certificate Assignment (証明書の割り当ての更新)**」を選択します。
   g)「**Server (サーバー)**」を選択して、「**Continue (続行)**」をクリックします。
   h)「QMGRNAME (WMQ)」を選択して、「**Update Certificate Assignment (証明書の割り当ての更新)**」をクリックします。
   i) 証明書を選択して、「**Assign New Certificate (新しい証明書の割り当て)**」をクリックします。 証明書がアプリケーションに割り当てられたことを通知するウィンドウが開きます。

IBM MQ の IBM i 用 SSL クライアント・ユーティリティー (amqrsslc)

IBM i 用の IBM MQ SSL クライアント・ユーティリティー (amqrsslc) は、 IBM i システム上の IBM MQ MQI client によって、クライアント・ユーザー・プロファイルの登録または登録抹消、あるいは証明書ストア・パスワードのスタッシュを行うために使用されます。 このユーティリティーを実行できるのは、*ALLOBJ 特殊権限のプロファイルがあるユーザー、またはデジタル証明書マネージャー (DCM) でアプリケーション登録を作成/削除するオプションがある QMQMADM のメンバーに限られます。

構文図



クライアント・ユーザー・プロファイルの登録

IBM MQ MQI client が *SYSTEM 証明書ストアを使用している場合は、アプリケーションとして使用するクライアント・ユーザー・プロファイル (ログオン・ユーザー) を デジタル Certificate Manager (DCM)に登録する必要があります。

クライアント・ユーザー・プロファイルを登録する場合は、 -r オプションで *UserProfile* を指定して **amqrsslc** プログラムを実行します。 **amqrsslc** を呼び出すときに使用するユーザー・プロファイルには、*USE 権限が必要です。 *UserProfile* に -r オプションを指定すると、*UserProfile* がサーバー・アプリケーションとして QIBM_WEBSPHERE_MQ_*UserProfile* という固有のアプリケーション・ラベルと *UserProfile* (WMQ) という記述のラベルで登録されます。 その後、このサーバー・アプリケーションは DCM に表示され、システム・ストアでこのアプリケーションにサーバー証明書やクライアント証明書を割り当てることができるようになります。

注 : -r オプションでユーザー・プロファイルを指定しない場合は、**amqrsslc** ツールを実行するユーザーのユーザー・プロファイルが登録されます。

amqrsslc を使用してユーザー・プロファイルを登録するコードを以下に示します。 最初の例では、指定したユーザー・プロファイルを登録し、2 番目の例では、ログイン・ユーザーのプロファイルを登録します。

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

クライアント・ユーザー・プロファイルの登録抹消

クライアント・プロファイルの登録を抹消する場合は、-u オプションで *UserProfile* を指定して **amqrsslc** プログラムを実行します。 **amqrsslc** を呼び出すときに使用するユーザー・プロファイルには、*USE 権限が必要です。 *UserProfile* に -u オプションを指定すると、 *UserProfile* が DCM からラベル QIBM_WEBSPHERE_MQ_*UserProfile* に登録抹消されます。

注 : -u オプションでユーザー・プロファイルを指定しない場合は、**amqrsslc** ツールを実行するユーザーのユーザー・プロファイルの登録が抹消されます。

amqrsslc を使用してユーザー・プロファイルの登録を抹消するコードを以下に示します。 最初の例では、指定したユーザー・プロファイルの登録を抹消し、2 番目の例では、ログイン・ユーザーのプロファイルの登録を抹消します。

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

証明書ストア・パスワードを隠しておく操作

IBM MQ MQI client が *SYSTEM 証明書ストアを使用しておらず、別の証明書ストアを使用している (つまり、MQSSLKEYR が *SYSTEM 以外の値に設定されている) 場合は、鍵データベースのパスワードをスタッ

シュする必要があります。 キー・データベースのパスワードを隠しておく場合は、-s オプションを使用します。

以下のコードでは、証明書ストアの完全修飾ファイル名は /Path/Of/KeyDatabase/MyKey.kdb です。

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

このコードを実行すると、このキー・データベースのパスワードに関する要求が実行されます。 このパスワードは、キー・データベースと同じ名前に .sth という拡張子を付けたファイルの中に隠しておかれます。このファイルは、キー・データベースと同じパスに格納されます。 このコード例では、/Path/Of/KeyDatabase/MyKey.sth の stash ファイルを生成します。 QMQM がこのファイルのユーザー所有者、QMQMADM がグループ所有者になります。 QMQM と QMQMADM には、読み取り権限と書き込み権限がありますが、他のプロファイルには、読み取り権限だけがあります。

IBM i 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。

証明書ストア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 TLS 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 TLS 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、IBM MQ TLS 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 TLS 接続が閉じられるとき。次回の TLS 接続で、証明書の変更が反映されます。
- 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、TLS 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 TLS 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、TLS 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 TLS 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、TLS 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 TLS 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 i での暗号ハードウェアの構成

IBM i で暗号コプロセッサーを構成する手順を取り上げます。

始める前に

コプロセッサー・ハードウェアを構成するには、ユーザー・プロファイルに *ALLOBJ および *SECADM の特殊権限が必要です。

手順

1. http://*machine*.domain:2001 または https://*machine*.domain:2010 のいずれかに移動します。ここで、 *machine* はご使用のコンピューターの名前です。
   ユーザー名とパスワードを要求するダイアログ・ボックスが表示されます。
2. 有効な IBM i ユーザー・プロファイルおよびパスワードを入力します。
3. 詳細については、暗号化に移動し、該当するリンクに従ってください。

次のタスク

4767 暗号化コプロセッサーの構成の詳細については、4767 暗号化コプロセッサーを参照してください。

ULW UNIX, Linux, and Windows での SSL/TLS の取り扱い

UNIX, Linux, and Windows システムでは、IBM MQ とともに Transport Layer Security (TLS) サポートが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。

ULW *runmqckm*、*runmqakm*、および *strmqikm* を使用したデジタル証明書の管理

UNIX, Linux, and Windows システムでは、**strmqikm** (iKeyman) GUI またはコマンド行から **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) を使用して、鍵およびデジタル証明書を管理します。

V 9.1.0

重要 : **runmqckm** コマンドと **strmqikm** コマンドはどちらも、 IBM MQ Java ランタイム環境 (JRE) に依存します。 IBM MQ 9.1 以降、JRE がインストールされていない場合、メッセージ AMQ9183 を受け取ります。

- **UNIX and Linux** システムの場合:
  - **strmqikm** (iKeyman) コマンドを使用し、iKeyman GUI を開始 する。
  - **runmqckm** (iKeycmd) 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** (GSKCapiCmd) 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。

    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あり、これらのプラットフォームの iKeyman プログラムおよび iKeycmd プログラムは 32 ビットです。

  詳しくは、GSKit: PKCS#11 および IBM MQ JRE アドレッシング・モードを参照してください。

  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。

    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。
  - *stashpw* または *stash* オプションを指定した **runmqakm -keydb** コマンドを使用します。

例えば、次のように **runmqakm -keydb** コマンドを使用した場合、

```
runmqakm -keydb -create -db key.kdb -pw secretpwd -stash
```

結果の .sth ファイルの読み取り権限は、mqm グループに付与されていません。

作成者のみがファイルを読み取ることができます。 **runmqakm** コマンドを使用して stash ファイルを作成した後は、ファイル許可を確認し、キュー・マネージャーを実行するサービス・アカウント、またはローカル mqm などのグループに権限を付与してください。

UNIX、Linux、または Windows システムで TLS 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
このセクションでは、runmqckm コマンドおよび runmqakm コマンドを、コマンドのオブジェクトに応じて説明します。

ULW *UNIX, Linux, and Windows での鍵リポジトリーのセットアップ*

鍵リポジトリーは、**strmqikm** (iKeyman) GUI を使用して、あるいは **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) コマンドを使用してコマンド行からセットアップできます。

このタスクについて

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 MQ キュー・マネージャーおよび IBM MQ MQI client には、鍵リポジトリーへのアクセス権が必要です。 詳しくは、23 ページの『SSL/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、**strmqikm** ユーザー・インターフェースを使用するか、あるいは **runmqckm** 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 MQ MQI client に関連付けます。 TLS は、認証のためにその証明書を使用します。 UNIX, Linux, and Windows のシステムでは、IBM MQ は、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client ユーザーのログオン ID をすべて小文字で付加した値のどちらかを使用します。 詳細については、デジタル証明書ラベルを参照してください。

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 (キュー・マネージャーの作成時に設定される) のデフォルト・パスは、/var/mqm/qmgrs/*queue_manager_name*/ssl です。

  Windows システムでは、デフォルト・パスは *MQ_INSTALLATION_PATH*\Qmgrs*queue_manager_name*\ssl です。ここで、*MQ_INSTALLATION_PATH* は IBM MQ がインストールされているディレクトリーです。 例えば、C:¥Program Files¥IBM¥MQ\Qmgrs\QM1\ssl などです。

  デフォルトの stem 名は key です。 オプションで、独自のパスとステム名を選択することができますが、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。
- IBM 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 でなければなりません。

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、289 ページの『UNIX, Linux, and Windows 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。

strmqikm または **runmqckm** 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、**strmqikm** または **runmqckm** 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 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 **strmqikm** または **runmqckm** 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 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。 詳細については、286 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 287 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。

strmqikm または **runmqckm** for IBM WebSphere MQ 7.0 では、新しい鍵データベースには、事前定義された認証局 (CA) 証明書のセットが自動的に取り込まれます。 **strmqikm** または **runmqckm** for IBM MQ 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ないため、必要な CA 証明書のみが鍵データベース・ファイルに含まれるため、初期セットアップのセキュリティーが強化されます。

注 : GSKit 8.0 の動作がこのように変更され、CA 証明書が自動的にリポジトリーに追加されなくなったため、優先 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。 287 ページの『GSKit 8.0 を使用した UNIX, Linux, and Windows で空の鍵リポジトリーにデフォルトの CA 証明書を追加する操作』 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。
   - **runmqckm** を使用する場合:

     ```
     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 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
   コマンドが FIPS モードで実行されるように指定します。 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、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テキスト key.kdb が含まれています。 ステム名が 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。 別のステム名を指定した場合は、key をご使用のステム名に置き換えてください。 ただし、.kdb 拡張機能を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 システム上) または C:¥ProgramData¥IBM¥MQ\qmgrs\QM1\ssl (Windows システム上)。
   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 MQ クライアントの場合: /var/mqm/ssl (UNIX and Linux システム上) または C:\mqm\ssl (Windows システム上)。
8. **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。
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、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. 286 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 287 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 *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。

制限されたユーザー・セットに権限を付与するには、ファイル *key*.kdb、*key*.sth、*key*.crl、および *key*.rdb にアクセス制御を設定します。「*key*」 は鍵データベースのステム名です。

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。

Linux UNIX *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。

ファイル *key*.kdb、*key*.sth、*key*.crl、および *key*.rdb に対する許可を設定します。ここで、「鍵」は、鍵データベースのステム名、ファイル所有者の「読み取り」および「書き込み」、および mqm またはクライアント・ユーザー・グループの「読み取り」(-rw-r-----) に対して設定します。

ULW *GSKit 8.0* を使用した *UNIX, Linux, and Windows* で空の鍵リポジトリーにデフォルトの *CA* 証明書を追加する操作

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 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。

strmqikm 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、GUI を開始する (UNIX, Linux, and 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 **runmqckm** 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、IBM 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。

ULW *UNIX, Linux, and Windows* 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 MQ Explorer 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: /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* は、IBM 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。

ULW *UNIX, Linux, and Windows 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。 例えば、UNIX and Linux では、以下を使用します。

```
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。

On Windows:

```
ALTER QMGR SSLKEYR('C:&#xa5;Program Files&#xa5;IBM&#xa5;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¥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。

IBM 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。 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、304 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)』 で説明されているように、必要な CA 証明書および必要な個人証明書をそのファイルに取り込む必要があります。

ULW *IBM MQ MQI client の鍵リポジトリーの位置確認 (UNIX, Linux, and Windows)*

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 MQ MQI client の鍵データベース・ファイルの位置を取得します。 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 (289 ページの『UNIX, Linux, and Windows 上の IBM MQ MQI client 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。

ULW *UNIX, Linux, and Windows 上の IBM MQ MQI client の鍵リポジトリーの場所の指定*

IBM MQ MQI client には、デフォルトの鍵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 MQ MQI client の鍵データベース・ファイルの位置は、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。 例えば、UNIX and Linux では、以下を使用します。

```
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
```

On Windows:

```
set MQSSLKEYR=C:&#xa5;Program Files&#xa5;IBM&#xa5;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:&#xa5;Program Files&#xa5;IBM&#xa5;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。

ULW *UNIX, Linux, and Windows* 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 TLS 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 TLS 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、TLS 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 TLS 接続が閉じられるとき。 次の TLS 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、TLS 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 TLS 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、TLS 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 TLS 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、TLS 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 TLS 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 MQ エクスプローラーまたは PCF コマンドを使用して IBM MQ TLS 環境を最新表示することもできます。

ULW *UNIX, Linux, and Windows* での自己署名個人証明書の作成

自己署名証明書は、**strmqikm** (iKeyman) GUI を使用して、あるいは **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) を使用してコマンド行から作成できます。

注 : IBM 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、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。 IBM 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、43 ページの『IBM MQ におけるデジタル証明書と CipherSpec の互換性』トピックの 44 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_ で始まるもの) 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。

-sig_alg ハッシュ・アルゴリズムで使用可能なオプションのリストについては、 526 ページの『UNIX, Linux, and Windows での runmqckm および runmqakm オプション』 を参照してください。

以下のとおりです。

- GUI。291 ページの『strmqikm ユーザー・インターフェースの使用』を参照してください。
- コマンド行。292 ページの『コマンド行の使用』を参照してください。

ULW ***strmqikm*** ユーザー・インターフェースの使用

strmqikm (iKeyman) GUI を使用して、個人証明書を作成できます。

このタスクについて

strmqikm は、FIPS 準拠のオプションを提供していません。 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

グラフィカル・ユーザー・インターフェースを使用して、キュー・マネージャーまたは IBM MQ MQI client の個人証明書を作成するには、以下の手順を実行します。

1. **strmqikm** コマンドを使用して、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. 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. **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。
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. 「**鍵ラベル**」フィールドに証明書ラベルを入力します。

ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client のログオン・ユーザー ID をすべて小文字で付加した値のどちらかです。 詳細については、デジタル証明書ラベルを参照してください。

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。
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。

次のタスク

CA に証明書要求を送信します。 詳しくは、298 ページの『個人証明書を鍵リポジトリーに受信する操作 (UNIX, Linux, and Windows)』を参照してください。

ULW コマンド行の使用

個人証明書は、**runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) コマンドを使用してコマンド行から作成できます。 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。

手順

runmqckm コマンドまたは **runmqakm** (GSKCapiCmd) コマンドを使用して、自己署名個人証明書を作成します。

- UNIX, Linux, and Windows 上で **runmqckm** 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_dnsname *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
```

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。 ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client のログオン・ユーザー ID をすべて小文字で付加した値のどちらかです。 詳しくは、25 ページの『デジタル証明書ラベルの要件に関する説明』 を参照してください。

-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。

注 : **runmqckm** および **runmqakm** ツールでは、郵便番号属性として PC ではなく POSTALCODE が参照されます。 これらの証明書管理コマンドを使用して郵便番号を含む証明書を要求する場合は、常に、**-dn** パラメーターに POSTALCODE を指定します。

-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。

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。

-file *filename*
認証要求のファイル名を指定します。

-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。 FIPS ICC コンポーネントのみが使用されます。このコンポーネントは FIPS モードで正常に初期化されている必要があります。 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、**runmqaqm** 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、SHA1WithECDSA、SHA1WithRSA、SHA2/ECDSA、SHA224WithECDSA、SHA256_WITH_RSA、SHA256WithECDSA、SHA256WithRSA、SHA2WithECDSA、SHA3/ECDSA、SHA384_WITH_RSA、SHA384WithECDSA、SHA384WithRSA、SHA3WithECDSA、SHA5/ECDSA、SHA512_WITH_RSA、SHA512WithECDSA、SHA512WithRSA、 SHA5WithECDSA、SHA_WITH_DSA、SHA_WITH_RSA、SHAWithDSA、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。

次のタスク

CA に証明書要求を送信します。 詳しくは、298 ページの『個人証明書を鍵リポジトリーに受信する操作 (UNIX, Linux, and Windows)』を参照してください。

ULW *UNIX, Linux, and Windows 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** (iKeycmd) または **runmqakm** (GSKCapiCmd) 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

strmqikm 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。

- IBM 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。 IBM 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、43 ページの『IBM MQ におけるデジタル証明書と CipherSpec の互換性』トピックの 44 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。

  -sig_alg ハッシュ・アルゴリズムで使用可能なオプションのリストについては、526 ページの『UNIX, Linux, and Windows での runmqckm および runmqakm オプション』 を参照してください。
- 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、313 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。

以下のとおりです。

- GUI。294 ページの『strmqikm ユーザー・インターフェースの使用』を参照してください。
- コマンド行。295 ページの『コマンド行の使用』を参照してください。

ULW *strmqikm* ユーザー・インターフェースの使用

個人証明書は、**strmqikm** (iKeyman) GUI を使用して、あるいは **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) 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

strmqikm は、FIPS 準拠のオプションを提供していません。 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**」フィールドに証明書ラベルを入力します。
   ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client のログオン・ユーザー ID をすべて小文字で付加した値のどちらかです。 詳細については、デジタル証明書ラベルを参照してください。
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. 「**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、絶対パスを指定して新しい値を入力します。
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 295 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。

ULW コマンド行の使用

個人証明書は、**runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) 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。

手順

runmqckm コマンドまたは **runmqakm** (GSKCapiCmd) コマンドを使用して、個人証明書を要求します。

- **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。 ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client のログオン・ユーザー ID をすべて小文字で付加した値のどちらかです。 詳しくは、25 ページの『デジタル証明書ラベルの要件に関する説明』 を参照してください。

-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。

注 : **runmqckm** および **runmqakm** ツールでは、郵便番号属性として PC ではなく POSTALCODE が参照されます。 これらの証明書管理コマンドを使用して郵便番号を含む証明書を要求する場合は、常に、**-dn** パラメーターに POSTALCODE を指定します。

-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。 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、SHA1WithECDSA、SHA1WithRSA、SHA2/ECDSA、SHA224WithECDSA、SHA256_WITH_RSA、SHA256WithECDSA、SHA256WithRSA、SHA2WithECDSA、SHA3/ECDSA、SHA384_WITH_RSA、SHA384WithECDSA、SHA384WithRSA、SHA3WithECDSA、SHA5/ECDSA、SHA512_WITH_RSA、SHA512WithECDSA、SHA512WithRSA、 SHA5WithECDSA、SHA_WITH_DSA、SHA_WITH_RSA、SHAWithDSA、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。

次のタスク

CA に証明書要求を送信します。 詳しくは、298 ページの『個人証明書を鍵リポジトリーに受信する操作 (UNIX, Linux, and Windows)』 を参照してください。

ULW *UNIX, Linux, and Windows での既存の個人証明書の更新*

個人証明書は、**strmqikm** (iKeyman) GUI を使用して、あるいは **runmqckm** (iKeycmd) または **runmqakm** (GSKCapiCmd) コマンドを使用してコマンド行から更新できます。

このタスクについて

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、既存の証明書を更新することはできません。294 ページの『UNIX, Linux, and Windows での個人証明書の要求』 に記載されているステップに従って既存の鍵を交換し、必要な鍵サイズを使用する新しい証明書要求を作成してください。

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。

strmqikm ユーザー・インターフェースの使用

このタスクについて

strmqikm は、FIPS 準拠のオプションを提供していません。 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

strmqikm 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 で **strmqikm** コマンドを使用して、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 297 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

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。

- UNIX, Linux, and Windows システムでの **runmqckm** の使用:

```
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、298 ページの『個人証明書を鍵リポジトリーに受信する操作 (UNIX, Linux, and Windows)』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。

ULW 個人証明書を鍵リポジトリーに受信する操作 *(UNIX, Linux, and Windows)*

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。

strmqikm の使用

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** 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 IBM MQ MQI client 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、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

個人証明書を鍵データベース・ファイルに追加するには、次のいずれかのコマンドを使用します。

- **runmqckm** を使用する場合:

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

- **runmqakm** を使用する場合:

```
runmqakm -cert -receive -file filename -db filename -pw password -fips
```

ここで、

-file *filename*
個人用証明書の完全修飾ファイル名を指定します。

-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。

-format 「*ascii*」
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 `ascii` 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。 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。

暗号ハードウェアを使用している場合は、314 ページの『PKCS #11 ハードウェアでの個人証明書の受け取り』を参照してください。

ULW 鍵リポジトリーから *CA* 証明書を抽出する操作 *(UNIX, Linux, and Windows)*

CA 証明書を取り出すには、次の手順に従います。

strmqikm の使用

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** (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、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

runmqckm を使用して CA 証明書を抽出するには、以下のコマンドを使用します。

- On UNIX, Linux, and Windows:

```
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。
-fips	コマンドが FIPS モードで実行されるように指定します。 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。

ULW 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操作 *(UNIX, Linux, and Windows)*

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。

strmqikm の使用

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** (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、GUI を開始する (UNIX, Linux, and 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. **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。
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

runmqckm 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。

- On UNIX, Linux, and Windows:

```
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。
-fips	コマンドが FIPS モードで実行されるように指定します。 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。

ULW *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 (UNIX, Linux, and Windows)*

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 : 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。

strmqikm の使用

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** 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、GUI を開始する (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. **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。
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. **追加** をクリックします。 「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

CA 証明書を鍵データベースに追加するには、次のいずれかのコマンドを使用します。

- **runmqckm** を使用する場合:

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

- **runmqakm** を使用する場合:

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

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*
証明書に付加するラベルを指定します。

-file *filename*
証明書を含むファイルの名前を指定します。

-format 「***ascii***」
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 `ascii` 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。 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。

ULW *UNIX, Linux, and Windows* で鍵リポジトリーから個人証明書をエクスポートする操作

個人証明書をエクスポートする手順を取り上げます。

strmqikm の使用

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** (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、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

runmqckm を使用して個人証明書をエクスポートするには、以下のコマンドを使用します。

- On UNIX, Linux, and Windows:

```
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。
-fips	コマンドが FIPS モードで実行されるように指定します。 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。
-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。

ULW 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)*

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 (302 ページの『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 (UNIX, Linux, and Windows)』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。

strmqikm の使用

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** 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、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、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。 証明書を特定のキュー・マネージャーまたは IBM MQ MQI client と関連付けるには、IBM MQ で **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client ユーザーのログオン ID をすべて小文字で付加した値のどちらかを使用します。 詳細については、デジタル証明書ラベルを参照してください。
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。 ラベルの変更ウィンドウが閉じ、元の IBM キー管理ウィンドウが再表示され、正しくラベル付けされた証明書で更新された 「**個人証明書**」 および 「**署名者証明書**」 フィールドが表示されます。
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

runmqckm を使用して個人用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。

- On UNIX, Linux, and Windows:

```
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。
-fips	コマンドが FIPS モードで実行されるように指定します。 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。

runmqckm 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する。 -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。

ULW *Microsoft .pfx* ファイルからの個人証明書のインポート

UNIX, Linux, and Windows 上の Microsoft.pfx ファイルからインポートするには、以下の手順に従います。

.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、GUI を開始する。 「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 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 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。 分かりやすい名前は、IBM MQ **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmwebspheremq にキュー・マネージャーの名前か IBM MQ MQI client ユーザーのログオン ID をすべて小文字で付加した値のどちらかにしなければなりません。 詳細については、デジタル証明書ラベルを参照してください。
21. **OK** をクリックします。 「ラベルの変更」ウィンドウが削除され、元の「IBM 鍵管理」ウィンドウが「個人証明書」および「署名者証明書」パネルで再表示され、正しくラベル付けされた個人証明書で更新されます。
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。

runmqckm または **runmqakm** を使用して証明書ラベルを変更することはできません。

コマンド行の使用

UNIX, Linux, and Windows で **runmqckm** を使用して個人証明書をインポートするには、以下のコマンドを使用します。

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

runmqakm を使用して個人用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。

```
runmqakm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label -fips -pfx
```

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。
-fips	コマンドが FIPS モードで実行されるように指定します。 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。
-pfx	PFX ファイルのフォーマットを指定します。

runmqckm 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する。 -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。

ULW *PKCS #7 ファイルからの個人証明書のインポート*

strmqikm (iKeyman) および **runmqckm** (iKeycmd) ツールは、PKCS #7 (.p7b) ファイルをサポートしていません。 **runmqckm** ツールを使用して、 UNIX, Linux, and Windows 上の PKCS #7 ファイルから証明書をインポートします。

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。 -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。

ULW *UNIX, Linux, and Windows* 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。

strmqikm の使用

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** (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、GUI を開始する (UNIX, Linux, and 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 (303 ページの『UNIX, Linux, and Windows で鍵リポジトリーから個人証明書をエクスポートする操作』 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

runmqckm を使用して証明書を削除するには、以下のコマンドを使用します。

- On UNIX, Linux, and Windows:

```
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。 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。

ULW 鍵リポジトリー保護のための強力なパスワードの生成 *(UNIX, Linux, and Windows)*

runmqakm (GSKCapiCmd) 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 **strmqikm** 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。

ULW 暗号ハードウェア *(UNIX, Linux, and Windows)*

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, and Windows 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 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, and Windows 上の IBM MQ クライアント用に暗号ハードウェアを構成することができます。

- 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。

  GSK_PKCS11 バージョンの SSLCRYP パラメーターを使用する場合、PKCS #11 トークン・ラベルは、ハードウェアの構成に使用したラベルと一致している必要があります。
- 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、310 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。

ULW *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 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。

   - **runmqckm** を使用する場合:

     ```
     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 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
   コマンドが FIPS モードで実行されるように指定します。 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. Java セキュリティー・プロパティー・ファイル `java.security` を開きます。
   - UNIX and Linux システムでは、Java セキュリティー・プロパティー・ファイルは、`java/jre64/jre/lib/security` インストール・ディレクトリーの IBM MQ サブディレクトリーにあります。
   - Windows システムでは、Java セキュリティー・プロパティー・ファイルは、`java\jre\lib\security` インストール・ディレクトリーの IBM MQ サブディレクトリーにあります。

   ファイル内にまだない場合は IBMPKCS11Impl セキュリティー・プロバイダーを追加します。 例えば、次の行を追加します。

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
   ```
4. **strmqikm** コマンドを実行して、ユーザー・インターフェースを開始する。
5. 「**鍵データベース・ファイル**」 > 「**オープン**」をクリックします。
6. 「**鍵データベース・タイプ**」をクリックして、**PKCS11Direct** を選択します。
7. 「**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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。
8. 「**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。
9. 証明書の保管に使用する暗号デバイス・トークン・ラベルを選択します。
10. 「**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。
11. 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 313 ページの『15』 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。
12. 「**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。
13. 「**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 MQ MQI client 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。
14. パスワードを入力してください。

    ステップ 312 ページの『11』 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 312 ページの『11』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、TLS 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)。

15. **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。

ULW *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 MQ MQI client 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。

このタスクについて

このタスクでは、**strmqikm** ユーザー・インターフェースを使用して個人証明書を要求する方法について説明します。コマンド・ライン・インターフェースを使用する場合は、295 ページの『コマンド行の使用』を参照してください。

注 : IBM 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

strmqikm (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。310 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**」フィールドに証明書ラベルを入力します。

   ラベルは、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの `ibmwebspheremq` にキュー・マネージャーの名前か IBM MQ MQI client のログオン・ユーザー ID をすべて小文字で付加した値のどちらかです。詳細については、デジタル証明書ラベルを参照してください。
4. 必要な「**鍵サイズ**」と「**署名アルゴリズム**」を選択します。
5.「**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の詳細は、11 ページの『識別名』を参照してください。
6.「**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、絶対パスを指定して新しい値を入力します。
7. **OK** をクリックします。

   確認ウィンドウが開きます。
8. **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 313 ページの『6』で選択したファイルに保管されます。

9. 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。

ULW *PKCS #11* ハードウェアでの個人証明書の受け取り

キュー・マネージャーまたは IBM MQ MQI client が個人証明書を暗号ハードウェアに受信するには、以下の手順を使用します。

始める前に

個人証明書への署名を行った CA の CA 証明書を追加します。 この証明書は、暗号ハードウェアまたは 2 次 CMS 鍵データベースに追加します。 この追加は、署名付き証明書を暗号ハードウェア内で受け取る前に行います。 CA 証明書を鍵リングに追加するには、302 ページの『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 (UNIX, Linux, and Windows)』の手順に従います。

手順

- **strmqikm** (iKeyman) ユーザー・インターフェースを使用して個人証明書を受信するには、以下の手順を実行します。
  a) 暗号ハードウェアを使用する手順を実行します。 310 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。
  b)「**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。
  c) 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。
  d) **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。
  e)「**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。
  f) **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。
- **runmqakm** (GSKCapiCmd) コマンドを使用して個人証明書を受信するには、以下の手順を実行します。
  a) 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。
  b) **runmqakm** (GSKCapiCmd) コマンドを使用して、個人証明書を受け取ります。

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

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 *module_name*
暗号化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名を指定します。

-tokenlabel *hardware_token*
PKCS #11 暗号デバイス・トークン・ラベルを指定します。

-pw *hardware_password*
暗号化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。 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。

-secondaryDB *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。

-secondaryDBpw *password*
CMS 鍵データベースのパスワードを指定します。

MQ Appliance IBM MQ Appliance での SSL/TLS の取り扱い

IBM MQ Appliance にはトランスポート層セキュリティー (TLS) サポートがあります。

IBM MQ Appliance には、証明書の管理のための別個のコマンドがあります。 証明書管理の詳細については、IBM MQ Appliance 資料の TLS 証明書管理 を参照してください。

z/OS z/OS での SSL/TLS の取り扱い

z/OS で Transport Layer Security (TLS) をセットアップして処理する方法を取り上げます。

各トピックには、RACF を使用して各タスクを実行する例が含まれています。 これ以外の外部セキュリティー・マネージャーを使用して同様のタスクを実行することもできます。

z/OS では、各キュー・マネージャーが TLS 呼び出しの処理に使用するサーバー・サブタスクの数を設定することも必要です。316 ページの『SSLTASKS パラメーターの設定 (z/OS)』 を参照してください。

z/OS の TLS サポートは、オペレーティング・システムに必須部分として組み込まれており、*System SSL* とも呼ばれます。 System SSL は、z/OS の Cryptographic Services Base エレメントの一部です。 暗号サービス・ベース・メンバーは、「*pdsname*」 にインストールされます。 「SIEALNKE 区分データ・セット (PDS)」System SSL をインストールする場合、必要な CipherSpec を提供するために適切なオプションを選択していることを確認してください。

z/OS *TLS でのユーザー ID に関する追加の要件 (z/OS)*

ご使用のユーザー ID が z/OS で TLS をセットアップして操作するのに必要な追加の要件を取り上げます。

ご使用のシステムに、該当する High Impact or Pervasive (HIPER) の更新版がすべてあることを確認してください。

以下の前提条件が設定されていることを確認してください。

- *ssidCHIN* ユーザー ID が RACF で正しく定義されており、*ssidCHIN* ユーザー ID が以下のプロファイルに対する READ アクセス権限を持っていること。
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  これらの変数は RACF FACILITY クラスで定義されています。
- *ssidCHIN* ユーザー ID が鍵リングの所有者である。
- RACDCERT コマンドによってキュー・マネージャーの個人証明書を作成した場合、作成時に指定した証明書タイプのユーザー ID が *ssidCHIN* のユーザー ID と同じである。
- 鍵リングに対して行った変更を反映するために、チャネル・イニシエーターがリサイクルされるか、コマンド **REFRESH SECURITY TYPE(SSL)** が発行されます。
- リンク・リスト、LPA、または STEPLIB DD ステートメントにより、IBM MQ チャネル・イニシエーター・プロシージャーがシステム SSL ランタイム・ライブラリー *pdsname*.SIEALNKE へのアクセス権を持っている。 このライブラリーは APF 許可を必要とします。
- ユーザー ID (そのユーザーの権限でチャネル・イニシエーターが実行されている) が UNIX システム・サービス (USS) を使用するように設定されている (設定方法については「z/OS UNIX システム・サービスの計画」の資料を参照してください)。

チャネル・イニシエーターは特別な許可を必要とせず、UNIX 内ではスーパーユーザーとして実行されないため、ゲスト/デフォルト UID および OMVS セグメントを使用して、チャネル・イニシエーターで UNIX システム・サービスを起動しないユーザーには、デフォルトのセグメントに基づいた新規 OMVS セグメントのモデル化のみが必要です。

z/OS *SSLTASKS パラメーターの設定 (z/OS)*

ALTER QMGR コマンドを使用して、TLS 呼び出しを処理するサーバー・サブタスクの数を設定します。

TLS チャネルを使用する場合は、ALTER QMGR コマンドを使用し、SSLTASKS パラメーターを設定することによって、2 つ以上のサーバー・サブタスクがあることを確認してください。 以下に例を示します。

```
ALTER QMGR SSLTASKS(5)
```

ストレージ割り振りの問題を避けるために、証明書取り消しリスト (CRL) 検査のない環境では、8 より大きな値に SSLTASKS 属性を設定しないでください。

CRL 検査が使用される場合、その検査期間にわたり該当するチャネルによって SSLTASK が保持されます。 それぞれの SSLTASK は z/OS タスク制御ブロックであるため、関連する LDAP サーバーへの接続中に経過時間がかなり長くなることがあります。

SSLTASKS 属性の値を変更した場合、チャネル・イニシエーターを再始動する必要があります。

z/OS *z/OS での鍵リポジトリーのセットアップ*

接続の両端で鍵リポジトリーをセットアップします。 それぞれの鍵リポジトリーにキュー・マネージャーを関連付けます。

TLS 接続では、接続の両端に鍵リポジトリー が必要です。 各キュー・マネージャーには、鍵リポジトリーへのアクセス権が必要です。 鍵リポジトリーをキュー・マネージャーに関連付けるには、ALTER QMGR コマンドで SSLKEYR パラメーターを使用します。 詳しくは、23 ページの『SSL/TLS 鍵リポジトリー』を参照してください。

z/OS では、外部セキュリティー・マネージャー (ESM) によって管理される鍵リング に、デジタル証明書が保管されます。 これらのデジタル証明書には、証明書をキュー・マネージャーに関連付けるラベルがあります。 TLS は、認証のためにこれらの証明書を使用します。 以下の例ではすべて RACF コマンドを使用します。 他の ESM プログラムにも同等のコマンドが存在します。

z/OS では、IBM MQ は、**CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmWebSphereMQ にキュー・マネージャーの名前を付加した値のどちらかを使用します。 詳細については、デジタル証明書ラベルを参照してください。

キュー・マネージャーの鍵リポジトリーの名前は、ご使用の RACF データベース内の鍵リングの名前です。 鍵リングの名前は、鍵リングの作成前または作成後のどちらでも指定できます。

キュー・マネージャー用に新しい鍵リングを作成するには、次の手順を使用してください。

1. RACDCERT コマンドを実行する適切な権限があることを確認する (詳しくは、「*SecureWay Security Server RACF* コマンド言語解説書」を参照)。
2. 以下のコマンドを発行します。

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

   ここで、

   - *userid1* は、チャネル・イニシエーター・アドレス・スペースのユーザー ID または鍵リングを所有するユーザー ID です (鍵リングが共有されている場合)。
   - *ring-name* は、鍵リングに指定したい名前です。 この名前の最大長は、237 文字です。 この名前は、大文字と小文字が区別されます。 問題を避けるために、*ring-name* は大文字で指定してください。

z/OS *z/OS で CA 証明書をキュー・マネージャーに対して有効にする操作*

鍵リングを作成したら、該当する CA 証明書をその鍵リングに接続します。

データ・セットに CA 証明書がある場合は、次のコマンドを使用して、まずその証明書を RACF データベースに追加する必要があります。

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

次に、My CA の CA 証明書を鍵リングに接続するには、次のコマンドを使用します。

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

ここで、*userid1* は、チャネル・イニシエーターのユーザー ID または共有鍵リングの所有者です。

CA 証明書の詳細については、9 ページの『デジタル証明書』を参照してください。

z/OS z/OS 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・マネージャーの鍵リングの位置を取得する手順を取り上げます。

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
   ```

2. コマンドの出力を調べて、鍵リングの位置を確認する。

z/OS キュー・マネージャーの鍵リポジトリーの位置の指定 (z/OS)

キュー・マネージャーの鍵リングの位置を指定するには、ALTER QMGR MQSC コマンドを使用して、キュー・マネージャーの鍵リポジトリー属性を設定します。

以下に例を示します。

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

鍵リングがチャネル・イニシエーター・アドレス・スペースによって所有されている場合、または

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

共有鍵リングである場合。ここで、*userid1* は鍵リングを所有するユーザー ID です。

z/OS チャネル・イニシエーターに正しいアクセス権を与える操作 (z/OS)

チャネル・イニシエーター (CHINIT) では、鍵リポジトリーと特定のセキュリティー・プロファイルに対するアクセス権が必要です。

鍵リポジトリーに対する読み取りアクセス権を CHINIT に与える操作

鍵リポジトリーを所有しているのが CHINIT ユーザー ID である場合は、このユーザー ID には FACILITY クラス内の IRR.DIGTCERT.LISTRING プロファイルに対する読み取りアクセス権が必要です。そうでない場合は、更新アクセス権が必要です。 それぞれのアクセス権を与えるには、ACCESS(UPDATE) または ACCESS(READ) を指定して PERMIT コマンドを実行します。

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
```

ここで *userid* は、チャネル・イニシエーター・アドレス・スペースのユーザー ID です。

該当する CSF* プロファイルに対する読み取りアクセス権を CHINIT に与える操作

Integrated Cryptographic Service Facility (ICSF) により提供されるハードウェア・サポートを使用するには、次のコマンドを使用して、CHINIT ユーザー ID に CSFSERV クラス内の適切な CSF* プロファイルへの読み取りアクセス権があることを確認します。

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

ここで、*csf-resource* は、 CSF* プロファイルの名前であり、*userid* は、チャネル・イニシエーター・アドレス・スペースのユーザー ID です。

次の CSF* プロファイルのそれぞれに対して、このコマンドを繰り返します。

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

CHINIT ユーザー ID には、他の CSF* プロファイルへの読み取りアクセスも必要になる場合があります。例えば、ECDHE_RSA_AES_256_GCM_SHA384 暗号仕様を使用している場合、CHINIT ユーザー ID には以下の CSF* プロファイルに対する読み取りアクセスも必要になります。

- CSF1DVK
- CSF1GAV
- CSF1GKP
- CSF1SKE
- CSF1TRC
- CSF1TRD

詳しくは、「RACF CSFSERV リソース要件」 を参照してください。

証明書の鍵が ICSF に保管されていて、ご使用のインストール済み環境に ICSF に保管されている鍵に対するアクセス制御が設定されている場合は、次のコマンドを使用して、CHINIT ユーザー ID に CSFKEYS クラス内のプロファイルへの読み取りアクセス権があることを確認します。

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

ここで *userid* は、チャネル・イニシエーター・アドレス・スペースのユーザー ID です。

Integrated Cryptographic Service Facility (ICSF) の使用

チャネル・イニシエーターでは、TLS が使用されていない場合に、パスワード保護アルゴリズムをシードしてクライアント・チャネルを介して流れるパスワードを難読化する際に、ICSF を使用して乱数を生成できます。

詳細については、260 ページの『Integrated Cryptographic Service Facility (ICSF) の使用』を参照してください。

z/OS *z/OS で証明書または鍵リポジトリーの変更が有効になる時点*

変更が有効になるのは、チャネル・イニシエーターを始動した時点またはリポジトリーをリフレッシュした時点です。

特に、鍵リングの証明書または鍵リポジトリー属性の変更が有効になるのは、以下のいずれかの時点です。

- チャネル・イニシエーターを始動/再始動したとき。
- REFRESH SECURITY TYPE(SSL) コマンドが発行されて、鍵リポジトリーの内容が最新表示されたとき。

z/OS z/OS での自己署名個人証明書の作成

自己署名個人証明書を作成する手順を取り上げます。

1. 次のコマンドを使用して、証明書、および公開鍵と秘密鍵のペアを生成する。

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
```

2. 次のコマンドを使用して、その証明書を鍵リングに接続する。

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

ここで、

- *userid1* は、チャネル・イニシエーター・アドレス・スペースまたは共有鍵リングの所有者のユーザー ID です。
- *userid2* は、証明書に関連したユーザー ID で、チャネル・イニシエーター・アドレス・スペースの ユーザー ID でなければなりません。

  userid1 と *userid2* は同じ ID にすることができます。
- *ring-name* は、316 ページの『z/OS での鍵リポジトリーのセットアップ』で鍵リングに指定した名前です。
- *label-name* は、IBM MQ **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmWebSphereMQ にキュー・マネージャーの名前を付加した値のどちらかでなければなりません。 詳細については、デジタル証明書ラベルを参照してください。

z/OS z/OS での個人証明書の要求

RACF を使用して個人証明書を申請します。

個人用証明書を申し込むには、次のように RACF を使用します。

1. 自己署名入り個人用証明書を作成する (319 ページの『z/OS での自己署名個人証明書の作成』を参照)。この証明書は、要求に、識別名の属性値を与えます。
2. 次のコマンドを使用して、データ・セットに書き込まれる PKCS #10 Base64-encoded 証明書要求を作成する。

```
RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
```

   説明:

   - *userid2* は、証明書に関連したユーザー ID で、チャネル・イニシエーター・アドレス・スペースのユーザー ID でなければなりません。
   - *label_name* は自己署名証明書の作成時に使用したラベルです。

   詳細については、25 ページの『デジタル証明書ラベルの要件に関する説明』を参照してください。
3. そのデータ・セットを認証局 (CA) に送信して、新しい個人用証明書を要求する。
4. 署名付きの証明書が認証局から返されたら、元のラベルを使用して、その証明書を RACF データベースに再び追加します (320 ページの『z/OS で鍵リポジトリーに個人証明書を追加する操作』を参照してください)。

z/OS *RACF* 署名入り個人用証明書の作成

RACF は、認証局の機能を果たし、独自の CA 証明書を発行することができます。

このセクションで使用される署名者証明書という用語は、RACF によって発行された CA 証明書を指します。

次の手順を実行する前に、署名者証明書の秘密鍵が、RACF データベース内に入っていなければなりません。

1. 次のコマンドを使用して、RACF データベースに含まれている署名者証明書を使用して、RACF によって署名される個人用証明書を生成する。

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
SIGNWITH(CERTAUTH LABEL('signer-label'))
```

2. 次のコマンドを使用して、その証明書を鍵リングに接続する。

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

ここで、

- *userid1* は、チャネル・イニシエーター・アドレス・スペースまたは共有鍵リングの所有者のユーザー ID です。
- *userid2* は、証明書に関連したユーザー ID で、チャネル・イニシエーター・アドレス・スペースの ユーザー ID でなければなりません。

  userid1 と *userid2* は同じ ID にすることができます。
- *ring-name* は、316 ページの『z/OS での鍵リポジトリーのセットアップ』で鍵リングに指定した名前です。
- *label-name* は、IBM MQ **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmWebSphereMQ にキュー・マネージャーかキュー共有グループの名前を付加した値のどちらかでなければなりません。 詳細は デジタル証明書ラベル を参照してください。
- *signer-label* は、独自の署名者証明書のラベルです。

z/OS *z/OS* で鍵リポジトリーに個人証明書を追加する操作

個人証明書を鍵リングに追加/インポートする手順を取り上げます。

認証局から新しい個人用証明書が送信された後、次の手順を使用して、その証明書を鍵リングに追加します。

1. 次のコマンドを使用して、RACF データベースに証明書を追加する。

```
RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
```

2. 次のコマンドを使用して、その証明書を鍵リングに接続する。

```
RACDCERT ID( userid1 )
CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
```

ここで、

- *userid1* は、チャネル・イニシエーター・アドレス・スペースまたは共有鍵リングの所有者のユーザー ID です。
- *userid2* は、証明書に関連したユーザー ID で、チャネル・イニシエーター・アドレス・スペースの ユーザー ID でなければなりません。
- *ring-name* は、316 ページの『z/OS での鍵リポジトリーのセットアップ』で鍵リングに指定した名前です。
- *input-data-set-name* は、CA によって署名された証明書を含むデータ・セットの名前です。 データ・セットはカタログされなければならず、PDS または PDS のメンバーであってはなりません。 RACDCERT で予期されるレコード・フォーマット (RECFM) は VB です。 RACDCERT はデータ・セットを動的に割り振ってから開き、データ・セットからバイナリー・データとして証明書を読み取ります。
- *label-name* は、最初の要求の作成時に使用したラベル名です。 これは、IBM MQ **CERTLABL** 属性が設定されている場合はその値、またはデフォルトの ibmWebSphereMQ にキュー・マネージャーかキュー共有グループの名前を付加した値のどちらかでなければなりません。 詳細は デジタル証明書ラベル を参照してください。

z/OS *z/OS で鍵リポジトリーから個人証明書をエクスポートする操作*

RACDCERT コマンドを使用して証明書をエクスポートします。

エクスポートする証明書が入っているシステムで以下のコマンドを実行します。

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

ここで、

- *userid2* は、証明書が鍵リングに追加された際に使用したユーザー ID です。
- *label-name* は、取り出したい証明書のラベルです。
- *output-data-set-name* は、証明書が置かれるデータ・セットです。
- CERTB64 は、Base64 形式の DER エンコード X.509 証明書です。 別の形式を選択することもできます。 例えば、次のようにします。

  CERTDER
  2 進形式の DER エンコード X.509 証明書

  PKCS12B64
  Base64 形式の PKCS #12 証明書

  PKCS12DER
  2 進形式の PKCS #12 証明書

z/OS *z/OS で鍵リポジトリーから個人証明書を削除する操作*

RACDCERT コマンドを使用して個人証明書を削除します。

個人用証明書を削除する場合は、事前にその証明書のコピーを保管する必要がある場合があります。 個人用証明書を削除する前に、データ・セットにコピーするには、321 ページの『z/OS で鍵リポジトリーから個人証明書をエクスポートする操作』の手順を実行してください。 その後、次のコマンドを使用して、個人用証明書を削除してください。

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

ここで、

- *userid2* は、証明書が鍵リングに追加された際に使用したユーザー ID です。
- *label-name* は、削除したい証明書の名前です。

z/OS *z/OS で鍵リポジトリーに含まれている個人証明書の名前を変更する操作*

RACDCERT コマンドを使用して証明書の名前を変更します。

特定のラベルを持つ証明書を検出したくはないが、その証明書を削除したくない場合は、次のコマンドを使用して、一時的にその証明書の名前を変更することができます。

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

ここで、

- *userid2* は、証明書が鍵リングに追加された際に使用したユーザー ID です。
- *label-name* は、名前を変更したい証明書の名前です。
- *new-label-name* は、新しい証明書名です。

これは、TLS クライアント認証をテストする際に有用です。

z/OS *z/OS でユーザー ID にデジタル証明書を関連付ける操作*

IBM MQ は、RACF 証明書に関連付けられているユーザー ID をチャネルのユーザー ID として使用できます。 ユーザー ID に証明書を関連付けるには、そのユーザー ID で証明書をインストールするか、証明書名フィルターを使用します。

このトピックで説明されている方式は、ユーザー ID をデジタル証明書と関連付けるための、プラットフォームに依存しない方式に代わるものであり、チャネル認証レコードを使用します。 チャネル認証レコードの詳細については、47 ページの『チャネル認証レコード』を参照してください。

TLS チャネルの一方の側のエンティティーが、リモート接続から証明書を受け取ると、そのエンティティーは、その証明書に関連したユーザー ID があるかどうか、RACF に問い合わせます。 そのエンティティーは、そのユーザー ID をチャネル・ユーザー ID として使用します。 証明書に関連したユーザー ID がない場合、エンティティーは、チャネル・イニシエーターが実行する際に使用されるユーザー ID を使用します。

以下のいずれかの方法でユーザー ID に証明書を関連付けます。

- その証明書を関連付けたいユーザー ID の下で、その証明書を RACF データベースにインストールする (320 ページの『z/OS で鍵リポジトリーに個人証明書を追加する操作』を参照)。
- Certificate Name Filter (CNF) を使用して、証明書の所有者または発行者の識別名を、ユーザー ID にマップする (322 ページの『証明書名フィルターのセットアップ (z/OS)』を参照)。

z/OS 証明書名フィルターのセットアップ *(z/OS)*

RACDCERT コマンドを使用して、証明書名フィルター (CNF) を定義します。CNF では、識別名とユーザー ID の対応関係を記述します。

CNF をセットアップするには、以下の手順を使用します。

1. 以下のコマンドを使用して CNF 機能を有効にします。 そのためには、DIGTNMAP クラスに対する更新権限が必要です。

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

2. CNF を定義する。 以下に例を示します。

   ```
   RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
   SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
   ```

   ここで、USER1 は、次の場合に使用されるユーザー ID です。

   - 所有者の DN には、組織名 IBM と国名 UK がある。
   - 発行者の DN には、組織名 ExampleCA と地域名 Internet がある。

3. CNF のマッピングをリフレッシュする。

   ```
   SETROPTS RACLIST(DIGTNMAP) REFRESH
   ```

注 :

1. 実際の証明書が RACF データベース内に保管されている場合、その証明書がインストールされる際に使用されるユーザー ID は、任意の CNF に関連付けられているユーザー ID よりも優先して使用されます。証明書が RACF データベース内に保管されていない場合、最も詳細なところまで一致する CNF に関連したユーザー ID が使用されます。 所有者 DN の一致は、発行者 DN の一致より詳細であると見なされます。
2. CNF に対する変更が適用されるのは、CNF のマッピングをリフレッシュした後です。
3. DN が、CNF 内の DN フィルターと一致するのは、DN フィルターが、DN の最下位部分 と等しい場合だけです。 DN の最下位部分は、通常、DN の右端にリストされ、証明書の先頭に表示される属性から構成されます。

   例えば、SDNFILTER `'O=IBM.C=UK'` の場合を考えてみましょう。 所有者の DN `'CN=QM1.O=IBM.C=UK'` は、そのフィルターと一致しますが、所有者の DN `'CN=QM1.O=IBM.L=Hursley.C=UK'` は、そのフィルターと一致しません。

   一部の証明書の最下位部分には、DN フィルターと一致しないフィールドが入る可能性があります。DEFINE CHANNEL コマンドの SSLPEER パターンで DN パターンを指定することによって、その証明書を除外することを検討してください。
4. 最も詳細に一致する CNF が、RACF に対して NOTRUST として定義される場合、エンティティーは、チャネル・イニシエーターが実行する際に使用されるユーザー ID を使用します。
5. RACF は、区切り文字として `'.'` 文字を使用します。 IBM MQ は、コンマかセミコロンのどちらかを使用します。

CNF を定義すると、エンティティーがチャネルのユーザー ID をデフォルトに設定しないことを確実にすることができます。このデフォルトは、チャネル・イニシエーターが実行する際に使用されるユーザー ID です。 エンティティーに関連した鍵リング内の CA 証明書ごとに、その CA 証明書の所有者 DN と正確に一致する IDNFILTER で、CNF を定義してください。 これにより、エンティティーが使用するすべての証明書が、これらの CNF の少なくとも 1 つと一致することが確実になります。 これは、こうした証明書がすべて、エンティティーに関連した鍵リングに接続されるか、エンティティーに関連した鍵リングに接続される証明書の CA によって発行されなければならないからです。

CNF の操作に使用するコマンドの詳細については、「*SecureWay Security Server RACF Security Administrator's Guide*」を参照してください。

z/OS *QMA での送信側チャネルと伝送キューの定義 (z/OS)*

DEFINE CHANNEL コマンドと **DEFINE QLOCAL** コマンドを使用して、必要なオブジェクトをセットアップします。

手順

QMA で以下の例のようなコマンドを実行します。

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

タスクの結果

送信側チャネル TO.QMB と伝送キュー QMB が作成されます。

z/OS *QMB での受信側チャネルの定義 (z/OS)*

DEFINE CHANNEL コマンドを使用して、必要なオブジェクトをセットアップします。

手順

QMB で以下の例のようなコマンドを実行します。

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

タスクの結果
受信側チャネル TO.QMB が作成されます。

z/OS *QMA での送信側チャネルの開始 (z/OS)*
必要に応じて、リスナー・プログラムを開始し、セキュリティーをリフレッシュします。 さらに、**START CHANNEL** コマンドを使用してチャネルを開始します。

手順

1. オプション: QMB でリスナー・プログラムをまだ開始していない場合は、開始します。
   リスナー・プログラムは、着信ネットワーク要求を listen し、必要に応じて受信側チャネルを開始します。 リスナーを開始する方法については、『チャネル・リスナーの開始』を参照してください。
2. オプション: SSL/TLS チャネルが以前に実行されている場合は、コマンド REFRESH SECURITY TYPE(SSL) を発行します。
   これにより、鍵リポジトリーに加えられたすべての変更が有効になります。
3. コマンド START CHANNEL(TO.QMB) を使用して、QMA でチャネルを開始します。

タスクの結果
送信側チャネルが開始されます。

z/OS 自己署名証明書の交換 (z/OS)
抽出した証明書を交換します。 FTP を使用する場合は、正しい形式を使用する必要があります。

手順

FTP などで QM1 証明書の CA 部分を QM2 システムに転送し、QM2 証明書の CA 部分を QM1 システムに転送します。

FTP を使用して証明書を転送する場合は、正しい形式を使用する必要があります。

次の証明書タイプは、*2* 進形式 で転送します。

- DER エンコード 2 進 X.509
- PKCS #7 (CA 証明書)
- PKCS #12 (個人用証明書)

以下の証明書タイプは、ASCII フォーマットで転送します。

- PEM (Privacy-Enhanced Mail)
- Base64 エンコード X.509

z/OS *QM1 での送信側チャネルと伝送キューの定義 (z/OS)*
DEFINE CHANNEL コマンドと **DEFINE QLOCAL** コマンドを使用して、必要なオブジェクトをセットアップします。

手順

QM1 で以下の例のようなコマンドを実行します。

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')
```

```
DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

チャネルの両端の CipherSpec は同じでなければなりません。

チャネルが TLS を使用するようにしたい場合は、SSLCIPH パラメーターだけが必須です。 SSLCIPH パラメーターに指定できる値については、38 ページの『IBM MQ での CipherSpec と CipherSuite』を参照してください。

タスクの結果
送信側チャネル QM1.TO.QM2 と伝送キュー QM2 が作成されます。

z/OS *QM2 での受信側チャネルの定義 (z/OS)*
DEFINE CHANNEL コマンドを使用して、必要なオブジェクトをセットアップします。

手順
QM2 で以下の例のようなコマンドを実行します。

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

このチャネルの名前は、324 ページの『QM1 での送信側チャネルと伝送キューの定義 (z/OS)』で作成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。

z/OS *QM1 での送信側チャネルの開始 (z/OS)*
必要に応じて、リスナー・プログラムを開始し、セキュリティーをリフレッシュします。 さらに、**START CHANNEL** コマンドを使用してチャネルを開始します。

手順
1. オプション: QM2 でリスナー・プログラムをまだ開始していない場合は、開始します。
   リスナー・プログラムは、着信ネットワーク要求を listen し、必要に応じて受信側チャネルを開始します。 リスナーを開始する方法については、『チャネル・リスナーの開始』を参照してください。
2. オプション: SSL/TLS チャネルを実行したことがある場合は、コマンド REFRESH SECURITY TYPE(SSL) を実行します。
   これにより、鍵リポジトリーに加えられたすべての変更が有効になります。
3. コマンド START CHANNEL(QM1.TO.QM2) を使用して、QM1 でチャネルを開始します。

タスクの結果
送信側チャネルが開始されます。

z/OS *SSL 環境または TLS 環境のリフレッシュ (z/OS)*
REFRESH SECURITY コマンドを使用して、キュー・マネージャー QMA の TLS 環境をリフレッシュします。

手順
QMA で、次のコマンドを入力します。

```
REFRESH SECURITY TYPE(SSL)
```

これにより、鍵リポジトリーに加えられたすべての変更が有効になります。

z/OS 受信側チャネルで匿名の接続を許可する操作 (z/OS)

ALTER CHANNEL コマンドを使用して、SSL または TLS クライアント認証をオプションにします。

手順

QMB で、次のコマンドを入力します。

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

z/OS *QM1* での送信側チャネルの開始 (z/OS)

必要に応じて、チャネル・イニシエーターを開始し、リスナー・プログラムを開始し、セキュリティーをリフレッシュします。 さらに、**START CHANNEL** コマンドを使用してチャネルを開始します。

手順

1. オプション: チャネル・イニシエーターをまだ開始していない場合は、開始します。
2. オプション: QM2 でリスナー・プログラムをまだ開始していない場合は、開始します。

   リスナー・プログラムは、着信ネットワーク要求を listen し、必要に応じて受信側チャネルを開始します。 リスナーを開始する方法については、『チャネル・リスナーの開始』を参照してください。
3. オプション: チャネル・イニシエーターを実行していた場合や SSL/TLS チャネルを実行したことがある場合は、コマンド REFRESH SECURITY TYPE(SSL) を実行します。

   これにより、鍵リポジトリーに加えられたすべての変更が有効になります。
4. コマンド START CHANNEL(QM1.TO.QM2) を使用して、QM1 でチャネルを開始します。

タスクの結果

送信側チャネルが開始されます。

z/OS *QMA* での送信側チャネルの開始 (z/OS)

必要に応じて、チャネル・イニシエーターを開始し、リスナー・プログラムを開始し、セキュリティーをリフレッシュします。 さらに、**START CHANNEL** コマンドを使用してチャネルを開始します。

手順

1. オプション: チャネル・イニシエーターをまだ開始していない場合は、開始します。
2. オプション: QMB でリスナー・プログラムをまだ開始していない場合は、開始します。

   リスナー・プログラムは、着信ネットワーク要求を listen し、必要に応じて受信側チャネルを開始します。 リスナーを開始する方法については、『チャネル・リスナーの開始』を参照してください。
3. オプション: チャネル・イニシエーターを実行していた場合や SSL/TLS チャネルを実行したことがある場合は、コマンド REFRESH SECURITY TYPE(SSL) を実行します。

   これにより、鍵リポジトリーに加えられたすべての変更が有効になります。
4. コマンド START CHANNEL(TO.QMB) を使用して、QMA でチャネルを開始します。

タスクの結果

送信側チャネルが開始されます。

z/OS z/OS での楕円曲線鍵の長さの変更

使用の優先順位順にリストされた 1 つまたは複数の 4 文字の数値で構成される文字列でクライアントによって指定されている楕円曲線またはサポートされているグループのリストを設定するための GSK_CLIENT_ECURVE_LIST 環境変数を変更する方法について説明します。

重要 : z/OS APAR OA61783 のフィックスを適用して、TLS 1.0、TLS 1.1 、および/または TLS 1.2 ネゴシエーション接続を使用する際に、特定の楕円曲線がオペレーティング・システムによって有効になることを許可する必要があります。

この TLS 環境変数は、CEEOPTS DD ステートメントを使用してチャネル・イニシエーターの始動 JCL で設定できます。

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

上述のデータ・セットに、 次のように使用するリストを指定します。

```
ENVAR("GSK_CLIENT_ECURVE_LIST=002300240025")
```

重要 : この CEEOPTS ステートメントはインストリーム・データと一緒に使用しないでください。このステートメントを使用するすべての TLS タスクに対して環境変数が設定されなくなります。

順次データ・セット、または区画データ・セット・メンバーを参照するようにすることで、このステートメントが 1 より大きい SSLTASKS 値を使用する場合に動作するようにしてください。

GSK_CLIENT_ECURVE_LIST (GSK_SERVER_ALLOWED_KEX_ECURVES) と同等のサーバー・アナログを使用することもできます。 詳しくは、 鍵交換楕円曲線の制限 を参照してください。

さらに、有効な 4 文字の楕円曲線とサポートされるグループの指定のリストについては、 暗号スイートの定義 の表 5 を参照してください。

デフォルト指定は 00210023002400250019 です。 TLS V1.3 が使用可能になっている場合、 0029 (x25519) がデフォルトのリストの末尾に付加されています。

ユーザーの識別および認証

X.509 証明書、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。

X.509 証明書の使用

x.509 証明書を **CHLAUTH** コマンドおよび **SSLPEER** パラメーターと共に使用して、ユーザーを識別および認証することができます。 **SSLPEER** パラメーターは、 チャネルの反対側にあるピアのキュー・マネージャーまたはクライアントの証明書のサブジェクト識別名と比較するために使用するフィルターを指定します。

CHLAUTH コマンドおよび **SSLPEER** パラメーターの使用について詳しくは、 SET CHLAUTH を参照してください。

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。

警告 : クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを経由してプレーン・テキストで送信される場合があります。 クライアント・アプリケーションのパスワードが適切に 保護されるようにするには、28 ページの『MQCSP パスワード保護』を参照してください。

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、IBM 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、IBM MQ MQI client・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。 詳しくは、150 ページの『SSPI チャネル出口プログラム (Windows)』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。

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。 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『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、444 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、115 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、333 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。

プラグ可能認証方式 (PAM)

Linux UNIX

PAM は現在 UNIX and Linux プラットフォームにおいて一般的になっており、ユーザー認証の詳細情報をサービスから隠す一般的なメカニズムを提供します。

規則を構成すれば、サービス自体を変更しなくても、さまざまなサービスにさまざまな認証規則を使用できます。

詳しくは、346 ページの『プラグ可能認証方式 (PAM) の使用法』を参照してください。

特権ユーザー

特権ユーザーは、IBM MQ の全管理権限を付与されたユーザーです。

キュー・マネージャーの整合性とセキュリティーを確保するため、以下の表にリストされているユーザーに加えて、アクセス権限を付与する場合に格別の注意を払う必要がある特定の対象と権限があります。 以下のいずれかの権限を付与する場合は、追加の検査を適用する必要があります。

- SYSTEM オブジェクトに対する権限
- オブジェクトの作成、変更、および削除を行うための管理権限。

  z/OS z/OS では、この権限は DEFINE、ALTER、および DELETE コマンドを発行するためのコマンド・セキュリティー権限およびコマンド・リソース・セキュリティー権限です。

  Multi 他のすべてのプラットフォームでは、これらの権限は `+crt`、`+chg`、`+dlt` などの管理権限です。
- キューを消去するための管理権限。

  z/OS z/OS では、この権限は CLEAR コマンドを発行するためのコマンド・セキュリティー権限およびコマンド・リソース・セキュリティー権限です。

  Multi 他のすべてのプラットフォームでは、この権限は `+clr` です。
- チャネルの停止、メッセージのバックアウトまたはコミットを行うための管理権限。

  z/OS z/OS では、この権限は RESET CHANNEL、START CHANNEL、STOP CHANNEL などのコマンドを発行するためのコマンド・セキュリティー権限またはコマンド・リソース・セキュリティー権限です。

  Multi 他のすべてのプラットフォームでは、これらの権限は `+ctrl` および `+ctrlx` です。
- アプリケーションで許可検査の特権をエスカレーションできるようにする代替ユーザー MQI 権限。

  z/OS z/OS では、この権限は代替ユーザー・セキュリティー・プロファイルに付与される任意の権限です。

  Multi 他のすべてのプラットフォームでは、この権限は `+altusr` です。
- アプリケーションでメッセージのセキュリティー・コンテキストを変更できるようにするコンテキスト権限。

  z/OS z/OS では、この権限はコンテキスト・セキュリティー・プロファイルに付与される任意の権限です。

Multi 他のすべてのプラットフォームでは、これらの権限は +setall および +setid です。

一般的なプリンシパルとして、メッセージング・アプリケーションには、必要なキューまたはトピックに対する基本 MQI 権限のみを付与する必要があります。 非特権ユーザー MCAUSER の下で実行される MCA チャネル、および送達不能キュー・ハンドラーなどのその他の特殊タイプのアプリケーションには、正常に作動するために通常アプリケーションには付与されない追加の権限が必要な場合があります。

表 67. プラットフォーム別の特権ユーザー

プラットフォーム	特権ユーザー
Windows システム	• SYSTEM • mqm グループのメンバー • 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
IBM i IBM i IBM i システム	• プロファイル qmqm および qmqmadm • qmqmadm グループのすべてのメンバー • *ALLOBJ を設定して定義されたすべてのユーザー
z/OS	チャネル・イニシエーター、キュー・マネージャー、拡張メッセージ・セキュリティーのアドレス・スペースの実行に使用されているユーザー ID。 これらのユーザー ID は IBM MQ に対する完全な管理権限を自動的には持ちませんが、通常これらのユーザー ID に付与されるアクセス・レベルの程度を考慮して、特権を持つと見なされます。

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。

MQCSP はセキュリティー出口で変更することができます。

警告 : クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを経由してプレーン・テキストで送信される場合があります。 クライアント・アプリケーションのパスワードが適切に 保護されるようにするには、28 ページの『MQCSP パスワード保護』を参照してください。

MQCSP と AdoptCTX 設定の関係

IBM MQ は、接続認証機能が有効になっていない場合を除き、常に MQCSP 構造を介して渡される資格情報を認証します。 信任状が正常に認証されると、IBM MQ は、ADOPTCTX が使用可能になっていない限り、将来の許可検査のためにユーザー ID を採用しようとします。

IBM MQ には、許可検査のためにユーザーが使用できるユーザー ID の長さの制限があります。 これらの制限は、81 ページの『ユーザー ID』で詳細化されています。 MQCSP 構造を介して渡されたユーザー ID を採用する場合、IBM MQ の動作は他の設定オプションによって異なります。

- LDAP 接続認証を使用する場合、 IBM MQ は、SHORTUSR で設定されたフィールドの値をそのユーザーの LDAP レコードから取得し、そのユーザー ID を採用します。

  例えば、SHORTUSR が 'CN' に設定され、LDAP レコードでユーザーが 'CN=Test,SN=MQ,O=IBM,C=UK' としてリストされる場合、ユーザー ID Test が使用されます。

- OS 接続認証または PAM 認証を使用する場合、ADOPTCTX が YES であれば、MQCSP 構造を介して渡されるユーザー ID は、接続コンテキストとして採用されたときの IBM MQ の 12 文字のユーザー ID 制限を満たすために切り捨てられます。

  ChlAuthEarlyAdopt が有効になっている場合、ユーザー資格情報が認証された後に切り捨てが行われます。

  ChlAuthEarlyAdopt が有効になっていない場合、切り捨ては採用前に行われます。 Windows では、ユーザーが `user@domain` の形式で指定された場合、これは、ユーザーが 12 文字未満のときにドメイン指定が無効になる可能性があることを意味します。

  例えば、ユーザー `ibmmq@windowsdomain` が MQCSP を介して提供される場合、このシナリオでは `ibmmq@window` に切り捨てられます。 これにより、以下のエラーが発生します。

  ```
  AMQ8074W: SID「SID」がエンティティー「ibmmq@window」と一致しないため、許可が失敗しました。
  ```

  これに基づいて、 `user@domain` の形式の Windows ドメイン・ユーザー ID など、12 文字より長いユーザー ID を MQCSP を介して渡す場合は、このエラーを回避するために qm.ini ファイルで **ChlAuthEarlyAdopt**=*Y* を構成する必要があります。

  あるいは、CONNAUTH AUTHINFO 構成で ADOPTCTX (NO) を使用し、CHLAUTH USERMAP 規則、セキュリティー出口、またはチャネル・オブジェクト MCAUSER 設定などの代替方法を使用して、チャネルのユーザー ID を設定します。

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、IBM MQ MQI client・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、IBM MQ MQI client・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、150 ページの『SSPI チャネル出口プログラム (Windows)』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。

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『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、444 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、115 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、333 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。

TLS 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。

以下のプラットフォームでは、IBM MQ SSL サポートによって、OCSP (Online Certificate Status Protocol) または LDAP (Lightweight Directory Access Protocol) サーバー上の CRL と ARL に基づいて、失効した証明書があるかどうか検査されます。 OCSP が推奨される方法です。

- Linux
- UNIX
- Windows

IBM MQ classes for Java および IBM MQ classes for JMS では、クライアント・チャネル定義テーブル・ファイルの 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 の使用を参照)。

以下のプラットフォームでは、IBM MQ SSL サポートによって、LDAP サーバー上の CRL と ARL のみに基づいて、失効した証明書があるかどうか検査されます。

- IBM i
- z/OS

認証局について詳しくは、9 ページの『デジタル証明書』を参照してください。

OCSP/CRL 検査

リモート着信証明書に対して OCSP (Online Certificate Status Protocol)/CRL (証明書失効リスト) 検査が実行されます。 このプロセスでは、リモート・システムの個人証明書からそのルート証明書までのチェーン全体が検査されます。

openSSL を使用した OCSP 検査の検証

自社で OCSP の検証に openSSL を使用している場合、GSKit TLS 接続の使用を試みると、不明 (unknown) 状況の警告を受け取ります。

これは、root 以外のチェーン内のすべての証明書が GSKit によって失効状況を検査されるためです。 GSKit の操作は RFC 5280 に従っており、これについては GSKit トラスト・ポリシーに記述されています。 GSKit アルゴリズムは、RFC 5280 および GSKit トラスト・ポリシーに記述されているとおり、使用可能なすべてのソースについて失効情報があるかどうかを確認しようとします。

IBM MQ の OCSP/CRL 検査の仕組み

IBM MQ は、指定された OCSP または CRL のエンドポイントで証明書を検査する時の動作を制御するメカニズムとして、証明書の拡張で制御する方法と、AUTHINFO オブジェクト内にある定義に基づいて制御する方法の 2 つをサポートします。

- qm.ini ファイルの SSL スタンザの **OCSPCheckExtensions** 属性、 **CDPCheckExtensions** 属性、および **OCSPAuthentication** 属性、および
- キュー・マネージャーの SSLCRLNL パラメーターと、AUTHINFO OCSP 構成と CRLLDAP 構成の使用。 詳細については、ALTER AUTHINFO および ALTER QMGR を参照してください。

重要 :

AUTHTYPE(OCSP) を指定した ALTER AUTHINFO コマンドは、IBM i と z/OS のキュー・マネージャーでの使用には適用されません。 しかし、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。

SSL スタンザの属性 **OCSPCheckExtensions** と **CDPCheckExtensions** は、IBM MQ が、証明書の AIA 拡張の中で詳細情報が記述されている OCSP サーバーや CRL サーバーで証明書を検証するかどうかを制御します。

それが有効になっていないと、証明書の拡張で指定されている OCSP サーバーや CRL サーバーへのアクセスは行われません。

OCSP サーバーや CRL サーバーの詳細情報が AUTHINFO オブジェクトに記述されていて、SSLCRLNL の **QMGR** 属性を使用して参照されている場合は、証明書失効のプロセス中に IBM MQ がそれらのサーバーにアクセスしようとします。

重要 : SSLCRLNL 名前リストで定義できるのは、1 つの OCSP AUTHINFO オブジェクトだけです。

次の場合

OCSPCheckExtensions=NO と **CDPCheckExtensions**=*NO* が設定されている
AUTHINFO オブジェクトで OCSP サーバーも CRL サーバーも定義されていない

この場合、証明書失効検査が実行されません。

証明書の失効状況の検査時に、指定されている OCSP サーバーや CRL サーバーに IBM MQ がアクセスする順序は以下のとおりです (ただし、関連する設定が有効になっている場合)。

1. **AUTHTYPE(OCSP)** オブジェクトで詳細情報が記述されていて、SSLCRLNL の **QMGR** 属性で参照されている OCSP サーバー。
2. 証明書の AIA 拡張で詳細情報が記述されている OCSP サーバー (**OCSPCheckExtensions**=*YES* の場合)。
3. 証明書の **CRLDistributionPoints** 拡張で詳細情報が記述されている CRL サーバー (**CDPCheckExtensions** =*YES* の場合)。
4. **AUTHINFO(CRLLDAP)** オブジェクトで詳細情報が記述されていて、SSLCRLNL の **QMGR** 属性で参照されている CRL サーバー。

証明書の検査時に、どれかのステップの結果として OCSP サーバーまたは CRL サーバーから証明書の照会に REVOKED または VALID の確定的な応答が返されると、それ以上の検査は実行されません。また、提示された証明書の状況に基づいて、その証明書を信頼するかしないかが決まります。

OCSP サーバーまたは CRL サーバーから UNKNOWN という結果が返された場合は、OCSP サーバーまたは CRL サーバーから確定的な結果が返されるか、すべてのオプションの検査が終わるまで、処理が続きます。

状況を確定できない場合に証明書を失効状態と見なすかどうかの動作は、OCSP サーバーの場合と CRL サーバーの場合とで異なります。

- CRL サーバーの場合は、CRL を取得できないと、証明書が NOT_REVOKED と見なされます。
- OCSP サーバーの場合は、指定された OCSP サーバーから失効状況を取得できないと、qm.ini ファイルの SSL スタンザの **OCSPAuthentication** 属性によって動作が制御されます。

  この属性は、接続のブロック、接続の許可、警告メッセージ付きでの接続の許可のいずれかに構成できます。

必要であれば、qm.ini ファイルと mqclient.ini ファイルの SSL スタンザにある **SSLHTTPProxyName**=*string* 属性を OCSP 検査のために使用できます。 このストリングは、OCSP 検査のために GSKit で使用する HTTP プロキシー・サーバーのホスト名またはネットワーク・アドレスのいずれかです。

IBM MQ 9.1.5 以降、失効検査の実行時に OCSP レスポンダーを待機する秒数を設定する **OCSPTimeout** 値を、 qm.ini ファイルまたは mqclient.ini ファイルの SSL スタンザに設定できます。

ULW 失効した証明書および OCSP

IBM 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、 UNIX, Linux, and Windows システム上の IBM MQ にのみ適用されます。

IBM 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 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 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、IBM 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 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、この属性は以下のいずれかの場所に格納されています。

- Linux UNIX `qm.ini` 上の UNIX and Linux ファイルの SSL スタンザ内。
- Windows Windows のレジストリー。

その属性を設定するには、IBM MQ Explorer を使用できます。 クライアントでは、属性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、IBM 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、IBM 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 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。TLS 接続を確立するために既に実行したステップで OCSP 応答を十分に検証できます。
- 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、TLS 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。302 ページの『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 (UNIX, Linux, and Windows)』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、IBM 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 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オンライン証明書状況プロトコル (OCSP)

Java API の制約により、IBM MQ が 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

プロパティー名	説明
	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、TLS 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

IBM 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。

IBM 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 IBM MQ MQI client 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。 IBM 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、IBM MQ は *SSLCRLNL* 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、IBM 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 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。 IBM MQ TLS 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。

340 ページの図 20 は、CA1 によって発行された CRL と ARL をロードするために LDAP サーバーへの入力として作成される可能性があるサンプル LDIF ファイルを示します。これは識別名 " CN=CA1 、OU=Test、O= IBM、C=GB " の仮想認証局です。IBM 内のテスト組織によってセットアップされます。

```
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 *20.* 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。

340 ページの図 21 は、340 ページの図 20 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。

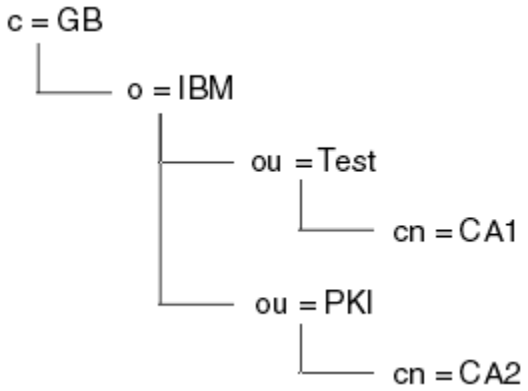


図 *21. LDAP Directory Information Tree* 構造例

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 341 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。 IBM i IBM i 上の IBM MQ の動作は、他のプラットフォームとは異なります。

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。

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、*SSLCRLNL* 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。 IBM i IBM i では、CRTMQMAUTI CL コマンドも使用できます。

   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。

   z/OS さらに、z/OS の場合のみ、すべての LDAP サーバーへのアクセスには同じユーザー ID およびパスワードを使用する必要があります。 ユーザー ID およびパスワードは、名前リストの最初の AUTHINFO オブジェクトに指定されたものを使用します。

   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。 z/OS z/OS では、NLTYPE 名前リスト属性が AUTHINFO に設定されていることを確認します。
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、*SSLCRLNL* 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。

IBM i IBM i では、認証情報オブジェクトを指定できますが、キュー・マネージャーは認証情報オブジェクトも認証情報オブジェクトの名前リストも使用しません。 IBM MQ キュー・マネージャーによって生成されたクライアント接続テーブルを使用する IBM i クライアントのみが、その IBM i キュー・マネージャーに指定された認証情報を使用します。 IBM i の *SSLCRLNL* キュー・マネージャー属性は、このようなクライアントが使用する認証情報を決定します。 IBM i キュー・マネージャーに CRL へのアクセス方法を指示する方法については、 342 ページの『IBM i での CRL および ARL へのアクセス』 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、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 i *IBM i での CRL および ARL へのアクセス*

この手順を使用して、IBM i 上の CRL または ARL にアクセスします。

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。

IBM i で特定の証明書に対する CRL ロケーションをセットアップするには、次のステップに従います。

1. DCM インターフェースにアクセスする (271 ページの『DCM へのアクセス』 を参照)。
2. ナビゲーション・パネルの「**Manage CRL locations (CRL ロケーションの管理)**」 タスク・カテゴリーで、「**Add CRL location (CRL ロケーションを追加)**」をクリックする。 タスク・フレームに「Manage CRL Locations (CRL ロケーションの管理)」ページが表示されます。
3. 「**CRL Location Name (CRL ロケーション名)**」 フィールドに、CRL ロケーション名 (例: LDAP Server #1) を入力します。
4. 「**LDAP Server (LDAP サーバー)**」フィールドに LDAP サーバー名を入力する。
5. TLS を使用して LDAP サーバーに接続する場合は、「**Use Secure Sockets Layer (SSL) (SSL (Secure Sockets Layer) を使用)**」フィールドで「**Yes (はい)**」を選択する。 それ以外の場合は「**No (いいえ)**」を選択する。
6. 「**Port Number (ポート番号)**」フィールドに LDAP サーバーの ポート番号を入力する (例: 389)。
7. 使用している LDAP サーバーで 匿名ユーザーがディレクトリーを照会できない場合は、サーバーのログイン識別名を 「**login distinguished name (ログイン識別名)**」フィールドに入力する。
8. **OK** をクリックします。 DCM から、CRL ロケーションを作成したことが通知されます。
9. ナビゲーション・パネルで、「**Select a Certificate Store (証明書ストアの選択)**」をクリックする。 タスク・フレームに「Select a Certificate Store (証明書ストアの選択)」ページが表示されます。
10. 「**Other System Certificate Store (他のシステム証明書ストア)**」チェック・ボックスを選択し、「**Continue (続行)**」をクリックする。 「Certificate Store and Password (証明書ストアおよびパスワード)」ページが表示されます。
11. 「**Certificate store path and filename (証明書ストアのパスおよびファイル名)**」フィールドに、272 ページの『IBM i での証明書ストアの作成』 で設定した IFS パスおよびファイル名を入力する。
12. 「**Certificate Store Password (証明書ストア・パスワード)**」フィールドに パスワードを入力する。「**Continue (続行)**」をクリックする。 タスク・フレームに「Current Certificate Store (現在の証明書ストア)」ページが表示されます。
13. ナビゲーション・パネルの「**Manage Certificates (証明書の管理)**」タスク・カテゴリーで、「**Update CRL location assignment (CRL ロケーション割り当ての更新)**」をクリックする。 タスク・フレームに「CRL Location Assignment (CRL ロケーション割り当て)」ページが表示されます。
14. CRL ロケーションを割り当てる CA 証明書のラジオ・ボタンを選択する。 「**Update CRL Location Assignment (CRL ロケーション割り当ての更新)**」をクリックする。 タスク・フレームに「Update CRL Location Assignment (CRL ロケーション割り当ての更新)」ページが表示されます。

15. 証明書に割り当てる CRL ロケーションのラジオ・ボタンを選択する。 「**Update Assignment (割り当てを更新)**」をクリックする。 DCM から、割り当てを更新したことが通知されます。

DCM では、認証局ごとに異なる LDAP サーバーを割り当てることができます。

IBM MQ Explorer を使用した *CRL* および *ARL* へのアクセス

IBM 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 343 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 343 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 MQ MQI client を使用した *CRL* および *ARL* へのアクセス

IBM MQ MQI client による検査のための CRL が格納されている 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 IBM MQ MQI client 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 MQ MQI client・システムでは、OCSP レスポンダーの位置、および証明書取り消しリスト (CRL) を保持する Lightweight Directory Access Protocol (LDAP) サーバーの位置を指定できます。

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。

IBM i IBM i については、IBM i での CRL および ARL へのアクセスを参照してください。

IBM MQ MQI client ・アプリケーションで、MQCONNX 呼び出しが発行される場合

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、IBM MQ MQI client 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

IBM MQ MQI client 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。

z/OS IBM i AUTHTYPE(OCSP) を指定した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) にコピーされるプラットフォーム上で指定することができます。

IBM MQ MQI client 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。

Multi

サーバー・プラットフォーム上では、AIX 、Linux、 IBM i 、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 **SSLCRLNL** 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、IBM MQ MQI client 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 **SSLCRLNL** 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 IBM MQ MQI client 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 **SSLCRLNL** 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で異なる証明書取り消し場所が必要で、かつ、証明書取り消し場所を作成するサーバー・キュー・マネージャーを使用する場合は、以下のようにします。

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。
4. クライアント・マシンで、**-n** パラメーターを指定して **runmqsc** コマンドを使用できます。

Multi

クライアント・プラットフォーム AIX、Linux、 IBM i、 Solaris、および Windows

runmqsc コマンドに **-n** パラメーターを指定し、CCDT ファイル内の **DEFINE AUTHINFO** オブジェクトを使用することにより、クライアント・マシン上で CCDT を作成できます。 オブジェクトが定義される順番は、それらがファイルで使用される順番になります。 **DEFINE AUTHINFO** オブジェクトで使用する名前は、ファイルでは保持されません。 CCDT ファイルで **AUTHINFO** オブジェクトの **DISPLAY** を実行する場合、定位置番号のみが使用されます。

注 : -n パラメーターを指定する場合は、その他のパラメーターを指定しないでください。

Windows での Active Directory の使用

Windows

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 MQ classes for Java および *IBM MQ classes for JMS* を使用した *CRL* および *ARL* へのアクセス

IBM MQ classes for Java および IBM MQ classes for JMS の CRL へのアクセスは、他のプラットフォームとは異なります。

IBM MQ classes for Java で CRL と ARL を操作するための情報については、証明書失効リストの使用を参照してください。

IBM MQ classes for JMS で CRL と ARL を操作するための情報については、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 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については、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な説明については、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。

それが使用可能なプラットフォームでは、IBM MQ Explorer も使用できます。

Linux UNIX プラグ可能認証方式 (PAM) の使用法

PAM は UNIX and Linux プラットフォームのみで使用できます。 典型的な UNIX システムには、従来型の認証メカニズムを実装している PAM モジュールがありますが、それ以上を行える場合もあります。 パスワードの検証という基本的な作業に加えて、追加の規則を実行するために PAM モジュールを呼び出すこともできます。

構成ファイルは、アプリケーションごとに使用する認証方式を定義します。 例のアプリケーションには、標準的な端末ログインの FTP と Telnet が組み込まれています。

PAM の利点には、実際にユーザー ID が認証される方法をアプリケーションが認識したり調べたりする必要がないことがあります。 アプリケーションが正しい形式の認証データを PAM に提供できる限り、背後のメカニズムは透過的です。

認証データの形成は、使用しているシステムに応じて異なります。 例えば、IBM MQ は、MQCONNX API 呼び出しで使用される MQCSP 構造などのパラメーターを介してパスワードを取得します。

重要 : IBM MQ 8.0.0 Fix Pack 3 をインストールしてから **-e CMDLEVEL=**レベル の *802* (strmqm コマンド) を使用して必要なコマンド・レベルを設定し、キュー・マネージャーを再始動するまで、 **AUTHENMD** 属性を設定することはできません。

PAM を使用するようにシステムを構成する

PAM の呼び出し時に IBM MQ によって使用されるサービス名は *ibmmq* です。

なお、IBM MQ のインストールはデフォルトの PAM 構成を維持しようとするので、さまざまなオペレーティング・システムの既知のデフォルトに基づいて、オペレーティング・システム・ユーザーからの接続が可能です。

ただし、システム管理者は、`/etc/pam.conf`、または `/etc/pam.d/ibmmq` で定義されているルールが依然として適切であることを確認する必要があります。

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。

ULW 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。

このセクションには、各プラットフォームのシステムにセキュリティーを適用する場合に実行する作業を確認できるチェックリストや、IBM MQ の管理権限および IBM MQ オブジェクトの操作権限をユーザーに付与する場合の注意点も含まれています。

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。

許可に使用されるユーザーの判別

リソースにアクセスする権限は、ユーザーがメンバーになっているグループに付与されるか、特定のモードで、接続に関連付けられているユーザーに直接付与されます。 接続プロセス中、特にリモート (クライアント) 接続の場合、この ID はキュー・マネージャーの構成によって変更される可能性があります。 このページには、接続アプリケーションの ID に影響を与える可能性がある IBM MQ のさまざまな機能とその構成オプション、およびそれらの機能が有効になる優先順位がリストされています。

どのユーザーを採用するかを変更できる機能

どのユーザーを許可するかを設定できるさまざまな機能は、以下のとおりです。

アプリケーションが表明したユーザー
IBM MQ によってリモート接続が開始されると、プロセスが実行されているオペレーティング・システム・ユーザーが受信側キュー・マネージャーに送信されます。 このユーザーは、ユーザーを変更する構成がこれ以上存在しない場合に、許可検査に使用できるユーザーが存在することを確認するために送信されます。

このユーザーを許可の基礎として使用することはお勧めしません。これにより、サーバー・サイドの検証なしで接続が ID を表明できるためです。 これには、管理ユーザー ('mqm ') が含まれる場合もあります。

チャネル MCAUSER 設定
ネットワーク・バインディングを介して接続するアプリケーションは、IBM MQ チャネル定義を使用してこれを行います。 チャネル定義は、 **MCAUSER** 属性をサポートします。この属性は、接続アプリケーションによって表明されたユーザーではなく、許可に使用される別のユーザーを指定するために使用できます。

接続認証 ADOPTCTX
アプリケーションは、認証のためにキュー・マネージャーに送信するユーザーとパスワードを指定できます。 これらの資格情報は、接続認証機能に指定された構成を使用して認証されます。 接続認証の **ADOPTCTX** オプションは、ユーザーが正常に検証された後に、そのユーザーを許可に使用するかどうかを制御します。 YES に設定すると、認証用に指定されたユーザーが許可検査に採用されます。

チャネル認証レコード MCAUSER
接続処理中に、キュー・マネージャーは、接続に一致するチャネル認証レコードを見つけようとします。 チャネル認証レコードが一致し、その **USERSRC** 属性値が MAP に設定されている場合、 IBM MQ は、許可に使用されるユーザーを **MCAUSER** 属性の値に変更します。

セキュリティー出口
セキュリティー出口は、IBM MQ セキュリティー処理中に作成して呼び出すことができるカスタム関数です。 この関数が呼び出されると、MQCD 構造体のコピーが提供されます。このコピーには、許可検査に使用される接続ユーザーに関連するいくつかのフィールドが含まれています。 セキュリティー出口は、これらのフィールドを変更して、許可されるユーザーを変更することができます。

優先順位

以下の表は、 IBM MQ が許可するユーザーを選択するときの、 347 ページの『どのユーザーを採用するかを変更できる機能』 で説明されている各セキュリティー機能の優先順位を示しています。 順序は、最も低いものから最も高いものの順になります。つまり、最初の行にユーザーを設定するセキュリティー機能は、他のいずれかの行によってオーバーライドされます。

表 *68.* セキュリティー機能の優先順位

順序	**フィーチャー**
1 (最低)	アプリケーション表明 ID
2	チャネル定義の **MCAUSER** 属性

表 68. セキュリティー機能の優先順位 (続き)	
順序	**フィーチャー**
3	**ADOPTCTX(YES)** での接続認証
4	**USERSRC(MAP)** を使用したチャネル認証レコード
5 (最高)	セキュリティー出口

早期採用の影響

接続認証レコードおよびチャネル認証レコードは、接続認証ユーザーの採用をいつ実行するかを制御する構成オプションを提供します。 この設定は、早期採用と呼ばれます。 早期採用が有効になっている場合、チャネル認証レコードが処理される前に接続認証 ID の採用が行われます (つまり、チャネル認証レコードが **CONNAUTH** の採用をオーバーライドします)。

無効にすると、順序が逆になります。つまり、チャネル認証レコードは **CONNAUTH** の採用前に処理されます。 この状況では、接続認証の採用は、チャネル認証が記録する優先順位よりも高くなります。

早期採用のデフォルト設定は enabled です。

ULW OAM によるオブジェクトへのアクセスの制御 (UNIX, Linux, and Windows)

オブジェクト権限マネージャー (OAM) には、IBM 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。

399 ページの『UNIX, Linux, and Windows 上の IBM 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。 IBM 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。

Linux UNIX UNIX and Linux での OAM ユーザーに基づく許可

IBM MQ 8.0 から、UNIX and Linux システム上のオブジェクト権限マネージャー (OAM) は、ユーザーに基づく許可とグループに基づく許可を使用できるようになりました。

IBM MQ 8.0 より前は、UNIX and Linux でのアクセス制御リスト (ACL) はグループのみに基づいていました。 IBM MQ 8.0 以降では、ACL はユーザー ID とグループの両方に基づいており、 インストール可能サービスの構成 および「UNIX および Linux での許可サービス・スタンザの構成」で説明されているように、**SecurityPolicy** 属性を適切な値に設定することにより、許可にユーザー・ベース・モデルまたはグループ・ベース・モデルのいずれかを使用できます。

IBM MQ 8.0 以降での動作の変更点

IBM MQ 8.0 以降、ユーザー・ベースのポリシーによる実行時に、一部のコマンドが以前のバージョンの製品とは異なる情報を返すようになりました。

- **dmpmqaut** および **dmpmqcfg** コマンドは、PCF の同等の操作と同様に、ユーザー・ベースのレコードを表示します。
- IBM MQ Explorer 用の OAM プラグインではユーザー・ベースのレコードが表示され、ユーザー・ベースの変更が可能です。
- OAM の **Inquire** 関数から返される結果には、ユーザー処置可能であることが表示されます。

qm.ini ファイルのサービス・スタンザで説明されているように qm.ini ファイルでユーザー・ベースの許可が有効になっている場合、 **setmqaut** コマンドで **-p** 属性を使用しても、同じ 1 次グループ内のすべてのユーザーにアクセス権限が付与されるわけではありません。

多数のユーザーが存在する状況でユーザー・ベースの許可を使用し始めた場合、グループ・ベース・モデルの場合に比べて AUTH キューに保管されるレコード数がおそらく増えることになり、検証対象のレコー

ドが多くなったために以前よりも許可プロセスに少し時間がかかる可能性があります。 この増加は大きな問題にはならないと予想されます。 必要に応じて、ユーザーとグループの許可を混合して使用することもできます。

移行の考慮事項

既存のキュー・マネージャーのモデルをグループからユーザーに変更した場合、直ちには影響が発生しません。 既に付与された許可は引き続き適用されます。 そのようなキュー・マネージャーに接続するすべてのユーザーは以前と同じ特権 (つまり、それらのユーザー ID が属するすべてのグループの組み合わせ) が付与されます。 ユーザー ID に対して新しい **setmqaut** コマンドが発行された場合は、直ちに効果が発生します。

ユーザー・ポリシーを使って新しいキュー・マネージャーを作成した場合、このキュー・マネージャーにはそれを作成したユーザー (通常はこのユーザー ID は mqm であるが異なる場合もある) の許可だけが含まれます。 さらに mqm グループに自動的に付与される許可もあります。 しかし mqm が 1 次グループではない場合、mqm グループは初期の許可セットには含まれません。

ユーザー・ポリシーからグループ・ポリシーに移行した場合、ユーザー・ベースの許可は自動的には削除されません。 ただし、これらは許可の検査で使用されなくなります。 ポリシーを戻す前に現在の構成を保存し、ポリシーを変更して、キュー・マネージャーを再始動した後、スクリプトを再生してください。 現在はグループ・ベースのキュー・マネージャーになったため、1 次グループに基づいてユーザー ID ルールが保管されます。

関連概念

オブジェクト権限マネージャー (OAM)

プリンシパルとグループ (UNIX、Linux、および Windows)

qm.ini ファイルの Service スタンザ

関連資料

crtmqm (キュー・マネージャーの作成) コマンド

ULW UNIX, Linux, and Windows 上の IBM MQ オブジェクトへのアクセス権限の付与

IBM MQ オブジェクトに対するアクセス権限をユーザーおよびユーザー・グループに付与するには、**setmqaut** 制御コマンド、 **SET AUTHREC** MQSC コマンド、または **MQCMD_SET_AUTH_REC** PCF コマンドを使用します。 IBM MQ Appliance では、 **SET AUTHREC** コマンドのみを使用できます。

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。

SET AUTHREC MQSC コマンドとその構文の詳細な定義については、SET AUTHREC 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、Set Authority Record 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、351 ページの『UNIX, Linux, and Windows 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 setmqaut 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。

SET AUTHREC の使用上の注意

追加する許可のリストと削除する許可のリストが重複しないようにしてください。 例えば、表示権限の追加と表示権限の削除を同じコマンドで行うことはできません。 権限が別々のオプションで表されている場

合でも、この規則は適用されます。 例えば次のようなコマンドは、DSP 権限が ALLADM 権限と重なり合っているため失敗します。

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

この重なり合いの動作の例外は、ALL 権限を指定した場合です。 以下のコマンドは、最初に ALL 権限を追加してから、SETID 権限を削除します。

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

以下のコマンドは、まず ALL 権限を削除してから、DSP 権限を追加します。

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

コマンドで指定されている順序に関係なく、ALL が最初に処理されます。

ULW UNIX, Linux, and Windows 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して、多数のオブジェクトに対するユーザーの特権を 1 回の操作で設定します。個々のオブジェクトの作成時に、個々のオブジェクトに対して個別の **setmqaut** コマンドまたは **SET AUTHREC** コマンドを発行する必要はありません。 IBM MQ Appliance では、 **SET AUTHREC** コマンドのみを使用できます。

setmqaut または SET AUTHREC 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、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、 `-t prcs` を使用してプロセスを識別し、プロファイル名として ** を使用する場合、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。

完全な修飾子として使用できるのは、二重アスタリスク ** のみです。

```
**.DEF
  ABC.**
  A*.**
```

しかし、

```
A**
```

そうでない場合は、メッセージ「 AMQ7226E: プロファイル名が無効です。 」を受け取ります。

注 : UNIX システムおよび 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、この例では、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。

DISPLAY AUTHREC MQSC コマンドとその構文の詳細な定義については、DISPLAY AUTHREC 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 および Linux のユーザーは、**dmpmqaut** コマンドに -p オプションを使用できますが、許可を定義するときは代わりに -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 : IBM 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
```

ULW *OAM プロファイルでのワイルドカード文字の使用 (UNIX, Linux, and Windows)*

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、 -t prcs を使用してプロセスを識別し、プロファイル名として ** を使用する場合、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**。

ULW プロファイルの優先順位 *(UNIX, Linux, and Windows)*

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、この例では、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. **

MQSC コマンドを使用する場合の同等の情報については、SET AUTHREC を参照してください。

ULW プロファイル設定のダンプ *(UNIX, Linux, and Windows)*

指定されたプロファイルに関連付けられている現在の許可をダンプするには、 **dmpmqaut** 制御コマンド、**DISPLAY AUTHREC** MQSC コマンド、または **MQCMD_INQUIRE_AUTH_RECS** PCF コマンドを使用します。 IBM MQ Appliance では、 **DISPLAY AUTHREC** コマンドのみを使用できます。

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。

DISPLAY AUTHREC MQSC コマンドとその構文の詳細な定義については、DISPLAY AUTHREC 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。その代わりに、-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 : IBM 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
```

ULW アクセス設定の表示 (UNIX, Linux, and Windows)

dspmqaut 制御コマンド、**DISPLAY AUTHREC** MQSC コマンド、または **MQCMD_INQUIRE_ENTITY_AUTH** PCF コマンドを使用して、特定のプリンシパルまたはグループが特定のオブジェクトに対して持っている権限を表示します。 IBM MQ アプライアンスで使用することができるのは、**DISPLAY AUTHREC** コマンドのみです。

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。

DISPLAY AUTHREC MQSC コマンドとその構文の詳細な定義については、DISPLAY AUTHREC 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
```

ULW IBM MQ オブジェクトへのアクセス権の変更と取り消し (UNIX, Linux, and Windows)

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** 制御コマンド、**DELETE AUTHREC** MQSC コマンド、または **MQCMD_DELETE_AUTH_REC** PCF コマンドを使用します。 MQ Appliance IBM MQ アプライアンスで使用することができるのは、**DELETE AUTHREC** コマンドのみです。

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。

- Windows 142 ページの『Windows でのグループの作成と管理』
- AIX 140 ページの『AIX でのグループの作成と管理』
- Solaris 142 ページの『Solaris でのグループの作成と管理』
- Linux 141 ページの『Linux でのグループの作成と管理』

.

IBM 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詳細な定義については、setmqaut を参照してください。

DELETE AUTHREC MQSC コマンドとその構文の詳細な定義については、DELETE AUTHREC を参照してください。

MQCMD_DELETE_AUTH_REC PCF コマンドとその構文の詳細な定義については、Delete Authority Record を参照してください。

Windows Windows では、 IBM MQ 8.0 以降、 **setmqaut** の **-u** *SID* パラメーターを使用して、特定の Windows ユーザー・アカウントに対応する OAM 項目をいつでも削除できます。

IBM MQ 8.0 より前では、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 項目を削除する必要がありました。 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でした。

ULW 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、オブジェクト権限マネージャー (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 を設定します。

  MQSNOAUT 変数の設定による影響、および Windows と UNIX での MQSNOAUT の設定方法については、環境変数の説明 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。

OAM が無効になっている状態で **setmqaut**、または **dspmqaut** コマンドを使用する場合の注意点を以下にまとめます。

- OAM は、指定のプリンシパルまたはグループを検証しません。つまり、コマンドで無効値が使用されていても、そのまま受け入れられてしまいます。
- 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。 それは、OAM がオブジェクト作成時に同じ場所に置かれている必要があるためです。 IBM MQ OAM を削除後に使用するには、キュー・マネージャーを再作成してください。

関連概念

UNIX、Linux、および Windows 用のインストール可能サービスとコンポーネント

関連タスク

インストール可能サービスの構成

関連資料

インストール可能サービスの参照情報

リソースへの必要なアクセス権限の付与

このトピックでは、 UNIX、 Linux、 Windows、 IBM i、および z/OS 上の IBM MQ システムにセキュリティーを適用するために実行するタスクについて説明します。

このタスクについて

このタスクでは、ご使用の IBM 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: 360 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: 369 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: 376 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: 377 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: 379 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: 380 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。

z/OS Multi 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 69. 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	360 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	362 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	363 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	364 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	365 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	367 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	368 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX

- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  UNIX, Linux, and Windows 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 ObjectProfile -t queue -g GroupName ReqdAction
```

- IBM i
  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(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS の場合は、次のコマンドを実行して、指定したキューに対するアクセス権限を付与します。

```
RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  キューに対してどの MQSC コマンドをユーザーが実行できるかを指定するには、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. QType UACC(NONE)
PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  ユーザーが DISPLAY QUEUE 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. QType UACC(NONE)
PERMIT QMgrName.DISPLAY. QType 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 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。

  - ULW UNIX, Linux, and 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。
  - IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMDLT、*ADMDSP。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。
  - z/OS z/OS では、値 ALTER、CLEAR、DELETE、または MOVE のうちの 1 つ。

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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  UNIX, Linux, and Windows 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 ObjectProfile -t topic -g GroupName ReqdAction
```

- IBM i
  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(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS の場合は、以下のコマンドを発行します。

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

これらのコマンドは、指定されたトピックへのアクセス権を付与します。 トピックに対してどの 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.TOPIC UACC(NONE)
PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

ユーザーが DISPLAY TOPIC 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.TOPIC UACC(NONE)
PERMIT QMgrName.DISPLAY.TOPIC 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 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。

- ULW UNIX, Linux, and Windows システムでは、+chg、+clr、+crt、+dsp、+dlt、+dsp. +ctrl. のいずれかの許可を任意に組み合わせたものがあります。 権限 +alladm は +chg +clr +dlt +dsp と等価です。
- IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。
- z/OS z/OS では、値 ALTER、CLEAR、DEFINE、DELETE、または MOVE のうちの 1 つ。

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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- IBM i
  On IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS On z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

これらのコマンドは、指定されたチャネルへのアクセス権を付与します。 チャネルに対してどの 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.CHANNEL UACC(NONE)
PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

ユーザーが DISPLAY CHANNEL 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.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL 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 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。

- ULW UNIX, Linux, and Windows では、以下の許可を任意に組み合わせることができます。+chg、+clr、+crt、+dlt、+dsp。+ctrl、+ctrlx。 権限 +alladm は +chg +clr +dlt +dsp と等価です。
- IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL、*CTRLX。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。
- z/OS z/OS では、値 ALTER、CLEAR、DEFINE、DELETE、または MOVE のうちの 1 つ。

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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX

- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
  ```

- IBM i
  On IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS
  On z/OS:

  キュー・マネージャーで実行できる MQSC コマンドを確認するには、MQSC コマンドごとに以下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。

  - ULW UNIX, Linux, and Windows 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。

  - IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。

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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
  ```

- IBM i
  On IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS On z/OS:

  ```
  RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  これらのコマンドは、指定されたチャネルへのアクセス権を付与します。 チャネルに対してどの 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.PROCESS UACC(NONE)
  PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  ユーザーが DISPLAY PROCESS 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.PROCESS UACC(NONE)
  PERMIT QMgrName.DISPLAY.PROCESS 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 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。

- ULW UNIX, Linux, and Windows 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。
- IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。
- z/OS z/OS では、値 ALTER、CLEAR、DEFINE、DELETE、または MOVE のうちの 1 つ。

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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- IBM i
  On IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS On z/OS:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  これらのコマンドは、指定された名前リストへのアクセス権を付与します。 名前リストに対してどの 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.NAMELIST UACC(NONE)
PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  ユーザーが DISPLAY NAMELIST 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.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST 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 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。

- ULW On UNIX, Linux, and Windows 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。
- IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL、*CTRLX。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。
- z/OS z/OS では、値 ALTER、CLEAR、DEFINE、DELETE、または MOVE のうちの 1 つ。

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。z/OS「ただし、z/OS ではサービス・オブジェクトが存在しません。」

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- On IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS On z/OS:

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。

- ULW UNIX, Linux, and 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。
- IBM i IBM i では、次の権限の任意の組み合わせ: *ADMCHG、*ADMCLR、*ADMCRT、*ADMDLT、*ADMDSP、*CTRL、*CTRLX。 権限 *ALLADM は、これらの個々の権限すべてを合わせたものと等価です。

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 70. 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	370 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	371 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	372 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	372 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	373 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

表 70. 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 (続き)	
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**
名前リスト	374 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	375 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
  ```

- IBM i
  On IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
  ')
  ```

- z/OS
  On z/OS:

  ```
  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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- IBM i

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW

  On UNIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
  ```

- IBM i

  On IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  On z/OS:

  ```
  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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
  ```

- IBM i
  On IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  On z/OS:

  ```
  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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +alladm
  ```

- IBM i
  On IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  On z/OS:

  ```
  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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- IBM i
  On IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  On z/OS:

```
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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- IBM i
  On IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  On z/OS:

```
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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- IBM i
  On IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  On z/OS:

```
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 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

許可の詳細を変更した後、REFRESH SECURITY コマンドを使用してセキュリティー・リフレッシュを実行します。

手順

- ウィザードの使用:
  a) IBM MQ Explorer 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。
- Windows UNIX
  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
+put
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 は、IBM MQ Explorer を使用する場合にのみ必要です。

- IBM i

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

z/OS の場合は、以下のコマンドを発行します。

```
RDEFINE MQQUEUE QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQQUEUE) ID(GroupName) ACCESS(READ)
RDEFINE MXTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MXTOPIC) ID(GroupName) ACCESS(READ)
RDEFINE MQPROC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQPROC) ID(GroupName) ACCESS(READ)
RDEFINE MQNLIST QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
```

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 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

役割ベースの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・システムに適したコマンドを使用できます。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i

- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

注 : ULW

1. IBM MQ Explorer の代わりに **runmqsc** を使用してキュー・マネージャーを管理する場合は、SYSTEM.MQSC.REPLY.QUEUE では、 SYSTEM.MQEXPLORER.REPLY.MODEL キュー。
2. キュー・マネージャー上のすべてのリソースに対するアクセス権限をユーザーに付与する場合、そのユーザーが qm.ini ファイルに対する読み取り権限を持っていない限り、ユーザーが実行できないコマンドがいくつかあります。 これは、mqm 以外のユーザーが qm.ini ファイルを読み取ることができるという制限のためです。

   qm.ini ファイルに対する読み取り権限をユーザーに付与していない場合、ユーザーは以下のコマンドを発行できません。

   - TLS を使用するように構成されたチャネルの定義
   - qm.ini で定義されている自動構成挿入変数を使用したチャネルの定義

手順

- ウィザードを使用している場合は IBM MQ Explorer Navigator ペインでキュー・マネージャーを右クリックして、「**オブジェクト権限**」 > 「**役割ベースの権限の追加**」 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。
- Linux UNIX
  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.MODEL -t queue -g GroupName +dsp +inq +get +put
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +connect
```

  @class について詳しくは、 **setmqaut** を参照してください。
- Windows
  Windows システムの場合は、UNIX and Linux システムの場合と同じコマンドを実行しますが、プロファイル名 @class の代わりに @CLASS を使用します。
- IBM i
  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
  z/OS の場合は、以下のコマンドを発行します。

```
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 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。

UNIX、Linux、Windows の各システム、および IBM i で、DELETE AUTHREC コマンドを使用することもできます。

注 : IBM MQ アプライアンスで使用することができるのは、**DELETE AUTHREC** コマンドのみです

手順

- ULW
  UNIX, Linux, and Windows システムの場合は、以下のコマンドを実行します。

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName -connect
  ```

- IBM i
  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
  z/OS の場合は、以下のコマンドを発行します。

  ```
  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 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 (387 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 (380 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、387 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	388 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	395 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	396 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	397 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	398 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、TLS 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. 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. 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 TLS 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. 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. 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ブロックするには、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
381 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、330 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。 以下の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、クライアント接続に関連付けられるユーザー ID です。値は、クライアント・アプリケーションによって表明されたり、早期採用を使用する接続認証によって変更されたり、チャネル出口を介して設定されたりする場合があります。

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

channels スタンザの属性 (ChlauthEarlyAdopt)

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) SSLCERTI(generic-issuer-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 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 IBM 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。
generic-issuer-name は、一致する証明書の発行者 DN を参照します。 このパラメーターはオプションですが、複数の認証局を使用している場合、正しくない証明書に誤ってマッチングしないようにするため、このパラメーターを使用する必要があります。

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。 以下の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・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・ユーザー ID がチャネル接続を確立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、クライアント接続に関連付けられるユーザー ID です。値は、クライアント・アプリケーションによって表明されたり、早期採用を使用する接続認証によって変更されたり、チャネル出口を介して設定されたりする場合があります。

関連資料
SET CHLAUTH

SSL または *TLS* 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、TLS 識別名 (DN) 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') SSLCERTI(generic-issuer-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 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 IBM MQ の規則を参照してください。

generic-issuer-name は、一致する証明書の発行者 DN を参照します。 このパラメーターはオプションですが、複数の認証局を使用している場合、正しくない証明書に誤ってマッチングしないようにするため、このパラメーターを使用する必要があります。

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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
  On UNIX, Linux, and Windows:

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

- IBM i
  On IBM i:

```
GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
```

- z/OS
  On z/OS:

```
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。

関連概念

196 ページの『チャネル・イニシエーターのための接続セキュリティー・プロファイル』
チャネル・イニシエーターからの接続をチェックするためのプロファイルは、キュー・マネージャー名またはキュー共有グループ名の後に *CHIN* という語が付いた形式になっています。 チャネル・イニシエーターの開始済みタスクのアドレス・スペースで使用するユーザー ID に、接続プロファイルに対する READ アクセス権を与えてください。

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	389 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	390 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	391 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	455 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	392 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	393 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	394 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- Windows

注 : IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- Windows

注 : IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows システムの場合、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:

    ```
    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
    ```

  注 : `setid` 権限または `setall` 権限を使用するには、該当するキュー・オブジェクトとキュー・マネージャー・オブジェクトの両方に対して許可が付与されている必要があります。
- 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- ULW
UNIX, Linux, and Windows システムの場合、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:

```
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
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
  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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- Windows

注 : IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
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。 392 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が該当するのは、セキュリティー・スタンザのトピックの説明に従って、qm.ini ファイルの **ClusterQueueAccessControl** 属性に *RQMName* を設定し、キュー・マネージャーを再始動した場合のみです。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i IBM i
- Linux Linux
- UNIX UNIX
- IBM i Windows

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows システムの場合は、以下のコマンドを実行します。

```
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、以下のコマンドを発行します。

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE.ObjectProfile CLASS(MQQUEUE)
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 71. 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	395 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	396 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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- Windows

注 : IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- 

IBM i
- 

Linux
- 

UNIX
- 

Windows

注 : 

IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- IBM i

注 : MQ Appliance IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i

- Linux
- UNIX
- Windows

注 : IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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 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、以下のコマンドを発行します。

```
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。

以下のプラットフォームでは、SET AUTHREC コマンドも使用できます。

- IBM i
- Linux
- UNIX
- Windows

注 : IBM MQ Appliance で使用することができるのは、**SET AUTHREC** コマンドのみです

手順

- UNIX, Linux, and Windows 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
```

```
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、以下のコマンドを発行します。

```
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。

ULW UNIX, Linux, and Windows 上の IBM MQ を管理する権限

IBM MQ 管理者は、すべての IBM MQ コマンドを使用できます。他のユーザーに権限を与えることもできます。 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 MQ 管理者には、すべての IBM MQ コマンド (他ユーザーに IBM 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 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にならなければなりません。

Windows Windows に限って、ローカル・アカウントで IBM MQ で管理することも可能です。ただし、Windows システムでそのアカウントが Administrators グループのメンバーになっていることが条件です。

重要 : 管理者コマンドを使用して、Azure AD ユーザーを mqm グループに追加できます。 例えば、コマンド net localgroup mqm AzureAD\<your userID> /add を使用します。 その後、IBM MQ 管理コマンドを実行するか、IBM MQ Explorer を使用します。

mqm グループは、IBM MQ がインストールされると自動的に作成されます。 グループに他のユーザーを追加すると、そのユーザーが管理を実行できるようになります。 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、IBM 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、IBM 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 コマンドを使用した権限の管理の詳細については、プログラマブル・コマンド・フォーマットを参照してください。

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。 IBM MQ Explorer 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、IBM MQ Explorer 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 MQ Explorer 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。

重要 : IBM MQ 8.0 以降では、IBM MQ Script (MQSC) コマンドを発行する制御コマンド **runmqsc** を使用する際に、管理者である必要はありません。

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。

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、『IBM 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。
- z/OS IBM MQ for z/OS 上のコマンド・サーバーによって処理される MQSC コマンドについては、z/OS のコマンド・セキュリティーとコマンド・リソース・セキュリティー 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 IBM 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 IBM 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 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、404 ページの『プリンシパルとグループ (UNIX, Linux, and Windows)』 で説明されています。

ユーザー・アカウント制御 (UAC) 機能がある Windows のバージョンでは、ユーザーが Administrators グループのメンバーである場合でも、ユーザーが特定のオペレーティング・システム機能に対して実行できる操作が制限されます。 ユーザー ID が管理者グループには属しているが、 **mqm** グループには属していない場合は、昇格されたコマンド・プロンプトを使用して **crtmqm** などの IBM MQ 管理コマンドを発行する必要があります。そうしないと、エラー AMQ7077: 「要求された操作を実行する権限がありません」が生成されます。 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。

以下のアクション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 109 ページの『チャネル・イニシエーター定義の保護』 で説明します。)
- 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、IBM 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 5.3 ではこの制限を上げていますが、IBM MQ では引き続きすべての UNIX and Linux プラットフォーム上で 12

文字という制限が課されています。 12 文字を超えるユーザー ID を使用すると、IBM 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。

ULW mqm グループの管理 (UNIX, Linux, and Windows)

mqm グループのユーザーには、IBM 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、IBM 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 Windows でのグループの作成と管理
- AIX AIX でのグループの作成と管理
- Solaris Solaris でのグループの作成と管理
- Linux Linux でのグループの作成と管理

Windows Windows 2000 か Windows 2003 以降でドメイン・コントローラーを実行している場合は、ドメイン管理者が IBM MQ 用の特別なアカウントをセットアップしなければならない場合があります。 詳しくは、「IBM MQ を使用した Prepare IBM MQ Wizard の構成」 および 「Windows 用の IBM MQ ドメイン・アカウントの作成およびセットアップ」を参照してください。

ULW IBM MQ 上で UNIX, Linux, and Windows 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 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 IBM 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

399 ページの『UNIX, Linux, and Windows 上の IBM MQ を管理する権限』
IBM MQ 管理者は、すべての IBM MQ コマンドを使用できます。他のユーザーに権限を与えることもできます。 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。

ULW セキュリティー検査が行われるタイミング (UNIX, Linux, and Windows)

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 IBM 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 IBM MQ にサインオンする必要はありません。IBM 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)
IBM 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。

IBM 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 128 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。 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。

ULW UNIX, Linux, and Windows 上の IBM MQ によってアクセス制御が実装される方法

IBM 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 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、IBM MQ の一部です。 IBM 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 (358 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、IBM MQ オブジェクトにアクセスできます。 348 ページの『OAM によるオブジェクトへのアクセスの制御 (UNIX, Linux, and Windows)』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。また、 Windows の場合は、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にも制限されます。 このキューへのユーザー・アクセス権は変更できません。

IBM 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、348 ページの『OAM によるオブジェクトへのアクセスの制御 (UNIX, Linux, and Windows)』を参照してください。

IBM 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。 IBM MQ は、OAM の決定に従います。OAM が決定できない場合は、IBM MQ はアクセスを許可しません。

ULW ユーザー ID の識別 (UNIX, Linux, and Windows)

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 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 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、406 ページの『コンテキスト権限 (UNIX, Linux, and Windows)』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。

ULW プリンシパルとグループ (UNIX, Linux, and Windows)

プリンシパルは、グループに属します。 リソース・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。 アクセス制御リスト (ACL) は、グループとユーザー ID の両方に基づきます。

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。

以下の特定のプラットフォームでグループを定義して管理するプロセスが説明されています。

- Windows Windows でのグループの作成と管理
- AIX AIX でのグループの作成と管理
- Solaris Solaris でのグループの作成と管理
- Linux Linux でのグループの作成と管理

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。

Linux UNIX UNIX and Linux システム

IBM MQ 8.0 以降では、アクセス制御リスト (ACL) はユーザー ID とグループの両方に基づいており、インストール可能サービスの構成 および UNIX および Linux での許可サービス・スタンザの構成で説明されているように、 **SecurityPolicy** 属性を適切な値に設定することによって許可に使用できます。

IBM MQ 8.0 以降、ユーザー・ベースのモデル を許可に使用できるようになり、これによってユーザーとグループの両方を使用できるようになりました。 しかし、setmqaut コマンドでユーザーを指定している場合は、新しい権限はそのユーザー単独に適用され、そのユーザーが属するグループには適用されません。 詳しくは、UNIX および Linux システムでの OAM ユーザーに基づく許可を参照してください。

グループ・ベース・モデル を許可に使用すると、ユーザー ID が属する 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 IBM 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 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 の場合と同じです。 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。 IBM MQ では、ユーザー ID をドメイン・ネームで修飾することにより、これらのユーザーに異なるレベルのアクセス権を付与できます。

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_name\group_name
```

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・セキュリティー・スタンザに GroupModel=GlobalGroups という設定が組み込まれている。 保護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、IBM 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を持つローカル・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 での **SecurityPolicy** 属性の設定について詳しくは、 インストール可能サービス および Windows での許可サービス・スタンザの構成を参照してください。

Windows *Windows セキュリティー ID (SID)*

Windows 上の IBM MQ は、SID が使用可能な場合はそれを使用します。 許可要求で Windows SID が指定されていなければ、IBM 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 IBM MQ for Windows に作成される場合、IBM MQ はメッセージ記述子に SID を保管します。 IBM MQ on Windows は、許可検査を実行するときに、SID を使用して SAM データベースから完全な情報を照会します。 (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、IBM 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 IBM 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 IBM 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。

ULW 代替ユーザー権限 (UNIX, Linux, and Windows)

1 つのユーザー ID で IBM 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 IBM 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。

ULW コンテキスト権限 (UNIX, Linux, and Windows)

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、IBM 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。

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。

Multi マルチプラットフォーム では、MCA がキュー・マネージャーに接続した後にキュー・マネージャーのリソースにアクセスしようとする時に、キュー・マネージャーは、*MCAUserIdentifier* の値がブランクになっている場合を除いて、*MCAUserIdentifier* の値を権限検査用のユーザー ID として使用します。*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 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、IBM 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、IBM MQ MQI client・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。
- TLS 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 : IBM WebSphere MQ 7.1 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。

IBM MQ オブジェクト権限マネージャーのユーザー認証

IBM MQ MQI client 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 IBM 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、333 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、IBM 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。

LDAP 許可

LDAP 許可を使用すれば、ローカル・ユーザー ID を使用する必要がなくなります。

サポート対象プラットフォームでの LDAP 許可の使用可否

LDAP 許可は以下のプラットフォームで使用可能です。

- UNIX
- IBM i
- Windows

重要:

IBM MQ 9.0 の一般出荷以降、新規リリースの場合も、旧リリースから移行した場合も、すべてのキュー・マネージャーでこの機能を使用できるようになりました。

LDAP 許可の概要

LDAP 許可を使用すると、 **setmqaut** および **DISPLAY AUTHREC** などの許可構成を処理するコマンドは、識別名を処理できます。 以前は、ローカル・オペレーティング・システム上のユーザーとグループに関する使用可能な最大文字数を資格情報と比較して、ユーザーが認証されていました。

重要: DEFINE AUTHINFO コマンドを実行した場合、キュー・マネージャーを再始動する必要があります。 キュー・マネージャーを再始動しないと、setmqaut コマンドは正しい結果を返しません。

ユーザーが識別名ではなくユーザー ID を提供すると、ユーザー ID が処理されます。 例えば、PUTAUT(CTX) のチャネルに着信メッセージが存在する場合、ユーザー ID の文字が LDAP 識別名にマップされ、適切な許可検査が行われます。

他のコマンド (**DISPLAY CONN** など) は、ユーザー ID がローカル OS に実際に存在しない場合でも、引き続きユーザー ID の実際の値を処理/表示します。

UNIX LDAP 許可が設定されている場合、キュー・マネージャーは、 qm.ini ファイル内の **SecurityPolicy** 属性に関係なく、常に UNIX プラットフォーム上のセキュリティーのユーザー・モデルを使用します。 したがって、個別のユーザーに関してアクセス権を設定するとそのユーザーだけが影響を受け、そのユーザーのグループに属する他のどのユーザーも影響を受けません。

OS モデルの場合と同様に、ユーザー個人と、そのユーザーが属するすべてのグループ (存在する場合) に割り当てられた権限の組み合わせが引き続きユーザーに与えられます。

例えば、LDAP リポジトリーで以下のレコードが定義されているとします。

- **inetOrgPerson** クラス:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- **groupOfNames** クラス:

```
dn="cn=Application Group A, ou=groups, o=yourcompany, c=yourcountry"
        longname=ApplicationGroupA [longer than 12 characters]
        members="cn=JaneDoe, ou=users, o=yourcompany, c=yourcountry",
                "cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
```

認証を行うためには、この LDAP サーバーを使用するキュー・マネージャーの定義において、**CONNAUTH** 値がタイプ IDPWLDAP の **AUTHINFO** オブジェクトを指し示し、関連する名前解決属性が例えば次のように設定される必要があります。

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

認証にこの構成を使用すると、アプリケーションは、MQCNO 呼び出しの中で使用される CSPUserID フィールドを以下のいずれかの値のセットで完成させることができます。

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

または

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

どちらの場合も、システムでは提供された値を使用して「jodoe」の OS コンテキストを認証できます。

許可の設定

短縮名または **USRFIELD** を使用して許可を設定する方法について説明します。

408 ページの『LDAP 許可』で説明されている複数の形式を処理する方法は、引き続き許可コマンドに組み込まれます。さらに、 shortname または USRFIELD のいずれかを非修飾形式で使用できるように拡張されています。

許可の設定のためにユーザー (プリンシパル) を指定する際に文字ストリングにすると、LDAP レコード内の特定の属性が指定されます。

重要 : = 文字はオペレーティング・システムのユーザー ID に使用できないので、文字ストリングに使用できません。

shortname の可能性がある許可のために OAM にプリンシパル名を渡す場合、文字ストリングは 12 文字に収まらなければなりません。 マッピング・アルゴリズムでは、まず SHORTUSR 属性を LDAP 照会で使用して、DN への解決を試みます。

これが UNKNOWN_ENTITY エラーで失敗した場合、または指定されたストリングが shortname ではない可能性がある場合は、USRFIELD 属性を使用して LDAP 照会を構成しようとします。

重要 : DEFINE AUTHINFO コマンドを実行した場合、キュー・マネージャーを再始動する必要があります。 キュー・マネージャーを再始動しないと、setmqaut コマンドは正しい結果を返しません。

ユーザー許可を処理する場合、以下の setmqaut コマンド設定はすべて同等です。

表 72. ユーザー許可の設定

コマンド	注記
`setmqaut -m QM -t qmgr -p jodoe +connect`	これは単純な非修飾名で、SHORTUSR によって解決されます。
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	これも単純な非修飾名で、USRFIELD によって同じエンティティーに解決されます。
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	指定された属性を使用しています。
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	指定された別の属性を使用しています。この属性は、AUTHINFO オブジェクトに構成されたものでなくても構いません。

setmqaut コマンドの代わりに、SET AUTHREC MQSC コマンドを使用することもできます。

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

または、以下のストリングを MQCACF_PRINCIPAL_ENTITY_NAMES エレメントに含めて、Set Authority Record (MQCMD_SET_AUTH_REC) PCF コマンドを使用できます。

```
  "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

グループを処理する場合、shortname 処理に関するあいまいさはありません。これは、任意の形式のグループ名を 12 文字に適合させる必要がないためです。 つまり、グループについては、SHORTUSR 属性に相当するものはありません。

したがって、拡張属性を含めて AUTHINFO オブジェクトを構成し、以下のように設定した場合、411 ページの表 73 に示された構文例が有効です。

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

表 73. グループ許可設定

コマンド	注記
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	GRPFIELD を使用して解決します
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	単一の属性を指定しています
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	完全な識別名を使用しています

上記の **setmqaut** コマンドの代わりに、SET AUTHREC MQSC コマンドを使用することもできます。

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

または、以下のストリングを MQCACF_GROUP_ENTITY_NAMES エレメントに含めて、Set Authority Record (MQCMD_SET_AUTH_REC) PCF コマンドを使用できます。

```
  "ApplicationGroupA"
```

重要 :

ユーザーの場合もグループの場合も、いずれの形式を使用して名前を表すとしても、固有の識別名が得られなければなりません。

例えば、別個の 2 つのレコードが両方とも "shortu=jodoe" を含んでいてはなりません。

1 つの固有の DN を求めることができない場合、OAM は MQRC_UNKNOWN_ENTITY を返します。

許可の表示

ユーザーまたはグループの許可を表示する様々な方法。

dspmqaut コマンド

ユーザーまたはグループが利用できる許可を表示する最もシンプルな方法は、dspmqaut コマンドを使用することです。

ユーザーまたはグループを特定する構文のバリエーションで照会を使用できます。 コマンド出力は、コマンド・ラインで指定された形式で ID を繰り返します。 出力は、完全に解決された DN を報告することはありません。

以下に例を示します。

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

または

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

dmpmqaut コマンドと dmpmqcfg コマンド

dmpmqaut コマンドと MQSC または PCF の同等のコマンドは、410 ページの『許可の設定』 で説明する **setmqaut** テーブルのように、サポートされるフォーマットのいずれかのプリンシパルまたはグループを指定できます。 ただし、**dspmqaut**、 とは異なり、**dmpmqaut** コマンドは常に完全 DN を報告します。

```
dmpmqaut -m QM -t qmgr -p jodoe
-----------------------------------
profile: self
object type:qmgr
entity:cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry
entity type: principal
authority: connect
```

同様に、 dmpmqcfg コマンドは選択したレコードのフィルターを持っていませんが、常に後でやり直すことができるフォーマットで完全 DN を表示します。

```
dmpmqcfg -m QM -x authrec
-----------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

LDAP 許可を使用する場合のその他の考慮事項

IBM MQ 9.0.0 以降で LDAP 許可を使用する場合に注意する必要のある Message Queue Interface (MQI) およびその他の MQSC コマンドや PCF コマンドの変更に関する簡単な説明。

ADOPTCTX

アプリケーションで認証情報を指定するための要件、または ADOPTCTX 属性を YES に設定するための要件はありません。

アプリケーションが明示的に認証しない場合、またはアクティブな CONNAUTH オブジェクトに対して **ADOPTCTX** が NO に設定されている場合、このアプリケーションに関連付けられた ID コンテキストはオペレーティング・システムユーザー ID から取られます。

許可を適用する必要がある場合、そのコンテキストは LDAP ID にマップされ、setmqaut コマンドに対するのと同じ規則を使用します。

MQI 呼び出しへの入力パラメーター

「MQOPEN」、「MQPUT1」、および 「MQSUB」 には、代替ユーザー ID を指定できる構造があります。

これらのフィールドを使用する場合、**setmqaut**、**dmpmqaut**、および **dspmqaut** コマンドと同じ規則を使用して 12 文字のユーザー ID が DN にマップされます。

MQPUT および MQPUT1 を使用して、適切に許可されたプログラムを MQMD UserIdentifier フィールドに設定することもできます。 PUT プロセスの間、このフィールドの値は監視されないため、任意の値を設定できます。

しかしながら、大抵 **UserIdentifier** 値は、メッセージ処理の後半での (例えば、送信側チャネルで PUTAUT(CTX) を定義するとき) 許可に使用できます。

その時点で、その受信側のキュー・マネージャーの構成 (LDAP または OS ベースにできます) を使用して、許可について ID が検査されます。

MQI 呼び出しへの出力パラメーター

ユーザー ID を MQI 構造のプログラムに指定するときは必ず、接続に関連付けられた 12 文字のショート・ネーム・バージョンになります。

例えば、API 出口の **MQAXC.UserId** 値は、LDAP マッピングから返されるショート・ネームになります。

その他の管理 MQSC および PCF コマンド

DISPLAY CONN USERID のようなオブジェクト状況のユーザー情報を表示するコマンドは、コンテキストに関連付けられた 12 文字のショート・ネームが返されます。 フル DN は表示されません。

CHLAUTH マッピング規則またはチャネルの MCAUSER 値など ID のアサーションを許可するコマンドは、それらの属性に対して定義された最大長 (現時点では 64 文字) まで値を指定できます。

構文への変更はありません。 許可にその ID が必要な場合、**setmqaut**、**dmpmqaut**、および **dspmqaut** コマンドと同じ規則を使用して DN に内部的にマップします。

つまり、チャネル定義での MCAUSER 値は DISPLAY CHSTATUS と同じストリングが表示されない可能性がありますが、同じ ID を参照します。

以下に例を示します。

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

DISPLAY CHSTATUS(*) ALL は SHORTUSR 値 (すべての接続の *MCAUSER(jodoe)*) を表示します。

OS と LDAP 認証モデルの切り替え

様々なプラットフォームで様々な認証方法を切り替える方法。

キュー・マネージャーの CONNAUTH 属性は、AUTHINFO オブジェクトでポイントします。 オブジェクトが IDPWLDAP タイプの場合、LDAP リポジトリーが認証に使用されます。

認証方法を同じオブジェクトに適用できるようになり、これにより OS ベースの認証を続行することも、LDAP 認証で処理することもできます。

UNIX プラットフォームと IBM i

IBM i UNIX

キュー・マネージャーは、OS と LDAP モデルの間でいつでも切り替えることができます。 REFRESH SECURITY TYPE (CONNAUTH) コマンドを使用して、構成を変更し、その構成をアクティブにすることができます。

例えば、このオブジェクトが既に認証の接続情報を使用して構成されている場合は、次のようになります。

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
```

```
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

権限構成の変更に OS と LDAP モデル間の切り替えが含まれる場合、その変更を有効にするためには、キュー・マネージャーを再始動する必要があります。 そうでない場合は、REFRESH SECURITY TYPE (CONNAUTH) コマンドを使用して、 変更を有効にすることができます。

ルールの処理

OS から LDAP 認証に切り替えると、設定済みの既存の OS の権限のルールは非アクティブになって非表示になります。

dmpmqaut などのコマンドは、これらの OS ルールを表示しません。 同様に、LDAP から OS に切り替えるとき、定義された LDAP 認証は非アクティブになって非表示になり、元の OS ルールを復元します。

何らかの理由で **dmpmqcfg** コマンドを使用してキュー・マネージャーの定義をバックアップする場合は、バックアップの時点で有効な認証方法に呈して定義されているルールのみが含まれます。

LDAP 管理

各プラットフォームでの LDAP 管理の概要

LDAP 許可を使用する場合、オペレーティング・システムにおける mqm グループ (またはそれに相当するもの) のメンバーシップはそれほど重要ではありません。 そのグループのメンバーであることにより制御されるのは、一部のコマンド行コマンドを処理できるかどうかということだけです。

具体的には、strmqm コマンドと endmqm コマンドを実行するには、このグループのメンバーでなければなりません。

キュー・マネージャーの実行中に、フルに特権を持つアカウントに制限が付くようになりました。 OS の mqm グループ (またはそれに相当するもの) に属していても、**strmqm** コマンドを実行する人のユーザー ID 以外では、特別の特権は付与されません。

他のユーザーの許可は、そのユーザーがどの LDAP グループに属するかに基づいて決まります。 **setmqaut** などのコマンドにおいて、mqm グループ名を修飾せずに使用していずれかの LDAP グループにマップすることはできません。

UNIX プラットフォーム

UNIX

キュー・マネージャーの実行中に自動的にフル特権を付与される唯一のアカウントは、キュー・マネージャーの開始中を開始した実ユーザーです。

mqm はキュー・マネージャーを実行できる有効な ID なので、mqm ID は存在し、ファイルなどの OS リソースの所有者として使用されます。 ただし、mqm ユーザーは、OAM によって制御される管理用タスクを自動的に実行することはできなくなります。

IBM i

IBM i

IBM i の場合、自動的に特権を付与されるアカウントは、キュー・マネージャーを開始するアカウントと QMQM ID です。

キュー・マネージャーを開始するユーザー ID はシステムを開始するためだけに必要なので、両方の ID が必要です。 いったん実行されると、キュー・マネージャー・プロセスは QMQM 権限のみを持ちます。

Windows プラットフォーム

Windows

Windows で、全権限が自動的に付与されるアカウントは、キュー・マネージャーを開始した OS ユーザーです。また、キュー・マネージャーを Windows サービスとして開始した場合は、キュー・マネージャーのコア・プロセスを実行するユーザー (MUSR_MQADMIN など) です。

LDAP 許可モードで実行する場合、Windows は UNIX プラットフォームと非常に類似した動作になります。12 文字のショート・ネームとフル DN を扱います。

サンプル・スクリプト

キュー・マネージャーで管理作業をフルに実行できるグループがあることは役に立つので、UNIX プラットフォームでは、以下のサンプル・スクリプトが同梱されます。

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

このサンプル・スクリプトは以下の 2 つのパラメーターを受け取ります。

- キュー・マネージャー名
- LDAP グループ名

このサンプル・スクリプトは setmqaut コマンドを処理し、すべてのオブジェクトに全権限を付与します。これは、管理ロール用の IBM MQ Explorer OAM ウィザードによって生成されるスクリプトと同じです。 例えば、コードは次のように始まります。

```
setmqaut -t q -m qmgr -n "**" +alladm +allmqi -g
      groupname
```

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。 IBM 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、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、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。

V 9.1.4 z/OS キュー・マネージャーでメッセージを保存状態のまま暗号化する必要がある場合は、そのキュー・マネージャーで z/OS データ・セット暗号化を使用することができます。

Advanced Message Security の詳細については、102 ページの『計画 Advanced Message Security』を参照してください。 IBM MQ での TLS の使用については、22 ページの『IBM MQ での TLS セキュリティー・プロトコル』を参照してください。 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、444 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。

データ・セット暗号化による IBM MQ for z/OS での保存データの機密性 のセクションを参照してください。z/OS データ・セット暗号化の詳細については、

関連タスク

TLS による 2 つのキュー・マネージャーの接続

キュー・マネージャーへのクライアントのセキュア接続

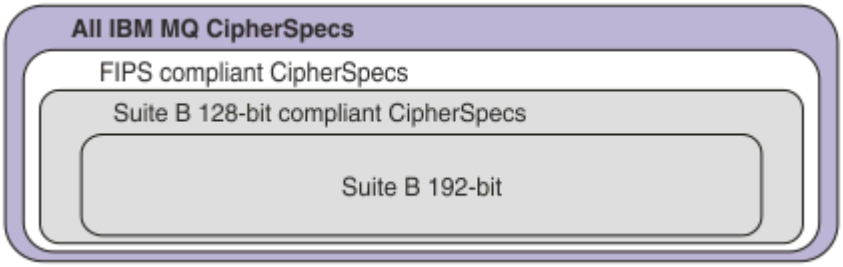
CipherSpecs の有効化

CipherSpec は、**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのどちらかにおいて、**SSLCIPH** パラメーターを使用することにより有効にします。

IBM MQ とともに使用できる CipherSpec の一部は FIPS 準拠です。 FIPS 準拠の CipherSpec の一部は Suite B 準拠でもありますが、それ以外 (TLS_RSA_WITH_AES_256_CBC_SHA など) 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。



IBM MQ 8.0.0 Fix Pack 3 以降、サポートされる CipherSpecs の数が削減されました。

V 9.1.1 デフォルト CipherSpec の構成について詳しくは、420 ページの『IBM MQ で有効化されているデフォルト CipherSpec 値』を参照してください。 MQ チャネルで使用可能な CipherSpec の代替セットを提供することもできます。 421 ページの『有効な CipherSpec のカスタム・リストの提供 (Multiplatforms)』を参照してください。

非推奨の CipherSpec の有効化について詳しくは、421 ページの『非推奨の CipherSpec の有効化 (Multiplatforms)』または 422 ページの『非推奨の CipherSpec の有効化 (z/OS)』を参照してください。 IBM MQ で使用するために再有効化できる CipherSpec のリストについては、425 ページの『推奨されない CipherSpec』を参照してください。

V 9.1.4 ULW IBM MQ 9.1.4 以降、IBM MQ は UNIX, Linux, and Windows で TLS 1.3 セキュリティー・プロトコルをサポートします。 これらの CipherSpecs の使用方法については、420 ページの『IBM MQ での TLS 1.3 の使用』 および 420 ページの『IBM MQ MQI client および TLS 1.3』 を参照してください。

IBM MQ の TLS サポートで使用できる CipherSpecs

次の表に、IBM MQ キュー・マネージャーで自動的に使用できる 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。 TLS 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。

表 *74. IBM MQ の TLS* サポートで使用できる *CipherSpecs*

プラットフォームのサポート 419 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 419 ページの『2』	Suite B
V 9.1.4 V 9.1.4 **別名 CipherSpecs**							

表 74. *IBM MQ* の *TLS* サポートで使用できる *CipherSpecs* (続き)

プラットフォームのサポート 419 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 419 ページの『2』	Suite B
すべて	ANY_TLS13_OR_HIGHER 419 ページの『3』 419 ページの『4』 419 ページの『5』	N/A	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み
すべて	ANY_TLS13 419 ページの『4』 419 ページの『5』 419 ページの『6』	N/A	TLS 1.3	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み
すべて	ANY_TLS12_OR_HIGHER 419 ページの『4』 419 ページの『5』 419 ページの『7』	N/A	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み
すべて	ANY_TLS12 419 ページの『8』	N/A	TLS 1.2	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み
すべて	ANY 419 ページの『9』	N/A	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み	ネゴシエーション済み
V 9.1.4 V 9.1.4 **TLS 1.3 の CipherSpec**							
すべて	TLS_AES_128_GCM_SHA256 419 ページの『4』	1301	TLS 1.3	GCM	AES-128 (GCM (128) を使用)	はい	いいえ
すべて	TLS_AES_256_GCM_SHA384 419 ページの『4』	1302	TLS 1.3	GCM	AES-256 と GCM (256)	はい	いいえ
すべて	TLS_CHACHA20_POLY1305_SHA256 419 ページの『4』	1303	TLS 1.3	POLY1305	CHACHA20 (256)	いいえ	いいえ
ULW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 (CTR (128) を使用)	はい	いいえ
ULW	TLS_AES_128_CCM_8_SHA256 419 ページの『11』	1305	TLS 1.3	CBC-MAC	AES-128 (CTR (128) を使用)	はい	いいえ
TLS 1.2 の CipherSpec							

表 74. *IBM MQ* の *TLS* サポートで使用できる *CipherSpecs* (続き)

プラットフォームのサポート 419 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 419 ページの『2』	Suite B
すべて	TLS_RSA_WITH_AES_128_CBC_SHA256 419 ページの『10』	003C	TLS 1.2	SHA-256	AES (128)	はい	いいえ
すべて	TLS_RSA_WITH_AES_256_CBC_SHA256 419 ページの『10』 419 ページの『12』	003D	TLS 1.2	SHA-256	AES (256)	はい	いいえ
すべて	TLS_RSA_WITH_AES_128_GCM_SHA256 419 ページの『10』 419 ページの『13』	009C	TLS 1.2	SHA-256 および AEAD GCM	AES (128)	はい	いいえ
すべて	TLS_RSA_WITH_AES_256_GCM_SHA384 419 ページの『10』 419 ページの『12』 419 ページの『13』	009D	TLS 1.2	SHA-384 および AEAD GCM	AES (256)	はい	いいえ
すべて	ECDHE_ECDSA_AES_128_CBC_SHA256 419 ページの『10』	C023	TLS 1.2	SHA-256	AES (128)	はい	いいえ
すべて	ECDHE_ECDSA_AES_256_CBC_SHA384 419 ページの『10』 419 ページの『12』	C024	TLS 1.2	SHA-384	AES (256)	はい	いいえ
すべて	ECDHE_RSA_AES_128_CBC_SHA256 419 ページの『10』	C027	TLS 1.2	SHA-256	AES (128)	はい	いいえ
すべて	ECDHE_RSA_AES_256_CBC_SHA384 419 ページの『10』 419 ページの『12』	C028	TLS 1.2	SHA-384	AES (256)	はい	いいえ
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 419 ページの『12』 419 ページの『13』	C02B	TLS 1.2	SHA-256 および AEAD GCM	AES (SHA384)	はい	128 ビット
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 419 ページの『12』 419 ページの『13』	C02C	TLS 1.2	SHA-384 および AEAD GCM	AES (SHA384)	はい	192 ビット
すべて	ECDHE_RSA_AES_128_GCM_SHA256 419 ページの『13』	C02F	TLS 1.2	SHA-256 および AEAD GCM	AES (128)	はい	いいえ
すべて	ECDHE_RSA_AES_256_GCM_SHA384 419 ページの『12』 419 ページの『13』	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	はい	いいえ

表 74. *IBM MQ* の *TLS* サポートで使用できる *CipherSpecs* (続き)

プラットフォームのサポート 419 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 419 ページの『2』	Suite B

注 :

1. 各プラットフォーム・アイコンでカバーしているプラットフォームのリストについては、製品資料で使用するリリースとプラットフォームのアイコンを参照してください。
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. ULW ANY_TLS13_OR_HIGHER エイリアス CipherSpec では、リモート・エンドで TLS 1.3 以上のプロトコルを使用した接続のみが許可される最上位のセキュリティーがネゴシエーションされます。
4. z/OS TLS 1.3 または ANY CipherSpec を IBM MQ for z/OS で使用するには、オペレーティング・システムが z/OS 2.4 以降である必要があります。
5. IBM i TLS 1.3 または ANY CipherSpec を IBM i で使用するには、基礎となるオペレーティング・システム・バージョンで TLS 1.3 がサポートされている必要があります。 詳しくは、TLSv1.3 のシステム TLS サポートを参照してください。
6. ULW ANY_TLS13 エイリアス CipherSpec は、TLS 1.3 プロトコルを使用する、受け入れ可能な CipherSpec のサブセットを表します。以下の表にプラットフォームごとのリストがあります。
7. ULW ANY_TLS12_OR_HIGHER エイリアス CipherSpec では、リモート・エンドで TLS 1.2 以上のプロトコルを使用した接続のみが許可される最上位のセキュリティーがネゴシエーションされます。
8. ANY_TLS12 CipherSpec は、TLS 1.2 プロトコルを使用する、受け入れ可能な CipherSpec のサブセットを表します。以下の表にプラットフォームごとのリストがあります。
9. ULW ANY エイリアス CipherSpec では、リモート・エンドで許可を与える最上位のセキュリティーがネゴシエーションされます。
10. IBM i これらの CipherSpec は、システム値 QSSLCSLCTL が *OPSSYS に設定されている IBM i 7.4 システムでは有効になっていません。
11. ULW これらの CipherSpecs は、16 オクテットの整合性検査値 (ICV) ではなく 8 オクテットの ICV を使用します。
12. IBM MQ Explorer 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。
13. Windows Linux GSKit の推奨に従って、TLS 1.2 GCM CipherSpecs には制限があります。つまり、同じセッション鍵を使用して 2 ^24.5 個の TLS レコードが送信されると、接続はメッセージ AMQ9288E で終了します。 この GCM 制限は、使用されている FIPS モードに関係なくアクティブです。

    このエラーが発生しないようにするには、TLS 1.2 GCM 暗号を使用しないようにするか、秘密鍵のリセットを有効にするか、環境変数 GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE を設定して IBM MQ キュー・マネージャーまたはクライアントを開始します。 GSKit ライブラリーの場合、この環境変数を接続の両側で設定し、クライアントからキュー・マネージャーへの接続とキュー・マネージャーからキュー・マネージャーへの接続の両方に適用する必要があります。 この設定は、非管理対象 .NET クライアントには影響しますが、 Java または管理対象 .NET クライアントには影響しないことに注意してください。 詳しくは、 AES-GCM 暗号制限を参照してください。

    この制限は、IBM MQ for z/OS には適用されません。

IBM MQ での TLS 1.3 の使用

V 9.1.4 ULW

IBM MQ 9.1.4 以降、IBM MQ は UNIX, Linux, and Windows で TLS 1.3 をサポートします。 サポートされるいずれのインストール済み環境でも、新規キュー・マネージャーが作成されるとき、qm.ini ファイルの SSL スタンザに、以下に示す項目が設定されます。

```
SSL:
  AllowTLSV13=TRUE
```

注: ファイル qm.ini は、ディレクトリー *<data directory>*/qmgrs/*<qmgr name>*にあります。

IBM MQ 9.1.4 より前のバージョンの IBM MQ を使用して作成されたキュー・マネージャーが、後で IBM MQ 9.1.4 以上を使用して開始された場合、 **AllowTLSV13** プロパティーは設定されません。 TLS 1.3 を有効にする場合は、 qm.ini file を編集し、例に示すようにプロパティーを追加する必要があります (「SSL:」スタンザがまだ存在しない場合は、このスタンザを含めます)。

この .ini ファイル・プロパティーによって TLS 1.3 が有効になり、TLS 1.3 CipherSpec を使用できるようになります。 TLS 1.3 仕様に従って、脆弱な CipherSpec を使用した通信の試みは、IBM MQ で有効かどうかに関係なく拒否されます。 TLS 1.3 で脆弱と見なされる CipherSpec は、以下の基準を 1 つ以上満たす CipherSpec です。

- SSL 3.0 プロトコルを使用している。
- 暗号化アルゴリズムとして RC4 または RC2 を使用している。
- 暗号鍵のサイズ (ビット) が 112 以下である。

これらの制限は、非推奨の CipherSpec の表 1 の注 [10] として示されています。

そのような CipherSpec を引き続き使用する必要がある場合は、TLS 1.3 モードを無効にする必要があります。 これを行うには、キュー・マネージャーの qm.ini ファイルを編集し、**AllowTLSV13** プロパティーの設定を次のように変更します。

```
SSL:
  AllowTLSV13=FALSE
```

注: これが設定されていると、TLS 1.3 CipherSpec を使用できません。

IBM MQ MQI client および TLS 1.3

V 9.1.4 ULW

IBM MQ MQI client クライアントを使用するとき、**AllowTLSV13** の値は、アプリケーションで使用されている mqclient.ini ファイルの SSL スタンザで明示的に指定されている場合を除き、暗黙的に設定されます。

- 脆弱な CipherSpec が有効になっている場合、**AllowTLSV13** は FALSE に設定され、TLS 1.3 CipherSpec を使用できなくなります。
- その他の場合、**AllowTLSV13** が TRUE に設定され、新しい TLS 1.3 CipherSpec および別名 CipherSpec を使用できるようになります。

IBM MQ で有効化されているデフォルト CipherSpec 値

V 9.1.1 Multi

デフォルト構成では、IBM MQ は、TLS 1.2 プロトコルと CipherSpec を使用するさまざまな暗号アルゴリズムのサポートを提供します。 互換性を維持するために、IBM MQ は、SSL 3.0 プロトコルおよび TLS 1.0 プロトコルと、セキュリティーの脆弱性の影響を受けやすいことが分かっている、いくつかの暗号アルゴリズムを使用するように構成することもできます。 デフォルト構成で有効化されている CipherSpec のリストは、保守の適用によって変更される場合があります。

以下の制御を使用して、CipherSpec の使用を制限または許可するように IBM MQ を構成できます。

- SSLFIPS を使用して、FIPS 140-2 準拠 CipherSpec のみを許可します。
- ULW SUITEB を使用して、NSA Suite B 準拠 CipherSpec のみを許可します。
- ULW **AllowedCipherSpecs** 環境変数または **AMQ_ALLOWED_CIPHERS** 環境変数を使用して、CipherSpec のカスタム・リストを許可します。
- ULW **AllowWeakCipher** または **AMQ_SSL_WEAK_CIPHER_ENABLE** 環境変数を使用して、非推奨の CipherSpec の使用を許可します。
- z/OS CHINIT JCL で DD ステートメントを使用して、非推奨の CipherSpec の使用を許可します。

注 : AllowedCipherSpecs または **AMQ_ALLOWED_CIPHERS** を使用して CipherSpec のカスタム・リストを指定すると、非推奨の CipherSpec の使用可能性はすべてオーバーライドされます。 NSA Suite B または FIPS 140-2 制限のいずれかを CipherSpec のカスタム・リストと組み合わせて使用する場合は、Suite B または FIPS 140-2 設定で許可されている CipherSpec のみがカスタム・リストに含まれていることを確認する必要があることに注意してください。

有効な CipherSpec のカスタム・リストの提供 (Multiplatforms)

V 9.1.1 Multi

AMQ_ALLOWED_CIPHERS 環境変数を使用するか、`.ini` ファイルの **AllowedCipherSpecs** SSL スタンザ属性を使用して、 IBM MQ チャネルで使用できる CipherSpecs の代替セットを提供することができます。 この設定を使用して、 IBM MQ リスナーが CipherSpecs という名前のいずれかを使用しない限り、着信チャネル開始要求を受け入れないように制限することができます。 この機能は、ANY* CipherSpecs に含まれる CipherSpec を制御するために使用できます。

AMQ_ALLOWED_CIPHERS 環境変数または **AllowedCipherSpecs** SSL スタンザ属性には、以下のものを使用できます。

- 単一の CipherSpec 名
- 再有効化する IBM MQ CipherSpec 名のコンマ区切りリスト
- すべての CipherSpec を表す特殊値 ALL (推奨されません)。

注 : ALL CipherSpecs を有効化すると、SSL 3.0 プロトコルおよび TLS 1.0 プロトコルと多数の脆弱な暗号アルゴリズムが有効化されるため、これは推奨されません。

この設定を構成すると、CipherSpec のデフォルト・リストがオーバーライドされ、IBM MQ は弱い暗号化非推奨設定を無視します (下記参照)。

- IBM MQ リスナーは、指定されたいずれかの CipherSpec を使用する SSL/TLS プロポーザルだけを受け入れます。
- IBM MQ チャネルは、ブランクの SSLCIPH 値、または指定された CipherSpec のいずれかのみを許可します。
- SSLCIPH 値の **runmqsc** タブ完了によって、完了値は指定された CipherSpec のいずれかに制限されます。

例えば、チャネルを定義/変更し、リスナーが ECDHE_RSA_AES_128_GCM_SHA256 または ECDHE_ECDSA_AES_256_GCM_SHA384 を受け入れることを許可するだけの場合は、`qm.ini` ファイルで以下のように設定することができます。

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

AMQP または MQTT チャネルで使用される暗号は、java.security ファイル設定を使用して制限できます。

非推奨の CipherSpec の有効化 (Multiplatforms)

Multi

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。 マルチプラットフォームで非推奨の CipherSpec を指定しようとすると、AMQ8242: SSLCIPH 定義が間違っています」というメッセージが表示され、PCF から MQRCCF_SSL_CIPHER_SPEC_ERROR が返されます。

推奨されない CipherSpec を使用してチャネルを開始することはできません。 非推奨の CipherSpec を使用して開始しようとすると、システムは MQCC_FAILED (2) と、**Reason** MQRC_SSL_INITIALIZATION_ERROR (2393) をクライアントに返します。

環境変数 **AMQ_SSL_WEAK_CIPHER_ENABLE** を設定して、非推奨の 1 つ以上の CipherSpec をサーバーでの実行時に再度有効にしてチャネルを定義することはできます。

AMQ_SSL_WEAK_CIPHER_ENABLE 環境変数には、以下のものを使用できます。

- 単一の CipherSpec 名
- 再有効化する IBM MQ CipherSpec 名のコンマ区切りリスト
- すべての CipherSpec を表す特殊値 ALL (推奨されません)。

注 : すべての CipherSpec を再有効化すると、SSL 3.0 プロトコルおよび TLS 1.0 プロトコルと多数の脆弱な暗号アルゴリズムが有効化されるため、これは推奨されません。

例えば、ECDHE_RSA_RC4_128_SHA256 を再有効化しようとしている場合、以下の環境変数を設定します。

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

または、次のように設定して、qm.ini ファイル内の SSL スタンザを代わりに変更します。

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

非推奨の CipherSpec の有効化 (z/OS)

z/OS

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。 z/OS で非推奨の CipherSpec を指定しようとすると、メッセージ CSQM102E かメッセージ CSQX674E が表示されます。

弱い (非推奨の) CipherSpec を有効にする場合は、CHINIT JCL で以下の DD ステートメントを定義する必要があります。

```
JCL: //CSQXWEAK DD DUMMY
```

注 : すべての非推奨 CipherSpecs がこの DD ステートメントの使用を必要とするわけではありません。 425 ページの『推奨されない CipherSpec』内の表の注 11 を参照してください。

非推奨の SSL 3.0 プロトコルを有効にする場合は、CHINIT JCL で以下の DD ステートメントを定義することも必要です。

```
JCL: //CSQXSSL3 DD DUMMY
```

V 9.1.0 非推奨の TLS 1.0 プロトコルを有効にする場合は、CHINIT JCL で以下の DD ステートメントを定義することも必要です。

```
JCL: //TLS10ON DD DUMMY
```

DD カードの名前が TLS10ON であることに注意してください。これは TLS100N ではなく、TLS 1.0 が ON になっているという意味です。

TLS 1.0 を OFF にするには、次のステートメントを使用します。

```
JCL: //TLS10OFF DD DUMMY
```

脆弱な、または壊れた暗号仕様を使用してリスナーとネゴシエーションをしないようにするには、CHINIT JCL で以下の DD ステートメントを定義する必要があります。

```
JCL: //WCIPSOFF DD DUMMY
```

デフォルトの暗号仕様リスト **System SSL** にリストされている暗号仕様のみを使用してリスナーとネゴシエーションするには、CHINIT JCL で以下の DD ステートメントを定義する必要があります。

```
JCL: //GSKDCIPS DD DUMMY
```

最小レベルおよび固定レベルの CipherSpec

V 9.1.4 ULW

IBM MQ は、以下の 2 つの異なるタイプの CipherSpec をサポートしています。

- **最小レベル**: これは、上限を設定しない CipherSpec です (例えば、ANY、ANY_TLS12_OR_HIGHER、ANY_TLS13_OR_HIGHER)。
- **固定レベル**: これは、特定のプロトコルを指定する CipherSpec です (例えば、ANY_TLS12 や ANY_TLS13、または ECDHE_ECDSA_3DES_EDE_CBC_SHA256 などの特定のアルゴリズム)

セキュリティーを維持しながら構成を最大限に単純化するには、チャネルの両側で**最小レベル**の CipherSpec を使用することをお勧めします。 これにより、新しいバージョンが両側でサポートされる場合に、どちらの側の構成も変更することなく、より高いレベルの TLS プロトコル・バージョンが通信で自動的にサポートされて使用できるようになります。

開始側で **最小レベル** CipherSpec を使用するが、受信側で **固定レベル** CipherSpec を使用すると、接続が拒否され、メッセージ AMQ9631 および AMQ9641 が発行される可能性があります。

別名 CipherSpec 設定のさまざまな結果を含む表については、429 ページの『別名 CipherSpec 設定間の関係』 を参照してください。

関連概念

43 ページの『IBM 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 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。

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。

41 ページの『Suite B 用 IBM MQ の構成』
IBM MQ は、Windows、UNIX and Linux の各プラットフォーム上で、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。

31 ページの『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 TLS 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。

関連タスク

ANY_TLS12_OR_HIGHER CipherSpec を使用するための既存のセキュリティー構成のマイグレーション

関連資料

DEFINE CHANNEL
ALTER CHANNEL
Change Channel、Copy Channel、および Create Channel

ULW AES-GCM 暗号制限

TLS 暗号化に使用される場合に AES-GCM 暗号に適用される制約事項に関するガイド。 これらの制限は IETF および NIST 組織によって課せられ、AES-GCM 暗号を使用する際に、2 つを超える $^{24.5}$ TLS レコードを安全に転送するために同じセッション鍵を使用してはならないことを要求します。

これらの制限について詳しくは、 RFC 9325 Section 4.4 Limits on Key Usage および RFC 8446 section 5.5 を参照してください。

IBM MQ は、暗号機能を直接実装しません。 代わりに、TLS と Advanced Message Security の機能を提供するために、いくつかの異なる暗号ライブラリーが使用されます。 Windows、 Linux、および AIX オペレーティング・システムでは、 IBM MQ が使用する暗号ライブラリーは GSKit です。 アプリケーションの場合、C および非管理 .NET ライブラリーは、暗号機能のために GSKit を使用します。 GSKit による AES-GCM 暗号化アルゴリズムの実装には、標準グループによって指定される制限が含まれます。 また、これらの制限はデフォルトで有効になっています。 そのため、AES-GCM 暗号を使用する場合、 IBM MQ TLS 通信は、2 つを超える $2^{24.5}$ TLS レコードが同じセッション鍵を使用して送信されると終了します。

注 : 異なる暗号ライブラリーが使用され、これらのライブラリーが同じ制限を実装していないため、この制限は IBM i、IBM Z または IBM MQ for HPE NonStop プラットフォーム、あるいは Java/JMS、管理対象 .NET アプリケーションには存在しません。

同じセッション鍵を使用して 2 つを超える $2^{24.5}$ TLS レコードが送信されるのに十分な時間、 IBM MQ チャネルが実行されたままになっている場合、基礎となる暗号ライブラリーは接続を終了します。 これにより、チャネルが終了し、AMQ9288E エラー・メッセージが生成されます。 この方法で通信が終了したアプリケーションは、実行されていた IBM MQ 操作から MQRC_CONNECTION_BROKEN 戻りコードを受け取ります。

接続の終了は、通信のいずれかの側で実行できますが、暗号機能に GSKit を使用している側でのみ実行できます。

制限を緩和するためのアドバイス

この制限のために終了した通信を防止または処理する方法について、いくつかのオプションを以下に示します。

再接続可能クライアントの使用

アプリケーションは、接続が失敗した場合に自動的に再接続を試行するように構成できます。 これには、 GCM の制限のために終了した接続が含まれます。 再接続用に構成されている場合、クライアント・アプリケーションは障害発生時に自動的に復元され、オープン・オブジェクトへのハンドルが復元されます。 これは、アプリケーション・コードに戻ることなく行われます。

詳細については、自動クライアント再接続を参照してください。

秘密鍵のリセット値の設定

IBM MQ は、構成可能なバイト数がチャネルを介して転送された後にセッション鍵のリセットを要求するように構成できます。 この制限に達すると、 IBM MQ は、暗号層がセッション鍵のリセットを実行することを要求します。これにより、新しいセッション鍵が生成されます。

指定される値は転送されたバイト数であり、これは IBM MQ によって送信されるメッセージのサイズに関連することに注意してください。 この制限は、送信される TLS レコードの数に基づきます。 TLS レコードはネットワークの最大伝送単位 (MTU) に依存する最大バイト数を送信できるため、メッセージ・バイトと TLS レコードの間に直接マッピングはありません。 この値より大きい送信メッセージは、複数の TLS レコードとして送信されます。 MTU 値はネットワーク間で異なります。 また、IBM MQ ハートビート・チェック、TLS アラート、その他の IBM MQ プロトコル・メッセージなど、 IBM MQ メッセージ・データの送信の外部で TLS レコードを送信する必要が生じる可能性があるその他の理由もあります。 これらの追加 TLS レコードは、TLS レコードの最大数にカウントされますが、 IBM MQ 秘密鍵リセット値にはカウントされません。

秘密鍵のリセットを使用してセッション鍵を定期的にリセットすると、AES-GCM 制限が原因でチャネルが終了しない可能性があります。

詳しくは、 SSL および TLS 秘密鍵のリセットを参照してください。

V 9.1.4 **TLS 1.3 CipherSpec を使用する**

TLS 1.3 プロトコルを使用する場合、AES-GCM 制限は引き続き存在しますが、TLS 1.3 プロトコルは、TLS 通信を中断することなく、セッション鍵リセットを自動的に実行することをサポートしています。これにより、IBM MQ が秘密鍵のリセットを要求しなくても、必要に応じて GSKit がセッション鍵のリセットを管理できるようになります。

詳しくは、 416 ページの『CipherSpecs の有効化』 の IBM MQ での TLS 1.3 の使用 を参照してください。

AES-GCM 制限を無効にします。
必要に応じて、環境変数 **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** を設定して AES-GCM 制限を無効にすることで、制限を無効にすることができます。 これにより、同じセッション鍵を使用して任意の数の TLS レコードを送信できます。 この緩和策を選択する場合は、セキュア通信に GSKit を使用する通信の両端で環境変数を設定する必要があります。

警告 : このオプションは推奨されません。2 つを超える $2^{24.5}$ TLS レコードが送信された後、攻撃者が送信されたレコードを分析して、使用中のセッション鍵を判別する可能性があるためです。セッション鍵が判別されると、そのセッション鍵を使用する既存および将来のすべての通信が危険にさらされます。

推奨されない CipherSpec

必要に応じて IBM MQ で使用できる、推奨されない CipherSpec のリスト。

非推奨の CipherSpec の有効化について詳しくは、421 ページの『非推奨の CipherSpec の有効化 (Multiplatforms)』 または 422 ページの『非推奨の CipherSpec の有効化 (z/OS)』 を参照してください。

次の表に、IBM 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。

表 *75. IBM MQ* で使用するために再び使用可能にできる非推奨の *CipherSpec*

プラットフォームのサポート 428 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 428 ページの『2』	Suite B	非推奨時の更新
SSL 3.0 の CipherSpec								
IBM i	AES_SHA_US 428 ページの『3』	002F	SSL 3.0	SHA-1	AES (128)	いいえ	いいえ	9.0.0.0
すべて	DES_SHA_EXPORT 428 ページの『3』 428 ページの『4』 428 ページの『5』	0009	SSL 3.0	SHA-1	DES (56)	いいえ	いいえ	9.0.0.0
ULW	DES_SHA_EXPORT1024 428 ページの『3』 428 ページの『6』	0062	SSL 3.0	SHA-1	DES (56)	いいえ	いいえ	9.0.0.0
ULW	FIPS_WITH_DES_CBC_SHA 428 ページの『3』	FEFE	SSL 3.0	SHA-1	DES (56)	いいえ 428 ページの『7』	いいえ	9.0.0.0
ULW	FIPS_WITH_3DES_EDE_CBC_SHA 428 ページの『3』	FEFF	SSL 3.0	SHA-1	3DES (168)	いいえ 428 ページの『8』	いいえ	9.0.0.1 および 9.0.1
すべて	NULL_MD5 428 ページの『3』	0001	SSL 3.0	MD5	なし	いいえ	いいえ	9.0.0.1
すべて	NULL_SHA 428 ページの『3』	0002	SSL 3.0	SHA-1	なし	いいえ	いいえ	9.0.0.1
すべて	RC2_MD5_EXPORT 428 ページの『3』 428 ページの『4』 428 ページの『5』	0006	SSL 3.0	MD5	RC2 (40)	いいえ	いいえ	9.0.0.0

表 75. *IBM MQ* で使用するために再び使用可能にできる非推奨の *CipherSpec* (続き)

プラットフォームのサポート 428 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 428 ページの『2』	Suite B	非推奨時の更新
すべて	RC4_MD5_EXPORT 428 ページの『4』 428 ページの『3』	0003	SSL 3.0	MD5	RC4 (40)	いいえ	いいえ	9.0.0.0
すべて	RC4_MD5_US 428 ページの『3』	0004	SSL 3.0	MD5	RC4 (128)	いいえ	いいえ	9.0.0.0
すべて	RC4_SHA_US 428 ページの『3』 428 ページの『5』	0005	SSL 3.0	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ULW	RC4_56_SHA_EXPORT1024 428 ページの『3』 428 ページの『6』	0064	SSL 3.0	SHA-1	RC4 (56)	いいえ	いいえ	9.0.0.0
すべて	TRIPLE_DES_SHA_US 428 ページの『3』 428 ページの『5』	000A	SSL 3.0	SHA-1	3DES (168)	いいえ	いいえ	9.0.0.1 および 9.0.1
TLS 1.0 の CipherSpec								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 428 ページの『3』	0006	TLS 1.0	MD5	RC2 (40)	いいえ	いいえ	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5 428 ページの『3』 428 ページの『4』	0003	TLS 1.0	MD5	RC4 (40)	いいえ	いいえ	9.0.0.0
すべて	TLS_RSA_WITH_DES_CBC_SHA 428 ページの『3』	0009	TLS 1.0	SHA-1	DES (56)	いいえ 428 ページの『9』	いいえ	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 428 ページの『3』	0001	TLS 1.0	MD5	なし	いいえ	いいえ	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA 428 ページの『3』	0002	TLS 1.0	SHA-1	なし	いいえ	いいえ	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 428 ページの『3』	0004	TLS 1.0	MD5	RC4 (128)	いいえ	いいえ	9.0.0.0
ULW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA 428 ページの『10』	002F	TLS 1.0	SHA-1	AES (128)	はい	いいえ	9.0.5
ULW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA 428 ページの『6』 428 ページの『10』	0035	TLS 1.0	SHA-1	AES (256)	はい	いいえ	9.0.5
すべて	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	はい	いいえ	9.0.0.1 および 9.0.1
TLS 1.2 の CipherSpec								
ULW	ECDHE_ECDSA_NULL_SHA256 428 ページの『3』	C006	TLS 1.2	SHA-1	なし	いいえ	いいえ	9.0.0.1

表 75. *IBM MQ* で使用するために再び使用可能にできる非推奨の *CipherSpec* (続き)

プラットフォームのサポート 428 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 428 ページの『2』	Suite B	非推奨時の更新
ULW	ECDHE_ECDSA_RC4_128_SHA256 428 ページの『3』	C007	TLS 1.2	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ULW IBM i	ECDHE_RSA_NULL_SHA256 428 ページの『3』	C010	TLS 1.2	SHA-1	なし	いいえ	いいえ	9.0.0.1
ULW IBM i	ECDHE_RSA_RC4_128_SHA256 428 ページの『3』	C011	TLS 1.2	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ULW	TLS_RSA_WITH_NULL_NULL 428 ページの『3』	0000	TLS 1.2	なし	なし	いいえ	いいえ	9.0.0.1
すべて	TLS_RSA_WITH_NULL_SHA256 428 ページの『3』	003B	TLS 1.2	SHA-256	なし	いいえ	いいえ	9.0.0.1
ULW	TLS_RSA_WITH_RC4_128_SHA256 428 ページの『3』	0005	TLS 1.2	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ULW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	はい	いいえ	9.0.0.1 および 9.0.1
ULW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	はい	いいえ	9.0.0.1 および 9.0.1

表 75. *IBM MQ* で使用するために再び使用可能にできる非推奨の *CipherSpec* (続き)

プラットフォームのサポート 428 ページの『1』	CipherSpec 名	16 進コード	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)	FIPS 428 ページの『2』	Suite B	非推奨時の更新

注 :

1. 各プラットフォーム・アイコンでカバーしているプラットフォームのリストについては、製品資料で使用するリリースとプラットフォームのアイコンを参照してください。
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. ULW これらの CipherSpec は、TLS 1.3 が (`qm.ini` の AllowTLSV13 プロパティーを使用して) 有効になっている場合は使用不可になります。

   z/OS IBM MQ for z/OS 9.2.0 以降で作成されたキュー・マネージャーは、デフォルトで TLS 1.3 を有効にします。これにより、これらの CipherSpecs が無効になります。 必要に応じて、TLS V1.3 を無効にすることで、これらの CipherSpecs を有効にできます。 これを行うには、**AllowTLSV13**=*FALSE* をキュー・マネージャー JCL の QMINI データ・セットの TransportSecurity スタンザに追加します。 以前のバージョンから IBM MQ for z/OS 9.2.0 にマイグレーションされたキュー・マネージャーでは、TLS 1.3 がデフォルトで有効になっていないため、これらの CipherSpecs が有効になっています。
4. 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。
5. これらの CipherSpec は、IBM MQ classes for Java または IBM MQ classes for JMS ではサポートされなくなりました。 詳しくは、IBM MQ classes for Java での SSL/TLS の CipherSpec と CipherSuite または IBM MQ classes for JMS での SSL/TLS の CipherSpec と CipherSuite を参照してください。
6. ハンドシェークの鍵サイズは 1024 ビットです。
7. 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。
8. `FIPS_WITH_3DES_EDE_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 の使用は推奨されません。
9. 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。
10. z/OS これらの CipherSpec のみを再度有効にする場合、CSQXWEAK DD ステートメントを使用する必要はありません。

関連概念

43 ページの『IBM 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 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

別名 CipherSpec 設定間の関係

以下の表は、クライアント、キュー・マネージャー、またはその両方で TLS1.3 が有効になっていない場合、およびクライアントとキュー・マネージャーの両方で TLS1.3 が有効になっている場合の予期される動作を示しています。

以下の表は、さまざまな別名 CipherSpec 設定と、予期される結果の関係を示しています。 429 ページの表 76 は、クライアント、サーバー、またはその両方で TLS 1.3 が有効になっていない場合に予期される動作を示しています。 429 ページの表 77 は、クライアントとサーバーの両方で TLS 1.3 が有効になっている場合に予期される動作を示しています。 どちらの場合も、クライアントの CipherSpecs が表の Y 軸に表示され、サーバーの CipherSpecs が表の X 軸に表示されます。

注 : エントリーに 「失敗する可能性が高い」 と示されているのは、使用されている特定の TLS 1.3 または TLS 1.2 CipherSpec が、クライアントおよびキュー・マネージャーにとって最も強い CipherSpec である場合、TLS ハンドシェークはそれを使用するように解決され、チャネル SSCIPH 値と一致するためです。

表 *76.* クライアント、サーバー、またはその両方で *TLS 1.3* が有効になっていない場合に予期される動作

	サーバー			
クライアント	**特定の TLS 1.2 CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ OR_HIGHER**
特定の TLS 1.2 CipherSpec	接続	接続	接続	接続
ANY	失敗する可能性が高い	接続	接続	接続
ANY_ TLS12	失敗する可能性が高い	接続	接続	接続
ANY_TLS12_ OR_HIGHER	失敗する可能性が高い	接続	接続	接続

表 *77.* クライアントとサーバーの両方で *TLS 1.3* が有効になっている場合に予期される動作

	サーバー						
クライアント	**特定の TLS 1.2 CipherSpec**	**特定の TLS 1.3 CipherSpec**	**ANY**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_HIGHER**	**ANY_TLS13_ OR_HIGHER**
特定の TLS 1.2 CipherSpec	接続	**失敗**	接続	接続	**失敗**	接続	**失敗**
特定の TLS 1.3 CipherSpec	**失敗**	接続	接続	**失敗**	接続	接続	接続
ANY	**失敗**	失敗する可能性が高い	接続	**失敗**	接続	接続	接続
ANY_TLS12	失敗する可能性が高い	**失敗**	接続	接続	**失敗**	接続	**失敗**
ANY_TLS13	**失敗**	失敗する可能性が高い	接続	**失敗**	接続	接続	接続
ANY_TLS12_ OR_HIGHER	**失敗**	失敗する可能性が高い	接続	**失敗**	接続	接続	接続

表 77. クライアントとサーバーの両方で *TLS 1.3* が有効になっている場合に予期される動作 (続き)							
	サーバー						
クライアント	**特定の TLS 1.2 CipherSpec**	**特定の TLS 1.3 CipherSpec**	**ANY**	**ANY_TLS12**	**ANY_TLS13**	**ANY_TLS12_OR_HIGHER**	**ANY_TLS13_OR_HIGHER**
ANY_TLS13_OR_HIGHER	**失敗**	失敗する可能性が高い	接続	**失敗**	接続	接続	接続

関連概念

43 ページの『IBM 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 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。

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。

416 ページの『CipherSpecs の有効化』
CipherSpec は、**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのどちらかにおいて、**SSLCIPH** パラメーターを使用することにより有効にします。

関連タスク

ANY_TLS12_OR_HIGHER CipherSpec を使用するための既存のセキュリティー構成のマイグレーション

IBM MQ Explorer を使用した CipherSpec についての情報の取得

IBM MQ Explorer を使用して、CipherSpec の説明を表示できます。

416 ページの『CipherSpecs の有効化』 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 MQ Explorer を開き、「**キュー・マネージャー**」フォルダーを展開します。
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 TLS 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 IBM 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 TLS 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、416 ページの『CipherSpecs の有効化』 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 TLS のバージョンによってサポートされ、かつ有効である TLS 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、「セキュア・セッションの文字情報の設定」 の使用上の注意セクションのポイント 3 を参照してください。

重要 : SSLCIPH では 16 進数の暗号値は指定しないでください。これは、どの暗号が使用されるかが値から不明確であることと、使用するプロトコルの選択が不確定になるためです。 16 進数の暗号値を使用すると、CipherSpec の不一致エラーが発生する可能性があります。

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 4 文字のストリング。 この 16 進コードは、TLS プロトコルで定義されている値に相当します。

詳細については、サポートされているすべての TLS 1.0、TLS 1.2、および TLS 1.3 暗号仕様のリストが 4 桁の 16 進数コード形式で示されている 「Cipher Suite Definitions」 を参照してください。

IBM MQ クラスターの考慮事項

IBM MQ クラスターを使用する場合は、416 ページの『CipherSpecs の有効化』 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、459 ページの『SSL/TLS とクラスター』を参照してください。

IBM MQ MQI client 用の CipherSpec の指定

IBM MQ MQI client の CipherSpec を指定するためのオプションが 3 つあります。

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 MQ classes for Java および IBM MQ classes for JMS を使用した CipherSuite の指定

IBM MQ classes for Java および IBM MQ classes for JMS の CipherSuite の指定は、他のプラットフォームとは異なります。

IBM MQ classes for Java を使用した CipherSuite の指定の詳細は、Java での Transport Layer Security (TLS) サポートを参照してください。

IBM MQ classes for JMS を使用した CipherSuite の指定の詳細は、IBM MQ classes for JMS での Transport Layer Security (TLS) の使用を参照してください。

IBM MQ.NET 用の CipherSpec の指定

IBM MQ.NET では、MQEnvironment クラスを使用するか、接続プロパティーのハッシュ・テーブルの MQC.SSL_CIPHER_SPEC_PROPERTY を使用して CipherSpec を指定できます。

.NET のアンマネージド・クライアント用の CipherSpec の指定については、.NET アンマネージド・クライアントの TLS の有効化を参照してください。

.NET マネージド・クライアント用の CipherSpec の指定については、.NET マネージド・クライアントの CipherSpec サポートを参照してください。

z/OS IBM MQ for z/OS での AT-TLS の使用

Application Transparent Transport Layer Security (AT-TLS) は、TLS サポートを実装するアプリケーションを使用せずに、または TLS が使用されていることを認識することなく、z/OS アプリケーションに TLS サポートを提供します。 AT-TLS は、z/OS でのみ使用可能です。

AT-TLS は、IBM MQ for z/OS のすべてのバージョンで使用できます。

AT-TLS を IBM MQ for z/OS で使用する前に、関係する 434 ページの『制限対象機能』 を必ず理解してください。

Application Transparent Transport Layer Security を使用するには、どの TCP/IP 接続で TLS を透過的に有効にするかを決定するために z/OS Communications Server によって使用される一連の規則を含むポリシーステートメントを定義します。

IBM MQ for z/OS には独自の TLS 実装があり、これにはサポートされる CipherSpec を使用して構成された SSLCIPH パラメーターがチャネルに必要です。

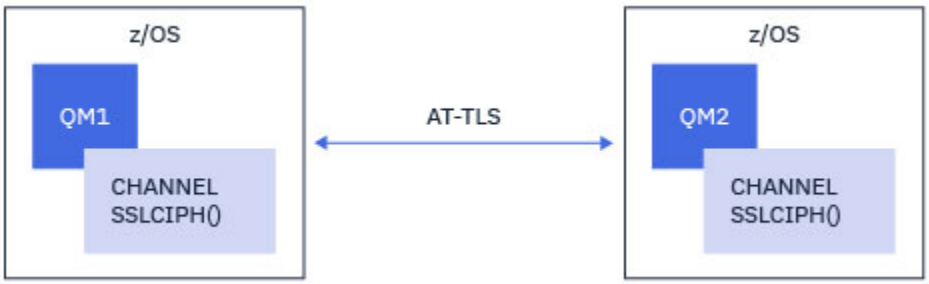
チャネルで TLS を使用可能にすることに決定したら、IBM MQ 管理者は、AT-TLS または IBM MQ TLS を使用するかどうかを決定できます。 多くの場合、この決定は AT-TLS が他のミドルウェアに使用されているか、パフォーマンスへの影響があるかに基づいて行われます。 AT-TLS と IBM MQ TLS のパフォーマンスの基本的な比較については、MP 16: の容量の計画と、IBM MQ for z/OS のチューニングを参照してください。

シナリオ

IBM MQ との AT-TLS の使用は、以下のシナリオでサポートされています。

シナリオ 1

チャネルの両側が AT-TLS を使用する 2 つの IBM MQ for z/OS キュー・マネージャー間。 つまり、どちらのチャネルも SSLCIPH 属性を指定していません。 この方法は、どのメッセージ・チャネルでも使用できます。

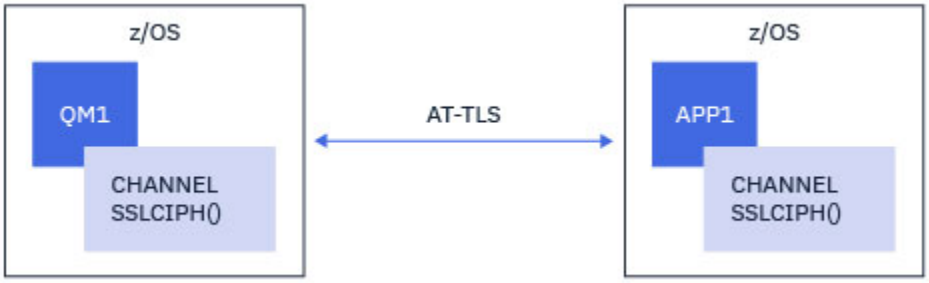


このシナリオの実装は、チャネルの両側に 1 つずつ、2 つの AT-TLS ポリシーを定義することから構成されます。 これらのポリシーは、 シナリオ 3 で使用したものと同じです。

例えば、チャネルが単一の名前の CipherSpec を使用して AT-TLS を使用するように変更されている場合、アウトバウンド・チャネルは 435 ページの『単一の名前付き CipherSpec を使用して、アウトバウンド・チャネル上の AT-TLS を IBM MQ for Multiplatforms キュー・マネージャーに構成する』 からのポリシーを使用し、インバウンド・チャネルは 438 ページの『単一の名前付き CipherSpec を使用して、IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネル上での AT-TLS の構成』 からのポリシーを使用します。

シナリオ 2

IBM MQ for z/OS キュー・マネージャーと、チャネルの両側で AT-TLS を使用する IBM MQ で実行されている Java z/OS クライアント・アプリケーションの間。 つまり、サーバー接続チャネルもクライアント接続チャネルも SSLCIPH 属性を指定することはありません。

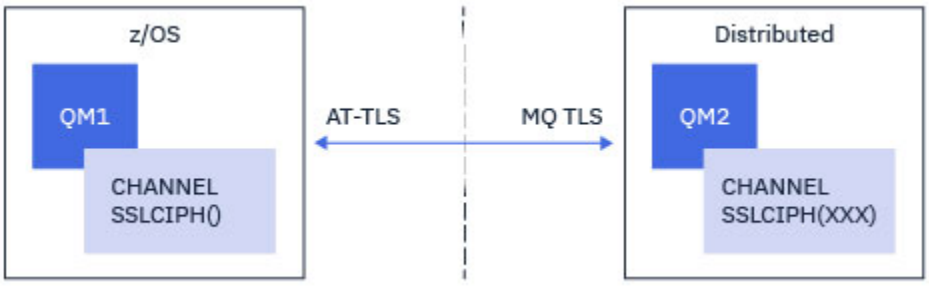


このシナリオの実装は、チャネルの両側に 1 つずつ、2 つの AT-TLS ポリシーを定義することから構成されます。 これらのポリシーは、 シナリオ 3 で使用したものと同じです。

例えば、チャネルが単一の名前の CipherSpec を使用して AT-TLS を使用するように変更されている場合、クライアント接続チャネルは 435 ページの『単一の名前付き CipherSpec を使用して、アウトバウンド・チャネル上の AT-TLS を IBM MQ for Multiplatforms キュー・マネージャーに構成する』 からのポリシーを使用し、サーバー接続チャネルは 438 ページの『単一の名前付き CipherSpec を使用して、IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネル上での AT-TLS の構成』 からのポリシーを使用します。

シナリオ 3

IBM MQ for z/OS キュー・マネージャーと IBM MQ for Multiplatforms で実行されているキュー・マネージャーの間で、IBM MQ for z/OS キュー・マネージャーは AT-TLS を使用し、IBM MQ for Multiplatforms キュー・マネージャーは IBM MQ TLS を使用します。 これは、クラスター送信側およびクラスター受信側以外のすべてのメッセージ・チャネル・タイプに適用されます。

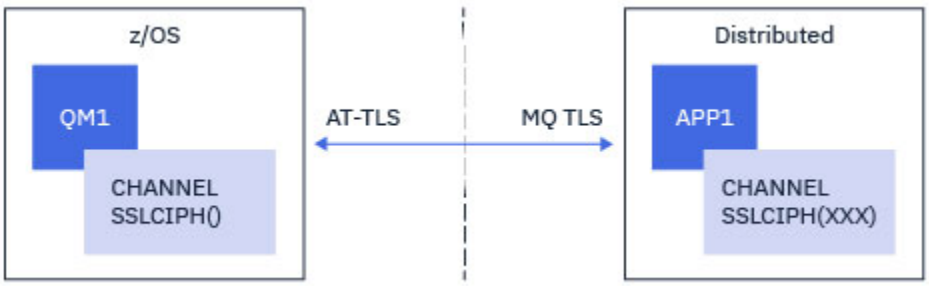


435 ページの『単一の名前付き CipherSpec を使用して、アウトバウンド・チャネル上の AT-TLS を IBM MQ for Multiplatforms キュー・マネージャーに構成する』 キュー・マネージャーから IBM MQ for z/OS キュー・マネージャーへのアウトバウンド・チャネル用の AT-TLS 構成例、および IBM MQ for Multiplatforms キュー・マネージャーから 438 ページの『単一の名前付き CipherSpec を使用して、IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネル上での AT-TLS の構成』 キュー・マネージャーへのインバウンド・チャネル用の IBM MQ for Multiplatforms 構成の例については、IBM MQ for z/OS を参照してください。

両方のキュー・マネージャーが z/OS 上にあるが、右側のキュー・マネージャーが AT-TLS を使用するように構成されていない場合は、同じ AT-TLS 構成を使用できます。

シナリオ 4

IBM MQ for z/OS キュー・マネージャーと IBM MQ for Multiplatforms 上で実行されているクライアント・アプリケーションの間で、IBM MQ for z/OS キュー・マネージャーは AT-TLS を使用し、クライアント・アプリケーションは CipherSpec という名前の単一の SSLCIPH 属性を指定して IBM MQ TLS を使用します。



このシナリオでは、インバウンド・メッセージ・チャネルによって使用される要件と同じ要件を満たす単一の AT-TLS ポリシーが必要です。438 ページの『単一の名前付き CipherSpec を使用して、IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネル上での AT-TLS の構成』 を参照してください。

クライアント・アプリケーションが Java アプリケーションで、また z/OS 上で実行されていても、AT-TLS を使用するように構成されていない場合は、同じ AT-TLS 構成を使用できます。

制限対象機能

IBM MQ for z/OS は AT-TLS 対応ではないため、前述のシナリオにはいくつかの制約事項があります。

- IBM MQ TLS との組み合わせで AT-TLS は、クラスター送信側チャネルおよびクラスター受信側チャネルでは機能しません。
- IBM MQ for z/OS キュー・マネージャーは、それらが AT-TLS を使用していて、パートナー・キュー・マネージャーまたはクライアントから証明書情報を受信しないことを認識していません。 したがって、以下の属性は AT-TLS を使用するチャネルの z/OS 側には影響しません。
  - SSLCAUTH チャネル属性および SSLPEER チャネル属性
  - SSLRKEYC キュー・マネージャー属性
  - CHLAUTH ルールの SSLPEERMAP 属性
- TLS 秘密鍵の再ネゴシエーションを使用するには、チャネルの両側が IBM MQ TLS を使用する必要があります。 したがって、IBM MQ for Multiplatforms キュー・マネージャー、またはクライアントは、AT-TLS を使用して IBM MQ for z/OS キュー・マネージャーに接続する場合は、TLS 秘密鍵の再ネゴシエーションを使用可能にしてはなりません。

  キュー・マネージャーの TLS 秘密鍵の再ネゴシエーションを無効にするには、キュー・マネージャーの SSLRKEYC パラメーターを `0` に設定します。 クライアントの場合は、クライアント・タイプに応じて、該当するパラメーターを `0` に設定します。 これを行う方法の詳細については、442 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。

AT-TLS 構成ステートメント

AT-TLS は、一連のステートメントを使用して構成されます。 このトピックで説明されているシナリオで使用されるものは以下のとおりです。

「TTLSRule」
TCP/IP connection を TLS 構成にマッチングするための基準のセットを指定します。 これは、他のステートメント・タイプを参照します。

TTLSGroupAction
参照 `TTLSRule` を使用可能にするかどうかを指定します。

TTLSEnvironmentAction
参照する `TTLSRule` の詳細な構成を指定し、他のいくつかのステートメントを参照します。

TTLSKeyringParms
AT-TLS によって使用される鍵リングを参照します。

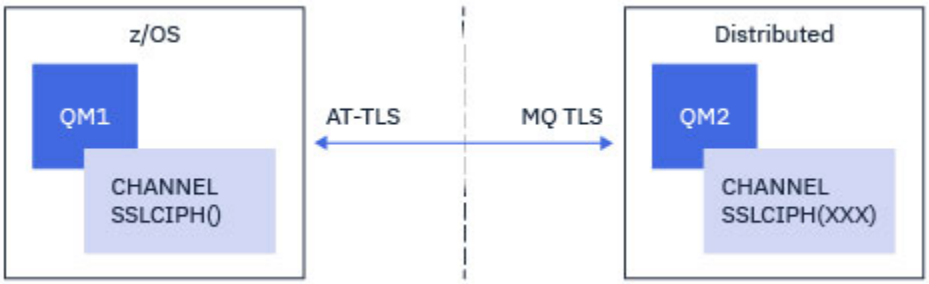
TTLSCipherParms
使用する暗号スイートを定義します。

「**TTLSEnvironmentAdvancedParms**」
使用可能にする TLS プロトコルまたは SSL プロトコルを定義します。

重要 : ここには記載されていない、AT-TLS の他の「AT-TLS ポリシー・ステートメント」があり、必要に応じて IBM MQ で使用することもできます。 ただし、IBM MQ は、このトピックで説明されているポリシーでのみテストされています。

単一の名前付き *CipherSpec* を使用して、アウトバウンド・チャネル上の *AT-TLS* を *IBM MQ for Multiplatforms* キュー・マネージャーに構成する

IBM MQ for z/OS キュー・マネージャーから IBM MQ for Multiplatforms キュー・マネージャーへのアウトバウンド・チャネルで AT-TLS をセットアップする方法。 この場合、z/OS キュー・マネージャー上のチャネルは、SSLCIPH 属性が設定されていない送信側チャネルであり、非 z/OS キュー・マネージャー上のチャネルは、SSLCIPH 属性が単一の名前付き CipherSpec に設定されている受信側チャネルです。



この例では、TLS 1.2 TLS_RSA_WITH_AES_256_GCM_SHA384 CipherSpec を使用する既存の送信側と受信側のチャネルのペアが、 IBM MQ TLS ではなく AT-TLS を使用するように調整されます。

その他の TLS プロトコルおよび CipherSpec を使用するには、構成に軽微調整を行うことができます。 クラスター送信側チャネルとクラスター受信側チャネル以外の他のメッセージ・チャネル・タイプは、AT-TLS 構成を変更せずに使用することができます。

手順

「**ステップ 1: チャネルを停止する**」

「**ステップ 2: AT-TLS ポリシーを作成して適用する**」

このシナリオには、以下の AT-TLS ステートメントを作成する必要があります。

1. チャネル・イニシエーター・アドレス・スペースからターゲット受信側チャネルの IP アドレスおよびポート番号へのアウトバウンド接続に一致する「TTLSRule」ステートメント。 これらの値は、送信側チャネルの CONNAME で使用される情報と一致する必要があります。 ここでは、特定のチャネル・イニシエーター・ジョブ名に一致するフィルター操作が追加されました。

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

このルールは、ポート 1414 で IP アドレス 123.456.78.9 に送信される接続に対して、CSQ1CHIN ジョブから突き合わせを行うものです。

より高度なフィルター・オプションについては、TTLSRule を参照してください。

2. ルールを使用可能にする 「TTLSGroupAction」 ステートメント。 TTLSRule は、**TTLSGroupActionRef** プロパティーを使用して TTLSGroupAction を参照します。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** プロパティーによって TTLSRule に関連付けられた TTLSEnvironmentAction ステートメント。 TTLSEnvironmentAction は TLS 環境を構成し、使用する鍵リングを指定します。

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. **TTLSKeyringParmsRef** プロパティーによって TTLSEnvironmentAction に関連付けられ、AT-TLS によって使用される鍵リングを定義する TTLSKeyringParms ステートメント。

   鍵リングには、リモートの非 z/OS キュー・マネージャーからトラステッド証明書が含まれている必要があります。 この鍵リングは、チャネル・イニシエーターが使用する鍵リングと同じ方法で定義することができます。252 ページの『z/OS システムで TLS を使用するための構成』 を参照してください。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** プロパティーによって TTLSEnvironmentAction に関連付けられた TTLSCipherParms ステートメント。

   このステートメントには、ターゲットの受信側チャネルで使用される IBM MQ CipherSpec 名と同等の名前を指定する必要がある単一の暗号スイート名が含まれている必要があります。

   注 : AT-TLS 暗号スイート名は、必ずしも IBM MQ CipherSpec 名と一致するわけではありません。 ただし、以下の表から IBM MQ CipherSpec 名を見つけ、4 文字のコード列を TTLSCipherParms トピックの表 2 の拡張文字列と相互参照することによって、IBM MQ CipherSpec 名と一致する AT-TLS 暗号スイート名を見つけることができます。

表 *78. 4* 文字コードから *CipherSpec* 名への変換

4 文字コード	**プロトコル**	**デフォルトで有効**	**CipherSpec 名**
0001	SSL 3.0	いいえ	NULL_MD5
0002	SSL 3.0	いいえ	NULL_SHA
0003	SSL 3.0	いいえ	RC4_MD5_EXPORT
0004	SSL 3.0	いいえ	RC4_MD5_US
0005	SSL 3.0	いいえ	RC4_SHA_US
0006	SSL 3.0	いいえ	RC2_MD5_EXPORT
0008	SSL 3.0	いいえ	DES_SHA_EXPORT
0009	TLS 1.0	はい	TLS_RSA_WITH_DES_CBC_SHA

表 78. 4 文字コードから CipherSpec 名への変換 (続き)			
4 文字コード	**プロトコル**	**デフォルトで有効**	**CipherSpec 名**
000A	SSL 3.0	いいえ	TRIPLE_DES_SHA_US
000A	TLS 1.0	はい	TLS_RSA_WITH_3DES_EDE_CBC_SHA
002F	TLS 1.0	はい	TLS_RSA_WITH_AES_128_CBC_SHA
0035	TLS 1.0	はい	TLS_RSA_WITH_AES_256_CBC_SHA
003B	TLS 1.2	はい	TLS_RSA_WITH_NULL_SHA256
003C	TLS 1.2	はい	TLS_RSA_WITH_AES_128_CBC_SHA256
003D	TLS 1.2	はい	TLS_RSA_WITH_AES_256_CBC_SHA256
C023	TLS 1.2	はい	ECDHE_ECDSA_AES_128_CBC_SHA256
C024	TLS 1.2	はい	ECDHE_ECDSA_AES_256_CBC_SHA384
C027	TLS 1.2	はい	ECDHE_RSA_AES_128_CBC_SHA256
C028	TLS 1.2	はい	ECDHE_RSA_AES_256_CBC_SHA384

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_RSA_WITH_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms ステートメントは、 **TTLSEnvironmentAdvancedParmsRef** プロパティーによって TTLSEnvironmentAction に関連付けられます。

   このステートメントを使用して、どの SSL プロトコルおよび TLS プロトコルを使用可能にするかを指定できます。 IBM MQ では、 TTLSCipherParms ステートメントで使用される暗号スイート名に一致する単一プロトコルのみを有効にする必要があります。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        ON
  TLSv1.3        OFF
}
```

   ステートメントの完全なセットは以下のとおりであり、ポリシー・エージェントに適用する必要があります。

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_RSA_WITH_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         ON
  TLSv1.3         OFF
}
```

ステップ 3: z/OS チャネルから SSLCIPH を削除する

以下のコマンドを使用して、 z/OS チャネルから CipherSpec を削除します。

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

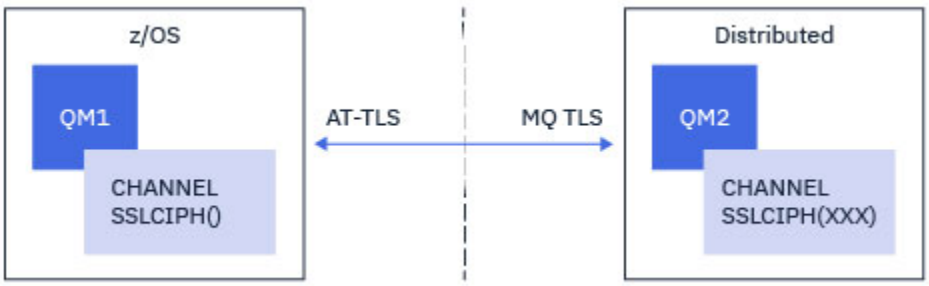
ステップ 4: チャネルを開始する

チャネルが開始されると、そのチャネルは AT-TLS と IBM MQ TLS の組み合わせを使用します。

重要 : 上記の AT-TLS ステートメントは、最小限の構成にすぎません。 ここには記載されていない、AT-TLS の他の「AT-TLS ポリシー・ステートメント」があり、必要に応じて IBM MQ で使用することもできます。 ただし、 IBM MQ は、記載されているポリシーでのみテストされています。

単一の名前付き *CipherSpec* を使用して、*IBM MQ for Multiplatforms* キュー・マネージャーからのインバウンド・チャネル上での *AT-TLS* の構成

IBM MQ for Multiplatforms キュー・マネージャーから IBM MQ for z/OS キュー・マネージャーへのインバウンド・チャネルで AT-TLS を構成する方法。 この場合、z/OS キュー・マネージャー上のチャネルは、SSLCIPH 属性が設定されていない受信側チャネルであり、非 z/OS キュー・マネージャー上のチャネルは、SSLCIPH 属性が単一の名前付き CipherSpec に設定されている送信側チャネルです。



この例では、TLS 1.2 TLS_RSA_WITH_AES_256_GCM_SHA384 CipherSpec を使用する既存の送信側と受信側のチャネルのペアが、 IBM MQ TLS ではなく AT-TLS を使用するように調整されます。

その他の TLS プロトコルおよび CipherSpec を使用するには、構成に軽微調整を行うことができます。 クラスター送信側チャネルとクラスター受信側チャネル以外の他のメッセージ・チャネル・タイプは、AT-TLS 構成を変更せずに使用することができます。

手順

「ステップ 1: チャネルを停止する」

「ステップ 2: AT-TLS ポリシーを作成して適用する」

このシナリオには、以下の AT-TLS ステートメントを作成する必要があります。

1. 「TLTSRule」 ステートメントは、送信側チャネルの IP アドレスからチャネル・イニシエーター・アドレス・スペースへのインバウンド接続を一致させます。 ここでは、特定のチャネル・イニシエーター・ジョブ名に一致するフィルター操作が追加されました。

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   上記のルールは、リモート IP アドレス 123.456.78.9 からローカル・ポート 1414 の CSQ 1 CHIN ジョブに入ってくる接続に対して一致します。

   より高度なフィルター・オプションについては、TTLSRule を参照してください。

2. ルールを使用可能にする 「TTLSGroupAction」 ステートメント。 TTLSRule は、**TTLSGroupActionRef** プロパティーを使用して TTLSGroupAction を参照します。

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction ステートメントは、 **TTLSEnvironmentActionRef** プロパティーによって TTLSRule に関連付けられます。 TTLSEnvironmentAction は TLS 環境を構成し、使用する鍵リングを指定します。

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS は、SSLCAUTH チャネル属性の使用と同等の相互認証機能を提供します。 これを行うには、インバウンド TTLSEnvironmentAction ステートメントの **HandshakeRole** 値に *ServerWithClientAuth* を指定した TTLSEnvironmentAction ステートメントを使用します。

4. TTLSKeyringParms ステートメントは、 **TTLSKeyringParmsRef** プロパティーによって TTLSEnvironmentAction に関連付けられ、AT-TLS によって使用される鍵リングを定義します。

   鍵リングには、リモートの非 z/OS キュー・マネージャーからトラステッド証明書が含まれている必要があります。 この鍵リングは、チャネル・イニシエーターが使用する鍵リングと同じ方法で定義することができます。252 ページの『z/OS システムで TLS を使用するための構成』 を参照してください。

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** プロパティーによって TTLSEnvironmentAction に関連付けられた TTLSCipherParms ステートメント。

   このステートメントには、ターゲットの受信側チャネルで使用される IBM MQ CipherSpec 名と同等の名前を指定する必要がある単一の暗号スイート名が含まれている必要があります。

   注 : AT-TLS 暗号スイート名は、必ずしも IBM MQ CipherSpec 名と一致するわけではありません。 ただし、以下の表から IBM MQ CipherSpec 名を見つけ、4 文字のコード列を TTLSCipherParms トピックの表 2 の拡張文字列と相互参照することによって、IBM MQ CipherSpec 名と一致する AT-TLS 暗号スイート名を見つけることができます。

表 *79. 4* 文字コードから *CipherSpec* 名への変換

4 文字コード	**プロトコル**	**デフォルトで有効**	**CipherSpec 名**
0001	SSL 3.0	いいえ	NULL_MD5
0002	SSL 3.0	いいえ	NULL_SHA
0003	SSL 3.0	いいえ	RC4_MD5_EXPORT
0004	SSL 3.0	いいえ	RC4_MD5_US
0005	SSL 3.0	いいえ	RC4_SHA_US
0006	SSL 3.0	いいえ	RC2_MD5_EXPORT
0008	SSL 3.0	いいえ	DES_SHA_EXPORT
0009	TLS 1.0	はい	TLS_RSA_WITH_DES_CBC_SHA
000A	SSL 3.0	いいえ	TRIPLE_DES_SHA_US
000A	TLS 1.0	はい	TLS_RSA_WITH_3DES_EDE_CBC_SHA
002F	TLS 1.0	はい	TLS_RSA_WITH_AES_128_CBC_SHA
0035	TLS 1.0	はい	TLS_RSA_WITH_AES_256_CBC_SHA
003B	TLS 1.2	はい	TLS_RSA_WITH_NULL_SHA256
003C	TLS 1.2	はい	TLS_RSA_WITH_AES_128_CBC_SHA256

表 79. 4 文字コードから CipherSpec 名への変換 (続き)			
4 文字コード	**プロトコル**	**デフォルトで有効**	**CipherSpec 名**
003D	TLS 1.2	はい	TLS_RSA_WITH_AES_256_CBC_SHA256
C023	TLS 1.2	はい	ECDHE_ECDSA_AES_128_CBC_SHA256
C024	TLS 1.2	はい	ECDHE_ECDSA_AES_256_CBC_SHA384
C027	TLS 1.2	はい	ECDHE_RSA_AES_128_CBC_SHA256
C028	TLS 1.2	はい	ECDHE_RSA_AES_256_CBC_SHA384

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_RSA_WITH_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms ステートメントは、 **TTLSEnvironmentAdvancedParmsRef** プロパティーによって TTLSEnvironmentAction に関連付けられます。

   このステートメントを使用して、どの SSL プロトコルおよび TLS プロトコルを使用可能にするかを指定できます。 IBM MQ では、 TTLSCipherParms ステートメントで使用される暗号スイート名に一致する単一プロトコルのみを有効にする必要があります。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         ON
  TLSv1.3         OFF
}
```

   ステートメントの完全なセットは以下のとおりであり、ポリシー・エージェントに適用する必要があります。

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_RSA_WITH_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

ステップ 3: z/OS チャネルから SSLCIPH を削除する

以下のコマンドを使用して、 z/OS チャネルから CipherSpec を削除します。

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

ステップ 4: チャネルを開始する

チャネルが開始されると、そのチャネルは AT-TLS と IBM MQ TLS の組み合わせを使用します。

重要 : 上記の AT-TLS ステートメントは、最小限の構成にすぎません。 ここには記載されていない、AT-TLS の他の「AT-TLS ポリシー・ステートメント」があり、必要に応じて IBM MQ で使用することもできます。 ただし、 IBM MQ は、記載されているポリシーでのみテストされています。

SSL および TLS 秘密鍵のリセット

IBM 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、秘密鍵がリセットされます。 チャネル・ハートビートが有効になっている場合は、チャネル・ハートビートに続いてデータが送受信される前に秘密鍵がリセットされます。

鍵リセット値は、IBM 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。

IBM i IBM i では、**CHGMQM** を **SSLRSTCNT** パラメーターと共に使用します。

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、TLS 秘密鍵が再ネゴシエーションされる前に、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、TLS 秘密鍵は絶対に再ネゴシエーションされません。 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、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、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 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 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。

sslResetCount 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 IBM MQ classes for Java クライアント・コードが送受信するバイトの総数を表します。 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 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ています。

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 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 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、TLS 対応の MQI チャネル (このチャネルの秘密鍵は、4 MB のデータが流れた後再ネゴシエーションされる) を介した接続の作成に使用できる ConnectionFactory オブジェクトを構成するには、JMSAdmin に対して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 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 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。

.NET の管理対象クライアントの場合、SSLStream クラスは、秘密鍵のリセットや再ネゴシエーションをサポートしません。 ただし、他の IBM MQ クライアントと一貫させるため、IBM MQ 管理対象 .NET クライアントでアプリケーションは SSLKeyResetCount を設定することができます。 詳細については、秘密鍵のリセットまたは再ネゴシエーション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 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

メッセージ・キュー・マネージャーの変更 (CHGMQM)

メッセージ・キュー・マネージャーの表示 (DSPMQM)

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、444 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。

V 9.1.4 z/OS データ・セット暗号化による IBM MQ for z/OS での保存データの機密性

IBM MQ for z/OS では、お客様のデータと構成データを堅牢にすることができます。これは、アクティブ・ログ・データ・セット、アーカイブ・ログ・データ・セット、ページ・セット、ブート・ストラップ・データ・セット (BSDS)、および V 9.1.5 共有メッセージ・データ・セット (SMDS) にデータを書き込むことによって行います。

z/OS には、ポリシー・ベースの効率的なデータ・セット暗号化機能があります。 IBM MQ for z/OS では、次のデータ・セットに対する z/OS データ・セット暗号化をサポートしています。

- アクティブ・ログ・データ・セット (注 446 ページの『1』 を参照)

- 保存ログ・データ・セット (注 446 ページの『2』 を参照)
- ページ・セット (注 446 ページの『1』 を参照)
- BSDS (注 446 ページの『2』 を参照)
- CSQINP* データ・セット (注 446 ページの『2』 を参照)
- V 9.1.5 SMDS (注 446 ページの『3』 を参照)

これにより、個々の z/OS キュー・マネージャー上の 保存データの機密性を保つことができます。

注 :

1. IBM MQ 9.1.4 以降、IBM MQ for z/OS は、アクティブ・ログとページ・セットに関する z/OS データ・セット暗号化をサポートしています。
2. アーカイブ・ログ、BSDS データ・セット、CSQINP* データ・セットに関するデータ・セット暗号化は、すべてのバージョンの IBM MQ for z/OS でサポートされています。
3. V 9.1.5 IBM MQ 9.1.5 以降、IBM MQ for z/OS は、SMDS に関する z/OS データ・セット暗号化をサポートしています。
4. IBM MQ Advanced Message Security には、保存データを保護するための代替メカニズムが用意されています。 さらに、AMS ではメモリー内のデータと移動中のデータも保護されます

z/OS データ・セットの暗号化についての詳細は、「z/OS データ・セット暗号化の機能強化」 を参照してください。

z/OS データ・セット暗号化の構成は、IBM MQ for z/OS の制御範囲外です。 暗号化設定は、データ・セットの作成時に有効になります。

つまり、新しいデータ・セットの暗号化ポリシーを使用するには、その前に既存のデータ・セットを再作成する必要があります。

IBM MQ for z/OS は、暗号化されたデータ・セットと暗号化されていないデータ・セットを混合して実行できますが、標準構成では、使用されるデータ・セットのすべてが暗号化されるか、またはまったく暗号化されないかです。

V 9.1.4 z/OS IBM MQ for z/OS のデータ・セットを暗号化するための手順の概要

IBM MQ for z/OS のデータ・セットを暗号化する方法。

始める前に

z/OS データ・セット暗号化が、自社で正しく構成済みであることを確認する必要があります。 キュー共有グループ内でデータ・セット暗号化をセットアップする場合は、データ共有を行えるように z/OS データ・セット暗号化を構成する必要があります。

注 : z/OS の暗号化されるデータ・セットは、拡張フォーマットのデータ・セットでなければなりません。

手順

1. データ・セットの暗号化に使用する暗号鍵と `key-label` を RACF でセットアップします。
2. RACF CSFKEYS クラスに `key-label` 用のプロファイルを作成します。
3. キュー・マネージャーのユーザー ID と、暗号化されたデータへのアクセスが必要な他のすべてのユーザー ID に READ アクセス権限を付与します。

   この権限付与の対象には、データ・セットに対して印刷ユーティリティーを実行するために使用するユーザー ID が含まれる場合があります。 例えば、CSQUTIL SCOPY を実行しているユーザーは、関係するページ・セットを復号する必要が生じます。
4. 暗号化の `key-label` をデータ・セット名に関連付けます。

   これを行うには、データ・セット名または高位修飾子に SMS データ・クラスまたは RACF DFP セグメントを使用します。

key-label は、データ・セットの割り振り時にそのデータ・セットに関連付けることもできます。

5. 既存のデータ・セットの名前を、IDCAMS ALTER を使用して変更します。
6. そのデータ・セットを、適切な属性を使用して再割り振りします。
7. 名前変更したデータ・セットの内容を、IDCAMS REPRO を使用して新しいデータ・セットにコピーします。

   データをそのデータ・セットにコピーする操作を行うと、そのデータが暗号化されます。
8. 暗号化する必要があるその他のデータ・セットについて、ステップ 446 ページの『4』 から 447 ページの『6』 までを繰り返します。

V 9.1.4 z/OS キュー・マネージャーのアクティブ・ログを暗号化する方法の例

以下の各トピックでは、既存のアクティブ・ログに対してデータ・セット暗号化を有効にするプロセスについて説明します。

注 : 他のデータ・セットに関するプロセスは、アクティブ・ログの場合に類似しています。

この例のそれぞれの指定の意味は次のとおりです。

- キュー・マネージャー CSQ1 が、ユーザー QMCSQ1 の下で実行され、アクティブ・ログ・データ・セット CSQ1.LOGS.LOGCOPY1.DS001、CSQ1.LOGS.LOGCOPY1.DS002 (以降、同様の順序で続く) を保持しています
- ハードウェアとソフトウェアの環境で、z/OS データ・セット暗号化を使用できます
- RACF は、SAF として使用されます
- キュー・マネージャーは停止しています

以下の順序でこの手順を実行します。

1. 447 ページの『キュー・マネージャーのデータ・セット暗号鍵の構成』
2. 448 ページの『ログ・データ・セットに対するデータ・セット暗号化の構成』

V 9.1.4 z/OS キュー・マネージャーのデータ・セット暗号鍵の構成

キュー・マネージャーのデータ・セット暗号鍵の構成方法。

このタスクについて

この作業は、448 ページの『ログ・データ・セットに対するデータ・セット暗号化の構成』の前提条件の 1 つです。

手順

1. z/OS 鍵生成ユーティリティー・プログラム (KGUP)を使用して、ラベル付きの AES-256 ビット暗号化 DATA 鍵 (例えば、 CSQ1DSKY) をセットアップします。
2. 次のコマンドを発行して、CSQ1DSKY 暗号鍵の RACF CSFKEYS プロファイルを定義します。

   ```
   RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
   ```
3. 次のコマンドを発行して、この鍵を保護された鍵として使用できるようにプロファイルの ICSF セグメントを構成します。

   ```
   RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
   ```
4. キュー・マネージャーがその暗号鍵を使用できるようにします。これは、次のコマンドを発行して、このプロファイルへの READ アクセス権限を QMCSQ1 に付与することによって行います。

   ```
   PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
   ```

暗号化されたデータ・セットの読み取りまたは書き込みが必要なすべての管理ユーザーに、同じアクセス権限を付与します。

5. 次のコマンドを発行して、CSFKEYS クラスをリフレッシュします。

```
SETROPTS RACLIST(CSFKEYS) REFRESH
```

次のタスク

448 ページの『ログ・データ・セットに対するデータ・セット暗号化の構成』の説明に従って、データ・セットに対してデータ・セット暗号化を構成します

V 9.1.4 z/OS ログ・データ・セットに対するデータ・セット暗号化の構成

ログ・データ・セットに対して暗号化を構成する方法。

始める前に

次の各トピックを必ずお読みください。

IBM MQ for z/OS のデータ・セットを暗号化するための手順の概要。さらに次のトピックに記載された手順を実行します
447 ページの『キュー・マネージャーのデータ・セット暗号鍵の構成』

このタスクについて

以下の方式では、RACF 総称プロファイルの DFP セグメントを使用して、このプロファイルに一致するすべての新規データ・セットに対して暗号鍵を使用できるようにしています。

また、SMS データ・クラスを構成して使用することも、データ・セットの割り振り時に鍵ラベルを直接指定することもできます。

前述のように、この例ではキュー・マネージャー CSQ1 が、ユーザー QMCSQ1 の下で実行され、アクティブ・ログ・データ・セット CSQ1.LOGS.LOGCOPY1.DS001、CSQ1.LOGS.LOGCOPY1.DS002 (以降、同様の順序で続く) を保持しています。

手順

1. 総称プロファイルが存在しない場合は、次のコマンドを発行してこれを作成します。

```
ADDSD 'CSQ1.LOGS.*' UACC(NONE)
```

2. 次のコマンドを発行して、キュー・マネージャーのユーザーに、このプロファイルに対する変更アクセス権限を許可します。

```
PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
```

   また、すべての管理ユーザーに必要な適切なアクセス権限も許可します。

3. 次のコマンドを発行して、暗号鍵ラベルを持つ DFP セグメントを追加します。

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

   注: キュー・マネージャーのデータ・セット暗号鍵の構成で使用した暗号鍵を使用する必要があります。

4. 次のコマンドを発行して、総称データ・セット・プロファイルをリフレッシュします。

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. 各ログ・データ・セットを名前変更してバックアップ用に変更した後、IDCAMS を使用してデータを再作成してリストアします。 次の JCL の断片では、CSQ1.LOGS.LOGCOPY1.DS001 を変換します。

   a) データ・セットを名前変更してバックアップ用に変更します

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
```

```
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME DATASET TO BACKUP                                    */
  /*-------------------------------------------------------------*/
  ALTER 'CSQ1.LOGS.LOGCOPY1.DS001'                    -
        NEWNAME('CSQ1.BAK.LOGS.LOGCOPY1.DS001')
```

b) データ・セットを再定義します。

新規データ・セットは、RACF プロファイルによって暗号化されます。

注: ++EXTDCLASS++ を、このデータ・セットのために使用する拡張フォーマットのデータ・クラスの名前に置き換えます。

```
//REDEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* REDEFINE THE DATASET                                        */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                      -
         (NAME(CSQ1.LOGS.LOGCOPY1.DS001)              -
         LINEAR                                       -
         SHAREOPTIONS(2 3)                            -
         MODEL(CSQ1.BAK.LOGS.LOGCOPY1.DS001)          -
         DATACLAS(++EXTDCLASS++))
```

c) データをバックアップから、再作成されたデータ・セット内にコピーします。

このステップで、データが暗号化されます。

```
//RESTORE  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RESTORE DATA INTO ENCRYPTED LOG                             */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)       -
        OUTDATASET(CSQ1.LOGS.LOGCOPY1.DS001)
```

次のタスク

すべてのアクティブ・ログ・データ・セットについて、ステップ 448 ページの『5』 を繰り返します。

1 つの暗号鍵だけが必要で、同じ鍵ラベルにすべてのデータ・セットを関連付けることができます。

キュー・マネージャー CSQ1 を再始動します。 DISPLAY LOG コマンドからの出力を使用して、ログ・データ・セットが暗号化されていることを確認します。

V 9.1.4 z/OS キュー共有グループ内での z/OS データ・セット暗号化に関する考慮事項

対象となる 1 つのキュー共有グループ (QSG) に属する各キュー・マネージャーは、その QSG 内の他のすべてのキュー・マネージャーのログ、BSDS V 9.1.5 と共有メッセージ・データ・セット (SMDS) を読み取れなければなりません。

これは、QSG のメンバーを実行できる各システムが z/OS データ・セット暗号化の要件を満たす必要があり、さらに、その QSG に属する各キュー・マネージャーのデータ・セットを保護するために使用されるすべての鍵ラベルと暗号鍵が各システム上で使用可能でなければならないことを意味しています。

IBM MQ for z/OS 9.1.3 より前のキュー・マネージャーは、暗号化アクティブ・ログ・データ・セットにアクセスすることはできません。

V 9.1.5 IBM MQ for z/OS 9.1.3 より前のキュー・マネージャーは、暗号化された SMDS にアクセスすることはできません。

V 9.1.5 z/OS データ・セット暗号化を使用する前に、QSG 内のすべてのキュー・マネージャーを少なくとも IBM MQ for z/OS 9.1.3 にマイグレーションする必要があります。

1 つの QSG の中で、あるキュー・マネージャーが開始済みであるが前回の開始時に暗号化アクティブ・ログをサポートするバージョンの IBM MQ for z/OS を使用していなかった場合、その QSG で暗号化アクティブ・ログ・データ・セットを使用する別のキュー・マネージャーを開始すると、暗号化アクティブ・ログを使用するキュー・マネージャーは、異常終了コード 5C6-00F50033 で異常終了します。

V 9.1.5 次の方法を使用すると、完全な停止なしで、暗号化アクティブ・ログおよび SMDS を使用するように QSG を変換できます。

1. 各キュー・マネージャーを少なくとも IBM MQ 9.1.5 に順番にマイグレーションする。
2. アクティブ・ログをキュー・マネージャーごとに順番に暗号化データ・セットに変換する。 キュー・マネージャーをシャットダウンしてから再始動する必要があります。

   同時に、ページ・セットとアーカイブ・ログで暗号化データ・セットを使用できるようになる可能性もありますが、これは QSG のマイグレーションには影響を与えません。

   各データ・セットを変換する手順は、447 ページの『キュー・マネージャーのアクティブ・ログを暗号化する方法の例』で説明されています
3. 以下のようにして、SMDS を個々の CF 構造ごとに順番に暗号化データ・セットに変換する。
   a. コマンド RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name) を発行して、SMDS へのキュー・マネージャー・アクセスを中断する。

      この間、SMDS に関連付けられている共有キュー上のデータが一時的に利用不可になることに注意してください。
   b. 447 ページの『キュー・マネージャーのアクティブ・ログを暗号化する方法の例』で説明されている手順を使用して、SMDS を構成する各データ・セットを暗号化データ・セットに変換する。
   c. コマンド RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name) を発行して、SMDS へのキュー・マネージャー・アクセスを再開する。

重要 : ログを変換する前に、キュー・マネージャーをクリーン・シャットダウンしてください。 この変換中に、カップリング・ファシリティー構造のリカバリーが不可能になるおそれがあります。これは、アクティブ・ログ・データ・セットが一時的に使用不可になるためです。

V 9.1.4 z/OS z/OS データ・セット暗号化の使用時のバックワード・マイグレーションに関する考慮事項

1 つ以上の暗号化されたデータ・セットのあるキュー・マネージャーをバックワード・マイグレーションする際には、以下の事項を考慮する必要があります。

z/OS データ・セット暗号化は、以下の IBM MQ for z/OS データ・セットでサポートされています。

- 活動ログ・データ・セット
- 保存ログ・データ・セット
- ページ・セット
- BSDS
- V 9.1.5 SMDS
- CSQINP* データ・セット

BSDS、アーカイブ・ログ、CSINP* データ・セットに関するバックワード・マイグレーションの考慮事項はありません。

しかし、以下のデータ・セットに関する考慮事項があります。

- V 9.1.5 SMDS
- ページ・セット
- アクティブ・ログ

z/OS データ・セット暗号化でこれらを使用するデータ・セットは、IBM MQ for z/OS 9.1.0 以前の長期サポート・リリースではサポートされません。

バックワード・マイグレーションを行う前に、V 9.1.5 SMDS、ページ・セット、アクティブ・ログ・データ・セットに関する暗号化ポリシーをすべて削除し、データを復号する必要があります。 この操作については、451 ページの『データ・セットからのデータ・セット暗号化の削除』を参照してください。

重要 : バックワード・マイグレーションするキュー・マネージャーがキュー共有グループ (QSG) の一部である場合は、452 ページの『キュー共有グループの考慮事項』のセクションを最初に読んでください。

データ・セットからのデータ・セット暗号化の削除

この例では、ログ・データ・セット CSQ1.LOGS.LOGCOPY1.DS001 からデータ・セット暗号化を削除する方法について説明します。 V 9.1.5 SMDS とページ・セットにも同等のプロセスを使用できます。

この例では、次のことを前提としています。

- RACF が SAF
- データ・セットを使用するキュー・マネージャーが停止している
- 暗号鍵ラベルが総称 RACF プロファイル CSQ1.LOGS.* と関連付けられている

以下の手順を実行します。

1. 当該データ・セットからバックアップ・データ・セットにデータをコピーします。
   a. 暗号鍵ラベルと関連付けないバックアップ・データ・セットを定義します。

      注 : ++EXTDCLASS++ を、このデータ・セットのために使用する拡張フォーマットのデータ・クラスの名前に置き換えます。

```
//DEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DEFINE UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                    -
         (NAME(CSQ1.BAK.LOGS.LOGCOPY1.DS001)        -
         LINEAR                                     -
         SHAREOPTIONS(2 3)                          -
         MODEL(CSQ1.LOGS.LOGCOPY1.DS001)            -
         DATACLAS(++EXTDCLASS++))
/*
```

   b. 元のデータ・セットからバックアップにデータをコピーします。 このステップで、データが復号されます。

```
//COPY     EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* COPY DATA INTO UNENCRYPTED DATA SET                         */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.LOGS.LOGCOPY1.DS001)          -
        OUTDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)
/*
```

   c. 元のデータ・セットを削除します

```
//DELETE   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DELETE ORIGINAL                                             */
  /*-------------------------------------------------------------*/
  DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
/*
```

d. バックアップの名前を元のデータ・セット名に変更します。 この時点でも、データは暗号化されません

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  ALTER CSQ1.BAK.LOGS.LOGCOPY1.DS001'                -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001)
  ALTER 'CSQ1.BAK.LOGS.LOGCOPY1.DS001.*'              -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001.*')
/*
```

2. オプションで、 CSQ1.LOGS.* 総称プロファイル。
3. CSQ1.LOGS.* 総称プロファイルが暗号化解除されました。次のコマンドを発行して、総称プロファイルに関連付けられている DATAKEY を削除してください。

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

4. 次のコマンドを発行して、総称データ・セット・プロファイルをリフレッシュします。

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. キュー・マネージャーを再始動する。
6. 暗号鍵が不要になった場合は、暗号鍵を削除し、関連付けられている RACF プロファイルを CSFKEYS クラスから削除します。

キュー共有グループの考慮事項

キュー共有グループの一部であるキュー・マネージャーを、データ・セット暗号化をサポートしていないバージョンの IBM MQ for z/OS にバックワード・マイグレーションしようとしている場合は、QSG 内の全キュー・マネージャーの全アクティブ・ログ・データ・セット V 9.1.5 と SMDS のデータ・セット暗号化ポリシーを削除し、データを復号する必要があります。

これは、バックワード・マイグレーションするのが QSG の単一のメンバーかすべてのメンバーかにかかわらず適用されます。

次の方法を使用すると、QSG の完全な停止なしで、暗号化ポリシーの削除とデータの復号を実行できます。

1. 451 ページの『データ・セットからのデータ・セット暗号化の削除』で説明されているプロセスを使用して、QSG 内の各キュー・マネージャーを順番にシャットダウンし、そのアクティブ・ログから暗号化ポリシーを削除してデータを復号します。

   キュー・マネージャーをバックワード・マイグレーションする場合、この時点でそのページ・セットを復号する必要もあります。 その後、キュー・マネージャーを再始動します。

2. V 9.1.5 次の方法を使用して、個々の CF 構造ごとに順番に SMDS の暗号化ポリシーを削除し、データを復号します。

   a. 次のコマンドを発行します。

   ```
   RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
   ```

   このコマンドで、SMDS へのキュー・マネージャー・アクセスを中断します。 この間、SMDS に関連付けられている共有キュー上のデータが一時的に利用不可になります。

   b. SMDS を構成するデータ・セットごとに、451 ページの『データ・セットからのデータ・セット暗号化の削除』のプロセスに従います。

   c. 次のコマンドを発行します。

   ```
   RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
   ```

   このコマンドで、SMDS へのキュー・マネージャー・アクセスを再開します。

z/OS データ・セット暗号化とそのサポート対象外のキュー・マネージャーとの併用

キュー・マネージャーを、データ・セット暗号化をサポートしていないバージョンの IBM MQ for z/OS に誤ってバックワード・マイグレーションし、暗号化ポリシーの削除とデータの復号をし忘れると、キュー・マネージャーがデータ・セットへのアクセスを試行した場合にエラーを受け取ります。

このエラーは、以下の表に示されているようにデータ・セットのタイプに応じて異なります。

注 : これらのエラーが 1 つ以上発生する場合、影響を受けているデータ・セットについて、451 ページの『データ・セットからのデータ・セット暗号化の削除』で説明されているプロセスに従う必要があります。このプロセスは、IBM MQ for z/OS のバージョンを変更せずに実行できます。

データ・セット	キュー・マネージャーが z/OS データ・セット暗号化をサポートしていない場合のエラー
ページ・セット 0	キュー・マネージャー開始時の異常終了 5C6-00C91400
ページ・セット 1 - 99	MQPUT などでページ・セットにアクセスする際の MQRC 2193「ページ・セット・エラー」
アクティブ・ログ	キュー・マネージャー開始時の異常終了 5C6-00E80084
V 9.1.5 SMDS	メッセージ IEC161I-122「データ・セットに KEYLABEL がありますが、ユーザーはアプリケーションが暗号化を処理できるように指定しませんでした (The data set has a KEYLABEL, but the user did not specify that the application could handle encryption)」がログに記録。 SMDS に AVAIL(ERROR) のマークが付けられる。

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

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。 IBM 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。

詳細情報

データ保全性の確保について詳しくは、416 ページの『CipherSpecs の有効化』のセクションを参照してください。

関連タスク

TLS による 2 つのキュー・マネージャーの接続

キュー・マネージャーへのクライアントのセキュア接続

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 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 TLS 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 IBM 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 (IBM 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

z/OS では、RACF を使用して、クラスター・キューに対する書き込み権限を設定します。 他のプラットフォームでは、キュー・マネージャーへの接続と、それらのキュー・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付与します。

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が該当するのは、セキュリティー・スタンザのトピックの説明に従って、qm.ini ファイルの **ClusterQueueAccessControl** 属性に *RQMName* を設定し、キュー・マネージャーを再始動した場合のみです。

手順

- z/OS の場合は、以下のコマンドを発行します。

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- UNIX, Linux, and Windows システムの場合は、以下のコマンドを実行します。

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

- 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)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

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。 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。

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。 394 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。

関連概念
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 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。

   z/OS **z/OS**

   セキュリティー出口のロード・モジュールは、チャネル・イニシエーターのアドレス・スペース・プロシージャーの CSQXLIB DD ステートメントで指定されたデータ・セット内になければなりません。

   ULW **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。

注 : RESET CLUSTER コマンドを使用するとクラスターからキュー・マネージャーが強制的に排除されますが、RESET CLUSTER を単独で使用しても、そのキュー・マネージャーがあとでクラスターに再加入することを防ぐことはできません。 キュー・マネージャーがクラスターに再加入しないようにするには、456 ページの『キュー・マネージャーのクラスターへの参加の防止』で詳細に説明されている手順に従ってください。

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)
```

注 : QMID はストリングであるため、 qmid の値は単一引用符で囲む必要があります (例: QMID('FR01_2019-07-15_14.42.42'))。

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/TLS とクラスター

クラスターの TLS 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 TLS 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 TLS を構成する必要があります。

TLS の詳細については、22 ページの『IBM MQ での TLS セキュリティー・プロトコル』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 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 TLS 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、TLS サポートへのアクセス権が必要です。 この TLS 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 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=IBM 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。 この例として、CipherSpec ストリング RC4_SHA_US の値が IBM i では 05 になっていますが、 UNIX、 Linux 、または Windows システムでは有効な指定ではありません。 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 TLS 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 TLS_RSA_WITH_AES_128_CBC_SHA)。

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/TLS 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 IBM 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。

  重要 : 1 つのクラスターで結合させるキュー・マネージャー同士の間で、楕円曲線暗号の署名の付いた証明書と RSA の署名の付いた証明書を混在させることはできません。

  クラスター内のキュー・マネージャーがすべて RSA の署名の付いた証明書を使用するか、すべて EC の署名の付いた証明書を使用するかのどちらかにしなければなりません。両方を混在させることはできません。

  詳しくは、43 ページの『IBM 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。

クラスター内のすべてのキュー・マネージャーを IBM MQ V8 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 TLS が作動するよう、証明書および鍵を配布します。

Tom をアップグレードするか、ANY_TLS12 CipherSpecs を使用する場合は、クラスター内のすべてのキュー・マネージャーを IBM MQ 9.1.2 以上にアップグレードする必要があります。

他のいずれかの別名 CipherSpecs (ANY_TLS13、 ANY_TLS12、 ANY_TLS12_OR_HIGHER など) にアップグレードするか、それを使用する場合は、クラスター内のすべてのキュー・マネージャーを IBM MQ 9.1.4 以上にアップグレードする必要があります。

このタスクについて

CLUSRCVR チャネルを変更した後に CLUSSDR チャネルを変更します。

手順

1. CLUSRCVR チャネルを任意の順番で TLS に切り替え、CLUSRCVR を一度に 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・パスは絶対に変更しないでください。
2. オプション: すべての手動 CLUSSDR チャネルを TLS 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、処理中のクラスターに **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。
3. DISPLAY CLUSQMGR コマンドを使用して、新しいセキュリティー構成がクラスター全体に伝搬していることを確認します。
4. チャネルを再始動し、TLS を使用して、REFRESH SECURITY (SSL) を実行します。

関連概念

416 ページの『CipherSpecs の有効化』
CipherSpec は、**DEFINE CHANNEL** MQSC コマンドまたは **ALTER CHANNEL** MQSC コマンドのどちらかにおいて、**SSLCIPH** パラメーターを使用することにより有効にします。

43 ページの『IBM 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 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/TLS を無効にする

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 ' ' (単一引用符で囲まれた空ストリング)IBM i、または *NONE (IBM i 上) に設定します。
   クラスター受信側チャネル上の TLS は任意の順番でオフにすることができます。

   変更は、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 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。 詳しくは、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 462 ページの図 22 に示し、続いて例を挙げて説明します。

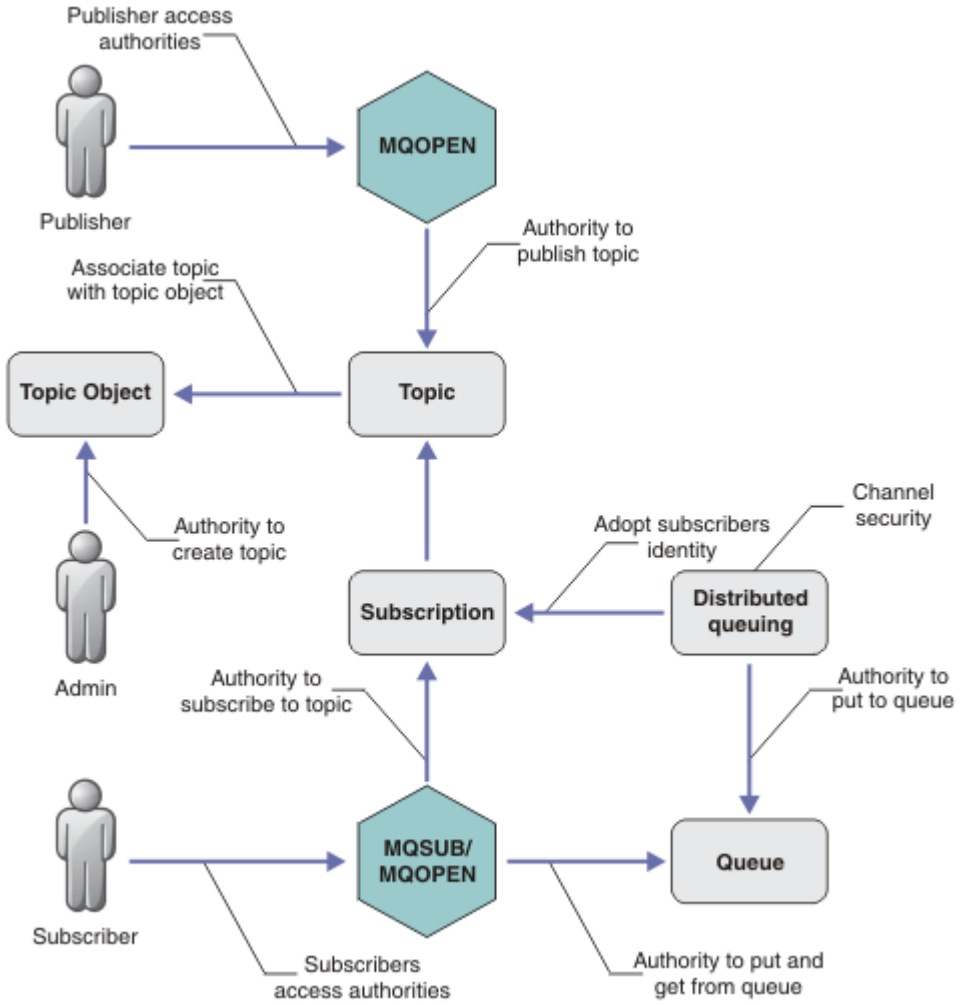


図 22. 各種コンポーネント間におけるパブリッシュ/サブスクライブのセキュリティー関係

「**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。 464 ページの『トピック・セキュリティー・モデル』では、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ています。

「**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。 469 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 477 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。 z/OS z/OS でのトピック・オブジェクトへのアクセスの制御については、トピック・セキュリティーのプロファイル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
465 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
466 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
467 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。 482 ページの『サブスクリプションのセキュリティー』を参照。

「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。 467 ページの『宛先キューに対する権限』を参照。

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。

484 ページの『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 IBM 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。

484 ページの『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. IBM 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。

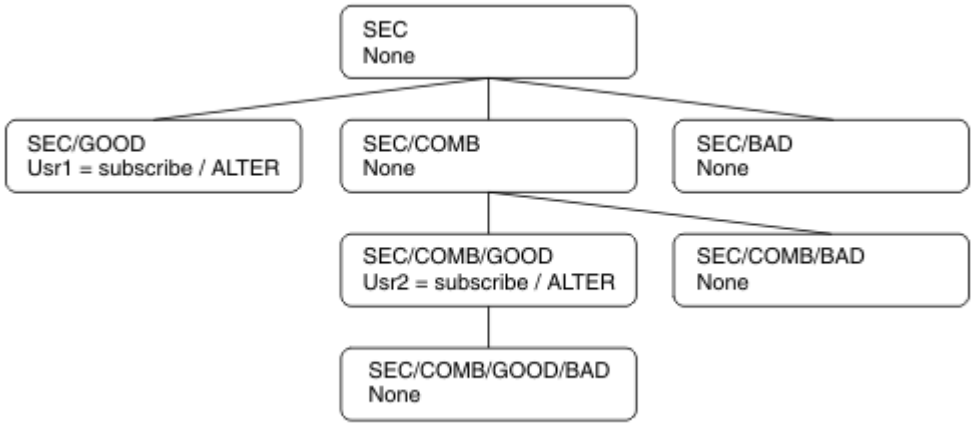
以下のようにセキュリティー属性または権限が定義されたトピック・オブジェクトの例を考えてみます。

表 *80.* 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 権限
SECROOT	SEC	なし	なし
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD

表 80. 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 権限**
SECBAD	SEC/BAD	なし	なし HLQ.SUBSCRIBE.SECBAD
SECCOMB	SEC/COMB	なし	なし HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/ GOOD/BAD	なし	なし HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	なし	なし 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。

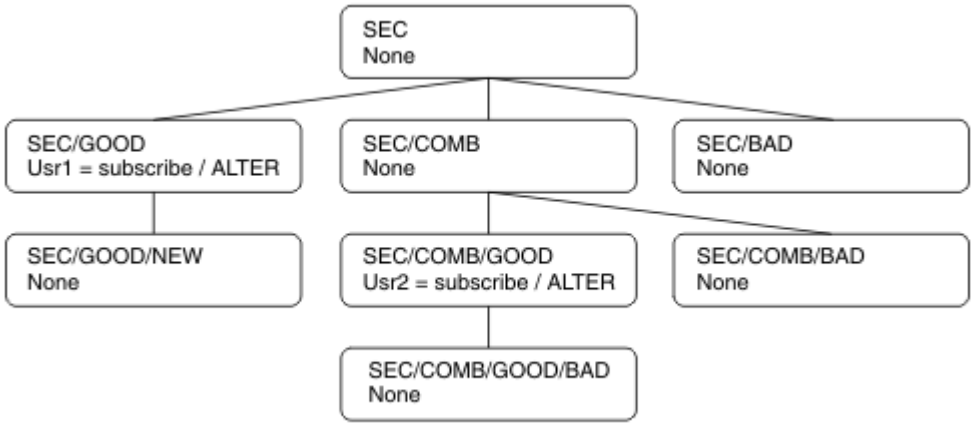
トピック・ストリングを使用したサブスクライブ操作 (トピック・ノードが存在する場合)

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と、いずれかのパラメーターが、パブリケーションを受信するために出力用にオープンされたキューのハンドル hobj になります。

hobj が指定されていないが、ブランクの場合、以下の条件が適用されると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場合、以前に指定された宛先キューは管理対象または非管理対象のいずれかになります。

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、469 ページの表 81 に示すように、検査の実行方法に影響を与えます。

表 *81.* 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		LIKE サブスクリプションのユーザー ID が使用されます
	管理者の ID が使用されます		SYSTEM.DEFAULT.SUB サブスクリプションのユーザー ID が使用され、ブランクの場合は管理者の ID が使用されます
	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。

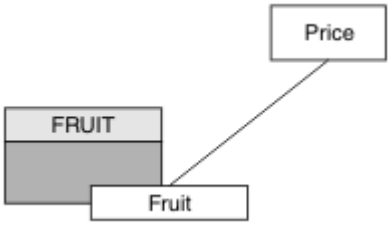


図 *23.* トピック・オブジェクトのアクセス権限の例

表 *82.* 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。

- z/OS **z/OS** :

  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、`hlq.SUBSCRIBE.FRUIT` 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、以下の RACF コマンドを使用します。

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
```

- 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。

  ULW **Windows、UNIX and Linux システム**

```
setmqaut -t topic -n FRUIT -p USER1 +sub
```

  IBM i **IBM i**

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*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。

- z/OS 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 ...
```

- ULW 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"
```

- IBM i IBMi 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、469 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。

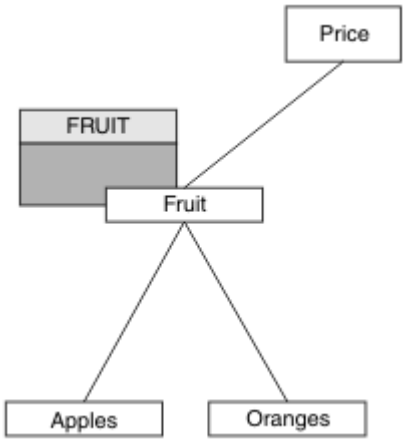


図 *24.* トピック・ツリー内のトピックへのアクセス権限を付与する例

表 *83.* 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 プロファイルへのアクセス権限と、他のプラットフォーム上の FRUIT プロファイルへのサブスクライブ・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" へのサブスクライブ権限が付与されました。 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付与されます。

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、471 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。

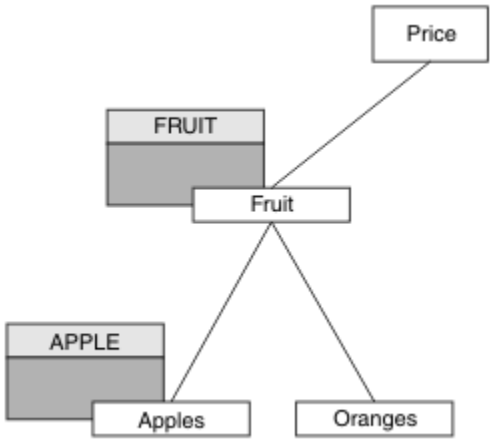


図 *25.* トピック・ツリー内の特定のトピックへのアクセス権限の付与

表 *84.* 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。

- z/OS **z/OS** :

  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、hlq.SUBSCRIBE.FRUIT プロファイルに対するアクセス権限をそのユーザーに与えました。

  この単一プロファイルにより、"Price/Fruit/Oranges" "Price/Fruit/#" にサブスクライブするための USER1 アクセス権限も付与されます。このアクセス権限は、新しいトピック・オブジェクトとそれに関連付けられたプロファイルが追加されても保持されます。

  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、hlq.SUBSCRIBE.APPLE 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、以下の RACF コマンドを使用します。

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
```

- 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。

  ULW **Windows、UNIX and Linux システム**

```
setmqaut -t topic -n APPLE -p USER2 +sub
```

  IBM i **IBM i**

```
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*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。

- z/OS 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 ...
```

- ULW Windows、UNIX and Linux 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/Oranges"
```

- IBM i IBMi 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 に関する追加の RACF ICH408I メッセージを回避する方法を示すタスクのリストの 4 番目のトピックです。

始める前に

このトピックでは、473 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。

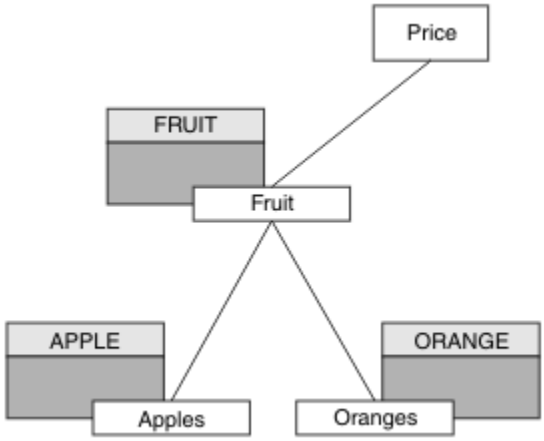


図 *26.* 追加のメッセージを回避するためにアクセス制御を付与する例。

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。

- z/OS **z/OS** :

  新しいプロファイルを定義し、そのプロファイルと既存のプロファイルへのアクセス権限を追加します。 そのために、以下の RACF コマンドを使用します。

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.ORANGE UACC(NONE)
PERMIT hlq.SUBSCRIBE.ORANGE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
PERMIT hlq.SUBSCRIBE.APPLE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
```

- 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。

  ULW **Windows、UNIX and Linux システム**

```
setmqaut -t topic -n ORANGE -p USER1 +sub
setmqaut -t topic -n APPLE -p USER1 +sub
```

  IBM i **IBM i**

```
GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
```

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。

- z/OS 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.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ULW 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"
```

- IBM i IBMi 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。

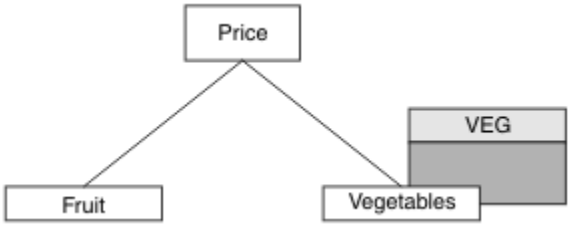


図 *27.* トピックへのパブリッシュ・アクセス権限の付与

表 *85.* 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。
   - z/OS **z/OS** :

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、hlq.PUBLISH.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、以下の RACF コマンドを使用します。

```
RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
```

- 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。

  ULW **Windows、UNIX and Linux システム**

  ```
  setmqaut -t topic -n VEG -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*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。

- z/OS 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 ...
  ```

- ULW 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"
  ```

- IBM i IBMi 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、477 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。

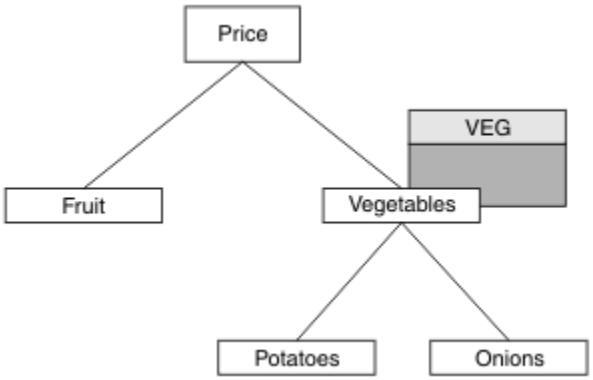


図 *28.* トピック・ツリー内のトピックへのパブリッシュ・アクセス権限の付与

表 *86.* 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、479 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。

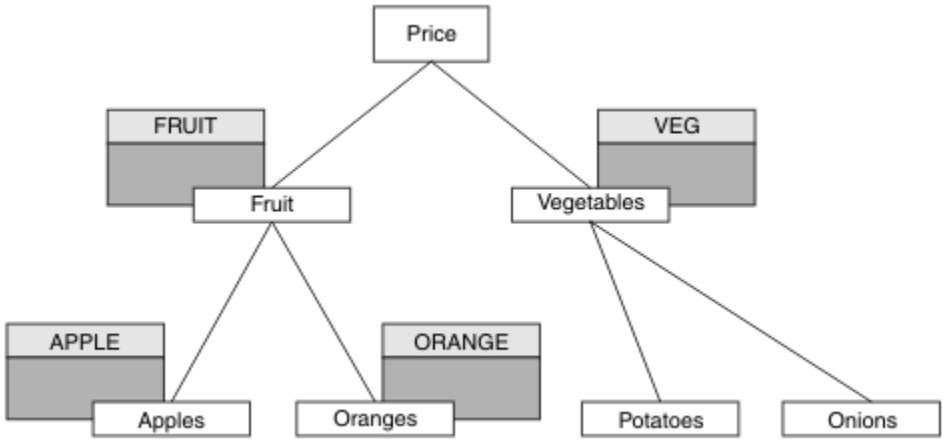


図 *29.* パブリッシュおよびサブスクライブのためのアクセス権限の付与

表 87. 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。

- z/OS **z/OS** :

  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 に与えるために、hlq.SUBSCRIBE.FRUIT プロファイルに対するアクセス権限をそのユーザーに与えました。

  ここでは、"Price/Fruit" トピックにパブリッシュするために、hlq.PUBLISH.FRUIT プロファイルに対するアクセス権限を USER1 に与えます。 そのために、以下の RACF コマンドを使用します。

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- 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。

  ULW **Windows、UNIX and Linux システム**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*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。

- z/OS 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.FRUIT ...
  ```

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ULW Windows、UNIX、および Linux 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      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i IBMi 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 *88.* 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
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 *89.* 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。

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

パブリッシュ/サブスクライブの内部メッセージ (プロキシー・サブスクリプションやパブリケーションなど) は、通常のチャネル・セキュリティー規則に基づいてパブリッシュ/サブスクライブのシステム・キューに書き込まれます。 このトピックでは、いくつかの図を交えながら、それらのメッセージの送信に必要な各種のプロセスとユーザー ID について特に説明します。

ローカル・アクセス制御

パブリケーションとサブスクリプションのためのトピックに対するアクセス権限を制御するには、ローカル・セキュリティー定義と規則を使用します (パブリッシュ/サブスクライブのセキュリティーを参照)。 z/OS では、アクセス制御を設定するためにローカル・トピック・オブジェクトは必要ありません。 その他のプラットフォームでも、アクセス制御のためのローカル・トピックは必要ありません。 管理者は、クラスター・トピック・オブジェクトがクラスターに依然として含まれているかどうかにかかわりなく、クラスター・トピック・オブジェクトにアクセス制御を適用することを選択できます。

システム管理者は、自身のローカル・システムのアクセス制御を担当します。 システム管理者は、階層またはクラスター集合の他のメンバーの管理者が、それぞれ責任を持ってアクセス制御ポリシーを実行していることを信頼する必要があります。 アクセス制御は個々のマシンごとに定義するので、細かいレベルでの制御が必要になると、作業が煩雑になるおそれがあります。 アクセス制御を実施する必要がない場合もありますが、トピック・ツリー内の上位オブジェクトでアクセス制御を定義することができます。 トピック名前空間のサブディビジョンごとに細かくアクセス制御を定義することもできます。

プロキシー・サブスクリプションの作成

他の組織のキュー・マネージャーがこちら側のキュー・マネージャーに接続する場合は、通常のチャネル認証手段によって、その組織を信頼できるかどうかが確認されます。 その組織が信頼でき、分散パブリッシュ/サブスクライブの実行も許可されると、権限検査が行われます。 権限検査は、チャネルが分散パブリッシュ/サブスクライブ・キューにメッセージを書き込む時点で行われます。 例えば、メッセージが SYSTEM.INTER.QMGR.CONTROL キューに書き込まれた場合などです。 キューの権限検査の対象になるユーザー ID は、受信側チャネルの PUTAUT 値によって決まります。 例えば、その値とプラットフォームに

よっても異なりますが、チャネルのユーザー ID、MCAUSER、メッセージ・コンテキストなどです。 チャネルのセキュリティーについて詳しくは、チャネル・セキュリティーを参照してください。

プロキシー・サブスクリプションは、リモート・キュー・マネージャーの分散パブリッシュ/サブスクライブ・エージェントのユーザー ID で作成されます。 例えば、485 ページの図 30 では QM2 がそれに該当します。 そのユーザー ID はシステムで定義されており、ドメインの競合がないため、ユーザーにはローカル・トピック・オブジェクト・プロファイルに対するアクセス権限がすぐに付与されます。

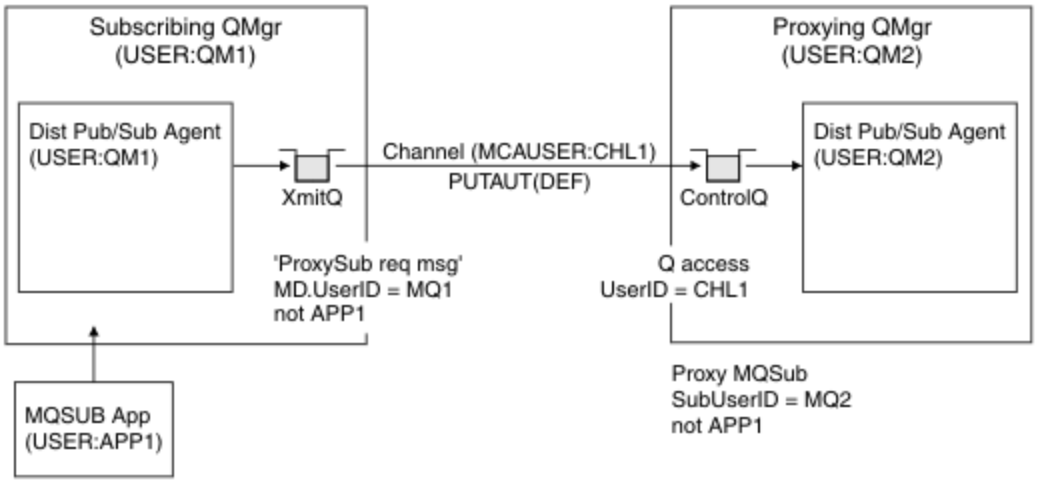


図 *30.* プロキシー・サブスクリプションのセキュリティー *(*サブスクリプションの作成*)*

リモート・パブリケーションを送り返す操作

パブリッシュ側のキュー・マネージャーでパブリケーションが作成されるときには、任意のプロキシー・サブスクリプションにパブリケーションのコピーが作成されます。 サブスクリプションを作成したユーザー ID (486 ページの図 31 では QM2) のコンテキストが、コピーされたパブリケーションのコンテキストに入ります。 プロキシー・サブスクリプションは、リモート・キューである宛先キューと共に作成されるため、パブリケーション・メッセージは伝送キューに解決されます。

他の組織のキュー・マネージャー QM2 が別のキュー・マネージャー QM1 に接続する場合、通常のチャネル認証手段によって、その組織を信頼できるかどうかが確認されます。 その組織が信頼でき、分散パブリッシュ/サブスクライブを実行することが許可されると、チャネルが分散パブリッシュ/サブスクライブ・パブリケーション・キュー SYSTEM.INTER.QMGR.PUBS にパブリケーション・メッセージを書き込む時点で、権限検査が行われます。 キューの権限検査の対象になるユーザー ID は、受信側チャネルの PUTAUT 値によって決まります (例えば、その値とプラットフォームによっても異なりますが、チャネルのユーザー ID、MCAUSER、メッセージ・コンテキストなどになります)。 チャネルのセキュリティーについて詳しくは、チャネル・セキュリティーを参照してください。

パブリケーション・メッセージがサブスクライブ側のキュー・マネージャーに達すると、そのキュー・マネージャーの権限で、そのトピックに対するもう 1 つの MQPUT が実行され、各ローカル・サブスクライバーがメッセージを受け取るたびに、メッセージのコンテキストが各ローカル・サブスクライバーのコンテキストに置き換えられます。

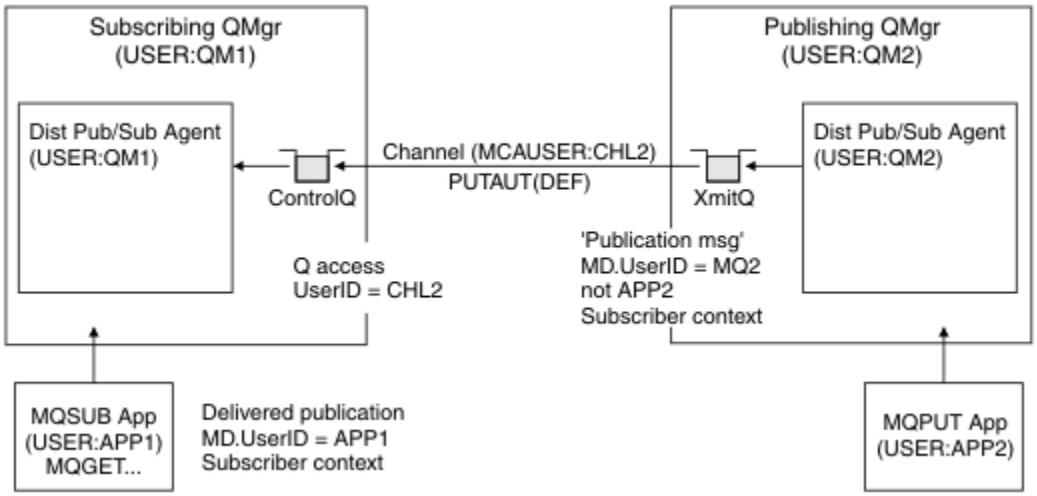


図 *31.* プロキシー・サブスクリプションのセキュリティー (パブリケーションの転送)

セキュリティーを重視していないシステムでは、ほとんどの場合、分散パブリッシュ/サブスクライブ・プロセスを mqm グループのユーザー ID で実行しており、チャネルの MCAUSER パラメーターはブランク (デフォルト) になっていて、メッセージは必要に応じてさまざまなシステム・キューに送信されます。 そのようなセキュリティーで保護されていないシステムでは、分散パブリッシュ/サブスクライブの PoC (概念検証) を簡単にセットアップできます。

セキュリティーを重視するシステムでは、それらの内部メッセージも、チャネルを通過する他のあらゆるメッセージと同じセキュリティー制御の対象になります。

チャネルのセットアップで非ブランクの MCAUSER を指定し、その MCAUSER をチェックするように PUTAUT 値を指定した場合は、対象の MCAUSER に SYSTEM.INTER.QMGR.* キューに対するアクセス権限を与える必要があります。 複数の異なるリモート・キュー・マネージャーがあり、それぞれが異なる MCAUSER ID でチャネルを実行する場合は、そのすべてのユーザー ID に SYSTEM.INTER.QMGR.* キューに対するアクセス権限を与える必要があります。 例えば、1 つのキュー・マネージャーで複数の階層接続を構成する場合に、異なる MCAUSER ID でチャネルが実行されることがあります。

チャネルのセットアップで、メッセージのコンテキストを使用するように PUTAUT 値を指定した場合は、内部メッセージの中で指定されているユーザー ID に基づいて、SYSTEM.INTER.QMGR.* キューに対するアクセス権限が検査されます。 それらのメッセージはすべて、内部メッセージまたはパブリケーション・メッセージを送信するキュー・マネージャーから、分散パブリッシュ/サブスクライブ・エージェントのユーザー ID と共に書き込まれるので (486 ページの図 31 を参照)、そのような方法で分散パブリッシュ/サブスクライブのセキュリティーをセットアップする場合は、さまざまなシステム・キューに対するアクセス権限を与えるユーザー ID のセットがそれほど大きくなりません (リモート・キュー・マネージャーごとに 1 つです)。 それでも、チャネル・コンテキスト・セキュリティーに伴う問題はすべて残ります。つまり、さまざまなユーザー ID ドメインの問題や、メッセージの中で指定されているユーザー ID が受信側のシステムで定義されていない場合がある、といった問題です。 それでも、必要であれば、このような実行方法は完全に有効です。

z/OS キューのリストと、分散パブリッシュ/サブスクライブ環境を安全にセットアップするために必要なアクセス権限については、システム・キュー・セキュリティーを参照してください。 セキュリティー違反のために内部メッセージまたはパブリケーションを書き込むことができなかった場合は、チャネルにより通常の方法でログにメッセージが書き込まれます。さらに、通常のチャネル・エラー処理に基づいて、メッセージを送達不能キューに送信できます。

分散パブリッシュ/サブスクライブのための内部キュー・マネージャー・メッセージングはすべて、通常のチャネル・セキュリティーで実行されます。

トピック・レベルでのパブリケーションとプロキシー・サブスクリプションの制限については、「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。

キュー・マネージャー階層でのデフォルトのユーザー ID の使用

異なるプラットフォームで実行されているキュー・マネージャーの階層があり、デフォルトのユーザー ID を使用している場合、そのデフォルトのユーザー ID はプラットフォームによって異なり、ターゲット・プラットフォームで認識されない可能性があることに注意してください。 結果として、一方のプラットフォーム上で実行されているキュー・マネージャーは、もう一方のプラットフォーム上のキュー・マネージャーから受信するメッセージを、理由コード MQRC_NOT_AUTHORIZED で拒否します。

メッセージが拒否されないようにするには、もう一方のプラットフォーム上で使用されるデフォルトのユーザー ID に対して、少なくとも以下の権限を追加する必要があります。

- SYSTEM.BROKER における *PUT *GET 権限 キュー
- SYSTEM.BROKER における *PUB *SUB 権限 トピック
- SYSTEM.BROKER.CONTROL.QUEUE キューにおける *ADMCRT *ADMDLT *ADMCHG 権限

キュー・マネージャー階層を持つデフォルトのユーザー ID は、以下のとおりです。

プラットフォーム	デフォルト・ユーザー ID
Windows	MUSR_MQADMIN
UNIX and Linux システム	mqm
IBM i	QMQM
z/OS	チャネル・イニシエーター・アドレス・スペースのユーザー ID

Windows、 UNIX、 Linux、および z/OS プラットフォーム上のキュー・マネージャーの IBM i 上のキュー・マネージャーに階層的に接続されている場合は、「qmqm」ユーザー ID を作成し、そのユーザー ID にアクセス権限を付与します。

IBM i および z/OS プラットフォーム上のキュー・マネージャーの Windows、 UNIX、または Linux のキュー・マネージャーに階層的に接続されている場合は、「mqm」ユーザー ID を作成し、そのユーザー ID にアクセス権限を付与します。

Windows、 UNIX、 Linux、および IBM i プラットフォーム上のキュー・マネージャーの z/OS 上のキュー・マネージャーに階層的に接続されている場合は、 z/OS チャネル・イニシエーター・アドレス・スペースのユーザー ID を作成し、そのユーザーにアクセス権限を付与します。

ユーザー ID には大/小文字の区別があります。 発信側のキュー・マネージャー (IBM i、Windows、UNIX、または Linux システムの場合) では、ユーザー ID をすべて強制的に大文字に変換します。 受信側のキュー・マネージャー (Windows、UNIX または Linux システムの場合) では、ユーザー ID をすべて強制的に小文字に変換します。 そのため、UNIX and Linux システムでは常に小文字の形式でユーザー ID を作成する必要があります。 メッセージ出口がインストールされている場合には、ユーザー ID が強制的に大文字または小文字に変換されることはありません。 メッセージ出口がユーザー ID を処理する方法をよく理解する必要があります。

ユーザー ID の変換での潜在的な問題を回避するには、以下のようにします。

- UNIX, Linux, and Windows システムでは、ユーザー ID が小文字で指定されていることを確認します。
- IBM i および z/OS では、ユーザー ID が大文字で指定されるようにしてください。

V 9.1.0 IBM MQ Console および REST API のセキュリティー

IBM MQ Console および REST API のセキュリティーは、mqwebuser.xml ファイル内の mqweb サーバー構成を編集することによって構成されます。

このタスクについて

mqweb サーバーのログ・ファイルを調べることで、ユーザーのアクションを追跡し、IBM MQ Console および REST API の使用状況を監査することができます。

IBM MQ Console および REST API のユーザーは、以下を使用して認証できます。

- 基本レジストリー
- LDAP レジストリー
- ローカル OS レジストリー
- z/OS の SAF
- WebSphere Liberty でサポートされるその他のレジストリー・タイプ

IBM MQ Console ユーザーおよび REST API ユーザーに役割を割り当て、それらのユーザーに付与する IBM MQ オブジェクトへのアクセス権限のレベルを決めることができます。 例えば、メッセージングを実行するには、ユーザーに `MQWebUser` 役割が割り当てられている必要があります。 使用可能な役割について詳しくは、498 ページの『IBM MQ Console および REST API の役割』を参照してください。

ユーザーに役割を割り当てた後、いくつかの方法でユーザーを認証することができます。 IBM MQ Console を使用する場合、ユーザーはユーザー名とパスワードでログインすることも、クライアント証明書認証を使用することもできます。 REST API を使用する場合、ユーザーは基本 HTTP 認証、トークン・ベース認証、またはクライアント証明書認証を使用できます。

手順

1. ユーザーを認証するためのユーザー・レジストリーを定義し、各ユーザーまたはグループに役割を割り当て、ユーザーおよびグループによる IBM MQ Console または REST API の使用を許可します。 詳しくは、489 ページの『ユーザーおよび役割の構成』を参照してください。
2. IBM MQ Console のユーザーが mqweb サーバーで認証を行う方法を選択します。 すべてのユーザーに対して同じ方法を使用する必要はありません。
   - トークン認証を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console ログイン画面でユーザー ID とパスワードを入力します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 この認証オプションを使用するための追加構成は不要ですが、オプションで LTPA トークンの有効期限時刻を構成することもできます。 詳しくは、LTPA トークンの有効期間の構成を参照してください。
   - クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。
3. REST API のユーザーが mqweb サーバーで認証を行う方法を選択します。 すべてのユーザーに対して同じ方法を使用する必要はありません。
   - HTTP 基本認証を使用してユーザーを認証する。 この場合、ユーザー名とパスワードはエンコードされても暗号化されず、REST API 要求ごとに送信されて、その要求に対してユーザーの認証と許可が行われます。 この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 詳しくは、504 ページの『REST API での HTTP 基本認証の使用』を参照してください。
   - トークン認証を使用してユーザーを認証する。 この場合、ユーザーは HTTP POST メソッドを使用して、REST API `login` リソースにユーザー ID とパスワードを指定します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 詳しくは、505 ページの『REST API でのトークン・ベースの認証の使用』を参照してください。

     この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 ただし、HTTP 接続を有効にしている場合は、HTTPS 接続のために発行されている LTPA トークンを HTTP 接続に使用できます。 詳しくは、LTPA トークンの構成を参照してください。
   - クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは REST API へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。
4. オプション: REST API のクロス・オリジン・リソース共有を構成します。

   デフォルトでは、スクリプトの発信元が REST API と同じでない場合には、Web ブラウザーで JavaScript などのスクリプトを使用して REST API を呼び出すことはできません。 つまり、クロス・オリジン要求

が有効になりません。 指定した URL からのクロス・オリジン要求を許可するようクロス・オリジン・リソース共有 (CORS) を構成することができます。 詳しくは、508 ページの『REST API の CORS の構成』を参照してください。

5. オプション: IBM MQ Console および REST API のためにホスト・ヘッダー検証を構成します。

   ホスト・ヘッダー検証を構成し、ホスト名とポートの許可リストを作成すると、特定のホスト・ヘッダーが設定された要求のみが IBM MQ Console および REST API によって処理されるようになります。 詳しくは、509 ページの『IBM MQ Console および REST API のホスト・ヘッダー検証の構成』を参照してください。

V 9.1.0 ユーザーおよび役割の構成

IBM MQ Console または REST API を利用するためには、ユーザーは mqweb サーバーに対して定義されたユーザー・レジストリーに基づいて認証される必要があります。

このタスクについて

認証されたユーザーは、IBM MQ Console および REST API の機能へのアクセス権限を与えるいずれかのグループのメンバーでなければなりません。 デフォルトでは、ユーザー・レジストリーにはユーザーは含まれません。これらのユーザーは、`mqwebuser.xml` ファイルを編集して追加する必要があります。

ユーザーおよびグループを構成するときは、ユーザーおよびグループを認証する際の基準となるユーザー・レジストリーをまず構成する必要があります。 このユーザー・レジストリーは、IBM MQ Console と REST API の間で共有されます。 ユーザーおよびグループのロールを構成するときに、ユーザーおよびグループが IBM MQ Console、REST API、またはその両方にアクセスできるかどうかを制御できます。

ユーザー・レジストリーを構成した後、ユーザーおよびグループの役割を構成して許可を付与します。 使用可能な役割はいくつもあり、REST API for Managed File Transfer の使用に特化した役割もあります。 各役割は別のレベルのアクセス権を付与します。 詳しくは、498 ページの『IBM MQ Console および REST API の役割』を参照してください。

ユーザーおよびグループの構成を簡単にするために、いくつかのサンプル XML ファイルが mqweb サーバーに付属しています。 WebSphere Liberty (WLP) でのセキュリティーの構成に精通しているユーザーは、サンプルを使用する必要はありません。 WLP では、ここで説明する機能のほかにも、複数の承認機能を利用できます。

手順

- `basic_registry.xml` ファイルを使用して、基本レジストリーのユーザーおよびグループを構成します。

  レジストリー内のユーザー名とパスワードが、IBM MQ Console および REST API ユーザーの認証と許可に使用されます。

  `basic_registry.xml` サンプル・ファイルを使用して基本レジストリーを構成するには、490 ページの『IBM MQ Console および REST API の基本レジストリーの構成』を参照してください。

- `ldap_registry.xml` ファイルを使用して、LDAP レジストリーのユーザーおよびグループを構成します。

  LDAP レジストリー内のユーザー名とパスワードが、IBM MQ Console および REST API の認証と使用許可に使用されます。

  `ldap_registry.xml` サンプル・ファイルを使用して LDAP レジストリーを構成するには、494 ページの『IBM MQ Console および REST API の LDAP レジストリーの構成』を参照してください。

- ULW

  `local_os_registry.xml` ファイルを使用して、ローカル・オペレーティング・システム・レジストリーのユーザーおよびグループを構成します。

  オペレーティング・システムのレジストリー内のユーザー名とパスワードが、IBM MQ Console および REST API のユーザーの認証と許可に使用されます。

local_os_registry.xml サンプル・ファイルを使用してローカル OS レジストリーを構成するには、493 ページの『IBM MQ Console および REST API のローカル OS レジストリーの構成』を参照してください。

- z/OS

  zos_saf_registry.xml ファイルを使用して、 z/OS 上の System Authorization Facility (SAF) インターフェースでユーザーおよびグループを構成します。

  RACF または他のセキュリティー製品のプロファイルを使用して、ユーザーとグループに役割へのアクセス権限を与えます。 RACF データベース内のユーザー名およびパスワードは、IBM MQ Console および REST API のユーザーを認証および許可するために使用されます。

  zos_saf_registry.xml サンプル・ファイルを使用して SAF インターフェースを構成するには、496 ページの『IBM MQ Console および REST API の SAF レジストリーの構成』を参照してください。
- no_security.xml ファイルを使用して、HTTPS を使用して IBM MQ Console または REST API にアクセスする機能を含むセキュリティーを無効にします。

次のタスク

ユーザー認証方法を選択します。

IBM MQ Console の認証オプション

- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console ログイン画面でユーザー ID とパスワードを入力します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 この認証オプションを使用するうえでこれ以上の構成は不要ですが、必要に応じて LTPA トークンの有効期間を構成できます。 詳しくは、 LTPA トークンの有効期限間隔の構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

REST API の認証オプション

- HTTP 基本認証を使用してユーザーを認証する。 この場合、ユーザー名とパスワードはエンコードされても暗号化されず、REST API 要求ごとに送信されて、その要求に対してユーザーの認証と許可が行われます。 この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 詳しくは、504 ページの『REST API での HTTP 基本認証の使用』を参照してください。
- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは HTTP POST メソッドを使用して、REST API login リソースにユーザー ID とパスワードを指定します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 詳しくは、505 ページの『REST API でのトークン・ベースの認証の使用』を参照してください。 LTPA トークンの有効期間を構成できます。 詳しくは、LTPA トークンの構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは REST API へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

V 9.1.0 IBM MQ Console および REST API の基本レジストリーの構成

基本レジストリーは mqwebuser.xml ファイル内で構成できます。 この xml ファイル内のユーザー名、パスワード、および役割が、IBM MQ Console と REST API のユーザーの 認証と許可に使用されます。

始める前に

- 基本レジストリー内でユーザーを構成する場合は、各ユーザーに役割を割り当てる必要があります。 それぞれの役割は、IBM MQ Console および REST API にアクセスするためのさまざまなレベルの特権を提供し、許可された操作が試行されるときに使用されるセキュリティー・コンテキストを決定します。 基

本レジストリーを構成する前に、これらの役割を理解しておく必要があります。 各役割の詳細については、498 ページの『IBM MQ Console および REST API の役割』を参照してください。

- このタスクを実行するには、mqwebuser.xml ファイルを編集するための十分な特権を持つユーザーでなければなりません。
  - z/OS z/OS では、mqwebuser.xml ファイルへの書き込みアクセス権限が必要です。
  - Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

手順

1. 以下のいずれかのパスからサンプルの XML ファイル basic_registry.xml をコピーします。
   - ULW UNIX, Linux, and Windows の場合、*MQ_INSTALLATION_PATH* /web/mq/samp/configuration
   - z/OS z/OS の場合、PathPrefix /web/mq/samp/configuration

     PathPrefix は、IBM MQ Unix System Services Components のインストール・パスです。
2. 以下のように、サンプル・ファイルを適切なディレクトリーに置きます。
   - ULW

     UNIX, Linux, and Windows 上: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
   - z/OS

     z/OS 上: *WLP_user_directory*/servers/mqweb

     ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
3. オプション: mqwebuser.xml の構成設定を変更した場合は、それらの設定をサンプル・ファイルにコピーします。
4. 既存の mqwebuser.xml ファイルを削除し、サンプル・ファイルの名前を mqwebuser.xml に変更します。
5. 新しい mqwebuser.xml ファイルを編集して、**basicRegistry** タグ内にユーザーとグループを追加します。

   MQWebUser 役割を持つすべてのユーザーは、ユーザー ID がキュー・マネージャーで実行を許可されている操作のみを実行できることに注意してください。 したがって、レジストリーで定義されているユーザー ID は、IBM MQ がインストールされているシステムで同一のユーザー ID を持っている必要があります。 これらのユーザー ID は大/小文字が同じである必要があります。同じでない場合、ユーザー ID 間のマッピングが失敗することがあります。

   基本ユーザー・レジストリーの構成について詳しくは、 WebSphere Liberty 資料の「Liberty の基本ユーザー・レジストリーの構成」を参照してください。
6. mqwebuser.xml ファイルを編集して、ユーザーとグループに役割を割り当てます。

   IBM MQ Console および REST API を使用する権限をユーザーとグループに与えるために、いくつかの役割を使用できます。 各役割は別のレベルのアクセス権を付与します。 詳しくは、498 ページの『IBM MQ Console および REST API の役割』を参照してください。
   - 役割を割り当て、IBM MQ Console に対するアクセス権限を付与するには、**<enterpriseApplication id="com.ibm.mq.console">** タグ内の適切な **security-role** タグの間にユーザーおよびグループを追加します。
   - 役割を割り当て、REST API に対するアクセス権限を付与するには、**<enterpriseApplication id="com.ibm.mq.rest">** タグ内の適切な **security-role** タグの間にユーザーおよびグループを追加します。

security-role タグ内のユーザーおよびグループの情報の形式については、例を参照してください。

7. mqwebuser.xml でユーザーにパスワードを指定した場合は、 WebSphere Liberty によって提供される **securityUtility encoding** コマンドを使用して、これらのパスワードをエンコードしてセキュリティーを強化する必要があります。 詳しくは、 WebSphere Liberty 製品資料の「 Liberty:securityUtility コマンド 」を参照してください。

例

以下の例では、グループ MQWebAdminGroup に、役割 MQWebAdmin を持つ IBM MQ Console へのアクセス権限が付与されます。 役割 MQWebAdminRO によってユーザー reader にアクセス権限が付与され、役割 MQWebUser によってユーザー guest にアクセス権限が付与されます。

```
<enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
            <security-role name="MQWebAdmin">
                <group name="MQWebAdminGroup" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebAdminRO">
                <user name="reader" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebUser">
                <user name="guest" realm="defaultRealm"/>
            </security-role>
        </application-bnd>
    </enterpriseApplication>
```

次の例では、ユーザー reader および guest に IBM MQ Console に対するアクセス権限が付与されます。 ユーザー user に REST API に対するアクセス権限が付与され、MQAdmin グループ内のどのユーザーにも IBM MQ Console と REST API に対するアクセス権限が付与されます。 mftadmin ユーザーには、REST API for MFT に対するアクセス権限が付与されます。

```
<enterpriseApplication id="com.ibm.mq.console">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="reader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="guest" realm="defaultRealm"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>

<enterpriseApplication id="com.ibm.mq.rest">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="user" realm="defaultRealm"/>
        </security-role>
        <security-role name="MFTWebAdmin">
            <user name="mftadmin" realm="defaultRealm"/>
        </security-role>
     </application-bnd>
</enterpriseApplication>
```

次のタスク

ユーザー認証方法を選択します。

IBM MQ Console の認証オプション

- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console ログイン画面でユーザー ID とパスワードを入力します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 この認証オプションを使用するうえでこれ以上の構成は不要です

が、必要に応じて LTPA トークンの有効期間を構成できます。 詳しくは、 LTPA トークンの有効期限間隔の構成を参照してください。

- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

REST API の認証オプション

- HTTP 基本認証を使用してユーザーを認証する。 この場合、ユーザー名とパスワードはエンコードされても暗号化されず、REST API 要求ごとに送信されて、その要求に対してユーザーの認証と許可が行われます。 この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 詳しくは、504 ページの『REST API での HTTP 基本認証の使用』を参照してください。
- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは HTTP POST メソッドを使用して、REST API `login` リソースにユーザー ID とパスワードを指定します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 詳しくは、505 ページの『REST API でのトークン・ベースの認証の使用』を参照してください。 LTPA トークンの有効期間を構成できます。 詳しくは、LTPA トークンの構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは REST API へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

V 9.1.0 ULW IBM MQ Console および REST API のローカル OS レジストリーの構成

ローカル・オペレーティング・システム・レジストリーは `mqwebuser.xml` ファイル内で構成できます。 ローカル・オペレーティング・システム上のユーザー名とパスワードが、IBM MQ Console と REST API のユーザーの認証と許可に使用されます。

始める前に

- ローカル OS 認証機能を使用するクライアント証明書認証の場合、ユーザー ID は、クライアント証明書の識別名 (DN) からの共通名 (CN) です。 ユーザー ID がオペレーティング・システム・ユーザーとして存在しない場合、クライアント証明書ログインは失敗し、パスワード・ベースの認証にフォールバックします。
- このタスクを実行するには、特権ユーザーでなければなりません。

このタスクについて

ローカル・オペレーティング・システム・レジストリーを使用すると、ユーザーとグループには次の役割が自動的に割り当てられます。

- 「mqm」グループ、または IBM i 上の「QMQMADM」グループの一部であるユーザーには、MQWebAdmin と MFTWebAdmin の役割が付与されます。
- 他のすべてのユーザーには、MQWebUser 役割が付与されます。

これらのロールについて詳しくは、498 ページの『IBM MQ Console および REST API の役割』を参照してください。

ローカル・オペレーティング・システム・レジストリーを使用できるのは、UNIX, Linux, and Windows 上だけです。 SAF レジストリーを構成することによって、同等の機能が z/OS 上で提供されます。 詳しくは、496 ページの『IBM MQ Console および REST API の SAF レジストリーの構成』を参照してください。

手順

1. 以下のパスからサンプルの XML ファイル `local_os_registry.xml` をコピーします。
   MQ_INSTALLATION_PATH`/web/mq/samp/configuration`

2. サンプル・ファイルを以下のディレクトリーに置きます。
   MQ_DATA_PATH/web/installations/*installationName*/servers/mqweb
3. オプション: mqwebuser.xml の構成設定を変更した場合は、それらの設定をサンプル・ファイルにコピーします。
4. 既存の mqwebuser.xml ファイルを削除し、サンプル・ファイルの名前を mqwebuser.xml に変更します。

次のタスク

ユーザー認証方法を選択します。

IBM MQ Console の認証オプション

- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console ログイン画面でユーザー ID とパスワードを入力します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 この認証オプションを使用するうえでこれ以上の構成は不要ですが、必要に応じて LTPA トークンの有効期間を構成できます。 詳しくは、 LTPA トークンの有効期限間隔の構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

REST API の認証オプション

- HTTP 基本認証を使用してユーザーを認証する。 この場合、ユーザー名とパスワードはエンコードされても暗号化されず、REST API 要求ごとに送信されて、その要求に対してユーザーの認証と許可が行われます。 この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 詳しくは、504 ページの『REST API での HTTP 基本認証の使用』を参照してください。
- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは HTTP POST メソッドを使用して、REST API login リソースにユーザー ID とパスワードを指定します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 詳しくは、505 ページの『REST API でのトークン・ベースの認証の使用』を参照してください。 LTPA トークンの有効期間を構成できます。 詳しくは、LTPA トークンの構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは REST API へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

V 9.1.0 IBM MQ Console および REST API の LDAP レジストリーの構成

LDAP レジストリーは mqwebuser.xml ファイル内で構成できます。 LDAP レジストリー内のユーザー名およびパスワードは、IBM MQ Console および REST API のユーザーを認証および許可するために使用されます。

始める前に

- LDAP レジストリーを構成する場合は、各ユーザーに役割を割り当てる必要があります。 それぞれの役割は、IBM MQ Console および REST API にアクセスするためのさまざまなレベルの特権を提供し、許可された操作が試行されるときに使用されるセキュリティー・コンテキストを決定します。 レジストリーを構成する前に、これらの役割を理解しておく必要があります。 各役割の詳細については、498 ページの『IBM MQ Console および REST API の役割』を参照してください。

  MQWebUser 役割を持つすべてのユーザーは、ユーザー ID がキュー・マネージャーで実行を許可されている操作のみを実行できることに注意してください。 したがって、LDAP サーバーで定義されているユーザー ID は、IBM MQ がインストールされているシステムで同一のユーザー ID を持っている必要があります。 これらのユーザー ID は大/小文字が同じである必要があります。同じでない場合、ユーザー ID 間のマッピングが失敗することがあります。

- このタスクを実行するには、mqwebuser.xml ファイルを編集するための十分な特権を持つユーザーでなければなりません。
  - z/OS z/OS では、mqwebuser.xml ファイルへの書き込みアクセス権限が必要です。
  - Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

手順

1. 以下のいずれかのパスからサンプルの XML ファイル ldap_registry.xml をコピーします。
   - ULW UNIX, Linux, and Windows の場合、*MQ_INSTALLATION_PATH* /web/mq/samp/configuration
   - z/OS z/OS の場合、PathPrefix /web/mq/samp/configuration

     PathPrefix は、IBM MQ Unix System Services Components のインストール・パスです。
2. 以下のように、サンプル・ファイルを適切なディレクトリーに置きます。
   - ULW

     UNIX, Linux, and Windows 上: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
   - z/OS

     z/OS 上: *WLP_user_directory*/servers/mqweb

     ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
3. オプション: mqwebuser.xml の構成設定を変更した場合は、それらの設定をサンプル・ファイルにコピーします。
4. 既存の mqwebuser.xml ファイルを削除し、サンプル・ファイルの名前を mqwebuser.xml に変更します。
5. 新しい mqwebuser.xml ファイルを編集して、**ldapRegistry** タグと **idsLdapFilterProperties** タグ内の LDAP レジストリーの設定を変更します。

   LDAP レジストリーの構成について詳しくは、 WebSphere Liberty 資料の「Liberty での LDAP ユーザー・レジストリーの構成」を参照してください。
6. mqwebuser.xml ファイルを編集して、ユーザーとグループに役割を割り当てます。

   IBM MQ Console および REST API を使用する権限をユーザーとグループに与えるために、いくつかの役割を使用できます。 各役割は別のレベルのアクセス権を付与します。 詳しくは、498 ページの『IBM MQ Console および REST API の役割』を参照してください。
   - 役割を割り当て、IBM MQ Console に対するアクセス権限を付与するには、**<enterpriseApplication id="com.ibm.mq.console">** タグ内の適切な **security-role** タグの間にユーザーおよびグループを追加します。
   - 役割を割り当て、REST API に対するアクセス権限を付与するには、**<enterpriseApplication id="com.ibm.mq.rest">** タグ内の適切な **security-role** タグの間にユーザーおよびグループを追加します。

次のタスク

ユーザー認証方法を選択します。

IBM MQ Console の認証オプション

- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console ログイン画面でユーザー ID とパスワードを入力します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 この認証オプションを使用するうえでこれ以上の構成は不要です

が、必要に応じて LTPA トークンの有効期間を構成できます。 詳しくは、 LTPA トークンの有効期限間隔の構成を参照してください。

- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

REST API の認証オプション

- HTTP 基本認証を使用してユーザーを認証する。 この場合、ユーザー名とパスワードはエンコードされても暗号化されず、REST API 要求ごとに送信されて、その要求に対してユーザーの認証と許可が行われます。 この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 詳しくは、504 ページの『REST API での HTTP 基本認証の使用』を参照してください。
- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは HTTP POST メソッドを使用して、REST API `login` リソースにユーザー ID とパスワードを指定します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 詳しくは、505 ページの『REST API でのトークン・ベースの認証の使用』を参照してください。 LTPA トークンの有効期間を構成できます。 詳しくは、LTPA トークンの構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは REST API へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

V 9.1.0 z/OS IBM MQ Console および REST API の SAF レジストリーの構成

System Authorization Facility (SAF) インターフェースにより、mqweb サーバーは外部セキュリティー・マネージャーを認証および許可検査のために呼び出すことができます。 その後、ユーザーは、IBM MQ Console と REST API に z/OS のユーザー ID とパスワードを使用してログインできます。

始める前に

- SAF レジストリーを構成する場合、ユーザーに役割を割り当てる必要があります。 それぞれの役割は、IBM MQ Console および REST API にアクセスするためのさまざまなレベルの特権を提供し、許可された操作が試行されるときに使用されるセキュリティー・コンテキストを決定します。 レジストリーを構成する前に、これらの役割を理解しておく必要があります。 各役割の詳細については、498 ページの『IBM MQ Console および REST API の役割』を参照してください。
- SAF への許可されたインターフェースを使用するために、WebSphere Liberty Angel プロセスを 実行する必要があります。 詳しくは、 z/OS 許可サービスを Liberty for z/OS を参照してください。
- このタスクを完了するには、`mqwebuser.xml` ファイルへの書き込みアクセス権限と、セキュリティー・マネージャー・プロファイルを定義する権限が必要です。

注 : V 9.1.0.20 IBM MQ 9.1.0 Fix Pack 20 以降、サンプル構成ファイル `zos_saf_registry.xml` が更新され、重複する safAuthorization 項目が削除されました。

この更新により、 MQ Console on z/OS が WebSphere Liberty Profile 22.0.0.12 以降 (つまり IBM MQ 9.1.0 Fix Pack 15 から) に付属するレベルにアップグレードされると、 ICH408I エラーが発生する可能性があるという問題が修正されました。 複数の safAuthorization ステートメントを指定することはサポートされておらず、EBJROLE クラスの MQWebAdmin または MQWebAdminRO ロールのいずれでもないユーザーが MQ Console を介して z/OS キュー・マネージャーにアクセスしようとすると、ICH408I エラーが発生する可能性があります。

ログに記録するアクセス試行のタイプを指定する **racRouteLog** のデフォルトは NONE です。 セキュリティー監査のために追加のレポートまたはレコードが必要な場合は、詳細について SAF 許可 (safAuthorization) を参照してください。

このタスクについて

SAF インターフェースにより、mqweb サーバーは、IBM MQ Console と REST API の両方の認証および許可検査のために外部セキュリティー・マネージャーを呼び出すことができます。

手順

1. z/OS に対する z/OS 許可サービスの Liberty の使用可能化のステップに従って、mqweb サーバーに z/OS 許可サービスを使用するためのアクセス権限を付与します。

   angel プロセスを開始するためのサンプル JCL は USS_ROOT/web/templates/zos/procs/bbgzangl.jcl にあります。ここで、USS_ROOT は、IBM MQ for z/OS USS コンポーネントがインストールされている Unix システム・サービス内のパスです。

   bbgzangl.jcl で、 USS_ROOT/web を指すように SET ROOT ステートメントを変更します (例: /usr/lpp/mqm/V9R1M0/web)。

   angel プロセスの停止および開始についての詳細は、「z/OS での Liberty の管理」を参照してください。
2. 「Liberty: システム許可機能 (SAF) 非認証ユーザーのセットアップ」の手順に従って、Liberty が必要とする非認証ユーザーを作成します。
3. zos_saf_registry.xml ファイルをパス PathPrefix /web/mq/samp/configuration からコピーします。ここで、 PathPrefix は IBM MQ Unix System Services Components のインストール・パスです。
4. サンプル・ファイルを *WLP_user_directory*/servers/mqweb ディレクトリーに置きます。ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトが実行される際に指定されたディレクトリーです。
5. オプション: mqwebuser.xml で構成設定を変更していた場合は、その設定をサンプル・ファイルにコピーします。
6. 既存の mqwebuser.xml ファイルを削除し、サンプル・ファイルの名前を mqwebuser.xml に変更します。
7. mqwebuser.xml の **safCredentials** エレメントをカスタマイズします。

   a. **profilePrefix** を Liberty サーバーで固有の名前に設定します。 単一のシステムで複数の mqweb サーバーが稼働している場合は、サーバーごとに異なる名前 (例えば、MQWEB910 と MQWEB905) を選択する必要があります。

   b. **unauthenticatedUser** は、ステップ 497 ページの『2』 で作成した非認証ユーザーの名前に設定してください。
8. mqweb サーバーの APPLID を RACF に定義します。

   APPLID リソース名は、ステップ 497 ページの『7』 で **profilePrefix** 属性に指定した値です。 RACF で mqweb サーバーの APPLID を定義する例を以下に示します。

   ```
   RDEFINE APPL profilePrefix UACC(NONE)
   ```
9. MQ Console や REST API で認証を行うすべてのユーザーやグループに、APPL クラスに含まれている mqweb サーバーの APPLID への READ アクセス権限を付与します。

   この作業は、ステップ 497 ページの『2』 で定義した非認証ユーザーに対しても行う必要があります。RACF で mqweb サーバーの APPLID に対する READ アクセス権限をユーザーに付与する例を以下に示します。

   ```
   PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
   ```
10. MQ Console と REST API の役割へのアクセス権限をユーザーに与えるために必要な EJBROLE クラスのプロファイルを定義します。

    以下の例では、 RACF でプロファイルを定義します。ここで、 **profilePrefix** は、ステップ 497 ページの『7』 で **profilePrefix** 属性に指定した値です。

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdminRO UACC(NONE)
```

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdminRO UACC(NONE)
```

11. MQ Console と REST API の役割へのアクセス権限をユーザーに付与します。

    そのために、手順 497 ページの『10』 で作成した EBJROLE クラスの 1 つ以上のプロファイルへの READ アクセス権限をユーザーやグループに付与します。 このロールについて詳しくは、498 ページの『IBM MQ Console および REST API の役割』を参照してください。

    以下の例では、 RACF 内の REST API に対する MQWebAdmin 役割へのアクセス権限をユーザーに付与します。ここで、 **profilePrefix** は、ステップ 497 ページの『7』で **profilePrefix** 属性に指定した値です。

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

タスクの結果

IBM MQ Console および REST API の SAF 認証のセットアップが完了しました。

次のタスク

ユーザー認証方法を選択します。

IBM MQ Console の認証オプション

- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console ログイン画面でユーザー ID とパスワードを入力します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 この認証オプションを使用するうえでこれ以上の構成は不要ですが、必要に応じて LTPA トークンの有効期間を構成できます。 詳しくは、 LTPA トークンの有効期限間隔の構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは IBM MQ Console へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

REST API の認証オプション

- HTTP 基本認証を使用してユーザーを認証する。 この場合、ユーザー名とパスワードはエンコードされても暗号化されず、REST API 要求ごとに送信されて、その要求に対してユーザーの認証と許可が行われます。 この認証を保護するには、セキュア接続を使用する必要があります。 つまり、HTTPS を使用する必要があります。 詳しくは、504 ページの『REST API での HTTP 基本認証の使用』を参照してください。
- トークン認証を使用してユーザーを認証する。 この場合、ユーザーは HTTP POST メソッドを使用して、REST API `login` リソースにユーザー ID とパスワードを指定します。 ユーザーが一定時間ログインと許可を維持するための LTPA トークンが生成されます。 詳しくは、505 ページの『REST API でのトークン・ベースの認証の使用』を参照してください。 LTPA トークンの有効期間を構成できます。 詳しくは、LTPA トークンの構成を参照してください。
- クライアント証明書を使用してユーザーを認証する。 この場合、ユーザーは REST API へのログインにユーザー ID もパスワードも使用せず、代わりにクライアント証明書を使用します。 詳しくは、500 ページの『REST API と IBM MQ Console でのクライアント証明書認証の使用』を参照してください。

V 9.1.0 IBM MQ Console および REST API の役割

ユーザーおよびグループに IBM MQ Console または REST API を使用する権限を与えるには、それらのユーザーおよびグループに **MQWebAdmin**、**MQWebAdminRO**、**MQWebUser**、**MFTWebAdmin**、および **MFTWebAdminRO** のいずれかの役割を割り当てる必要があります。 それぞれの役割は、IBM MQ Console および REST API にアクセスするためのさまざまなレベルの特権を提供し、許可された操作が試行されるときに使用されるセキュリティー・コンテキストを決定します。

注 : MQWebUser 役割を除き、ユーザー ID には大/小文字の区別はありません。 この役割の具体的な要件については、499 ページの『MQWebUser』 を参照してください。

MQWebAdmin

この役割を割り当てられたユーザーおよびグループはすべての管理操作を実行できます。また、mqweb サーバーの始動に使用されたオペレーティング・システムのユーザー ID のセキュリティー・コンテキストで操作を行います。

この役割を持つユーザーまたはグループには、次の REST サービスに対するアクセス権限がありません。

- MFT の REST API 。 これらのサービスを使用するには、ユーザーまたはグループに **MFTWebAdmin** 役割または **MFTWebAdminRO** 役割も割り当てる必要があります。
- messaging REST API。 messaging REST API を使用するには、ユーザーに **MQWebUser** 役割を割り当てる必要があります。

MQWebAdminRO

この役割は、IBM MQ Console または REST API への読み取り専用アクセス権を付与します。 この役割を割り当てられたユーザーおよびグループは、以下の操作を実行できます。

- キューやチャネルなどの IBM MQ オブジェクトに対する表示操作および照会操作。
- キューのメッセージの参照。

この役割を割り当てられたユーザーおよびグループは、mqweb サーバーの始動に使用されたオペレーティング・システムのユーザー ID のセキュリティー・コンテキストで操作を行います。

この役割を持つユーザーまたはグループには、次の REST サービスに対するアクセス権限がありません。

- MFT の REST API 。 これらのサービスを使用するには、ユーザーまたはグループに **MFTWebAdmin** 役割または **MFTWebAdminRO** 役割も割り当てる必要があります。
- messaging REST API。 messaging REST API を使用するには、ユーザーに **MQWebUser** 役割を割り当てる必要があります。

MQWebUser

この役割を割り当てられたユーザーおよびグループは、ユーザー ID がキュー・マネージャーで実行を許可されている操作をすべて実行できます。 以下に例を示します。

- チャネルなどの IBM MQ オブジェクトに対する開始操作および停止操作。
- キューやチャネルなどの IBM MQ オブジェクトに対する定義および設定操作。
- キューやチャネルなどの IBM MQ オブジェクトに対する表示操作および照会操作。
- messaging REST API を使用してメッセージを書き込み、取得します。

この役割を割り当てられたユーザーおよびグループは、プリンシパルのセキュリティー・コンテキストで操作を行い、 そのユーザー ID がキュー・マネージャーで実行を許可されている操作だけを実行できます。

そのため、ユーザーが操作を実行するためには、mqweb ユーザー・レジストリーで定義されているそのユーザーまたはグループに対して IBM MQ 内で事前に権限を付与しておく必要があります。 この役割を使用すると、IBM MQ Console および REST API を使用するときに、どのユーザーが特定の IBM MQ リソースに対してどのタイプのアクセス権限を持つかを細かく制御できます。

注 :

- この役割を割り当てられるユーザー ID の最大長は 12 文字です。
- ユーザー ID の大/小文字は、mqweb ユーザー・レジストリーおよび IBM MQ システムで同じである必要があります。 ユーザー ID の大文字と小文字が異なる場合、ユーザーは IBM MQ Console および REST API によって認証されますが、IBM MQ リソースを使用する権限は与えられません。

この役割を持つユーザーまたはグループには、REST API for MFT のどのサービスに対するアクセス権限もありません。 これらのサービスを使用するには、ユーザーまたはグループに **MFTWebAdmin** 役割または **MFTWebAdminRO** 役割も割り当てる必要があります。

MFTWebAdmin
この役割を割り当てられたユーザーまたはグループは、すべての MFT REST 操作を実行でき、mqweb サーバーの始動に使用されるオペレーティング・システム・ユーザー ID のセキュリティー・コンテキストで操作を行います。

この役割を持つユーザーまたはグループには、IBM MQ REST API のどのサービスに対するアクセス権限もありません。 これらのサービスを使用するには、ユーザーまたはグループに **MQWebAdmin** 役割、**MQWebAdminRO** 役割、または **MQWebUser** 役割も割り当てる必要があります。

MFTWebAdminRO
この役割は、REST API for MFT への読み取り専用アクセス権を付与します。 このロールを割り当てられたユーザーやグループは、転送やエージェントのリスト表示などの読み取り専用操作 (GET 要求) を実行できます。

この役割を割り当てられたユーザーおよびグループは、mqweb サーバーの始動に使用されたオペレーティング・システムのユーザー ID のセキュリティー・コンテキストで操作を行います。

この役割を持つユーザーまたはグループには、IBM MQ REST API のどのサービスに対するアクセス権限もありません。 これらのサービスを使用するには、ユーザーまたはグループに **MQWebAdmin** 役割、**MQWebAdminRO** 役割、または **MQWebUser** 役割も割り当てる必要があります。

これらの役割を使用するようにユーザーとグループを構成する方法について詳しくは、489 ページの『ユーザーおよび役割の構成』を参照してください。

オーバーラップする役割

1 つのユーザーまたはグループに複数の役割を割り当てることができます。 この状態でユーザーが操作を実行すると、その操作に適用可能な最高の特権の役割が使用されます。 例えば、役割 **MQWebAdminRO** および **MQWebUser** を持つユーザーがキューの照会操作を実行した場合は、**MQWebAdminRO** 役割が使用され、Web サーバーを始動したシステム・ユーザー ID のコンテキストで操作が試行されます。 その同じユーザーが定義操作を実行した場合は、**MQWebUser** 役割が使用され、プリンシパルのコンテキストで操作が試行されます。

V 9.1.0 ULW REST API と IBM MQ Console でのクライアント証明書認証の使用

クライアント証明書をプリンシパルにマップして、IBM MQ Console および REST API ユーザーを認証することができます。

始める前に

- IBM MQ Console と REST API の使用が許可されるようユーザー、グループ、および役割を構成します。 詳しくは、489 ページの『ユーザーおよび役割の構成』を参照してください。
- REST API を使用するときに、`login` リソースで HTTP GET メソッドを使用して、現在のユーザーの資格情報を照会することができます。その際、クライアント証明書を提供して要求を認証します。 この要求は、ユーザー名、およびユーザーに割り当てられている役割に関する情報を返します。 詳しくは、GET /login を参照してください。
- ユーザー認証のためにクライアント証明書をプリンシパルにマップした場合、構成されたユーザー・レジストリー内のユーザーと照合するために、クライアント証明書の識別名が使用されます。
  - 基本レジストリーの場合は、共通名 (CN) がユーザーと照合されます。 例えば、`CN=Fred, O=IBM, C=GB` の場合は `Fred` というユーザー名が照合されます。
  - LDAP レジストリーの場合は、デフォルトでは、完全識別名が LDAP で照合されます。 フィルターとマッピングをセットアップして、照合方法をカスタマイズすることができます。 詳しくは、WebSphere Liberty 資料の Liberty :LDAP 証明書マップ・モード を参照してください。

このタスクについて

クライアント証明書を使用してユーザーを認証する場合は、ユーザー名とパスワードの代わりに証明書が使用されます。 REST API では、ユーザーを認証するために REST 要求が行われるたびにクライアント証明書が提供されます。 IBM MQ Console では、ユーザーが証明書を使用してログインすると、そのユーザーはログアウトできなくなります。

この手順では、次の情報を前提としています。

- `mqwebuser.xml` ファイルが以下のサンプルのいずれかに基づいている。
  - `basic_registry.xml`
  - `local_os_registry.xml`
  - `ldap_registry.xml`
- UNIX、Linux、または Windows システムを使用している。
- 特権ユーザーである。

z/OS で RACF 鍵リングを使用したクライアント証明書認証を構成するには、513 ページの『REST API および IBM MQ Console の TLS を z/OS で設定します。』の手順に従います。

注 : 次の手順では、IBM MQ Console と REST API でクライアント証明書を使用するために必要なステップが概略されています。 開発者の便宜を考慮して、手順では自己署名証明書を作成および使用する方法について詳しく説明します。 ただし、実動では、認証局から取得した証明書を使用します。

手順

1. コマンド行で **strmqweb** コマンドを入力して、mqweb サーバーを始動します。
2. クライアント証明書を作成します。
   a) PKCS#12 鍵ストアを作成します。
      i) コマンド行で **strmqikm** コマンドを入力して、IBM 鍵管理ツールを開きます。
      ii) IBM 鍵管理ツールの「**鍵データベース・ファイル**」メニューで、「**新規**」をクリックします。
      iii)「**鍵データベース・タイプ**」リストから「**PKCS12**」を選択します。
      iv) 鍵ストアを保存する場所を選択して、「**ファイル名**」フィールドに適切な名前を入力します。 例えば、`user.p12`
      v) プロンプトが表示されたら、パスワードを設定します。
   b) 自己署名証明書を作成するか、認証局から証明書を取得することによって、証明書を作成します。
      - 自己署名証明書を作成します。
        i)「**新規自己署名**」をクリックします。
        ii)「**鍵ラベル**」フィールドに `user` を入力します。
        iii) 基本ユーザー・レジストリーを使用している場合、ユーザー・レジストリーのユーザーの名前を「**共通名**」フィールドに入力します。 例えば、`mqadmin` です。 LDAP ユーザー・レジストリーの場合、証明書の識別名が LDAP レジストリーの識別名と一致していることを確認してください。
        iv) **OK** をクリックします。
      - 認証局から証明書を取得します。 CA 証明書では、識別名 (DN) フィールドの共通名 (CN) に適切なユーザー名を含める必要があります。
        i) 新しい証明書を要求します。 「**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。
        ii)「**鍵ラベル**」フィールドに証明書ラベルを入力します。
        iii) 基本ユーザー・レジストリーを使用している場合、「**共通名**」フィールドに、証明書ユーザーのユーザー名を入力します。

ローカル OS レジストリーを使用している場合、「**共通名**」フィールドがローカル OS ユーザー ID と一致している必要があります。

LDAP ユーザー・レジストリーの場合、証明書の識別名が LDAP レジストリーの識別名と一致していることを確認してください。

iv) 必要に応じて、残りのフィールドに値を入力または選択します。

v) 証明書要求の保存場所、および証明書要求のファイル名を選択し、「**OK**」をクリックします。

vi) この証明書要求ファイルを認証局 (CA) に送信します。

vii) CA からの証明書がある場合、コマンド行で **strmqikm** コマンドを入力して、IBM 鍵管理ツールを開きます。

viii) IBM 鍵管理ツールの「**鍵データベース・ファイル**」メニューで、「**Open (オープン)**」をクリックします。

ix) クライアント証明書を保持する PKCS#12 鍵ストアを選択します。 例: `user.p12`

x)「**Receive (受信)**」をクリックし、適切な証明書を選択し、「**OK**」をクリックします。

3. クライアント証明書の公開部分を抽出します。
   a) コマンド行で **strmqikm** コマンドを入力して、IBM 鍵管理ツールを開きます。
   b) IBM 鍵管理ツールの「**鍵データベース・ファイル**」メニューで、「**Open (オープン)**」をクリックします。
   c) クライアント証明書を保持する PKCS#12 鍵ストアを選択します。 例: `user.p12`
   d) IBM 鍵管理ツールの証明書リストからクライアント証明書を選択します。
   e)「**Extract Certificate (証明書の取り出し)**」をクリックします。
   f) 証明書を保存する場所を選択して、「**Certificate file name (証明書ファイル名)**」フィールドに適切な名前を入力します。 例えば、`user.arm` です。
4. クライアント証明書の公開部分を mqweb サーバー・トラスト鍵ストアに署名者証明書としてインポートして、サーバーでクライアント証明書を検証できるようにします。
   a) mqweb サーバーで使用する `trust.jks` 鍵ストアがまだ存在していない場合は、それを作成します。
      i) IBM 鍵管理ツールの「**鍵データベース・ファイル**」メニューで、「**新規**」をクリックします。
      ii)「**鍵データベース・タイプ**」リストから「**JKS**」を選択します。
      iii) **参照** をクリックし、 `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb/resources/security` にナビゲートします

      このディレクトリーには既に `key.jks` ファイルが含まれています。 `trust.jks` ファイルが既に存在する場合は、それを上書きするのではなく、既存のファイルを開きます。
      iv)「**ファイル名**」フィールドに `trust.jks` と入力します。
      v) プロンプトが表示されたら、パスワードを設定します。
   b) ドロップダウン・メニューから、「**署名者証明書**」を選択します。
   c) **追加** をクリックします。
   d) 適切な arm ファイルを選択して、「**OK**」をクリックします。 例えば、`user.arm` を選択します。
   e) 証明書のラベルを入力します。
5. mqweb サーバーの鍵ストアのパスワードを変更します。
   a)「**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。
   b)「**鍵データベース・タイプ**」リストから「**JKS**」を選択します。
   c)「**参照**」 をクリックし、*MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb/resources/security にナビゲートします
   d) `key.jks` 鍵ストアを選択し、「**Open (オープン)**」をクリックします。
   e) プロンプトが表示されたらパスワードを入力します。 デフォルトのパスワードは `password` です。

f)「**Key Database File (鍵データベース・ファイル)**」メニューから、「**パスワード変更**」をクリックします。

g) 鍵ストアの新規パスワードを入力します。

6. mqwebuser.xml ファイルでクライアント証明書認証を有効にします。

   mqwebuser.xml ファイルは、次のパス上にあります。*MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   a) mqwebuser.xml ファイル内のセクションのコメントを外し、クライアント証明書認証を有効にします。 そのセクションには、以下のテキストが含まれています。

   ```
   <keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
       <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
       <ssl id="thisSSLConfig" clientAuthenticationSupported="true"
   keyStoreRef="defaultKeyStore"
               trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"
   serverKeyAlias="default"/>
       <sslDefault sslRef="thisSSLConfig"/>
   ```

   b)「**serverKeyAlias**」の値がサーバー証明書の名前と一致することを確認します。 デフォルトのサーバー証明書を使用する場合、正しい値になります。

   c) defaultKeyStore の「**パスワード**」の値を、key.jks 鍵ストアのパスワードのエンコード・バージョンに変更します。

      i) *MQ_INSTALLATION_PATH* /web/bin ディレクトリから、コマンドラインに次のコマンドを入力します。

      ```
      securityUtility encode password
      ```

      ii) このコマンドの出力を defaultKeyStore の「**パスワード**」フィールドに置きます。

   d) defaultTrustStore の「**パスワード**」の値を、trust.jks 鍵ストアのパスワードと一致するように変更します。

      i) *MQ_INSTALLATION_PATH* /web/bin ディレクトリから、コマンドラインに次のコマンドを入力します。

      ```
      securityUtility encode password
      ```

      ii) このコマンドの出力を defaultTrustStore の「**パスワード**」フィールドに置きます。

   e) mqwebuser.xml ファイルから、以下の行を除去するか、コメント化します。

   ```
   <sslDefault sslRef="mqDefaultSSLConfig"/>
   ```

7. コマンド行で **endmqweb** コマンドを入力して、mqweb サーバーを停止します。
8. コマンド行で **strmqweb** コマンドを入力して、mqweb サーバーを始動します。
9. クライアント証明書を使用して認証を行います。

   - IBM MQ Console でクライアント証明書を使用するには、IBM MQ Console へのアクセスに使用する Web ブラウザーにクライアント証明書をインストールします。 例えば、クライアント証明書 user.p12 を個人証明書としてインストールします。
   - REST API でクライアント証明書を使用するには、REST 要求が行われるたびにクライアント証明書を提供します。 HTTP POST、PATCH、または DELETE メソッドを使用する場合、クロスサイト・リクエスト・フォージェリー攻撃を防止するために、クライアント証明書に追加認証を提供する必要があります。 つまり、要求の認証に使用する資格情報がその所有者によって使用されていることを確認するために、追加認証を使用します。

     この追加認証は、ibm-mq-rest-csrf-token HTTP ヘッダーで指定します。 ibm-mq-csrf-token ヘッダーの値を、任意の値 (ブランクでも構いません) に設定してから、要求を実行依頼します。

例

重要 : この例では、すべての cURL 実装が自己署名証明書をサポートしているわけではないので、サポートしている cURL 実装を使用する必要があります。

次の cURL の例では、クライアント証明書認証を使用してキュー・マネージャー QM1 で新規キュー Q1 を作成する方法を示しています。 この cURL コマンドの正確な構成は、cURL がビルドされたライブラリーによって異なります。 この例は、cURL が OpenSSL に対してビルドされている Windows システムに基づいています。

- キュー・リソースを指定して HTTP POST メソッドを実行し、クライアント証明書で認証を行い、任意の値を指定した ibm-mq-rest-csrf-token HTTP ヘッダーを組み込みます。 この値は任意の値にすることができます (ブランクでも構いません)。 --cert-type フラグは、証明書が PKCS#12 証明書であることを指定します。 --cert フラグは、証明書の場所を指定し、その後にコロン (:)、証明書のパスワードの順に指定します。

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -
-cert-type P12 --cert c:\user.p12:password
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

V 9.1.0 REST API での HTTP 基本認証の使用

REST API のユーザーは、HTTP ヘッダー内に自分のユーザー ID とパスワードを指定することで認証できます。 HTTP メソッド (POST、PATCH、DELETE など) によってこの認証方式を使用する場合は、ユーザー ID とパスワードのほかに ibm-mq-rest-csrf-token HTTP ヘッダーも指定する必要があります。

始める前に

- REST API を使用する権限を与えるユーザー、グループ、および役割を構成します。 詳しくは、489 ページの『ユーザーおよび役割の構成』を参照してください。
- HTTP 基本認証を有効にしておきます。 mqwebuser.xml ファイルに次の XML が存在し、コメント化されていないことを確認してください。 この XML は、<featureManager> タグ内になければなりません。

```
<feature>basicAuthenticationMQ-1.0</feature>
```

z/OS z/OS では、このファイルを編集するために mqwebuser.xml への書き込み権限を持つユーザーでなければなりません。

Multi その他のすべてのオペレーティング・システムでは、mqwebuser.xml ファイルを編集するには、「特権ユーザー」 でなければなりません。

- REST 要求を送信するときは、セキュア接続を使用していることを確認してください。 ユーザー名とパスワードの組み合わせはエンコードされても暗号化されないため、REST API で HTTP 基本認証を使用するときは、セキュア接続 (HTTPS) を使用する必要があります。
- login リソースに対する HTTP GET メソッドを使用することにより、現行ユーザーの資格情報を照会できます。このメソッドを使用する際、その要求を認証するための基本認証情報を指定する必要があります。 この要求は、ユーザー名、およびユーザーに割り当てられている役割に関する情報を返します。 詳しくは、 GET /login を参照してください。

手順

1. ユーザー名とパスワードをコロンで連結します。 ユーザー名は大/小文字が区別されることに注意してください。
   例えば、ユーザー名が admin でパスワードが admin の場合は、次のストリングになります。

   ```
   admin:admin
   ```

2. ユーザー名とパスワードのこのストリングを base64 エンコードでエンコードします。

3. エンコードされたこのユーザー名とパスワードを HTTP `Authorization: Basic` ヘッダーに組み込みます。

   例えば、ユーザー名 admin とパスワード admin がエンコードされた場合、次のヘッダーが作成されます。

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```

4. HTTP の POST、PATCH、DELETE のいずれかのメソッドを使用する場合は、ユーザー名とパスワードと一緒に追加の認証を指定する必要があります。

   この追加認証は、`ibm-mq-rest-csrf-token` HTTP ヘッダーで指定します。 `ibm-mq-rest-csrf-token` HTTP ヘッダーは要求の中に存在する必要がありますが、その値はブランクを含めどのような値でも構いません。

5. 適切なヘッダーとともに REST 要求を IBM MQ に実行依頼します。

例

以下の例は、基本認証を使用して、 Windows システム上のキュー・マネージャー QM1 に新しいキュー Q1 を作成する方法を示しています。 この例では cURL を使用しています。

- キュー・リソースを指定して HTTP POST メソッドを実行し、基本認証で認証を行い、任意の値を指定した `ibm-mq-rest-csrf-token` HTTP ヘッダーを組み込みます。 この値は任意の値にすることができます (ブランクでも構いません)。

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST
-u mqadmin:mqadmin
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

V 9.1.0 REST API でのトークン・ベースの認証の使用

REST API のユーザーは、HTTP POST メソッドを使用して、REST API の `login` リソースにユーザー ID とパスワードを指定することによって認証できます。 ユーザーが今後の要求を認証できるようにする LTPA トークンが生成されます。 この LTPA トークンの接頭部は、`LtpaToken2` です。 HTTP DELETE メソッドを使用するとログアウトでき、HTTP GET メソッドを使用すると現行ユーザーのログイン情報を照会できます。

始める前に

- REST API を使用する権限を与えるユーザー、グループ、および役割を構成します。 詳しくは、489 ページの『ユーザーおよび役割の構成』を参照してください。
- LTPA トークンが含まれる Cookie の名前のデフォルトは、先頭が `LtpaToken2` で、mqweb サーバーの再始動時に変更される可能性がある接尾部が付きます。 このように Cookie 名がランダム化されているので、複数の mqweb サーバーを同一のシステム上で実行できます。 しかし、Cookie 名を一定の値にしておく場合は、**setmqweb** コマンドを使用して Cookie の名前を指定することができます。 詳しくは、LTPA トークンの構成を参照してください。
- デフォルトでは、LTPA トークンの Cookie は 120 分後に有効期限が切れます。 LTPA トークンの Cookie の有効期限の時間は、**setmqweb** コマンドを使用して構成できます。 詳しくは、LTPA トークンの構成を参照してください。
- REST 要求を送信するときは、セキュア接続を使用していることを確認してください。 `login` リソースで HTTP POST メソッドを使用するとき、要求とともに送信されるユーザー名とパスワードの組み合わせは暗号化されません。 したがって、REST API によるトークン・ベースの認証を使用するときは、セキュア接続 (HTTPS) を使用する必要があります。 デフォルトでは、HTTP を LTPA トークン認証で使用することはできません。 **secureLTPA** を `False` に設定することによって、セキュアでない HTTP 接続で LTPA トークンを使用できます。 詳しくは、LTPA トークンの構成を参照してください。
- `login` リソースに対する HTTP GET メソッドを使用することにより、現行ユーザーの資格情報を照会できます。このメソッドを使用する際、その要求を認証するための LTPA トークンを指定する必要がありま

す。 この要求は、ユーザー名、およびユーザーに割り当てられている役割に関する情報を返します。 詳しくは、GET /login を参照してください。

手順

1. ユーザーをログインします。
   a) login リソースで HTTP POST メソッドを使用します。

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   JSON 要求の本体にユーザー名とパスワードを、以下の形式で組み込みます。

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) その要求から返された LTPA トークンをローカル Cookie ストアに保管します。 この LTPA トークンの接頭部は、デフォルトでは LtpaToken2 です。
2. すべての要求の Cookie として、保管した LTPA トークンを使用して、REST 要求の認証を行います。
   HTTP の PUT、PATCH、DELETE のいずれかのメソッドを使用する要求には ibm-mq-rest-csrf-token ヘッダーを組み込みます。 このヘッダーの値は、ブランクでも他のどんな値でも構いません。
3. ユーザーをログアウトします。
   a) login リソースで HTTP DELETE メソッドを使用します。

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

   要求の認証を行うための Cookie として LTPA トークンを指定する必要があります。ibm-mq-rest-csrf-token ヘッダーも組み込んでください。 このヘッダーの値は、ブランクでも他のどんな値でも構いません。

   b) ローカルの Cookie ストアから LTPA トークンを削除するための命令を処理します。

   注 : この命令を処理せず、LTPA トークンがローカル Cookie ストアに残っている場合、その LTPA トークンを使用して以降の REST 要求の認証を受けることができます。 つまり、セッションの終了後にユーザーがその LTPA トークンを使用して認証を試みると、既存のトークンを使用する新しいセッションが作成されます。

例

以下の cURL の例は、 Windows システムで、トークン・ベースの認証を使用して、キュー・マネージャー QM1 に新しいキュー Q1 を作成する方法を示しています。

- ログインして、接頭部が LtpaToken2 の LTPA トークンをローカルの Cookie ストアに追加します。 ユーザー名とパスワード情報は JSON 本体に組み込まれています。 -c フラグはトークンを保管するファイルの場所を指定するためのフラグです。

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/login -X POST
  -H "Content-Type: application/json" --data
  "{\"username\":\"mqadmin\",\"password\":\"mqadmin\"}"
  -c c:\cookiejar.txt
  ```

- キューを作成します。 キュー・リソースを使用して HTTP POST メソッドを実行し、LTPA トークンで認証を行います。 接頭部が LtpaToken2 の LTPA トークンは、-b フラグを使用して cookiejar.txt ファイルから取得されます。 ibm-mq-rest-csrf-token HTTP ヘッダーを組み込むことによって、CSRF 保護を指定します。

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -b
  c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json"
  --data "{\"name\":\"Q1\"}"
  ```

- ローカルの Cookie ストアからログアウトし、LTPA トークンを削除します。 -b フラグを使用して、cookiejar.txt ファイルから LTPA トークンを取得します。 ibm-mq-rest-csrf-token HTTP ヘッダーを組み込むことによって、CSRF 保護を指定します。 以下のように、cookiejar.txt ファイルの場所は -c フラグによって指定されるため、LTPA トークンはファイルから削除されます。

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X DELETE
-H "ibm-mq-rest-csrf-token: value" -b c:\cookiejar.txt
-c c:\cookiejar.txt
```

関連資料

POST /login

GET /login

/login の削除

V 9.1.3 IFrame による IBM MQ Console の組み込み

HTML <iframe> 要素を使用して、1 つの Web ページをインライン・フレーム (IFrame) によって別のページに組み込むことができます。 セキュリティー上の理由により、デフォルトでは、IBM MQ Console を IFrame に組み込むことはできません。 ただし、mqweb サーバーの **mqConsoleFrameAncestors** 構成プロパティーを使用して IFrame を有効にできます。

このタスクについて

mqweb サーバーは、IFrame を使用して IBM MQ Console を埋め込むことができる Web ページのオリジンの許可リストを維持します。 オリジンは、URL スキーム、ドメイン、ポートの組み合わせです (例: https://example.com:1234)。

mqweb サーバーで **mqConsoleFrameAncestors** 構成プロパティーを使用して、リスト内の項目を指定できます。

デフォルトでは、**mqConsoleFrameAncestors** はブランクです。つまり、IBM MQ Console を IFrame で組み込むことはできません。

手順

以下のコマンドを入力して、IFrame で IBM MQ Console を組み込むことができる Web ページのオリジンのリストを指定します。

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

ここで、*allowedOrigins* はオリジンのコンマ区切りリストです。 各オリジンは以下の内容で構成されます。

- ホスト名または IP アドレス
- URL スキーム (オプション)
- ポート番号 (オプション)

ホスト名の先頭をワイルドカード文字 (*) にしたり、ポート番号でワイルドカード文字 (*) を使用したりもできます。
オリジンの例:

```
https://example.com:1234
```

この場合、https://example.com:1234 からのすべての Web ページで、IFrame を使用して IBM MQ Console を組み込むことができます。

```
https://*.example.com:*
```

この場合、ホスト名の末尾が example.com で、任意のポートを使用する HTTPS Web ページにおいて、IFrame を使用して IBM MQ Console を組み込むことができます。

例

以下の例では、 https://site2.example.com:1234 または https://site2.example.com:1235 から提供される Web ページから IFrame に IBM MQ Console を組み込むことができます。

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

V 9.1.0 REST API の CORS の構成

デフォルトでは、スクリプトの発信元が REST API と同じでない場合には、Web ブラウザーで JavaScript などのスクリプトを使用して REST API を呼び出すことはできません。 つまり、クロス・オリジン要求が有効になりません。 指定した発信元からのクロス・オリジン要求を許可するようクロス・オリジン・リソース共有 (CORS) を構成することができます。

このタスクについて

Web ブラウザーを介して、スクリプトなどを使用して REST API にアクセスできます。 これらはさまざまな発信元から REST API に対する要求、つまりクロス・オリジン要求であるため、Web ブラウザーに拒否されます。 ドメイン、ポート、またはスキームが同一でない場合、発信元は異なります。

例えば、http://localhost:1999/ でホストされているスクリプトがある場合、https://localhost:9443/ でホストされている Web サイトで HTTP GET を発行すると、クロス・オリジン要求を作成できます。 この要求がクロス・オリジン要求になるのは、ポート番号とスキーム (HTTP) が異なるためです。

CORS を構成し、REST API へのアクセスが許可されている発信元を指定することにより、クロス・オリジン要求を有効にできます。

CORS について詳しくは、https://www.w3.org/TR/cors/ および https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS を参照してください。

手順

1. 以下のコマンドを入力して、現在の構成を表示します。

   ```
   dspmqweb properties -a
   ```

   mqRestCorsAllowedOrigins エントリーは、許可される発信元を指定します。 mqRestCorsMaxAgeInSeconds エントリーでは、Web ブラウザーが CORS プリフライト検査の結果をキャッシュできる時間 (秒数) を指定します。
2. 以下のコマンドを入力して、REST API へのアクセスが許可される発信元を指定します。

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   ここで、*allowedOrigins* は、クロス・オリジン要求の発行を許可する発信元を指定します。 二重引用符で囲まれたアスタリスク ("*") を使用すると、すべてのクロス・オリジン要求を許可することができます。 二重引用符で囲まれたコンマ区切りリストで複数の発信元を入力することができます。 クロス・オリジン要求を許可しない場合は、*allowedOrigins* の値に空の引用符を入力します。
3. 以下のコマンドを入力して、Web ブラウザーが CORS プリフライト検査の結果をキャッシュできるようにする時間 (秒) を指定します。

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

例

以下の例は、http://localhost:9883、https://localhost:1999、および https://localhost:9663 に対して有効化されたクロス・オリジン要求を示しています。 CORS プリフライト検査の結果がキャッシュされる最長期間は、90 秒に設定されます。

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

IBM MQ Console および REST API のホスト・ヘッダー検証の構成

指定された許可リストと一致するホスト・ヘッダーを付けて送信された要求のみが処理されるように、IBM MQ Console および REST API へのアクセスを制限するよう mqweb サーバーを構成できます。 許可リストにないホスト・ヘッダー値が使用されている場合は、エラーが返されます。

このタスクについて

mqweb サーバーでは、仮想ホストを使用して、許容可能なホスト・ヘッダーの許可リストを定義します。仮想ホストについて詳しくは、WebSphere Liberty の資料を参照してください。「https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html」

このタスクを実行するには、mqwebuser.xml ファイルを編集するための十分な特権を持つユーザーでなければなりません。

- z/OS z/OS では、mqwebuser.xml ファイルへの書き込みアクセス権限が必要です。
- Multi 他のすべてのオペレーティング・システムでは、特権ユーザーでなければなりません。

手順

1. mqwebuser.xml ファイルを開きます。 このファイルは、次のいずれかのロケーションにあります。
   - ULW

     UNIX, Linux, and Windows 上: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
   - z/OS

     z/OS 上: *WLP_user_directory*/servers/mqweb

     ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行したときに指定したディレクトリーです。
2. mqwebuser.xml ファイル内の以下のコードを追加またはコメント解除する

   ```
   <virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
         <hostAlias>localhost:9080</hostAlias>
   </virtualHost>
   ```
3. **<hostAlias>** フィールドを編集して、許可するホスト名とポートの組み合わせを挿入します。

   この組み合わせは、mqweb サーバーの構成で使用したホスト名とポート名である可能性があります。例えば、デフォルト構成の localhost:9443 を使用する場合、**<hostAlias>** フィールドで localhost:9443 を使用する可能性があります。

   必要な場合は、ホスト名とポートの組み合わせをさらに許可するために、**<virtualHost>** タグ内に複数の **<hostAlias>** フィールドを追加することもできます。 例えば、HTTP ポートを使用するホスト・ヘッダーと、HTTPS ポートを使用するホスト・ヘッダーを許可するために、これを追加できます。

V 9.1.0 監査

キュー・マネージャーのコマンド・イベントと構成イベントを有効にして、IBM MQ Console と REST API で実行する操作の監査レコードを作成できます。UNIX, Linux, and Windows では、重要な状態変更が mqweb サーバーのログ・ファイルに記録されます。

重要な状態変更

ULW

UNIX, Linux, and Windows では、IBM MQ Console は、重要な状態変更を mqweb サーバーのログ内にメッセージとして記録します。 各メッセージには、操作を要求した認証済みのプリンシパル名が示されます。

キュー・マネージャーが作成、開始、終了、削除されるなどの重要な状態変更が、 mqweb サーバーの `messages.log` および `console.log` ファイルに [AUDIT] ロギング・レベルで記録されます。 各ログ項目には、その操作を要求した認証済みのプリンシパル名が示されます。

`messages.log` および `console.log` ファイルは、次の場所にあります。

- ULW UNIX, Linux, and Windows 上: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb\logs

mqweb サーバーのロギング・レベルの構成の詳細については、ロギングの構成を参照してください。

コマンド・イベントと構成イベント

オプションでキュー・マネージャー上のコマンドおよび構成イベントを使用可能にして、ほとんどの IBM MQ Console および REST API アクティビティーに関する情報を提供することができます。 例えば、チャネルの作成やキューの照会は、コマンド・イベントおよび構成イベントを生成します。 コマンド・イベントおよび構成イベントを有効にする方法について詳しくは、『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』を参照してください。

これらのコマンド・イベントおよび構成イベントのメッセージでは、MQIACF_EVENT_ORIGIN フィールドに MQEVO_REST が設定され、MQCACF_EVENT_APPL_IDENTITY フィールドには認証済みのプリンシパル名の最初の 32 文字が報告されます。 ユーザーに **MQWebAdmin** 役割または **MQWebAdminRO** 役割がある場合、 MQCACF_EVENT_USER_ID フィールドには、mqweb サーバー・ユーザー ID ではなく、Web サーバーを始動したプリンシパルのユーザー名が報告されます。 ただし、ユーザーに **MQWebUser** 役割がある場合は、MQCACF_EVENT_USER_ID にはコマンドを発行したプリンシパルのユーザー名が報告されます。

関連概念

454 ページの『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 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。

z/OS 上の IBM MQ Console および REST API のセキュリティーに関する考慮事項

IBM MQ Console および REST API には、ユーザーがコマンドを発行、表示、または変更できるかどうかを制御するセキュリティー機能があります。 その後、コマンドはキュー・マネージャーに渡され、キュー・マネージャー・セキュリティーは、ユーザーがその特定のキュー・マネージャーに対してコマンドを発行できるかどうかを制御するために使用されます。

手順

1. mqweb サーバー開始タスクのユーザー ID に、特定の PCF コマンドを発行し、特定のキューにアクセスするための適切な権限があることを確認します。 詳しくは、511 ページの『mqweb サーバー開始タスクのユーザー ID に必要な権限』を参照してください。
2. `MQWebUser` 役割が付与されたすべてのユーザーが適切な権限を持っていることを確認します。

MQWebUser 役割に割り当てられた IBM MQ Console および REST API ユーザーは、プリンシパルのセキュリティー・コンテキストの下で操作します。 このようなユーザー ID は、そのユーザー ID がキュー・マネージャーで実行を許可されている操作のみを実行できます。また、mqweb サーバーのアドレス・スペースと同じシステム・キューに対するアクセス権限を付与される必要があります。

mqweb サーバー開始タスクのユーザー ID には、MQWebUser 役割に割り当てられたすべてのユーザーに対する 代替ユーザー・アクセス権限が付与される必要があります。

MQWebUser 役割を持つユーザーに対する適切な権限付与について詳しくは、511 ページの『IBM MQ または MQ Console の使用に必要な REST API リソースへのアクセス』 を参照してください。

3. オプション: IBM MQ Console と REST API に TLS を構成します。 詳しくは、513 ページの『REST API および IBM MQ Console の TLS を z/OS で設定します。』 を参照してください。

z/OS mqweb サーバー開始タスクのユーザー ID に必要な権限

z/OS では、mqweb サーバー開始タスクのユーザー ID に、PCF コマンドを発行し、システム・リソースにアクセスするための特定の権限が必要になります。

mqweb サーバー開始タスクのユーザー ID には、以下が必要です。

- z/OS UNIX System Services を使用できるようにするための z/OS UNIX ユーザー ID (UID)。
- hlq.SCSQAUTH インストール内の hlq.SCSQANL* および IBM MQ データ・セットへのアクセス。
- z/OS UNIX システム・サービスの IBM MQ インストール・ファイルに対する読み取り権限。
- **crtmqweb** スクリプトによって作成された Liberty ユーザー・ディレクトリーに対する読み取り権限と書き込み権限。
- キュー・マネージャーに接続するための権限。 MQCONN クラス内の hlq.BATCH プロファイルへの「*READ*」 アクセス権を mqweb サーバー開始タスク・ユーザー ID に付与します。
- IBM MQ コマンドを発行し、特定のキューにアクセスするための権限。 これらの詳細は、225 ページの『IBM MQ Console - 必要なコマンドセキュリティー・プロファイル』、203 ページの『システム・キュー・セキュリティー』、および 214 ページの『コンテキスト・セキュリティーのためのプロファイル』 で説明されています。
- REST API for MFT を使用するために、 SYSTEM.FTE トピックにサブスクライブする権限。 mqweb サーバーに対して、MXTOPIC クラス内の hlq.SUBSCRIBE.SYSTEM.FTE プロファイルへのアクセス権を、ユーザー ID *ALTER* に付与します。
- SAF レジストリーを構成する場合は、さまざまなセキュリティー・プロファイルにアクセスします。 詳しくは、 496 ページの『IBM MQ Console および REST API の SAF レジストリーの構成』 を参照してください。

接続認証

すべてのバッチ・アプリケーションが有効なユーザー ID とパスワードを提供するようにキュー・マネージャーを構成している場合は、CHKLOCL (REQUIRED) を設定して、MQCONN クラスの hlq.BATCH プロファイルへの 「*UPDATE*」 アクセス権を mqweb サーバー開始タスク・ユーザー ID に与える必要があります。

この権限により、mqweb サーバー開始タスクのユーザー ID に対する接続認証が、CHKLOCL(OPTIONAL) モードで 動作するようになります。

すべてのバッチ・アプリケーションが有効なユーザー ID とパスワードを提供することを要求するようにキュー・マネージャーを構成していない場合は、MQCONN クラス内の hlq.BATCH プロファイルへの mqweb サーバー・タスク 「*READ*」 アクセスを開始するユーザー ID を使用するだけで十分です。

CHCKLOCL について詳しくは、193 ページの『ローカルでバインドされたアプリケーションでの CHCKLOCL の使用』 を参照してください。

IBM MQ または MQ Console の使用に必要な REST API リソースへのアクセス

MQWebUser 役割のユーザーによって MQ Console または REST API で実行される操作は、そのユーザーのセキュリティー・コンテキストの下で実行されます。

このタスクについて

MQ Console および REST API の役割について詳しくは、498 ページの『IBM MQ Console および REST API の役割』 を参照してください。

以下の手順を使用して、MQWebUser 役割のユーザーに、MQ Console や REST API を使用するのに必要なキュー・マネージャー・リソースへのアクセス権限を付与します。

手順

1. mqweb server started task ユーザー ID に、MQWebUser 役割の各ユーザー ID に対する代替ユーザー・アクセス権限を付与します。

   ユーザーが MQ Console や REST API によって管理するすべてのキュー・マネージャーで、その作業を実行してください。

   以下のサンプル RACF コマンドを使用して、 MQWebUser 役割のユーザーに mqweb server started task ユーザー ID 代替ユーザー・アクセス権限を付与できます。

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   ここで、

   hlq
   プロファイル接頭部。キュー・マネージャー名かキュー共有グループ名のいずれかです。

   userId
   MQWebUser 役割のユーザーです。

   mqwebUserId
   mqweb server started task ユーザー ID です。

   注 : 大/小文字混合セキュリティーを使用している場合は、MQADMIN クラスではなく MXADMIN クラスを使用してください。
2. MQWebUser ロール・アクセスの各ユーザーに、MQ Console および REST API の使用に必要なシステム・キューへのアクセス権限を付与します。

   そのために、SYSTEM.ADMIN.COMMAND.QUEUE と SYSTEM.REST.REPLY.QUEUE の両方について、大/小文字混合セキュリティーを使用しているかどうかに応じて、MQQUEUE クラスか MXQUEUE クラスへの UPDATE アクセス権限を各ユーザーに付与してください。

   この権限付与は、ユーザーが REST API によって管理するすべてのキュー・マネージャー (administrative REST API ゲートウェイで管理するリモート・キュー・マネージャーを含む) で実行する必要があります。
3. MQWebUser 役割のユーザーがリモート・キュー・マネージャーを管理できるようにするには、MQQUEUE クラスか MXQUEUE クラスのプロファイルに対する UPDATE アクセス権限をそのユーザーに付与して、リモート・キュー・マネージャーにコマンドを送信する時に使用する伝送キューを保護する必要があります。 また、ゲートウェイ・キュー・マネージャーに対する UPDATE 権限をユーザーに付与することも必要です。

   リモート・キュー・マネージャーで、同じユーザーに、コマンド応答メッセージをゲートウェイ・キュー・マネージャーに送り返すのに使用する伝送キューへの書き込みアクセス権限を付与します。
4. MQWebUser 役割のユーザーに、MQ Console と REST API で サポートされている操作を実行するために必要な他のすべてのリソースに対するアクセス権限を付与します。

   必要なアクセス権限:

   - REST API での操作の実行については、個々の REST API リソース の「 セキュリティー要件 」セクションで説明されています。
   - MQ Console でコマンドを実行するための権限。225 ページの『IBM MQ Console - 必要なコマンドセキュリティー・プロファイル』を参照してください。

V 9.1.0 REST API および IBM MQ Console の TLS を z/OS で設定します。

z/OS では、TLS でのセキュア接続とクライアント証明書認証のための証明書を保管するときに RACF 鍵リングを使用するように、mqweb サーバーを構成できます。

始める前に

この手順を実行するには、`mqwebuser.xml` ファイルへの書き込みアクセス権限を持つユーザー、および SAF 鍵リングを処理する権限を持っている必要があります。

このタスクについて

デフォルトの mqweb サーバー構成では、サーバー証明書とトラステッド証明書に Java 鍵ストアを使用します。 z/OS では、Java 鍵ストアの代わりに RACF 鍵リングを使用するように mqweb サーバーを構成できます。 また、このサーバーは、ユーザーがクライアント証明書を使用して認証できるように構成することもできます。

Liberty での RACF 鍵リングの使用については、 Liberty: 鍵ストア を参照してください。

RACF 鍵リングを使用するように mqweb サーバーを構成し、オプションでクライアント証明書認証を構成するには、次の手順を実行します。

手順

1. サーバー証明書に署名するために使用される認証局 (CA) 証明書を作成します。 例えば、以下の RACF コマンドを入力します。

```
RACDCERT GENCERT
  CERTAUTH
  SUBJECTSDN(CN('mqweb Certification Authority')
    O('IBM')
    OU('MQ'))
  SIZE(2048)
  WITHLABEL('mqwebCertauth')
```

2. 以下のコマンドを入力して、ステップ 1 で作成した CA 証明書で署名されたサーバー証明書を作成します。

```
RACDCERT ID(mqwebUserId) GENCERT
  SUBJECTSDN(CN('hostname')
   O('IBM')
   OU('MQ'))
   SIZE(2048)
   SIGNWITH (CERTAUTH LABEL('mqwebCertauth'))
  WITHLABEL('mqwebServerCert')
```

   ここで、*mqwebUserId* は mqweb サーバー開始タスクのユーザー ID、*hostname* は mqweb サーバーのホスト名です。

3. 以下のコマンドを入力して、CA 証明書とサーバー証明書を SAF 鍵リングに接続します。

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

   ここで、*mqwebUserId* は mqweb サーバー開始タスクのユーザー ID、*keyring* は使用する鍵リングの名前です。

4. 以下のコマンドを入力して、CA 証明書を CER ファイルにエクスポートします。

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth'))
        DSN('hlq.CERT.MQWEBCA')
        FORMAT(CERTDER)
        PASSWORD('password')
```

5. エクスポートされた CA 証明書をバイナリー形式でワークステーションに FTP 転送し、認証局証明書としてブラウザーにインポートします。
6. オプション: クライアント証明書認証を構成する場合は、クライアント証明書を作成してエクスポートします。
   a) クライアント証明書に署名するために使用される認証局 (CA) 証明書を作成します。 例えば、以下の RACF コマンドを入力します。

```
RACDCERT GENCERT
  CERTAUTH
  SUBJECTSDN(CN('mqweb User CA')
   O('IBM')
   OU('MQ'))
   SIZE(2048)
  WITHLABEL('mqwebUserCertauth')
```

   b) 以下のコマンドを入力して、CA 証明書を SAF 鍵リングに接続します。

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

   ここで、*mqwebUserId* は mqweb サーバー開始タスクのユーザー ID、*keyring* は使用する鍵リングの名前です。
   c) CA 証明書で署名されたクライアント証明書を作成します。 例えば、以下のコマンドを入力します。

```
RACDCERT ID(clientUserId) GENCERT
  SUBJECTSDN(CN('clientUserId')
     O('IBM')
     OU('MQ'))
   SIZE(2048)
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth'))
  WITHLABEL('userCertLabel')
```

   ここで *clientUserId* は、ユーザー名です。

   証明書をプリンシパルにマップするために使用される方式は、構成されたユーザー・レジストリーのタイプによって異なります。

   - 基本レジストリーを使用している場合は、証明書の「Common Name」フィールドがレジストリー内のユーザーと突き合わされます。
   - SAF レジストリーを使用していて、証明書が RACF データベースに存在する場合、証明書の作成時に **ID** パラメーターで指定した証明書の所有者が使用されます。
   - LDAP レジストリーを使用している場合は、証明書の完全識別名が LDAP レジストリーと突き合わされます。

   d) 以下のコマンドを入力して、クライアント証明書を PKCS #12 ファイルにエクスポートします。

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) PASSWORD('password')
DSN('hlq.USER.CERT')
```

   e) エクスポートされた証明書をバイナリー形式でワークステーションに FTP 転送します。 IBM MQ Console でクライアント証明書を使用するには、IBM MQ Console にアクセスするために使用する Web ブラウザーにそのクライアント証明書を個人証明書としてインポートします。
7. ファイル *WLP_user_directory*/servers/mqweb/mqwebuser.xml を編集します。ここで、*WLP_user_directory* は、mqweb サーバー定義を作成するために **crtmqweb** スクリプトを実行した際に指定したディレクトリーです。

   次の変更を行って、RACF 鍵リングを使用するように mqweb サーバーを構成します。
   a) 次の行を削除するかまたはコメント化します。

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   b) 次のステートメントを追加します。

```
<keyStore id="defaultKeyStore" filebased="false" location="safkeyring://mqwebUserId/
keyring"
          password="password" readOnly="true" type="JCERACFKS" />
<ssl id="thisSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1.2"
          serverKeyAlias="mqwebServerCert" clientAuthenticationSupported="true" />
<sslDefault sslRef="thisSSLConfig"/>
```

ここで、

- *mqwebUserId* は、mqweb サーバー開始タスク・ユーザー ID です。
- *keyring* は、RACF 鍵リングの名前です。
- *mqwebServerCert* は、mqweb サーバー証明書のラベルです。

注 : keyStore password の値は無視されます。

8. mqweb サーバー開始タスクを停止してから再始動することによって、mqweb サーバーを再始動します。
9. オプション: クライアント証明書を使用して認証を行います。
   - IBM MQ Console でクライアント証明書を使用するには、クライアント証明書をインストールした Web ブラウザーで MQ Console の URL を入力します。
   - REST API でクライアント証明書を使用するには、REST 要求が行われるたびにクライアント証明書を提供します。

   注 :

   a. 証明書のみを使用して IBM MQ Console を認証している場合、ブラウザーが選択するための証明書のリストを表示する場合があります。

   b. 別の証明書を使用する場合、ブラウザーを閉じてから再始動することが必要になることがあります。

   c. RACF データベースに含まれていないクライアント証明書を使用している場合は、RACF の証明書名フィルターを使用して、証明書属性をユーザー ID にマップすることができます。 以下に例を示します。

   ```
   RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
   ```

   これにより、OU=DEPT1 と C=US が含まれているサブジェクト識別名を持つ証明書が、ユーザー ID DEPT3USR にマップされます。

タスクの結果

IBM MQ Console および REST API の TLS インターフェースのセットアップが完了しました。

ULW 鍵と証明書の管理 (UNIX, Linux, and Windows)

runmqckm コマンド (UNIX および Windows)、および runmqakm コマンド (UNIX, Linux, and Windows) を使用して、鍵、証明書、および認証要求を管理します。

runmqckm コマンド

runmqckm コマンドは、UNIX および Windows で使用可能です。

runmqckm コマンドは、5 ページの『セキュリティー IBM MQ』に記載されている iKeyman の機能と同様の機能を提供します。

runmqckm コマンドを使用するには、 **setmqenv** コマンドを実行して、システム環境変数が正しく構成されていることを確認します。

V 9.1.0 **runmqckm** コマンドを実行するには、IBM MQ JRE コンポーネントがインストールされている必要があります。 このコンポーネントがインストールされていない場合、代わりに **runmqackm** コマンドを使用できます。

runmqakm コマンド
runmqakm コマンドは、UNIX、Linux、および Windows で使用可能です。

runmqaqm コマンドを使用するには、 **setmqenv** コマンドを実行して、システム環境変数が正しく構成されていることを確認します。

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には、**runmqckm** コマンドではなく、**runmqakm** コマンドを使用します。 これは、**runmqakm** コマンドが、強い暗号化をサポートするためです。

runmqckm コマンドおよび runmqakm コマンドを使用して、以下を行います。

- IBM MQ が必要とする CMS キー・データベース・ファイルのタイプの作成
- 証明書要求の作成
- 個人証明書のインポート
- CA 証明書のインポート
- 自己署名証明書の管理

関連情報
Keytool

ULW UNIX, Linux, and Windows での runmqckm および runmqakm コマンド

このセクションでは、runmqckm コマンドおよび runmqakm コマンドを、コマンドのオブジェクトに応じて説明します。

2 つのコマンドの主な違いは以下のとおりです。

- ULW **runmqakm**
  - UNIX、Linux、Windows で利用できます。
  - 楕円曲線公開鍵を使用する証明書および証明書要求の作成をサポートします。これに対し、**runmqckm** コマンドはサポートしません。
  - **-strong** パラメーターを使用することによって、鍵リポジトリー・ファイルに関して **runmqckm** コマンドより強い暗号化をサポートします。
  - **runmqckm** コマンドとは異なり、FIPS 140-2 準拠と認定されていて、**-fips** パラメーターを使用することによって、FIPS 準拠で動作するように構成できます。
- Windows UNIX **runmqckm**
  - UNIX と Windows で利用できます。
  - JKS および JCEKS 鍵リポジトリー・ファイル形式を サポートしますが、**runmqakm** コマンドはサポートしません。

重要 : V 9.1.0 **runmqckm** コマンドを実行するには、IBM MQ Java runtime environment (JRE) の機能がインストールされている必要があります。

各コマンドは、少なくとも 1 つのオブジェクト を指定します。 PKCS #11 デバイス操作用のコマンドは、追加のオブジェクトを指定できます。 鍵データベース、証明書、証明書要求オブジェクトのコマンドでは、アクション も指定します。 オブジェクトは、次のいずれかになります。

-keydb
アクションは鍵データベースに適用されます。

-cert
アクションは証明書に適用されます。

-certreq
アクションは証明書要求に適用されます。

-help
ヘルプを表示します。

-バージョン
バージョン情報を表示します。

以下のサブトピックでは、鍵データベース、証明書、証明書要求オブジェクトに対して実行できるアクションについて説明します。 コマンドのオプションの説明については、526 ページの『UNIX, Linux, and Windows での runmqckm および runmqakm オプション』を参照してください。

ULW CMS 鍵データベースのコマンド (UNIX, Linux, and Windows のみ)

runmqckm コマンドおよび **runmqakm** コマンドを使用して、CMS 鍵データベースの鍵および証明書を管理できます。

-keydb -changepw
CMS キー・データベースのパスワードを変更します。

```
-keydb -changepw -db filename -pw password -new_pw new_password


-stash
```

-keydb -create
CMS キー・データベースを作成します。

```
-keydb -create -db filename
-pw password -type cms -expire days -stash
```

-keydb -stashpw
CMS キー・データベースのパスワードをファイルに stash します。

```
-keydb -stashpw -db filename
-pw password
```

-cert -getdefault

注 : デフォルトの証明書は、IBM MQ 8.0 ではサポートされていません。 25 ページの『デジタル証明書ラベルの要件に関する説明』の説明に従って、証明書ラベル構成を使用する必要があります。

デフォルトの個人証明書を入手します。

```
-cert -getdefault -db filename
-pw password
```

-cert -modify
証明書を変更します。

注 : 現在、変更できるフィールドは、「Certificate Trust (証明書トラスト)」フィールドのみです。

```
-cert -modify -db filename
-pw password -label label
-trust enable|disable
```

-cert -setdefault

注 : デフォルトの証明書は、IBM MQ 8.0 以降ではサポートされていません。 25 ページの『デジタル証明書ラベルの要件に関する説明』の説明に従って、証明書ラベル構成を使用する必要があります。

デフォルトの個人証明書を設定します。

```
-cert -setdefault -db filename
-pw password -label label
```

ULW UNIX, Linux, and Windows 上の CMS または PKCS #12 鍵データベース用のコマンド

runmqckm コマンドおよび runmqakm コマンドを使用して、CMS キー・データベースまたは PKCS #12 キー・データベースのキーおよび証明書を管理できます。

注 : IBM 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。

-keydb -changepw
キー・データベースのパスワードを変更します。

```
-keydb -changepw -db filename -pw password -new_pw
new_password -expire days
```

-keydb -convert
キー・データベースをある形式から別の形式に次のように変換します。

```
-keydb -convert -db filename -pw password
-old_format cms | pkcs12 -new_format cms
```

-keydb -create
キー・データベースを作成します。

```
-keydb -create -db filename -pw password -type cms
| pkcs12
```

-keydb -delete
キー・データベースを削除します。

```
-keydb -delete -db filename -pw password
```

-keydb -list
現在サポートされているキー・データベースのタイプをリストします。

```
-keydb -list
```

-cert -add
証明書をファイルからキー・データベースに追加します。

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

-cert -create
自己署名証明書を作成します。

```
-cert -create -db filename -pw password -label label
-dn distinguished_name
-size 1024 | 512 -x509version 3 | 1
```

```
| 2
-expire days -sig_alg 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
|
SHAWithRSA
```

-cert -delete
証明書を削除します。

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

-cert -details
特定の証明書の詳細情報をリストします。

```
-cert -details -db filename -pw password -label label
```

-cert -export
個人証明書とその関連する専用キーをキー・データベースから PKCS #12 ファイルまたは別のキー・データベースにエクスポートします。

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

-cert -extract
証明書をキー・データベースから取り出します。

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

-cert -import
個人証明書をキー・データベースからインポートします。

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

-label オプションは必須で、ソース・キー・データベースからインポートする証明書のラベルを指定します。

-new_label オプションはオプションで、ソース・データベース内のラベルとは異なるラベルを、インポートした証明書にターゲット・キー・データベース内で付けられるようにします。

-cert -list
キー・データベース内のすべての証明書をリストします。

```
-cert -list all | personal | CA
-db filename -pw password
```

-cert -receive
ファイルから証明書を受け取ります。

```
-cert -receive -file filename -db filename -pw password

-format ascii | binary -default_cert yes |
no
```

-cert -sign
証明書に署名します。

```
-cert -sign -db filename -file filename -pw password
-label label -target filename
-format ascii | binary -expire days
-sig_alg 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 |
SHAWithRSA
```

-certreq -create
証明書要求を作成します。

```
-certreq -create -db filename -pw password
-label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg 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 |
SHAWithRSA
```

-certreq -delete
証明書要求を削除します。

```
-certreq -delete -db filename -pw password -label
label
```

-certreq -details
特定の証明書要求の詳細情報をリストします。

```
-certreq -details -db filename -pw password -label
label
```

証明書要求に関する詳細情報をリストし、完全証明書要求を表示します。

```
-certreq -details -showOID -db filename
-pw password -label label
```

-certreq -extract
証明書要求を証明書要求データベースから取り出してファイルに入れます。

```
-certreq -extract -db filename -pw password
-label label -target filename
```

-certreq -list
証明書要求データベース内のすべての証明書要求をリストします。

```
-certreq -list -db filename -pw password
```

-certreq -recreate
証明書要求を再作成します。

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

ULW UNIX, Linux, and Windows で暗号装置を操作するためのコマンド

runmqckm コマンドおよび runmqakm コマンドを使用して、暗号デバイス操作用のキーおよび証明書を管理できます。

注 : IBM 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。

-keydb -changepw
暗号デバイスのパスワードを変更します。

```
-keydb -changepw -crypto module_name -tokenlabel token_label
-pw password -new_pw new_password
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-keydb -list
現在サポートされているキー・データベースのタイプをリストします。

```
-keydb -list
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -add
証明書をファイルから暗号デバイスに追加します。

```
-cert -add -crypto module_name -tokenlabel token_label
-pw password -label label -file filename -format
ascii | binary
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -create
暗号デバイス上で自己署名証明書を作成します。

```
-cert -create -crypto module_name -tokenlabel token_label

-pw password -label label -dn distinguished_name
-size 1024 | 512
-x509version 3 | 1 | 2 -default_cert no
| yes -expire days
-sig_alg 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 |
SHAWithRSA
```

注 : 識別名に複数の OU (組織単位) 属性を含む証明書をインポートすることはできません。

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -delete
暗号デバイス上で証明書を削除します。

```
-cert -delete -crypto module_name -tokenlabel token_label
-pw password -label label
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -details
暗号デバイス上で特定の証明書の詳細情報をリストします。

```
-cert -details -crypto module_name -tokenlabel token_label

-pw password -label label
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

暗号デバイス上で特定の証明書の詳細情報をリストし、完全証明書を表示します。

```
-cert -details -showOID -crypto module_name -tokenlabel
token_label
-pw password -label label
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -extract
証明書をキー・データベースから取り出します。

```
-cert -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename
-format ascii | binary
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -import
2 次キー・データベース・サポートを使用して証明書を暗号デバイスにインポートします。

```
-cert -import -db filename -pw password -label label
-type cms
-crypto module_name -tokenlabel token_label -pw
password
-secondaryDB filename -secondaryDBpw password
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

```
-cert -import -db filename -pw password -label label
-type cms
-crypto module_name -tokenlabel token_label -pw
password
-secondaryDB filename -secondaryDBpw password -fips
```

2 次キー・データベース・サポートを使用して PKCS #12 証明書を暗号デバイスにインポートします。

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw
password
-secondaryDB filename -secondaryDBpw password
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw
```

```
password
-secondaryDB filename -secondaryDBpw password -fips
```

注 : 識別名に複数の OU (組織単位) 属性を含む証明書をインポートすることはできません。

-cert -list
暗号デバイス上ですべての証明書をリストします。

```
-cert -list all | personal | CA
-crypto module_name -tokenlabel token_label -pw
password
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-cert -receive
2 次キー・データベース・サポートを使用して証明書をファイルから暗号デバイスに受け取ります。

```
-cert -receive -file filename -crypto module_name -tokenlabel
token_label
-pw password -default_cert yes | no
-secondaryDB filename -secondaryDBpw password -format
ascii | binary
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

runmqakm コマンドを使用する場合:

-certreq -create
暗号デバイス上で証明書要求を作成します。

```
-certreq -create -crypto module_name -tokenlabel token_label

-pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg 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 |
SHAWithRSA
```

注 : 識別名に複数の OU (組織単位) 属性を含む証明書をインポートすることはできません。

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-certreq -delete
暗号デバイスから証明書要求を削除します。

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-certreq -details
暗号デバイス上で特定の証明書要求の詳細情報をリストします。

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

暗号デバイス上で証明書要求に関する詳細情報をリストし、完全証明書要求を表示します。

```
-certreq -details -showOID -crypto module_name -tokenlabel
token_label
-pw password -label label
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-certreq -extract
暗号デバイス上で証明書要求を証明書要求データベースから取り出してファイルに入れます。

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

-certreq -list
暗号デバイス上で証明書要求データベース内のすべての証明書要求をリストします。

```
-certreq -list -crypto module_name -tokenlabel token_label
```

```
-pw password
```

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、**runmqckm** および **strmqikm** 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あり、これらのプラットフォームの **strmqikm** プログラムおよび **runmqckm** プログラムは 32 ビットです。

ULW UNIX, Linux, and Windows での runmqckm および runmqakm オプション

runmqckm (iKeycmd) および **runmqakm** コマンド・ライン・オプションを使用して、鍵、証明書、および認証要求を管理できます。

ULW runmqakm コマンドは、UNIX, Linux, and Windows で使用できます。

Windows UNIX runmqckm コマンドは、UNIX および Windows で使用可能です。

注 : IBM 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。

各オプションの意味は、コマンドに指定されたオブジェクトおよびアクションによって異なります。

表 *90.* ***runmqckm*** および ***runmqakm*** と共に使用できるオプション

パラメーター	説明
-作成	鍵データベースを作成するためのオプション。
-crypto	PKCS #11 暗号デバイスを管理するためのモジュールの名前。 プロパティー・ファイルにモジュール名を指定した場合、**-crypto** の後の値はオプションです。 PKCS #11 暗号ハードウェアに保管されている証明書または鍵を使用している場合、 **runmqckm** および **strmqikm** は、 IBM MQ インストールで提供される Java 仮想マシン (JVM) を使用して実行され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 JVM プロセスにロードされるため、JVM のビット数に一致する暗号ハードウェアの管理用に PKCS #11 ライブラリーがインストールされている必要があります。また、このライブラリーを **runmqckm** または **strmqikm** に指定する必要があります。
-db	キー・データベースの完全修飾パス名。
-default_cert	デフォルトの証明書として証明書を設定します。 値は 「はい」 または 「いいえ」 にすることができます。 デフォルトは no です。
-dn	X.500 識別名。 値は、二重引用符で囲んだストリングです (例: "CN=John Smith,O=IBM,OU=Test,C=GB")。 必須の属性は O と C です。 共通名 (CN) の指定はオプションです。
-encryption	証明書エクスポート・コマンドで使用される暗号化の強度。 値は 「強」 または 「弱」 にすることができます。 デフォルトは strong です。

表 90. ***runmqckm*** および ***runmqakm*** と共に使用できるオプション (続き)	
パラメーター	**説明**
-expire	証明書またはデータベース・パスワードの日単位の有効期限。 証明書パスワードのデフォルトは 365 日です。 データベース・パスワードにはデフォルトの期限はありません。データベース・パスワードの有効期限は **-expire** パラメーターを使用して明示的に設定します。
-file	証明書または証明書要求のファイル名。
-fips	コマンドが FIPS モードで実行されるように指定します。 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。
-format	証明書の形式。 値は、Base64_encoded ASCII の場合は `ascii` 、Binary DER データの場合は `binary` です。 デフォルトは `ascii` です。
-label	証明書または証明書要求に付加されるラベル。 証明書が、IBM MQ クライアント・アプリケーションまたはキュー・マネージャーを識別するために使用される個人証明書である場合、ラベルは IBM MQ 証明書ラベル (CERTLABL) 設定に対応している必要があります。詳しくは、25 ページの『デジタル証明書ラベルの要件に関する説明』を参照してください。
-new_format	鍵データベースの新しい形式。
-new_label	このオプションは、証明書のインポート・コマンドに使用され、ソース・キー・データベースで持っていたラベルとは異なるラベルで証明書がインポートされるようにします。 証明書が、IBM MQ クライアント・アプリケーションまたはキュー・マネージャーを識別するために使用される個人証明書である場合、ラベルは IBM MQ 証明書ラベル (CERTLABL) 設定に対応している必要があります。詳しくは、25 ページの『デジタル証明書ラベルの要件に関する説明』を参照してください。
-new_pw	新しいデータベース・パスワード。
-old_format	鍵データベースの古い形式。
-pw	キー・データベースまたは PKCS #12 ファイル用のパスワード。
-secondaryDB	PKCS #11 デバイス操作用の 2 次キー・データベースの名前。
-secondaryDBpw	PKCS #11 デバイス操作用の 2 次キー・データベースのパスワード。
-showOID	完全証明書または証明書要求を表示します。

表 90. ***runmqckm*** および ***runmqakm*** と共に使用できるオプション (続き)

パラメーター	説明
-sig_alg	認証要求、自己署名証明書、または証明書の署名の作成中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、新しく作成された証明書または認証要求に関連した署名の作成に使用されます。 **runmqckm** の場合、値は MD2_WITH_RSA、MD2WithRSA、MD5_WITH_RSA、MD5WithRSA、SHA1WithDSA、SHA1WithECDSA、SHA1WithRSA、SHA2/ECDSA、SHA224WithECDSA、SHA256_WITH_RSA、SHA256WithECDSA、SHA256WithRSA、SHA2WithECDSA、SHA3/ECDSA、SHA384_WITH_RSA、SHA384WithECDSA、SHA384WithRSA、SHA3WithECDSA、SHA5/ECDSA、SHA512_WITH_RSA、SHA512WithECDSA、SHA512WithRSA、SHA5WithECDSA、SHA_WITH_DSA、SHA_WITH_RSA、SHAWithDSA、SHAWithRSA のいずれかです。 デフォルト値は SHA1WithRSA です。 **runmqakm** の場合、指定可能な値は 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 です。 デフォルト値は SHA1WithRSA です。
-size	キー・サイズ。 **runmqckm** の場合、値は 512、1024、または 2048 にすることができます。 デフォルト値は 1024 ビットです。 **runmqakm** の場合、値は署名アルゴリズムに応じて次のように異なります。 • RSA 署名アルゴリズム (**-sig_alg** が指定されない場合に使用されるデフォルトのアルゴリズム) の場合、指定可能な値は 512、1024、2048、または 4096 です。 **-fips** パラメーターが使用可能になっている場合、512 ビット の RSA 鍵サイズは許可されません。 デフォルトの RSA 鍵サイズは 1024 ビットです。 • 楕円曲線アルゴリズムの場合、指定可能な値は 256、384、または 512 です。 デフォルトの楕円曲線鍵サイズは、署名アルゴリズムによって異なります。 SHA256 の場合は 256、SHA384 の場合は 384、SHA512 の場合は 512 です。
-stash	鍵データベース・パスワードをファイルに隠しておきます。 タイプ CMS および「PKCS12」 のデータベースにのみ適用されます。 **注 : -stash** は **-keydb** -create コマンドで有効で、パスワードを含む stash ファイルを作成するように **runmqckm**/**runmqakm** に指示します。 コマンド $ runmqakm -help を発行すると、ハイレベル・ヘルプ・パラメーターのみがリストされます。

表 90. ***runmqckm*** および ***runmqakm*** と共に使用できるオプション (続き)	
パラメーター	**説明**
-stashed	鍵データベースまたは PKCS #12 ファイルのパスワードが stash ファイルにあることを示します。 **注 : -stashed** オプションは、 **-keydb** -create コマンド以外の呼び出しで有効です。 このオプションを指定しない場合は、 **-pw** を使用してパスワードを指定する必要があります。 さらに、どの種類のアクションを実行するかをコマンドに指示した場合にのみ、**-stashed** を示す詳細なヘルプが表示されます。
-target	宛先ファイルまたはデータベース。
-target_pw	**-target** がキー・データベースを指定する場合のキー・データベースのパスワード。
-target_type	**-target** オペランドによって指定されるデータベースのタイプ。 許される値については、**-type** パラメーターを参照してください。
-tokenLabel	PKCS #11 暗号デバイスのラベル。
-trust	CA 証明書の信頼状況。 値は enable または disable です。 デフォルトは enable です。
-type	データベースのタイプ。 値には以下のいずれかの値を指定できます。 • CMS キー・データベースの場合は cms • PKCS #12 ファイルの場合は pkcs12
-x509version	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。

-rfc3339	このパラメーターは、runmqakm -cert -details コマンドでの日付を次のような RFC 3339 形式で出力するときに使用します。
	`Not Before : 2015-08-26T08:53:37Z`
	`Not After : 2016-08-26T08:53:37Z`
	-rfc3339 パラメーターは、次のようにコマンドの追加パラメーターの後に置く必要があります。
	`runmqakm -cert -details -db exampleDB -stashed -label certficateLabel -rfc3339`

注 : runmqckm ユーティリティーの IBM Global Security Kit (GSKit) で提供される対称鍵暗号化 **-seckey** パラメーターに関連するプロパティーは無視され、 IBM MQ ではサポートされません。

ULW UNIX, Linux, and Windows での runmqakm エラー・コード

runmqakm によって送出される数値エラー・コードとその意味を示す表。

エラー・コード	エラー・メッセージ
0	成功
1	不明なエラーが発生しました

エラー・コード	エラー・メッセージ
2	ASN.1 エンコード/デコード・エラーが発生しました。
3	ASN.1 エンコーダー/デコーダーの初期化中にエラーが発生しました。
4	インデックスが範囲外であるか、オプション・フィールドが存在しないことが原因で、ASN.1 エンコード/デコード・エラーが発生しました。
5	データベース・エラーが発生しました。
6	データベース・ファイルを開こうとしてエラーが発生しました。ファイルが存在していること、およびそのアクセス権を確認してください。
7	データベース・ファイルを再度開こうとしてエラーが発生しました。
8	データベースの作成に失敗しました。
9	そのデータベースは既に存在します。
10	データベース・ファイルの削除中にエラーが発生しました。
11	データベースを開くことができませんでした。
12	データベース・ファイルの読み取り中にエラーが発生しました。
13	データベース・ファイルにデータを書き込み中にエラーが発生しました。
14	データベース妥当性検査エラーが発生しました。
15	無効なデータベース・バージョンが検出されました。
16	無効なデータベース・パスワードが検出されました。
17	無効なデータベース・ファイル・タイプが検出されました。
18	指定されたデータベースが破損しています。
19	無効なパスワードが指定されたか、鍵データベースに改ざんまたは破損があります。
20	データベースのキー項目の保全性エラーが発生しました。
21	データベースに、重複する証明書が既に存在しています。
22	データベースに、重複するキーが既に存在しています (レコード ID)。
23	鍵データベースに、同じラベルの証明書が既に存在しています。
24	データベースに、重複するキーが既に存在しています (署名)。
25	データベースに、重複するキーが既に存在しています (未署名証明書)。

エラー・コード	エラー・メッセージ
26	データベースに、重複するキーが既に存在しています (発行者および通し番号)。
27	データベースに、重複するキーが既に存在しています (サブジェクト公開鍵情報)。
28	データベースに、重複するキーが既に存在しています (未署名 CRL)。
29	このラベルは、データベースで使用されています。
30	パスワードの暗号化エラーが発生しました。
31	LDAP 関連のエラーが発生しました。 (このプログラムは LDAP をサポートしていません)
32	暗号エラーが発生しました。
33	暗号化/暗号化解除エラーが発生しました。
34	無効な暗号アルゴリズムが検出されました。
35	データ署名中にエラーが発生しました。
36	データの検証中にエラーが発生しました。
37	データのダイジェストを計算中にエラーが発生しました。
38	無効な暗号パラメーターが検出されました。
39	サポートされない暗号アルゴリズムが検出されました。
40	サポートされている係数サイズよりも大きな入力サイズが指定されました。
41	サポートされない係数サイズを検出しました。
42	データベース妥当性検査エラーが発生しました。
43	キー項目妥当性検査が失敗しました。
44	重複する拡張フィールドが存在します。
45	鍵のバージョンが間違っています。
46	必要な拡張フィールドが存在していません。
47	今日が有効期間に含まれていないか、発行者の有効期間内に含まれていません。
48	今日が有効期間に含まれていないか、発行者の有効期間内に含まれていません。
49	秘密鍵使用拡張の妥当性検査中にエラーが発生しました。
50	鍵の発行者が見つかりませんでした。
51	必要な証明書拡張がありません。
52	無効な基本制約拡張が検出されました。
53	鍵署名の妥当性検査が失敗しました。
54	鍵のルート・キーがトラステッドではありません。

エラー・コード	エラー・メッセージ
55	この鍵は取り消されました。
56	権限キー ID 拡張の妥当性検査中にエラーが発生しました。
57	秘密鍵使用拡張の妥当性検査中にエラーが発生しました。
58	サブジェクト代替名拡張の妥当性検査中にエラーが発生しました。
59	発行者代替名拡張の妥当性検査中にエラーが発生しました。
60	鍵使用拡張の妥当性検査中にエラーが発生しました。
61	不明なクリティカル拡張が検出されました。
62	鍵ペア項目の妥当性検査中にエラーが発生しました。
63	CRL の妥当性検査中にエラーが発生しました。
64	mutex エラーが発生しました。
65	無効なパラメーターが見つかりました。
66	ヌル・パラメーターまたはメモリー割り振りエラーが検出されました。
67	数またはサイズが大きすぎるか、小さすぎます。
68	旧パスワードが無効です。
69	新規パスワードが無効です。
70	パスワードの有効期限が切れています。
71	スレッド関連のエラーが発生しました。
72	スレッドの作成中にエラーが発生しました。
73	スレッドが終了を待機中にエラーが発生しました。
74	入出力エラーが発生しました。
75	CMS のロード中にエラーが発生しました。
76	暗号化ハードウェア関連のエラーが発生しました。
77	ライブラリーの初期化ルーチンが正常に呼び出されませんでした。
78	内部データベース・ハンドル・テーブルが壊れています。
79	メモリーの割り振りエラーが発生しました。
80	認識されないオプションが検出されました。
81	時刻情報取得中にエラーが発生しました。
82	mutex 作成エラーが発生しました。
83	メッセージ・カタログを開こうとしてエラーが発生しました。

エラー・コード	エラー・メッセージ
84	エラー・メッセージ・カタログを開こうとしてエラーが発生しました。
85	ヌル・ファイル名を検出しました。
86	ファイルを開こうとしてエラーが発生しました。ファイルが存在していること、およびそのアクセス権を確認してください。
87	ファイルを読み取り用に開こうとしてエラーが発生しました。
88	ファイルを書き込み用に開こうとしてエラーが発生しました。
89	そのようなファイルは存在しません。
90	ファイルのアクセス権の設定が原因で、そのファイルを開けません。
91	ファイルにデータを書き込もうとしてエラーが発生しました。
92	ファイルの削除中にエラーが発生しました。
93	無効な Base64 エンコード・データが検出されました。
94	無効な Base64 メッセージ・タイプが検出されました。
95	Base64 エンコード規則を使用したデータのエンコード中にエラーが発生しました。
96	Base64 エンコード・データのデコード中にエラーが発生しました。
97	識別名タグを取得中にエラーが発生しました。
98	必要な共通名フィールドが空になっています。
99	必要な国または地域名フィールドが空になっています。
100	無効なデータベース・ハンドルが検出されました。
101	鍵データベースが存在しません。
102	要求鍵ペア・データベースが存在しません。
103	パスワード・ファイルが存在しません。
104	新規パスワードが旧パスワードと同じです。
105	鍵データベースに鍵が見つかりませんでした。
106	要求鍵が見つかりませんでした。
107	トラステッド CA が見つかりませんでした。
108	証明書の要求鍵が見つかりませんでした。
109	鍵データベースに秘密鍵がありません。
110	鍵データベースにデフォルト鍵がありません。
111	鍵レコードに秘密鍵がありません。

エラー・コード	エラー・メッセージ
112	鍵レコードに証明書がありません。
113	CRL 項目がありません。
114	無効な鍵データベース・ファイル名が検出されました。
115	認識されない秘密鍵タイプが検出されました。
116	無効な識別名の入力が検出されました。
117	指定された鍵ラベルを持つ鍵項目が見つかりませんでした。
118	鍵ラベル・リストが破損しています。
119	入力データが有効な PKCS12 データではありません。
120	パスワードが無効です。あるいは、PKCS12 データが破損しているか、より新しいバージョンの PKCS12 で作成されています。
121	認識されない鍵エクスポート・タイプが検出されました。
122	サポートされていないパスワード・ベースの暗号化アルゴリズムが検出されました。
123	鍵リング・ファイルを CMS 鍵データベースに変換中にエラーが発生しました。
124	CMS 鍵データベースを鍵リング・ファイルに変換中にエラーが発生しました。
125	認証要求のための証明書を作成中にエラーが発生しました。
126	完全な発行者チェーンを作成できません。
127	無効な WEBDB データが検出されました。
128	鍵リング・ファイルに書き込むデータがありません。
129	入力した日数が、許可された有効期間を超えています。
130	パスワードが短すぎます。少なくとも {0} 文字必要です。
131	パスワードには、少なくとも 1 つの数字を含める必要があります。
132	パスワードのすべての文字が、英字または数字になっています。
133	認識されない、またはサポートされない署名アルゴリズムが指定されました。
134	無効なデータベース・タイプが検出されました。
135	指定された 2 次鍵データベースは、別の PKCS#11 デバイスで使用中です。
136	2 次鍵データベースが指定されていません。

エラー・コード	エラー・メッセージ
137	PKCS#11 デバイス上にラベルが存在しません。
138	この PKCS#11 デバイスにアクセスするにはパスワードが必要です。
139	この PKCS#11 デバイスにアクセスするのに、パスワードは不要です。
140	暗号ライブラリーをロードできません。
141	この操作に対して PKCS#11 はサポートされていません。
142	PKCS#11 デバイスでの操作が失敗しました。
143	LDAP ユーザーは有効なユーザーではありません。(このプログラムは LDAP をサポートしていません)
144	LDAP ユーザーは有効なユーザーではありません。(このプログラムは LDAP をサポートしていません)
145	LDAP 照会が失敗しました。 (このプログラムは LDAP をサポートしていません)
146	無効な証明書チェーンが検出されました。
147	ルート証明書がトラステッドではありません。
148	取り消された証明書が検出されました。
149	暗号オブジェクト関数が失敗しました。
150	使用可能な証明書失効リスト・データ・ソースがありません。
151	使用可能な暗号トークンがありません。
152	FIPS モードは使用できません。
153	FIPS モードの設定値との競合があります。
154	入力されたパスワードは、必要な最小強度を満たしていません。
200	プログラムの初期化中に障害が発生しました。
201	runmqakm プログラムに渡された引数のトークン化に失敗しました。
202	コマンドに指定されたオブジェクトが、認識済みのオブジェクトではありません。
203	渡されたアクションが、既知の -keydb アクションではありません。
204	渡されたアクションが、既知の -cert アクションではありません。
205	渡されたアクションが、既知の -certreq アクションではありません。
206	要求されたコマンドに欠落しているタグがあります。
207	-version タグによって渡された値が、認識済みの値ではありません。

エラー・コード	エラー・メッセージ
208	-size タグによって渡された値が、認識済みの値ではありません。
209	-dn タグによって渡された値が、正しい形式ではありません。
210	-format タグによって渡された値が、認識済みの値ではありません。
211	ファイルのオープンに関連するエラーが発生しました。
212	PKCS12 は、この段階ではサポートされていません。
213	パスワードを変更しようとしている暗号トークンは、パスワードで保護されていません。
214	PKCS12 は、この段階ではサポートされていません。
215	入力されたパスワードは、必要な最小強度を満たしていません。
216	FIPS モードは使用できません。
217	有効期限日付として入力した日数が、許容範囲を超えています。
218	パスワードの強度が、最小要件を満たしていません。
219	要求された鍵データベースにデフォルト証明書が見つかりませんでした。
220	無効なトラステッド状況が検出されました。
221	サポートされない署名アルゴリズムを検出しました。 この段階では、MD5 および SHA1 のみがサポートされています。
222	その特定の操作に対して、PCKS11 はサポートされていません。
223	渡されたアクションが、既知の -random アクションではありません。
224	ゼロより小さい長さは許可されていません。
225	-strong タグを使用する場合のパスワードの最小長は、14 文字です。
226	-strong タグを使用する場合のパスワードの最大長は、300 文字です。
227	FIPS モードでは、MD5 アルゴリズムはサポートされません。
228	-cert -list コマンドに対する site タグはサポートされません。 この属性は、後方互換性および今後の機能拡張のために追加されています。
229	-ca タグに関連付けられている値が認識されません。 この値は、「true」または「false」のいずれかである必要があります。
230	-type タグによって渡された値が無効です。

エラー・コード	エラー・メッセージ
231	-expire タグによって渡された値が、許容範囲を下回っています。
232	使用または要求された暗号化アルゴリズムはサポートされていません。
233	ターゲットが既に存在しています。

データベース認証の保護の詳細

データベース・マネージャーに接続するためにユーザー名とパスワードの認証を使用している場合は、それらを MQ XA 資格情報ストアに保管して、パスワードがプレーン・テキストで qm.ini ファイルに保管されないようにすることができます。

リソース・マネージャー用に XAOpenString を更新する

資格情報ストアを使用するには、qm.ini ファイル内の XAOpenString を変更する必要があります。 このストリングは、データベース・マネージャーに接続するために使用されます。 置き換え可能なフィールドを指定して、XAOpenString ストリング内のユーザー名とパスワードが置換される場所を特定します。

- +USER+ フィールドは、XACredentials ストアに保管されているユーザー名の値に置き換えられます。
- +PASSWORD+ フィールドは、XACredentials ストアに保管されているパスワードの値に置き換えられます。

以下の例は、資格情報ファイルを使用してデータベースに接続するように XAOpenString を変更する方法を示しています。

Db2 データベースへの接続

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

Oracle データベースへの接続

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

MQ XA 資格情報ストアに対するデータベースの資格情報の処理

置き換え可能な資格情報ストリングを使用して qm.ini ファイルを更新した後、 **setmqxacred** コマンドを使用して、ユーザー名とパスワードを MQ 資格情報ストアに追加する必要があります。 **setmqxacred** を使用して既存の資格情報を変更したり、資格情報を削除したり、資格情報をリストしたりすることもできます。 以下の例は、典型的なユース・ケースを示しています。

資格情報の追加

以下のコマンドは、リソース mqdb2 に関するキュー・マネージャー QM1 のユーザー名とパスワードを安全に保存します。

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

資格情報の更新

データベースへの接続に使用するユーザー名とパスワードを更新するには、新しいユーザー名とパスワードを使用して **setmqxacred** コマンドを再発行します。

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

変更を有効にするには、キュー・マネージャーを再始動する必要があります。

資格情報の削除

以下のコマンドは、資格情報を削除します。

```
setmqxacred -m QM1 -x mydb2 -d
```

資格情報のリスト

以下のコマンドは、資格情報をリストします。

```
setmqxacred -m QM1 -l
```

関連資料

setmqxacred

Managed File Transfer の保護

インストール直後の変更のない状態では、Managed File Transfer のセキュリティー・レベルは、保護された環境におけるテストまたは評価のためには適している可能性があります。 ただし、実稼働環境では、ファイル転送操作を開始できるユーザー、転送されているファイルの読み取りおよび書き込みができるユーザー、およびファイルの保全性を 保護する方法の適切な管理について考慮する必要があります。

関連タスク

MFT 固有リソースのグループ権限の制限

MFT 固有リソースの権限の管理

600 ページの『Advanced Message Security と Managed File Transfer の使用』
このシナリオでは、 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。

関連資料

ファイル・システムにアクセスするための MFT 権限

commandPath MFT プロパティー

MFT エージェント・ログ・メッセージおよび状況メッセージをパブリッシュする権限

MFT と IBM MQ の接続認証

接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。 関連付けられたキュー・マネージャーのセキュリティーが使用可能に設定されており、資格情報の詳細 (ユーザー ID とパスワード) が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。 接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

資格情報の詳細を提供する方法

多くの Managed File Transfer コマンドで、資格情報の詳細を提供するための以下の方法がサポートされています。

コマンド行引数で詳細を提供する。

資格情報の詳細は、**-mquserid** パラメーターおよび **-mqpassword** をパラメーター使用して指定できます。 **-mqpassword** が指定されないと、ユーザーはパスワードの入力を求められます。ここで入力された内容は表示されません。

資格情報ファイルから提供される詳細 MQMFTCredentials.xml。
資格情報の詳細は、平文または難読化されたテキストとして、MQMFTCredentials.xml ファイル内に事前定義できます。

IBM MQ for Multiplatforms で MQMFTCredentials.xml ファイルをセットアップする方法については、539 ページの『マルチプラットフォームでの MQMFTCredentials.xml の構成』を参照してください。

IBM MQ for z/OS で MQMFTCredentials.xml ファイルをセットアップする方法については、541 ページの『z/OS での MQMFTCredentials.xml の構成』を参照してください。

優先順位

資格情報の詳細は、次の優先順位で決まります。

1. コマンド行引数。
2. 関連するキュー・マネージャーおよびコマンドを実行するユーザーによる MQMFTCredentials.xml 索引。
3. 関連するキュー・マネージャーによる MQMFTCredentials.xml 索引。
4. 以前のリリースの IBM MQ または IBM WebSphere MQ との互換性を許可する資格情報の詳細が提供されないデフォルトの後方互換性モード

注 :

- **fteStartAgent** コマンドおよび **fteStartLogger** コマンドは、コマンド行引数 **-mquserid** と **-mqpassword** をサポートしておらず、資格情報の詳細は MQMFTCredentials.xml ファイルで指定する方法のみが可能です。
- z/OS

  z/OS では、ユーザーのパスワードに小文字が含まれている場合でも、パスワードを大文字にする必要があります。 例えば、ユーザーのパスワードが「password」であれば、「PASSWORD」と入力する必要があります。

関連資料
MFT コマンドとその接続先のキュー・マネージャー
MFT の資格情報ファイルのフォーマット

マルチプラットフォームでの MQMFTCredentials.xml の構成

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。 同様に、 MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要がある場合があります。 この資格情報は、 MFT 資格情報ファイルに保管できます。

このタスクについて

MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。 MQMFTCredentials.xml のフォーマットについては、MFT 資格情報ファイルのフォーマットを参照してください。

資格情報ファイルの例は、 MQ_INSTALLATION_PATH/mqft/samples/credentials ディレクトリーにあります。

MFT 資格情報ファイルは、調整キュー・マネージャー用に 1 つ、コマンド・キュー・マネージャー用に 1 つ、各エージェントに 1 つ、各ロガーに 1 つ使用できます。 あるいは、トポロジー内のすべてのもので使用される 1 つのファイルを使用することもできます。

MFT 資格情報ファイルのデフォルトの場所は以下のとおりです。

Linux UNIX **UNIX and Linux**
$HOME

Windows

%USERPROFILE%または%HOMEDRIVE%%HOMEPATH%

資格情報ファイルが別の場所に保管されている場合は、以下のプロパティーを使用して、コマンドが検索する場所を指定できます。

表 *91.* : 各種コマンドの *MQMFTCredentials.xml* ファイルの場所を定義するプロパティー。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
調整キュー・マネージャーに接続するコマンド	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
コマンド・キュー・マネージャーに接続するコマンド	connection.properties	connectionQMgrAuthenticationCredentialsFile
エージェント・プロセスに接続するコマンド	agent.properties	agentQMgrAuthenticationCredentialsFile
ロガー・プロセスに接続するコマンド	logger.properties	loggerQMgrAuthenticationCredentialsFile

表 *92.* : エージェントおよびロガー・プロセスの *MQMFTCredentials.xml* ファイルの場所を定義するプロパティー。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
MFT エージェント	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

どのコマンドおよびプロセスがどのキュー・マネージャーに接続するかについて詳しくは、どの MFT コマンドおよびプロセスがどのキュー・マネージャーに接続するかを参照してください。

資格情報ファイルにはユーザー ID とパスワードの情報が含まれているため、このファイルへの無許可アクセスを防止するには特別な権限が必要です。

UNIX and Linux

```
chown <agent owner userid>
chmod 600
```

Windows

継承が有効になっていないことを確認してから、資格情報ファイルを使用するエージェントまたはロガーを実行しているユーザー ID を除き、すべてのユーザー ID を削除してください。

IBM MQ Explorer Managed File Transfer プラグインで MFT 調整キュー・マネージャーに接続するために使用される資格情報の詳細は、構成のタイプによって異なります。

グローバル (ローカル・ディスク上の構成)

グローバル構成は、調整プロパティーおよびコマンド・プロパティーで指定された資格情報ファイルを使用します。

ローカル (IBM MQ Explorer 内で定義する)

ローカル構成は、IBM MQ Explorer 内で関連付けられたキュー・マネージャーの接続詳細のプロパティーを使用します。

関連タスク

542 ページの『MFT の接続認証の有効化』

調整キュー・マネージャーまたはコマンド・キュー・マネージャーと接続する IBM MQ Explorer MFT プラグインの接続認証、および調整キュー・マネージャーまたはコマンド・キュー・マネージャーと接続する Managed File Transfer エージェントの接続認証は、互換モードまたは MQCSP 認証モードで実行できます。

関連資料
MFT の資格情報ファイルのフォーマット
fteObfuscate: 機密データの暗号化

z/OS での MQMFTCredentials.xml の構成

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、ユーザー ID とパスワードの資格情報を提供するために、すべての MFT エージェント、およびキュー・マネージャーに接続するコマンドが必要になります。

同様に、 MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要がある場合があります。

この資格情報は、 MFT 資格情報ファイルに保管できます。 資格情報ファイルはオプションですが、環境をカスタマイズする前に必要な 1 つ以上のファイルを定義する方が簡単です。

これに加えて、資格情報ファイルがある場合は、受け取る警告メッセージの数が少なくなります。 警告メッセージは、MFT でキュー・マネージャーのセキュリティーがオフであると見なされたために認証の詳細が指定されないということを通知します。

資格情報ファイルの例は、 MQ_INSTALLATION_PATH/mqft/samples/credentials ディレクトリーにあります。

MQMFTCredentials.xml ファイルの例を以下に示します。

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

ユーザー ID ADMIN を持つジョブは、キュー・マネージャー MQPH に接続する必要があるときに、ユーザー ID *JOHNDOEH* を渡し、パスワード *cXXXX* を使用します。

ジョブが他のユーザー ID で実行され、MQPH に接続する場合、そのジョブはユーザー ID *NONEH* とパスワード *yXXXX* を渡します。

MQMFTCredentials.xml ファイルのデフォルトの場所は、z/OS UNIX System Services (USS) 上のユーザーのホーム・ディレクトリーです。 USS 上の別の場所、または区分データ・セット内のメンバーのいずれかにファイルを保管することもできます。

資格情報ファイルが別の場所に保管されている場合は、以下のプロパティーを使用して、コマンドが検索する場所を指定できます。

表 *93. :* 各種コマンドの *MQMFTCredentials.xml* ファイルの場所を定義するプロパティー。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
調整キュー・マネージャーに接続するコマンド	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
コマンド・キュー・マネージャーに接続するコマンド	connection.properties	connectionQMgrAuthenticationCredentialsFile
エージェント・プロセスに接続するコマンド	agent.properties	agentQMgrAuthenticationCredentialsFile
ロガー・プロセスに接続するコマンド	logger.properties	loggerQMgrAuthenticationCredentialsFile

表 94. : エージェントおよびロガー・プロセスの *MQMFTCredentials.xml* ファイルの場所を定義するプロパティー。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
MFT エージェント	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

どのコマンドおよびプロセスがどのキュー・マネージャーに接続するかについて詳しくは、どの MFT コマンドおよびプロセスがどのキュー・マネージャーに接続するかを参照してください。

区分データ・セット内に資格情報ファイルを作成するには、以下のステップを実行します。

- VB の形式と論理レコード長 (Lrecl) 200 で PDSE を作成します。
- データ・セット内にメンバーを作成し、データ・セットとメンバーのメモを取り、以下のコードをメンバーに追加します。

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!--credentials information goes here-->
</tns:mqmftCredentials>
```

資格情報ファイルは、RACF などのセキュリティー製品を使用して保護できますが、Managed File Transfer コマンドを実行するユーザー ID、およびエージェントとロガーのプロセスを管理するユーザー ID には、このファイルに対する読み取り権限が必要です。

このファイルの情報は、メンバー BFGCROBS の JCL を使って覆い隠すことができます。 これはファイルを取り、IBM MQ ユーザー ID とパスワードを暗号化します。 例えば、メンバー BFGCROBS は次の行を取ります。

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

そして、次のものを作成します。

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

このユーザー ID から IBM MQ ユーザー ID へのマッピングを保持する場合、ファイルにコメントを追加できます。 例えば

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

これらのコメントは、覆い隠すプロセスでは変更されません。

注: 内容は覆い隠されますが、高い強度で暗号化されるわけではありません。 そのファイルにアクセスできるユーザー ID を制限する必要があります。

関連タスク

539 ページの『マルチプラットフォームでの MQMFTCredentials.xml の構成』
Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。 同様に、 MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要がある場合があります。 この資格情報は、 MFT 資格情報ファイルに保管できます。

MFT の接続認証の有効化

調整キュー・マネージャーまたはコマンド・キュー・マネージャーと接続する IBM MQ Explorer MFT プラグインの接続認証、および調整キュー・マネージャーまたはコマンド・キュー・マネージャーと接続する Managed File Transfer エージェントの接続認証は、互換モードまたは MQCSP 認証モードで実行できます。

このタスクについて

IBM MQ 9.1.1 より前では、互換モードが接続認証のデフォルト設定です。 ただし、デフォルトの互換モードを無効にして、MQCSP 認証モードを有効にできます。

V 9.1.1 IBM MQ 9.1.1 以降では、MQCSP 認証モードがデフォルトです。

CLIENT トランスポートを使用してキュー・マネージャーに接続する IBM MQ Explorer Managed File Transfer プラグインまたは Managed File Transfer エージェントの接続認証では、12 文字より長いパスワードは MQCSP 認証モードでのみサポートされます。 互換モードを使用して権限を付与するときに 12 文字より長いパスワードを指定すると、エラーが発生し、エージェントはキュー・マネージャーで認証されません。 診断メッセージ: BFGAG0001 - BFGAG9999 の BFGAG0187E メッセージを参照してください。

手順

- IBM MQ Explorer の調整キュー・マネージャーまたはコマンド・キュー・マネージャーの接続認証モードを選択するには、以下の手順を実行します。

  a) 接続先キュー・マネージャーを選択します。

  b) 右クリックして、ポップアップ・メニューから「**接続詳細**」>「**プロパティー**」を選択します。

  c)「**ユーザー ID**」タブをクリックします。

  d) 使用する接続認証モードのチェック・ボックスが選択されていることを確認します。

  - V 9.1.0 IBM MQ 9.1.0 以降、デフォルトで「**ユーザー ID の互換モード**」チェック・ボックスが選択解除されています。 これは、「**ユーザー ID を有効にする**」チェック・ボックスが選択されている場合は、IBM MQ Explorer はキュー・マネージャーに接続するときに MQCSP 認証を使用するということです。 IBM MQ Explorer が、MQCSP 認証ではなく互換モードを使用してキュー・マネージャーに接続する必要がある場合は、「**ユーザー ID を有効にする**」と「**ユーザー ID の互換モード**」のチェック・ボックスが両方とも選択されていることを確認してください。
  - IBM MQ 9.1.0 より前では、デフォルトで「**ユーザー ID の互換モード**」チェック・ボックスが選択されています。 この場合、「**ユーザー ID を有効にする**」チェック・ボックスが選択されているなら、IBM MQ Explorer はキュー・マネージャーに接続するときに互換モードを使用します。 IBM MQ Explorer が MQCSP 認証を使用してキュー・マネージャーに接続する必要がある場合は、「**ユーザー ID を有効にする**」チェック・ボックスが選択され、「**ユーザー ID の互換モード**」チェック・ボックスが選択解除されていることを確認してください。

- `MQMFTCredentials.xml` ファイルを使用して Managed File Transfer エージェントの MQCSP 認証モードを有効または無効にするには、パラメーター **useMQCSPAuthentication** を関連ユーザーの `MQMFTCredentials.xml` ファイルに追加します。

  useMQCSPAuthentication パラメーターの値は次のとおりです。

  true
  MQCSP 認証モードを使用してキュー・マネージャーでユーザーを認証します。

  V 9.1.1 IBM MQ 9.1.1 以降では、`true` がデフォルト値です。 **useMQCSPAuthentication** パラメーターが指定されていない場合は、デフォルトでそこに `true` が設定され、キュー・マネージャーでのユーザーの認証に MQCSP 認証モードが使用されます。

  false
  互換モードを使用してキュー・マネージャーでユーザーを認証します。

  IBM MQ 9.1.1 より前では、**useMQCSPAuthentication** パラメーターが指定されていない場合は、デフォルトでそこに `false` が設定され、キュー・マネージャーでのユーザーの認証に互換モードが使用されます。

  次の例は、`MQMFTCredentials.xml` ファイルで **useMQCSPAuthentication** パラメーターを設定する方法を示しています。

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

関連概念

28 ページの『MQCSP パスワード保護』
IBM MQ 8.0 以降、パスワードは、MQCSP 構造に組み込んで、IBM MQ 機能を使用して保護するか、または TLS 暗号化を使用するかのどちらかにより送信することができます。

関連資料

538 ページの『MFT と IBM MQ の接続認証』
接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。 関連付けられたキュー・マネージャーのセキュリティーが使用可能に設定されており、資格情報の詳細 (ユーザー ID とパスワード) が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。 接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

MFT の資格情報ファイルのフォーマット

MFT のサンドボックス

ファイル・システムの中で、エージェントが転送処理時にアクセスできる領域を制限できます。 エージェントがアクセスできる制限領域のことをサンドボックスといいます。 制限の適用対象は、エージェントにすることも、転送を要求するユーザーにすることも可能です。

エージェントがプロトコル・ブリッジ・エージェントまたは Connect:Direct® ブリッジ・エージェントである場合は、サンドボックスはサポートされません。 IBM MQ キューとの間で転送する必要のあるエージェントに、エージェント sandboxing を使用することはできません。

関連資料

544 ページの『MFT エージェント・サンドボックスの処理』
追加のセキュリティー・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

546 ページの『MFT ユーザー・サンドボックスの処理』
ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、 転送を要求する MQMD ユーザー名に基づいて制限することができます。

MFT エージェント・サンドボックスの処理

追加のセキュリティー・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

エージェント・サンドボックス機能は、IBM MQ キューとの間で転送を行うエージェントに対して使用することはできません。 サンドボックス機能によって IBM MQ キューへのアクセスの制限を実装するには、代わりにユーザー・サンドボックス機能を使用します。これはすべてのサンドボックス機能要件で推奨されるソリューションです。 ユーザー・サンドボックス機能について詳しくは、546 ページの『MFT ユーザー・サンドボックスの処理』を参照してください。

エージェントのサンドボックス化を使用可能にするには、制限するエージェントの `agent.properties` ファイルに以下のプロパティーを追加します。

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

ここで、

- *restricted_directory_name* は、許可あるいは拒否されるディレクトリー・パスです。
- ！はオプションであり、 *restricted_directory_name* の以下の値が拒否される (除外される) ことを指定します。 ！が指定されていない場合、 *restricted_directory_name* は許可された (組み込まれた) パスです。
- *separator* は、プラットフォーム固有の分離文字です。

例えば、AGENT1 が /tmp ディレクトリーに対してのみアクセスを制限するが、サブディレクトリー private にアクセスすることを許可しない場合は、AGENT1: sandboxRoot=/tmp:!/tmp/private に属する agent.properties ファイルのプロパティーを以下のように設定します。

sandboxRoot プロパティーは、『拡張エージェント・プロパティー』で説明されています。

エージェント・サンドボックス機能、およびユーザー・サンドボックス機能は、いずれもプロトコル・ブリッジ・エージェント、または Connect:Direct ブリッジ・エージェントではサポートされていません。

UNIX、Linux、および Windows プラットフォームでのサンドボックスの機能

ULW UNIX、Linux、および Windows プラットフォームでは、サンドボックス機能により Managed File Transfer Agent の読み取り元および書き込み先のディレクトリーを制限します。 サンドボックスがアクティブな場合、Managed File Transfer Agent は、許可すると指定されたディレクトリーと、その指定されたディレクトリーに含まれるサブディレクトリー (ただし、そのサブディレクトリーが sandboxRoot で拒否すると指定されていない場合) への読み取りと書き込みができます。 Managed File Transfer のサンドボックスは、オペレーティング・システムのセキュリティーより優先順位が上ではありません。 Managed File Transfer Agent を開始したユーザーには、ディレクトリーからの読み取りとディレクトリーへの書き込みができるように、そのディレクトリーに対するオペレーティング・システムの適切なレベルのアクセス権限が必要です。 リンクしているディレクトリーが指定された sandboxRoot ディレクトリー (およびサブディレクトリー) 以外にある場合、ディレクトリーへのシンボリック・リンクをたどることはできません。

z/OS でのサンドボックスの機能

z/OS z/OS の場合、サンドボックスは Managed File Transfer Agent の読み取り元および書き込み先であるデータ・セット名修飾子を制限します。 Managed File Transfer Agent を開始したユーザーには、関係するどのデータ・セットに対しても正しいオペレーティング・システムの権限がなければなりません。 sandboxRoot データ・セット名修飾子の値を二重引用符で囲むと、その値は通常の z/OS 規則に従い、完全修飾として処理されます。 二重引用符を省略すると、sandboxRoot の前に現在のユーザー ID が接頭部として付けられます。 例えば、sandboxRoot プロパティーを sandboxRoot=//test に設定すると、エージェントは、次のデータ・セットに (標準 z/OS 表記で) アクセスすることができます。 //*username*.test.** ランタイムでは、完全に解決したデータ・セット名の初期レベルが sandboxRoot と一致しない場合、転送要求は拒否されます。

IBM i システムでのサンドボックスの機能

IBM i IBM i システムの統合ファイル・システムのファイルの場合は、サンドボックスは、Managed File Transfer Agent の読み取り元および書き込み先のディレクトリーを制限します。 サンドボックスがアクティブな場合、Managed File Transfer Agent は、許可すると指定されたディレクトリーと、その指定されたディレクトリーに含まれるサブディレクトリー (ただし、そのサブディレクトリーが sandboxRoot で拒否すると指定されていない場合) への読み取りと書き込みができます。 Managed File Transfer のサンドボックスは、オペレーティング・システムのセキュリティーより優先順位が上ではありません。 Managed File Transfer Agent を開始したユーザーには、ディレクトリーからの読み取りとディレクトリーへの書き込みができるように、そのディレクトリーに対するオペレーティング・システムの適切なレベルのアクセス権限が必要です。 リンクしているディレクトリーが指定された sandboxRoot ディレクトリー (およびサブディレクトリー) 以外にある場合、ディレクトリーへのシンボリック・リンクをたどることはできません。

関連資料

549 ページの『ワイルドカード転送の追加検査』
エージェントがファイルの転送先/転送元として使用できる場所を制限するために、そのエージェントにユーザー・サンドボックスまたはエージェント・サンドボックスが構成されている場合、そのエージェントのワイルドカード転送で追加の検査が行われるように指定できます。

544 ページの『MFT エージェント・サンドボックスの処理』
追加のセキュリティー・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

MFT agent.properties ファイル

MFT ユーザー・サンドボックスの処理

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、 転送を要求する MQMD ユーザー名に基づいて制限することができます。

ユーザー・サンドボックスは、エージェントがプロトコル・ブリッジ・エージェント または Connect:Direct ブリッジ・エージェントである場合は サポートされません。

ユーザー・サンドボックス化を使用可能にするには、制限するエージェントの agent.properties ファイルに以下のプロパティーを追加します。

```
userSandboxes=true
```

このプロパティーが存在し、true に設定されている場合、エージェントは *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name*/UserSandboxes.xml ファイル内の情報を使用して、転送を要求するユーザーがアクセスできるファイル・システムの部分を判別します。

UserSandboxes.xml XML は、 <sandbox> エレメントを 0 個以上含んでいる 1 つの <agent> エレメントで構成されています。 これらのエレメントは、どの規則がどのユーザーに適用されるかを記述します。 <sandbox> エレメントの user 属性は、要求の MQMD ユーザーと突き合わせるために使用されるパターンです。

ファイル UserSandboxes.xml は、エージェントによって定期的に再ロードされ、ファイルへの有効な変更は、エージェントの動作に影響します。 デフォルトの再ロード間隔は 30 秒です。 この間隔は、 agent.properties ファイルにエージェント・プロパティー xmlConfigReloadInterval を指定することによって変更できます。

userPattern="regex" 属性または値を指定する場合、user 属性は Java 正規表現として解釈されます。 詳しくは、MFT が使用する正規表現を参照してください。

userPattern="regex" 属性も値も指定しない場合、user 属性は次のワイルドカード文字を持つパターンとして解釈されます。

- アスタリスク。0 個以上の文字を表します。
- 疑問符 (?)。ちょうど 1 文字を表します。

マッチングは、ファイル内で <sandbox> エレメントがリストされている順序で実行されます。 最初のマッチングのみが使用され、ファイル内にあるかもしれないそれ以降の他のマッチングはすべて無視されます。 ファイルで指定された <sandbox> エレメントが、転送要求メッセージに関連付けられた MQMD ユーザーとマッチングしない場合、その転送はファイル・システムにアクセスできません。 MQMD ユーザー名と user 属性の間にマッチングが検出された場合、そのマッチング項目を基にして、転送に適用される規則セットが <sandbox> エレメント内で識別されます。 この規則セットを使用して、 転送の一環として読み取りまたは書き込みが可能であるファイルやデータ・セットが判別されます。

規則セットごとに、読み取り可能なファイルを識別する <read> エレメント および書き込み可能なファイルを識別する <write> エレメントを指定できます。 規則セットから <read> または <write> エレメントを省略した場合、その規則セットに関連付けられたユーザーは、それぞれ読み取りまたは書き込みの実行を許可されないと想定されます。

注: UserSandboxes.xml ファイルの中で、<read> エレメントは <write> エレメントよりも前に、<include> エレメントは <exclude> エレメントよりも前に配置する必要があります。

<read> または <write> のそれぞれのエレメントには、ファイルがサンドボックス内にあるかどうかと、転送可能であるかどうかを決定するために使用されるパターンが 1 つ以上含まれています。 これらのパターンは、<include> および <exclude> エレメントを使用して指定します。 <include> または <exclude> エレメントの name 属性は、 突き合わせ対象となるパターンを指定します。 オプションの type 属性は、名前値がファイルまたはキュー・パターンであるかを指定します。 type 属性が指定されない場合、エージェントはパターンをファイルまたはディレクトリー・パス・パターンとして扱います。 以下に例を示します。

```
<tns:read>
```

```
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

エージェントは <include> および <exclude> name パターンを 使用して、ファイル、データセット、またはキューの読み取りまたは書き込みが可能であるかどうかを判別します。 操作は、正規のファイル・パス、データ・セット、またはキュー名が、 包含パターンの少なくとも 1 つにマッチングし、除外パターンに 1 つもマッチングしない 場合にのみ許可されます。 <include> および <exclude> エレメントの name 属性を 使用して指定するパターンには、エージェントを実行しているプラットフォームに適したパス分離文字および規則が使用されます。 相対ファイル・パスを指定する場合、パスはエージェントの transferRoot プロパティーを基準にして相対的に解決されます。

キューの制限を指定する場合、QUEUE@QUEUEMANAGER の構文がサポートされ、以下のルールが使用されます。

- アットマーク文字 (@) が項目から欠落している場合、パターンはいずれかのキュー・マネージャーでアクセスされるキュー名として扱われます。 例えば、パターンが name である場合、 name@** と同じように扱われます。
- アットマーク文字 (@) が項目の最初の文字である場合、パターンはキュー・マネージャー名として扱われ、キュー・マネージャーにあるすべてのキューにアクセスできます。 例えば、パターンが @name である場合、 **@name と同じように扱われます。

以下のワイルドカード文字は、<include> および <exclude> エレメントの name 属性の一部として指定した場合、特別な意味を持ちます。

*****
単一のアスタリスクは、ディレクトリー名、または データ・セット名または キュー名の修飾子の中の 0 個以上の文字と一致します。

?
疑問符 (?) は、ディレクトリー名、または データ・セット名か キュー名の修飾子の中の 1 文字にのみ一致します。

******
2 つのアスタリスク文字は、ゼロ個以上のディレクトリー名、または データ・セット名または キュー名のゼロ個以上の修飾子に一致します。 また、パス分離文字で終わるパスには、パスの終わりに暗黙の "**" が追加されています。 したがって、/home/user/ は /home/user/**と同じです。

以下に例を示します。

- /**/test/** は、パス内に test ディレクトリーを持つすべてのファイルに一致します。
- /test/file? は、/test ディレクトリー内のすべてのファイルと一致し、先頭にはストリング file の後に任意の単一文字が続きます。
- c:\test*.txt は、c:\test ディレクトリー内のすべてのファイルを .txt 拡張子で一致させます。
- c:\test***.txt は、'c:\test ディレクトリー内の任意のファイル、または .txt 拡張子を持つサブディレクトリーの 1 つに一致します。
- z/OS //'TEST.*.DATA' は、TEST の最初の修飾子、2 番目の修飾子、および DATA の 3 番目の修飾子を持つすべてのデータセットに一致します。
- *@QM1 は、単一修飾子を持つキュー・マネージャー QM1 上のすべてのキューと一致します。
- TEST.*.QUEUE@QM1 は、TEST の最初の修飾子、2 番目の修飾子、および QUEUE の 3 番目の修飾子を持つキュー・マネージャー QM1 上のすべてのキューと一致します。
- **@QM1 は、キュー・マネージャー QM1 上の任意のキューと一致します。

シンボリック・リンク

UserSandboxes.xml ファイル内のファイル・パスで使用するシンボリック・リンクは、<include> および <exclude> エレメント内でハード・リンクを指定して、完全に解決する必要があります。 例えば、/var が /SYSTEM/var にマップするシンボリック・リンクがある場合は、このパスを <tns:include name="/SYSTEM/var"/>として指定する必要があります。そうしないと、意図した転送はユーザー・サンドボックス・セキュリティー・エラーで失敗します。

例

この例は、以下の <sandbox> エレメントを AGENT_JUPITER の構成ディレクトリー内のファイル UserSandboxes.xml に追加することにより、MQMD ユーザー名 guest を持つユーザーが、エージェント AGENT_JUPITER が実行されているシステム上の /home/user/public ディレクトリーまたはそのサブディレクトリーから任意のファイルを転送できるようにする方法を示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
        xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="guest">
            <tns:read>
                <tns:include name="/home/user/public/**"/>
             </tns:read>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

例

次の例は、account に 1 つの数字が続く MQMD ユーザー名を持つユーザー (例: account4) に、以下のアクションの実行を許可する方法を示しています。

- /home/account ディレクトリから、またはそのサブディレクトリから、エージェント AGENT_SATURN が実行されているシステム上の /home/account/private ディレクトリを除いて、任意のファイルを転送します。
- エージェントの AGENT_SATURN が実行されているシステム上の /home/account/output ディレクトリーまたはそのサブディレクトリーのいずれかにファイルを転送します。
- ローカル・キュー・マネージャー上のキューから、接頭部 ACCOUNT. で始まるメッセージを読み取ります。ただし、ACCOUNT.PRIVATE. で始まる (第 2 レベルに PRIVATE がある) 場合は除きます。
- キュー・マネージャー上の接頭部 ACCOUNT.OUTPUT. で始まるキューにデータを転送します。

MQMD ユーザー名 account のユーザーがこれらのアクションを実行できるようにするには、AGENT_SATURN の構成ディレクトリーにあるファイル UserSandboxes.xml に以下の <sandbox> エレメントを追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
            xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="account[0-9]" userPattern="regex">
            <tns:read>
                <tns:include name="/home/account/**"/>
                <tns:include name="ACCOUNT.**" type="queue"/>
                <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
                <tns:exclude name="/home/account/private/**"/>
                                                    </tns:read>
            <tns:write>
                <tns:include name="/home/account/output/**"/>
                <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
            </tns:write>
        </tns:sandbox>
```

```
    </tns:agent>
</tns:userSandboxes>
```

関連資料

549 ページの『ワイルドカード転送の追加検査』
エージェントがファイルの転送先/転送元として使用できる場所を制限するために、そのエージェントにユーザー・サンドボックスまたはエージェント・サンドボックスが構成されている場合、そのエージェントのワイルドカード転送で追加の検査が行われるように指定できます。

MFT agent.properties ファイル

ワイルドカード転送の追加検査

エージェントがファイルの転送先/転送元として使用できる場所を制限するために、そのエージェントにユーザー・サンドボックスまたはエージェント・サンドボックスが構成されている場合、そのエージェントのワイルドカード転送で追加の検査が行われるように指定できます。

additionalWildcardSandboxChecking プロパティー

ワイルドカード転送の追加検査を使用可能にするには、検査するエージェントの agent.properties ファイルに以下のプロパティーを追加します。

```
additionalWildcardSandboxChecking=true
```

このプロパティーが true に設定されている場合、ワイルドカードのファイル・マッチング用に定義されたサンドボックスの外側にある場所の読み取りを試行する転送要求がエージェントによって行われると、転送は失敗します。 1 つの転送要求内に複数の転送があり、サンドボックスの外側にある場所を読み取ろうとしたためにこれらの要求のいずれかが失敗した場合、転送全体が失敗します。 検査が失敗すると、失敗の理由がエラー・メッセージに示されます。

もし、additionalWildcardSandboxChecking プロパティーがエージェントの agent.properties ファイルから省略されているか、false に設定されている場合は、そのエージェントのワイルドカード転送に対する追加検査は行われません。

ワイルドカード検査のエラー・メッセージ

構成済みのサンドボックス・ロケーションの外側にある場所にワイルドカード転送要求が行われたときに報告されるメッセージが変更されました。

転送要求のワイルドカード・ファイル・パスが、制限されたサンドボックスの外にある場合、次のメッセージが表示されます。

BFGSS0077E: ファイル・パスを読み取ろうとしました: 「パス」 が拒否されました。
ファイル・パスは、制限された転送サンドボックスの外にあります。

複数転送要求内の転送に、制限されたサンドボックスの外にパスがあるワイルドカード転送要求が含まれている場合、次のメッセージが出されます。

BFGSS0078E: ファイル・パス *path* を読み取ろうとしましたが、別の転送として無視されました。
管理対象転送の項目が、制限された転送サンドボックスの外部で読み取ろうとしました。

制限された転送サンドボックスの外にファイルがある場合、次のメッセージが表示されます。

BFGSS0079E: ファイル *file path* を読み取ろうとしましたが、拒否されました。
ファイルは、制限された転送サンドボックスの外にあります。

次のメッセージは、別のワイルドカード転送要求が原因となりこの転送が無視された複数転送要求で表示されます。

BFGSS0080E: ファイル *file path* を読み取ろうとしましたが、別の転送として無視されました。
管理対象転送の項目が、制限された転送サンドボックスの外部で読み取ろうとしました。

ワイルドカードを含まない単一ファイル転送の場合、転送にサンドボックスの外にあるファイルが含まれているときに報告されるメッセージは前のリリースから変更されていません。

BFGIO0056E で失敗: ファイル "*FILE*" を読み取ろうとしましたが、拒否されました。
ファイルは、制限された転送サンドボックスの外にあります。

関連資料

546 ページの『MFT ユーザー・サンドボックスの処理』
ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、 転送を要求する MQMD ユーザー名に基づいて制限することができます。

544 ページの『MFT エージェント・サンドボックスの処理』
追加のセキュリティー・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

MFT agent.properties ファイル

MFT の SSL または TLS 暗号化の構成

IBM MQ Managed File Transfer で SSL または TLS を使用して、エージェントとエージェント・キュー・マネージャーの間の通信、接続先のコマンドとキュー・マネージャー、およびトポロジー内のさまざまなキュー・マネージャーとキュー・マネージャーの間の接続を保護することができます。

始める前に

SSL または TLS 暗号化を使用して、 IBM MQ Managed File Transfer トポロジーを流れるメッセージを暗号化できます。 これには以下が含まれます。

- エージェントとそのエージェント・キュー・マネージャーの間で受け渡されるメッセージ。
- 接続先のコマンドおよびキュー・マネージャーに関するメッセージ。
- トポロジー内のエージェント・キュー・マネージャー、コマンド・キュー・マネージャー、および調整キュー・マネージャーの間を流れる内部メッセージ。

このタスクについて

IBM MQ で SSL を使用する一般情報については、270 ページの『SSL/TLS の取り扱い』を参照してください。 IBM MQ の観点からすると、Managed File Transfer は、標準的な Java クライアント・アプリケーションです。

Managed File Transfer で SSL を使用するには、以下のステップを実行します。

手順

1. トラストストア・ファイルを作成し、オプションで鍵ストア・ファイルを作成します (これらのファイルは同じファイルにすることが可能です)。 クライアント認証を必要としない場合 (つまりチャネル上で SSLCAUTH=OPTIONAL)、鍵ストアを準備する必要はありません。 トラストストアは、キュー・マネージャーの証明書を認証するためにのみ必要です。

   IBM MQ で作業するには、トラストストアと鍵ストアの証明書を作成するために使用する鍵アルゴリズムが RSA でなければなりません。
2. SSL を使用するよう IBM MQ キュー・マネージャーをセットアップします。

   例えば、IBM MQ Explorer を使用して SSL を使用するようにキュー・マネージャーをセットアップする方法については、『キュー・マネージャーでの SSL の構成』を参照してください。
3. トラストストア・ファイルおよび鍵ストア・ファイル (存在する場合) を適切な場所に保存します。 推奨されるロケーションは、*config_directory*/*coordination_qmgr*/agents/*agent_name* ディレクトリーです。
4. 各 SSL 対応キュー・マネージャーの必要に応じて、SSL プロパティーを Managed File Transfer の該当するプロパティー・ファイルに設定します。 各プロパティー・セットは別個のキュー・マネージャー (エージェント、調整、およびコマンド) を参照します。ただし、1 つのキュー・マネージャーがこれらの複数のロールを担う可能性はあります。

   CipherSpec または **CipherSuite** プロパティーのいずれかが必要です。ない場合にはクライアントは SSL を使用せずに接続を試行します。 IBM MQ と Java の用語に違いがあるため、**CipherSpec** プロパティーと **CipherSuite** プロパティーの両方が提供されています。 Managed File Transfer は、どちらのプロパティーも受け入れて必要な変換を行うため、両方のプロパティーを設定する必要はありませ

ん。 **CipherSpec** と **CipherSuite** の両方のプロパティーを指定した場合は、**CipherSpec** が優先されます。

PeerName プロパティーはオプションです。 このプロパティーを、接続先キュー・マネージャーの識別名に設定できます。 Managed File Transfer は、識別名が一致しない不正確な SSL サーバーへの接続をリジェクトします。

SslTrustStore および **SslKeyStore** プロパティーを、トラストストア・ファイルおよび鍵ストア・ファイルを指すファイル名に設定します。 これらのプロパティーを既に実行中のエージェントに対してセットアップする場合、エージェントを停止してから再開し、SSL モードで再接続します。

プロパティー・ファイルにはプレーン・テキスト・パスワードが含まれるため、ファイル・システムの適切な許可を設定することを考慮してください。

SSL プロパティーについて詳しくは、MFT の SSL プロパティーを参照してください。

5. エージェントのキュー・マネージャーが SSL を使用する場合、そのエージェントを作成するときに必要な詳細を提供することはできません。 そのエージェントを作成するには次のステップを実行します。
   a) **fteCreateAgent** コマンドを使用してエージェントを作成します。 エージェントの存在を調整キュー・マネージャーにパブリッシュできないことに関する警告を受け取ります。
   b) 前のステップで作成された `agent.properties` ファイルを編集して、SSL 情報を追加します。 エージェントが正常に開始すると、パブリッシュが再度試行されます。
6. IBM MQ ファイルまたは `agent.properties` ファイル内の SSL プロパティーが変更されている間に、`coordination.properties` エクスプローラーのエージェントまたはインスタンスが実行されている場合は、エージェントまたは IBM MQ Explorer を再始動する必要があります。

関連資料
MFT `agent.properties` ファイル

クライアント・モードでチャネル認証を使用してキュー・マネージャーに接続する操作

チャネル・レベルでアクセスをより正確に制御するために、IBM WebSphere MQ 7.1 ではチャネル認証レコードが導入されました。 動作がこのように変更されたことにより、新しく作成された IBM WebSphere MQ 7.1 以降のキュー・マネージャーは、Managed File Transfer コンポーネントからのクライアント接続をデフォルトで拒否します。

チャネル認証の詳細については、47 ページの『チャネル認証レコード』を参照してください。

Managed File Transfer によって使用される SVRCONN のチャネル認証構成が非特権 MCAUSER ID を指定している場合は、Managed File Transfer Agent とコマンドが正しく動作するように、キュー・マネージャー、キュー、およびトピックに特定の権限レコードを付与する必要があります。 チャネル認証レコードを作成、変更、または削除するには、MQSC コマンド SET CHLAUTH または PCF コマンド Set Channel Authentication Record を使用します。 IBM WebSphere MQ 7.1 以降のキュー・マネージャーに接続するすべての Managed File Transfer エージェントについて、すべてのエージェントに使用する MCAUSER ID をセットアップするか、エージェントごとに別個の MCAUSER ID をセットアップすることができます。

各 MCAUSER ID に以下の権限を付与します。

- キュー・マネージャーに必要な権限レコード:
  - connect
  - setid
  - inq
- キューに必要な権限レコード:

  すべてのエージェント固有キュー (以下のリストでキュー名の最後が *agent_name* になっているもの) に関して、クライアント接続を使用して IBM WebSphere MQ 7.1 以降のキュー・マネージャーに接続するエージェントごとに、これらのキュー権限レコードを作成する必要があります。

  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)

  - put, get, setid, browse (SYSTEM.FTE.COMMAND.*agent_name*)
  - put, get (SYSTEM.FTE.DATA.*agent_name*)
  - put, get (SYSTEM.FTE.REPLY.*agent_name*)
  - put, get, inq, browse (SYSTEM.FTE.STATE.*agent_name*)
  - put, get, browse (SYSTEM.FTE.EVENT.*agent_name*)
  - put, get (SYSTEM.FTE)
- トピックに必要な権限レコード:
  - sub, pub (SYSTEM.FTE)
- ファイル転送に必要な権限レコード。

  ソース・エージェントと宛先エージェントで MCAUSER ID が異なる場合には、ソースと宛先の両方のエージェント・キューに対して権限レコードを作成します。

  例えば、ソース・エージェントの MCAUSER ID が **user1** で、宛先エージェントの MCAUSER ID が **user2** の場合、 それぞれのエージェント・ユーザーに対して以下の権限を設定します。

エージェント・ユーザー	キュー	必要な権限
user1	SYSTEM.FTE.DATA.*destination_agent_name*	put
user1	SYSTEM.FTE.COMMAND.*destination_agent_name*	put
user2	SYSTEM.FTE.REPLY.*source_agent_name*	put
user2	SYSTEM.FTE.COMMAND.*source_agent_name*	put

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードが SSL プロトコルを使用して相互に接続するように構成します。そのためには、鍵ストアとトラストストアを作成し、Connect:Direct ブリッジ・エージェントのプロパティー・ファイルでプロパティーを設定します。

このタスクについて

ここでは、認証局から鍵の署名を得るための手順を含めています。 認証局を使用しない場合は、自己署名証明書を生成できます。 自己署名証明書の生成について詳しくは、283 ページの『UNIX, Linux, and Windows での SSL/TLS の取り扱い』を参照してください。

ここでは、Connect:Direct ブリッジ・エージェントの新しい鍵ストアとトラストストアを作成するための手順を含めています。 Connect:Direct ブリッジ・エージェントに、IBM MQ キュー・マネージャーへのセキュア接続で使用できる鍵ストアとトラストストアが既にある場合は、Connect:Direct ノードへのセキュア接続で既存の鍵ストアとトラストストアを使用できます。 詳細については、550 ページの『MFT の SSL または TLS 暗号化の構成』を参照してください。

手順

Connect:Direct ノードの場合、以下のステップを実行します。

1. Connect:Direct ノードの鍵と署名付きの証明書を生成します。

   これは、IBM MQ で提供される IBM 鍵管理ツールを使用して行うことができます。 詳しくは、270 ページの『SSL/TLS の取り扱い』を参照してください。
2. 鍵の署名を得るための要求を認証局に送信します。 返ってくる証明書を受け取ります。
3. テキストファイルの作成。たとえば、証明機関の公開鍵を含む `/test/ssl/certs/CAcert` などです。
4. Connect:Direct ノードに Secure+ オプションをインストールします。

ノードが既に存在している場合は、インストーラーを再び実行し、既存のインストール環境の場所を指定し、Secure+ オプションだけのインストールを選択することによって、Secure+ オプションをインストールできます。

5. 新規テキスト・ファイルを作成します。たとえば、/test/ssl/cd/keyCertFile/*node_name*.txt です。
6. 認証局から受信した証明書と、/test/ssl/cd/privateKeys/*node_name*.key 内にある秘密鍵をテキスト・ファイルにコピーします。

   /test/ssl/cd/keyCertFile/*node_name*.txt の内容は、以下の形式になっている必要があります。

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHQjES
MBAGA1UECBMJSGFtcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xDjAMBgNVBAsTBU1RSVBUMQswCQYDVQQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEwlIYW1wc2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvgP1QIklU9ypSKD1XoODo1yk
EyMFXBOUpZRrDVxjoSEC0vtWNcJ199e+Vc4UpNybDyBu+NkDlMNofX4QxeQcLAFj
WnhakqCiQ+JIAD5Aurhnrwche0MV3kjA84GKH/rOSVqtl984mu/lDyS819XcfSSn
cOOMsK1KbneVSCIV2XECAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0E
HxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFNXMIpSc
csBXUniW4A3UrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj4lVz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7klXa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/vl8+iv01OCL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIiEk9Dlz2U6WO9lICwn
l7PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qHO9ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRUOIQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDWoMNt5fj51v7aPmVeS60bOm+UlGre8B/Ze18JVj2O4K2Uh72rDCXE
5e6eFxSdUM207sQDy20euBVELJtM2kOkL1ROdoQQSlU3XQNgJw/t3ZIx5hPXWEQT
rjRQO64BEhb+PzzxPF8uwzZ9IrUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI8OGvBvWXO4fHyvIX5aslwhBoArXIS1AtNTrptPvoaP1zyIAeZ6OCVo/
SFo+A2UhmtEJeOJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWDyJBVhUOjXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+lYcX8OVZ6
sHFPNlHluCNy/riUcBy9iviVeodX8IomOchSyO5DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uUOAeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM5OoWXa6U5+AYuGUMg
/itPZmUmNrHjTk7ghT6i1IQOaBowXXKJBlMmq/6BQXN2IhkD9ys2qrvM1hdi5nAI
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lzQ==
-----END RSA PRIVATE KEY-----
```

7. Secure+ 管理ツールを開始します。
   - Linux または UNIX のシステムでは、**spadmin.sh** コマンドを実行します。
   - Windows のシステムでは、「**スタート**」 > 「**プログラム**」 > 「**Sterling Commerce Connect:Direct**」 > 「**CD Secure+ 管理ツール**」をクリックします。

   CD Secure+ 管理ツールが開始します。
8. CD Secure+ 管理ツールで、**.Local** の行をダブルクリックして、SSL または TLS のメイン設定を編集します。
   a) 使用するプロトコルに応じて、「**SSL プロトコルを有効にする**」または「**TLS プロトコルを有効にする**」を選択します。
   b)「**オーバーライドを無効にする**」を選択します。
   c) 少なくとも 1 つの暗号スイートを選択します。
   d) 両方向認証が必要な場合は、 「**クライアント認証を有効にする**」 の値を Yes に変更します。
   e)「**トラステッド・ルート証明書**」フィールドに、認証局 /test/ssl/certs/CAcert の公開証明書ファイルへのパスを入力します。
   f)「**鍵証明書ファイル**」フィールドに、作成したファイル /test/ssl/cd/keyCertFile/*node_name*.txt へのパスを入力します。

9. **.Client** の行をダブルクリックして、SSL または TLS のメイン設定を編集します。
   a) 使用するプロトコルに応じて、「**SSL プロトコルを有効にする**」または「**TLS プロトコルを有効にする**」を選択します。
   b)「**オーバーライドを無効にする**」を選択します。

Connect:Direct ブリッジ・エージェントの場合は、以下の手順を実行します。

10. トラストストアを作成します。 そのためには、ダミーの鍵を作成してから、そのダミーの鍵を削除します。
    以下のコマンドを使用できます。

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 認証局の公開証明書をトラストストアにインポートします。
    以下のコマンドを使用できます。

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. Connect:Direct ブリッジ・エージェントのプロパティー・ファイルを編集します。
    ファイルの任意の場所に以下の行を組み込みます。

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

    この手順の例では、*protocol* は使用するプロトコル (SSL または TLS) で、*password* はトラストストアの作成時に指定したパスワードです。

13. 双方向認証を設定する場合は、Connect:Direct ブリッジ・エージェントの鍵と証明書を作成します。
    a) 鍵ストアと鍵を作成します。
       以下のコマンドを使用できます。

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

    b) 署名要求を生成します。
       以下のコマンドを使用できます。

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

    c) 前の手順で受け取った証明書を鍵ストアにインポートします。 証明書は、x.509 形式でなければなりません。
       以下のコマンドを使用できます。

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

    d) Connect:Direct ブリッジ・エージェントのプロパティー・ファイルを編集します。
       ファイルの任意の場所に以下の行を組み込みます。

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

この手順の例では、*password* は鍵ストアの作成時に指定したパスワードです。

関連タスク

Connect:Direct ブリッジの構成

ULW AMQP クライアントの保護

さまざまなセキュリティー・メカニズムを使用して、AMQP クライアントからの接続を保護し、データがネットワーク上で適切に保護されるようにします。 MQ Light アプリケーションにセキュリティーを組み込むことができます。 また、IBM MQ の既存のセキュリティー機能を、他のアプリケーションに使用するのと同様の方法で、AMQP クライアントでも使用することができます。

チャネル認証規則 (CHLAUTH)

チャネル認証規則を使用して、キュー・マネージャーへの TCP 接続を制限できます。 AMQP チャネルは、キュー・マネージャー用に構成されたチャネル認証規則の使用をサポートします。 チャネル認証規則が、キュー・マネージャー上のいずれかの AMQP チャネルと一致するプロファイルによって定義されている場合、これらの規則はそれらのチャネルに適用されます。 デフォルトでは、新しい IBM® MQ キュー・マネージャーでチャネル認証が有効になっているので、AMQP チャネルを使用するためにはその前に少なくとも一部の構成を行う必要があります。

キュー・マネージャーへの AMQP 接続を許可するようにチャネル認証規則を構成する方法について詳しくは、AMQP チャネルの作成および使用を参照してください。

接続認証 (CONNAUTH)

接続認証を使用して、キュー・マネージャーへの接続を認証できます。 AMQP チャネルは、AMQP アプリケーションからキュー・マネージャーへのアクセスを制御するための接続認証の使用をサポートします。

AMQP プロトコルは SASL (Simple Authentication and Security Layer) フレームワークを使用して、接続が認証される方法を指定します。 さまざまな SASL メカニズムがありますが、IBM MQ は 2 つの SASL メカニズム (ANONYMOUS および PLAIN) をサポートします。

ANONYMOUS の場合、クライアントからキュー・マネージャーに認証のための資格情報は渡されません。 CONNAUTH 属性に指定された MQ AUTHINFO オブジェクトの CHCKCLNT 値が REQUIRED または REQDADM (管理ユーザーとして接続しているとき) の場合、接続は拒否されます。 CHCKCLNT の値が NONE または OPTIONAL の場合、接続は受け入れられます。

PLAIN の場合、クライアントからキュー・マネージャーに認証のためのユーザー名とパスワードが渡されます。 CONNAUTH 属性に指定された MQ AUTHINFO オブジェクトの CHCKCLNT 値が NONE の場合、接続は拒否されます。 CHCKCLNT の値が OPTIONAL、REQUIRED、または REQDADM (管理ユーザーとして接続しているとき) の場合、ユーザー名とパスワードはキュー・マネージャーによって検査されます。 キュー・マネージャーは、オペレーティング・システム (AUTHINFO オブジェクトのタイプが IDPWOS の場合) または LDAP リポジトリー (AUTHINFO オブジェクトのタイプが IDPWLDAP の場合) を検査します。

以下の表に、この認証の動作の要約を示します。

表 95. SASL メカニズムおよび接続認証の要約		
SASL メカニズム	**資格情報がクライアントからキュー・マネージャーに渡されるかどうか**	**CHKCLNT 値**
ANONYMOUS	いいえ	REQUIRED または REQDADM - 接続は拒否されます NONE または OPTIONAL - 接続は受け入れられます
PLAIN	はい。ユーザー名とパスワード。	REQUIRED、REQDADM、または OPTIONAL - キュー・マネージャーが検査するユーザー名とパスワード NONE - 接続は拒否されます

MQ Light クライアントを使用している場合は、接続先の AMQP アドレスに含めることによって資格情報を指定できます。次に例を示します。

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

チャネルでの MCAUSER 設定

AMQP チャネルの MCAUSER 属性で IBM MQ ユーザー ID を設定することにより、その ID でチャネルへのすべての接続が認証されるようにすることができます。 AMQP クライアントからそのチャネルへのすべての接続は、構成されたその MCAUSER ID を採用します。 そのユーザー ID は、さまざまなトピックでのメッセージングを許可するために使用されます。

チャネル認証 (CHLAUTH) を使用してキュー・マネージャーへの接続を保護することが推奨されています。チャネル認証を使用している場合、MCAUSER の値を、特権を持たないユーザーに構成することが推奨されます。 これにより、チャネルへの接続が CHLAUTH 規則にマッチングしない場合、その接続はキュー・マネージャーでメッセージングを行うことを許可されないようになります。

注 : Windows Windows の場合、IBM MQ 9.1.1 より前では、MCAUSER ユーザー ID の設定は、最大 12 文字のユーザー ID に対してのみサポートされます。

V 9.1.1 IBM MQ 9.1.1 より、12 文字の長さ制限は取り除かれています。

SSL/TLS のサポート

AMQP チャネルは、キュー・マネージャー用に構成された鍵リポジトリーにある鍵を使用する SSL/TLS 暗号化をサポートします。 SSL/TLS 暗号化の AMQP チャネル構成オプションは、他のタイプの MQ チャネルと同じオプションをサポートします。 暗号仕様、およびキュー・マネージャーが AMQP クライアント接続からの証明書を必要とするかどうかを指定できます。

キュー・マネージャーの FIPS 属性を使用して、SSL/TLS 暗号スイートを制御できます。 これを使用して、AMQP クライアントからの接続を保護することができます。

キュー・マネージャーの鍵リポジトリーをセットアップする方法については、UNIX、Linux および 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。

AMQP クライアント接続のための SSL/TLS サポートを構成する方法については、AMQP チャネルの作成および使用を参照してください。

Java 認証・承認サービス (JAAS) (Java Authentication and Authorization Service (JAAS))

オプションで、AMQP チャネルに JAAS ログイン・モジュールを構成し、AMQP クライアントから指定されるユーザー名とパスワードを検査することもできます。 558 ページの『AMQP チャネルのための JAAS の構成』を参照してください。

関連タスク

AMQP クライアント・アプリケーションの開発

AMQP チャネルの作成および使用

ULW AMQP クライアント・テークオーバーの制限

既存の AMQP クライアント接続と同じクライアント ID を持つ AMQP クライアント接続が行われると、既存のクライアント接続はデフォルトで切断されます。 ただし、クライアントのテークオーバー動作を制限するようにキュー・マネージャーを構成して、特定の基準が満たされた場合にのみテークオーバーが可能になるようにすることができます。

例えば、複数の異なるチームによって開発されている AMQP アプリケーションがあり、 それらがたまたま同じクライアント ID を使用している場合、既存のクライアント接続を切断することが適切ではないことがあります。 この問題に取り組むため、使用されている AMQP チャネルの名前、クライアントの IP アドレス、およびクライアントのユーザー ID (SASL 認証が有効な場合) に基づいてクライアント・テークオーバーを制限できます。

キュー・マネージャー属性 **AdoptNewMCA** および **AdoptNewMCACheck** の設定を使用して、次の表に示されているように、必要なクライアント・テークオーバー制限のレベルを指定します。

表 96. クライアント・テークオーバーを制限するための ***AdoptNewMCA*** および ***AdoptNewMCACheck*** の設定

AdoptNewMCA	**AdoptNewMCACheck**	**クライアント・テークオーバーが許可される前に検査される基準**
NO または未定義	適用外	なし。 クライアント・テークオーバーは、認証されていて、CHLAUTH 規則のすべてに合格するクライアント接続で許可されます。
ALL (または NO 以外の値)	QM または未定義	なし。 クライアント・テークオーバーは、認証されていて、CHLAUTH 規則のすべてに合格するクライアント接続で許可されます。
ALL (または NO 以外の値)	名前	ユーザー ID (SASL が有効のとき) チャネル名
ALL (または NO 以外の値)	ADDRESS	ユーザー ID (SASL が有効のとき) IP アドレス
ALL (または NO 以外の値)	ALL	ユーザー ID (SASL が有効のとき) チャネル名 IP アドレス

キュー・マネージャー属性 **AdoptNewMCA** および **AdoptNewMCACheck** は、CHANNELS スタンザで定義されるキュー・マネージャー構成の一部です。 IBM MQ for Windows システムおよび IBM MQ for Linux x86-64 システムでは、IBM MQ Explorer を使用して構成情報を変更します。 その他のシステムでは、

qm.ini 構成ファイルを編集して情報を変更します。 キュー・マネージャー・チャネルの情報を変更する方法について詳しくは、 「チャネルの属性」を参照してください。

関連タスク
AMQP クライアント・アプリケーションの開発
AMQP チャネルの作成および使用

ULW AMQP チャネルのための JAAS の構成

Java 認証・承認サービス (JAAS) のカスタム・モジュールを使用して、接続時に AMQP クライアントから AMQP チャネルに渡されるユーザー名とパスワードの資格情報を認証することができます。

このタスクについて

他の Java ベース・システムの認証で既に JAAS モジュールを使用していて、そのモジュールを MQ への AMQP 接続の認証にも使用したい場合などに、カスタム JAAS モジュールを使用できます。 また、MQ の標準の認証機能では、使用したい認証メカニズムがサポートされていない場合などにも、カスタム JAAS モジュールを作成できます。

AMQP チャネルに対する JAAS モジュールの構成は、キュー・マネージャー・レベルで行われます。 つまり、キュー・マネージャーへの AMQP 接続を認証するように JAAS モジュールを構成すると、そのモジュールがすべての AMQP チャネルに適用されます。 JAAS モジュールを呼び出したチャネルの名前がモジュールに渡されるため、チャネルごとに異なる JAAS ログイン動作をコーディングできます。

その他に、次の情報も JAAS モジュールに渡されます。

- 認証を試行している AMQP クライアントのクライアント ID。
- AMQP クライアントのネットワーク・アドレス。
- JAAS モジュールを呼び出したチャネルの名前。

手順

以下の手順を実行して、AMQP チャネルに対して JAAS 構成モジュールを構成します。

1. 1 つ以上の JAAS モジュール構成スタンザを含む jaas.config ファイルを定義します。 スタンザでは、JAAS のインターフェース javax.security.auth.spi.LoginModule を実装する Java クラスの完全修飾名を指定する必要があります。
   - デフォルトの jaas.config ファイルは、製品と共に出荷され、 *QM_data_directory*/amqp/jaas.config にあります。
   - このデフォルトの jaas.config ファイルには、MQXRConfig という名前の構成済みのスタンザが既に定義されています。
2. AMQP チャネルで使用するスタンザの名前を指定します。
   - UNIX amqp_unix.properties ファイルにプロパティーを追加します。
   - Windows amqp_win.properties ファイルにプロパティーを追加します。

   プロパティーの形式を次に示します。

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   以下に例を示します。

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```
3. キュー・マネージャー環境を、カスタム・モジュールのクラスを含むように構成します。 AMQP サービスが、JAAS 構成スタンザに構成された Java クラスを使用できなければなりません。

これを実行するには、JAAS クラスへのパスを MQ service.env ファイルに追加します。 MQ 構成ディレクトリー (*MQ_config_directory*) またはキュー・マネージャー構成ディレクトリー (*QM_config_directory*) 内の service.env ファイルを編集して、CLASSPATH 変数を JAAS モジュール・クラスのロケーションに設定します。

次のタスク

JAAS ログイン・モジュールのサンプルは、*mq_installation_directory*/amqp/samples ディレクトリーに製品と一緒に出荷されています。 このサンプル JAAS ログイン・モジュールは、クライアントの接続時に使用されたユーザー名またはパスワードにかかわらず、すべてのクライアント接続を認証します。

特定のパスワードを持つ特定のユーザーのみを認証するように、サンプルのソース・コードを変更して再コンパイルすることができます。 製品に付属するサンプル JAAS ログイン・モジュールを使用するように UNIX システム上の AMQP チャネルを構成するには、次のようにします。

1. ファイル /var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties を編集し、プロパティー com.ibm.mq.MQXR.JAASConfig=MQXRConfig を設定します。
2. ファイル /var/mqm/service.env を編集し、プロパティー CLASSPATH=*mq_installation_location*/amqp/samples を設定します。

jaas.config ファイルには、ログイン・モジュール・クラスとしてサンプル・クラス samples.JAASLoginModule を指定する MQXRConfig という名前のスタンザが既に含まれています。 サンプル・モジュールを試行する前に jaas.config に対して変更を行う必要はありません。

関連タスク

AMQP クライアント・アプリケーションの開発

AMQP チャネルの作成および使用

Advanced Message Security

Advanced Message Security (AMS) は、IBM 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 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。

Advanced Message Security の概要

IBM MQ アプリケーションは、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

AMS メッセージで使用される GSKit 戻りコード

Advanced Message Security のフィーチャーおよび機能

Advanced Message Security は、IBM 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っ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。

AMS は、以下の機能を提供します。

- IBM 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。
- IBM MQ 用の既存の専有アプリケーションおよび顧客作成アプリケーションを保護する

- z/OS V 9.1.3 IBM MQ 9.1.3 以降、IBM MQ for z/OS には、ネットワークを流れるメッセージの AMS 保護を解除したり追加したりする機能が用意されています。 この機能は、サーバー間メッセージ・チャネル・エージェント *(MCA)* インターセプト と呼ばれています。
- V 9.1.0.4 V 9.1.4 ULW IBM MQ 9.1.4 および IBM MQ 9.1.0 Fix Pack 4 以降、お客様のアプリケーション・プログラム内で実行される IBM MQ ライブラリー・コードに、ある検査が追加されています。 この検査は初期化の早い段階で実行されて環境変数 *AMQ_AMS_FIPS_OFF* の値を読み取り、それがいずれかの値に設定されている場合は、そのアプリケーションで GSKit コードが非 FIPS モードで実行されます。

AMS で使用可能な保護品質

Advanced Message Security には、`Integrity`、`Privacy`、および `Confidentiality` という 3 つの保護品質があります。

`Integrity` 保護は、デジタル署名によって提供されます。これにより、誰がメッセージを作成したかが明らかとなります。また、メッセージが変更または改ざんされないようにします。

`Privacy` 保護は、デジタル署名と暗号化の組み合わせによって提供されます。 暗号化により、対象の受信者だけがメッセージ・データを表示できるようにします。 許可されていない受信者が暗号化されたメッセージ・データのコピーを取得したとしても、実際のメッセージ・データ自体を表示することはできません。

`Confidentiality` 保護は、オプションの鍵再利用による暗号化でのみ提供されます。

パフォーマンスへの影響

AMS は、対称暗号ルーチンと非対称暗号ルーチンの組み合わせを使用して、デジタル署名と暗号化を提供します。 対称鍵操作は、CPU 使用率の高い非対称鍵操作と比べて非常に高速であり、AMS で大量のメッセージを保護する際のコストに重大な影響を与える可能性があります。

非対称暗号ルーチン

例えば、署名されたメッセージを送信する際、非対称鍵操作を使用してメッセージ・ハッシュに署名されます。

署名メッセージを取得する際にも、非対称鍵操作を使用して、署名されたハッシュを確認します。

したがって、メッセージ・データに署名し、確認するために、1 つのメッセージにつき少なくとも 2 回の非対称鍵操作が必要です。

非対称暗号ルーチンと対称暗号ルーチン

暗号化されたメッセージを送信する際、対称鍵が生成され、メッセージの対象受信者ごとに非対称鍵操作を使用して暗号化されます。

その後、メッセージ・データは、対称鍵によって暗号化されます。 暗号化されたメッセージを取得する際、対象受信者は非対称鍵操作を使用して、メッセージで使用されている対称鍵を発見する必要があります。

したがって、3 つの保護品質にはすべて、CPU の使用率が高い非対称鍵操作のさまざまな要素が含まれています。これは、メッセージの送信および取得を行うアプリケーションの最大到達可能メッセージング・レートに大きな影響を与えます。

しかし、`Confidentiality` ポリシーにより、一連のメッセージで対称鍵を再利用することができます。`Confidentiality` ポリシーでは、対称鍵の再使用によって CPU コストを大幅に節約できます。 この操作モードでは、対称暗号鍵を共有するために PKCS#7 フォーマットが引き続き使用されます。 ただし、デジタル署名がないので、メッセージごとの非対称鍵操作がいくつかなくなります。 やはり受信者ごとに非対称鍵操作で対称鍵を暗号化する必要がありますが、同じ受信者宛ての複数のメッセージで、必要に応じて対称鍵を再使用できます。 鍵の再使用がポリシーで許可されていれば、非対称鍵操作を必要とするのは、最初のメッセージのみです。 後続のメッセージでは、対称鍵操作を使用するだけで済みます。

鍵の再使用

`Confidentiality` ポリシーを使用すると、対称鍵再使用アプローチを使用して、同じキューに書き込まれ、同じ受信者を対象とする複数のメッセージの暗号化に関連するコストを大幅に削減できます。

例えば、10 通の暗号化されたメッセージを同じ受信者のセットに送信する場合、1 つの対称鍵が生成され、メッセージの対象受信者ごとに非対称鍵操作を使用して最初のメッセージで暗号化されます。

ポリシーによって制御された制限に基づいて、暗号化された対称鍵を、同じ受信者宛ての後続のメッセージで再利用することができます。 暗号化されたメッセージを取得するアプリケーションは、対称鍵が変更されていないときを判別して対称鍵を取得するためのコストを回避できるという点で、同じ最適化を適用できます。

この例では、同じ鍵を再利用することにより、送信を行うアプリケーションと取得を行うアプリケーションの両方で非対称鍵操作の 90% を回避できます。

鍵の再利用の詳細については、以下の資料を参照してください。

- MQSC コマンド SET POLICY
- 制御コマンド setmqspl
- IBM i コマンド SETMQMSPL

AMS の基本概念

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と *Advanced Message Security*

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

603 ページの『鍵ストアおよび証明書の使用』
IBM 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。 z/OS では、鍵ストア・ファイルの代わりに SAF 鍵リングを使用します。

AMS におけるデジタル証明書

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。

Multi 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、Multiplatforms の IBM 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、IBM 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。

- メッセージ・キュー・インターフェース (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
Message Queue Interface の概要

Advanced Message Security で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 IBM 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)
- IBM MQ Java Message Service (JMS) 1.0.2 および 1.1。
- IBM MQ 基本クラス Java

- IBM 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。

AMS の既知の制限

サポートされていないか、 Advanced Message Security に制限がある IBM MQ オプションがいくつかあります。

- 以下の IBM MQ オプションはサポートされていないか、または制限されています。

  パブリッシュ/サブスクライブ
  Point-to-Point と比較した場合のパブリッシュ/サブスクライブ・メッセージング・モデルの大きな利点の 1 つは、送信側および受信側のアプリケーションがデータを送受信するために互いについて識別する必要がない点です。 この利点は、意図された受信者または許可署名者を定義する必要がある Advanced Message Security ポリシーの使用によって否定されます。 アプリケーションが、ポリシーによって保護されている別名キュー定義を経由してトピックにパブリッシュすることは可能であり、サブスクライブ側のアプリケーションが、ポリシーで保護されたキューからメッセージを読み取ることも可能です。 ポリシーをトピック・ストリングに直接割り当てることはできず、ポリシーをキュー定義にのみ割り当てることができます。

  チャネル・データの変換
  Advanced Message Security で保護されたメッセージの保護されたペイロードは、バイナリー形式を使用して送信されるため、アプリケーション間のチャネルでのデータ変換によってメッセージ・ダイジェストが無効にされることがなくなります。 ポリシーで保護されたキューからメッセージを取り出すアプリケーションは、データ変換を要求する必要があり、保護されたペイロードの変換は、メッセージが正常に検査されて保護を解除された後に試行されます。

  配布リスト
  Advanced Message Security ポリシーは、メッセージを配布リストに書き込むアプリケーションを保護する場合に使用できますが、リスト内の各宛先キューに同じポリシーが定義されている場合に限ります。 アプリケーションが配布リストを開いたときに矛盾するポリシーが識別された場合、オープン操作は失敗して、セキュリティー・エラーがアプリケーションに返されます。

  アプリケーション・メッセージのセグメンテーション
  ポリシーで保護されたメッセージのサイズが増大すると、アプリケーションでメッセージのセグメント境界を正確に指定できません。

  管理対象モード (クライアント接続) で IBM MQ classes for .NET を使用するアプリケーション
  管理対象モード (クライアント接続) で IBM MQ classes for .NET を使用するアプリケーションはサポートされていません。

  注 : MCA インターセプトを使用すると、サポートされないクライアントで AMS を使用できるようになります。

  管理対象モードでの Message Service client for .NET (XMS) アプリケーション
  管理対象モードでの Message Service client for .NET (XMS) アプリケーションはサポートされていません。

  注 : MCA インターセプトを使用すると、サポートされないクライアントで AMS を使用できるようになります。

  IMS ブリッジによって処理される IBM MQ キュー
  IMS ブリッジによって処理される IBM MQ キューはサポートされません。

  注 : AMS は CICS ブリッジ・キューでサポートされています。 CICS ブリッジ・キューでの MQPUT (暗号化) および MQGET (暗号化解除) には、同じユーザー ID を使用する必要があります。

  待機中の getter への書き込み
  AMS ポリシーが定義されているキューに対する getter アプリケーションでは、待機中の getter への書き込みができません。

  V 9.1.3 **サーバー間 MCA インターセプト**
  IBM MQ 9.1.3 以降、IBM MQ for z/OS では、サーバー間 MCA インターセプトは、送信側、サーバー、受信側、要求側の各チャネル・タイプでのみサポートされています。

- ユーザーは同じ識別名を持つ複数の証明書を単一の鍵ストア・ファイルに置かないでください。メッセージを保護するときに使用する証明書の選択が未定義になるからです。
- **WMQ_PROVIDER_VERSION** プロパティーが 6 に設定されている場合、 JMS では AMS はサポートされません。
- AMS インターセプターは、AMQP または MQTT チャネルでサポートされていません。

z/OS V 9.1.3 メッセージ・チャネルでの Advanced Message Security インターセプトの概要

z/OS では、 Advanced Message Security (AMS) インターセプトにより、送信側チャネル、サーバー・チャネル、受信側チャネル、および要求側チャネルにセキュリティー・ポリシー保護 (SPLPROT) の追加オプションが追加され、既存のオファリングが拡張されます。

現在、 図 1 に示すように、銀行と通信するクリアリングハウスの例を使用すると、システムの両側で AMS をサポートする必要があります。

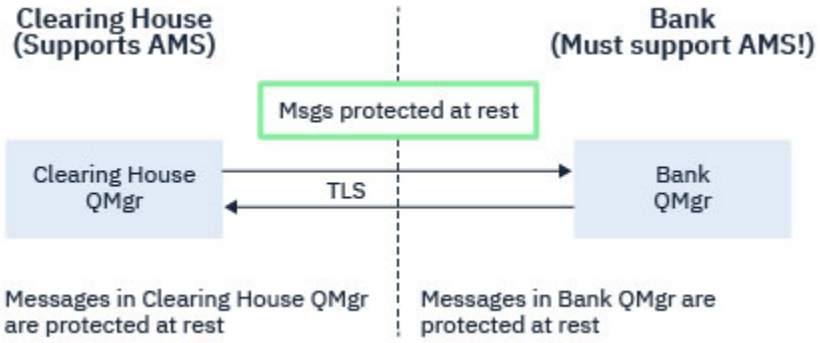


図 *32.* 現在の *AMS* 使用状況

この追加オプションの主要な利点としては、企業で AMS が構成されているものの、すべてのビジネス・パートナーが AMS をサポートしているわけではない場合、 AMS をサポートしていないビジネス・パートナーとの間のチャネルでアウトバウンド・メッセージの保護を削除し、インバウンド・メッセージを保護できるという点があります。

手形交換所と銀行の例を使用し、このシナリオを図 2 に示します。手形交換所、銀行、ビジネス・パートナーそれぞれの間にメッセージ・フローがあり、一部の事業体には AMS があり、一部にはありません。

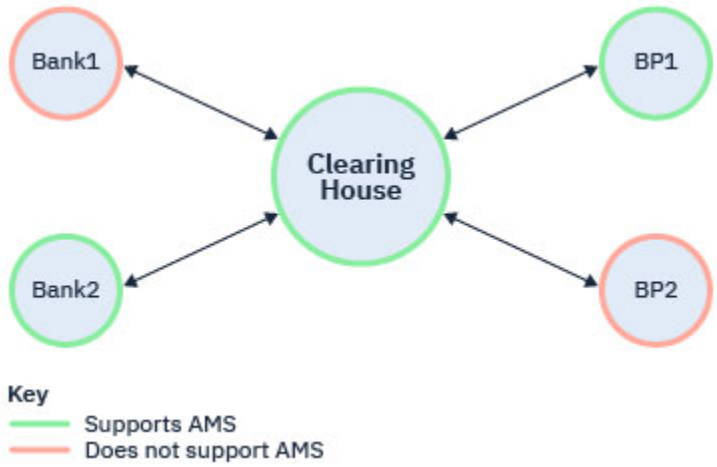


図 *33.* 一部のパートナーは *AMS* をサポートし、一部はサポートしていない

通常、チャネルでは TLS が有効です。

ただし、一部の銀行とビジネス・パートナーが AMS をサポートしていないものの、すべての銀行とビジネス・パートナー間でメッセージ交換できなければならないという状況が生じる可能性があります。 このシナリオを図 3 に示します。

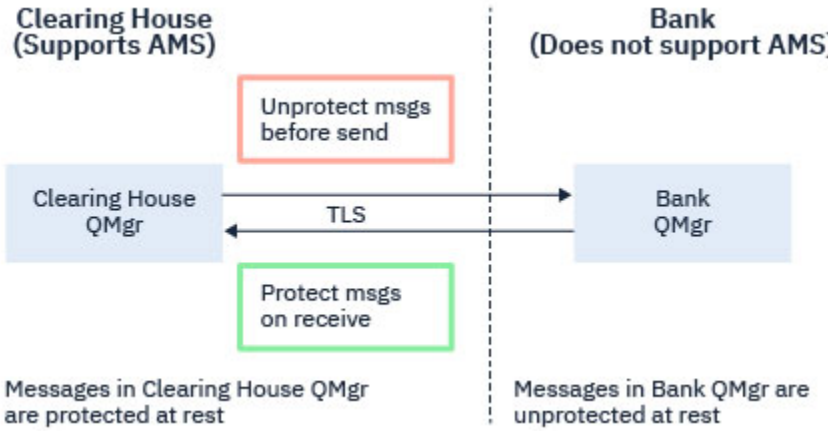


図 *34.* ビジネス・パートナー間のメッセージ・フロー

関連タスク

サーバー間メッセージ・チャネル・インターセプトの構成例

z/OS V 9.1.3 サーバー間メッセージ・チャネルでの *AMS* インターセプト

サーバー間メッセージ・チャネル・インターセプトを使用すると、メッセージに該当する Advanced Message Security (AMS) ポリシーを適用するかどうか、送信側タイプのメッセージ・チャネル・エージェントが伝送キューからメッセージを取得したり、受信側タイプのメッセージ・チャネル・エージェントがターゲット・キューにメッセージを書き込んだりするタイミングを制御できます。

これにより、AMS が有効になっていないキュー・マネージャーで、送信側、サーバー、受信側、要求側のタイプのサーバー間メッセージ・チャネルを使用して通信するときに、キュー・マネージャーで AMS 保護を有効な状態にすることができます。

つまり、AMS が有効なキュー・マネージャーの AMS 保護メッセージは AMS が有効ではないキュー・マネージャーに送信される前には保護されていない状態であることが可能で、AMS が有効ではないキュー・マネージャーから受信する保護されていないメッセージは AMS が有効なキュー・マネージャーで該当する AMS ポリシーを使用することにより保護することができます。

サーバー間メッセージ・チャネル・インターセプトの構成

サーバー間メッセージ・チャネル・インターセプトは、送信側、サーバー、受信側、または要求側のチャネル・タイプで SPLPROT 属性を使用して構成します。 動作の構成に選択できるオプションは、指定されているチャネル・タイプによって異なります。

PASSTHRU

このチャネルでメッセージング・チャネル・エージェントが送受信するメッセージを変更なしでパススルーします。

この値は、チャネル・タイプ (**CHLTYPE**) が SDR、SVR、RCVR、または RQSTR であるチャネルに有効で、これがデフォルト値です。

REMOVE

メッセージ・チャネル・エージェントが伝送キューから受け取ったメッセージの AMS 保護を解除し、そのメッセージをパートナーに送信します。

メッセージング・チャネル・エージェントが伝送キューからメッセージを受け取り、その伝送キューに AMS ポリシーが定義されていた場合、チャネルでメッセージを送信する前に、そのポリシーが適用されてメッセージの AMS 保護が解除されます。 伝送キューに AMS ポリシーが定義されていない場合、メッセージはそのまま送信されます。

この値は、チャネル・タイプが SDR または SVR のチャネルにのみ有効です。

ASPOLICY
ターゲット・キューに定義されたポリシーに基づいて、インバウンド・メッセージに AMS 保護を適用してからターゲット・キューに書き込まれるようにします。

メッセージ・チャネル・エージェントがインバウンド・メッセージを受信するときに、ターゲット・キューに AMS ポリシーが定義されている場合、メッセージがターゲット・キューに書き込まれる前に、AMS 保護がメッセージに適用されます。 ターゲット・キューに AMS ポリシーが定義されていない場合、メッセージはそのままターゲット・キューに書き込まれます。

この値は、チャネル・タイプが RCVR または RQSTR のチャネルにのみ有効です。

メッセージ・チャネル・インターセプトのユーザー ID

サーバー間メッセージ・チャネル・インターセプトで使用するユーザー ID の要件は、既存の AMS が有効なアプリケーションと同じです。 実行中のチャネルの場合、送信側のメッセージ・チャネル・エージェントは伝送キューからメッセージを取得し、受信側のメッセージ・チャネル・エージェントがターゲット・キューにメッセージを書き込みます。 サーバー間チャネルで設定されるメッセージ・チャネル・エージェント・ユーザー ID (MCAUSER) フィールドで、メッセージ・チャネル・エージェントが書き込みと取得の要求を行うユーザー ID を定義します。

サーバー間メッセージ・チャネル・インターセプトを使用する場合、他の AMS が有効なアプリケーションと同様、取得と書き込みの要求中に AMS 機能が実行されます。 そのため、メッセージ・チャネル・エージェントのユーザー ID の要件は、 AMS アプリケーションのユーザー ID の要件と同じになります。

書き込みと取得を実行する MCAUSER は構成可能で、アウトバウンド・チャネルとインバウンド・チャネルのどちらであるかによって異なります。 メッセージ・チャネル・エージェントにおける選択したユーザー ID による操作の詳細については、MCAUSER を参照してください。 また、チャネル・イニシエーターを実行しているユーザー ID が、サーバー間メッセージ・チャネル・インターセプトで AMS 機能を実行するために使用するユーザー ID になります。 そのため、こうしたユーザーの要件は、AMS アプリケーションのユーザー ID と同じです。

認証は、PUTAUT 構成を持つチャネルの詳細に関する既存の規則を使って実行されます。 詳しくは、チャネル・イニシエーターで使用されるユーザー ID を参照してください。

注 : サーバー間メッセージ・チャネル・インターセプトでは、PUTAUT チャネル属性の値は考慮されません。

メッセージ・サイズと MAXMSGL

AMS 保護のため、保護対象メッセージのメッセージ・サイズは元のメッセージ・サイズよりも大きくなります。

保護対象メッセージは、保護されていないメッセージよりも大きくなります。 そのため、保護対象メッセージのサイズを考慮に入れるには、キューとチャネルのどちらにおいても **MAXMSGL** 属性の値を変更する必要があります。

関連資料
サーバー間メッセージ・チャネル・インターセプトの構成例

エラーの処理

IBM MQ 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 - Advanced Message Security (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。

AMS は SYSTEM.PROTECTION.ERROR.QUEUE 。 IBM MQ AMS によって SYSTEM.PROTECTION.ERROR.QUEUE の前には MQDLH ヘッダーがあります。

IBM MQ 管理者は、 SYSTEM.PROTECTION.ERROR.QUEUE 。

z/OS V 9.1.3 IBM MQ 9.1.3 以降、 IBM MQ for z/OS で、サーバー間メッセージ・チャネル・エージェント (MCA) インターセプトが使用されている場合:

- 前述の理由で IBM MQ AMS が伝送キューのメッセージをエラー処理キューに移動させると、送信側の MCA は単純に伝送キューで次に使用可能になるメッセージを処理します。
- 通常は、既存のチャネル・ルールが以下の処理に適用されます。
  - メッセージを送達不能キューに書き込み処理
  - 送達不能キューへの書き込みが失敗した場合に実行されるアクション

具体的なシナリオについては、567 ページの『z/OS 上の AMS の未配信メッセージ』を参照してください。

z/OS V 9.1.3 *z/OS 上の AMS の未配信メッセージ*

IBM MQ for z/OS のサーバー間メッセージ・チャネル・エージェント・インターセプトに関連した具体的なシナリオ

IBM MQ 9.1.3 以降、 IBM MQ for z/OS で、サーバー間メッセージ・チャネル・エージェント (MCA) インターセプトが使用されている場合:

- 送信側 MCA がメッセージを受け取り無保護にした後に、チャネルに対してメッセージが大きすぎるなどの理由でメッセージを送信できなかった場合、USEDLQ 送信側チャネル属性が YES に設定されていると、送信側 MCA はそのメッセージをローカル送達不能キュー (DLQ) に移動します。

  SYSTEM.DEAD.LETTER.QUEUE をローカル DLQ として使用していると、そのメッセージは無保護状態になります。

  注 : IBM MQ AMS は、システム・キューに書き込むメッセージの保護をサポートしていません。

  名前付きの DLQ をローカル DLQ として使用している場合、その DLQ と同じ名前の IBM MQ AMS ポリシーが定義されていれば、メッセージは保護状態になり、適切なポリシーが定義されていなければ、無保護状態になります。
- 何かの理由でメッセージをローカル DLQ に書き込めない場合、チャネルの NPMSPEED が NORMAL に設定されているか、そのメッセージが持続メッセージであれば、現在のメッセージ・バッチがバックアウトされ、チャネルが RETRY 状態になります。 そうでなければ、そのメッセージは破棄され、送信側 MCA が伝送キューにある次のメッセージの処理に進みます。
- セキュリティー・ポリシーが SYSTEM.DEAD.LETTER.QUEUE や 638 ページの『AMS でのシステム・キューの保護』 に挙げられている他の SYSTEM キューに影響を及ぼさない場合、SYSTEM.DEAD.LETTER.QUEUE が使用中になっていると、MCA によってそのキューに書き込まれるメッセージは、現状のままの状態になります。 つまり、保護状態だったメッセージは保護状態のままになり、そうでないメッセージは無保護状態のままになります。

  キュー・マネージャーの DEADQ 属性が代替 (非システム) 送達不能キューの名前に設定されていて、同じ名前の AMS ポリシーが存在しない場合は、MCA によってそのキューに書き込まれるメッセージが現状のままの状態になります。 つまり、保護状態だったメッセージは保護状態のままになり、そうでないメッセージは無保護状態のままになります。

  キュー・マネージャーの DEADQ 属性が代替 (非システム) 送達不能キューの名前に設定されていて、DLQ と同じ名前の AMS ポリシーが存在する場合は、MCA によってそのキューに書き込まれるメッセージがそのポリシーによって保護されます。 メッセージがすでに保護されていれば、再び保護されることはありり

ません。保護の重複を避けるためです。 同じ名前の AMS ポリシーが存在しなければ、メッセージは現状のままの状態になります。

- setmqspl コマンドで許容オプションがオフに設定されていた (**-t** O) ポリシーが DLQ に存在する場合、メッセージが AMS で保護されていなければ (このため、PDMQ ヘッダーがなければ)、DLQ への書き込みは失敗します。 そのようになるのは、メッセージが PDMQ ヘッダーなしで受信側に届いた場合です。 つまり、メッセージの書き込み元に宛先のポリシーがなく、受信側で SPLPROT(ASPOLICY) が設定されていない場合です。
- DLQ に定義されている AMS ポリシーで、メッセージの保護のためにチャネル・イニシエーターの実行に使用されているユーザー ID が許可されていない場合、MCA による DLQ へのメッセージの書き込みが失敗する可能性があります。
- 受信側チャネルは通常、未配布メッセージをローカル DLQ に書き込みます。一方、送信側チャネルは通常、キューに対してメッセージが大きすぎる、無効な MQXQH ヘッダーになっている、といった理由で処理できないメッセージをローカル DLQ に書き込みます。
- DLQ ハンドラーは通常、DLQ ヘッダー (DLH) だけを確認し、メッセージ・ペイロード自体は確認しません。 したがって、メッセージ・ペイロードが保護されていたとしても、メッセージが DLQ に書き込まれた理由をハンドラーが判別することの障害にはなりません。
- DLQ が定義されていない場合、チャネルは以下のようにします。
  - 持続メッセージを送達できない場合は、異常終了します (再試行状態になります)。
  - 非持続未配布メッセージは破棄して、実行処理を続けます。

関連概念

566 ページの『エラーの処理』
IBM MQ 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。

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 *AMS プラットフォーム上の Windows のクイック・スタート・ガイド*

このガイドを使用して、 Windows 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 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 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 IBM 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 (IBM MQ 環境の設定)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 は、標準の IBM MQ インターフェースを介してメッセージが IBM MQ インフラストラクチャーに入る時点で、インターセプターを使用してメッセージに対して署名および暗号化を行います。 基本的なセットアップは IBM MQ で行い、以下のステップで構成されます。

IBM MQ Explorer 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、 C:¥Program Files¥IBM¥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
```

重要 : IBM MQ は、 SYSTEM.PROTECTION.POLICY.QUEUE をキューに入れます。

IBM MQ は使用可能なすべてのポリシーを必ずしもキャッシュに入れるわけではありません。 ポリシー数が多い場合、IBM MQ は限られた数のポリシーをキャッシュします。 そのため、キュ

ー・マネージャーに含まれる定義済みポリシー数が少ない場合には、SYSTEM.PROTECTION.POLICY.QUEUE に対する参照オプションを提供する必要はありません。

しかし、定義されているポリシー数が多い場合や古いクライアントを使用している場合には、このキューに対する参照権限を付与してください。 SYSTEM.PROTECTION.ERROR.QUEUE は、AMS コードが生成するエラー・メッセージを入れるときに使用されます。 このキューに対する書き込み権限がチェックされるのは、このキューにエラー・メッセージを書き込もうとする場合のみです。 AMS 保護キューに対するメッセージの書き込みや取得を行うときには、このキューに対する書き込み権限はチェックされません。

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、セクション 573 ページの『7. セットアップのテスト』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

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した Java アプリケーションを使用する予定の場合、JRE の一部である **keytool** コマンドを使用して、JKS 鍵ストアおよび証明書を作成する必要があります (詳しくは、591 ページの『Java クライアントを使用する AMS のクイック・スタート・ガイド』を参照)。 その他すべての言語、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、このガイドに示されているステップで問題ありません。

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 インターセプターは、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、keystore.conf フォルダー内に別個の .mqs ファイルを持っている必要があります。 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、以下の形式にする必要があります。

```
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 」(末尾にスペースがある) はそれぞれ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。
- CMS (暗号メッセージ構文)、JKS (Java 鍵ストア)、および JCEKS (Java 暗号拡張鍵ストア) という鍵ストア・フォーマットがあります。 詳細については、604 ページの『AMS 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 (eg. 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 にデフォルト以外のロケーションを使用する方法については、603 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/bob/AMS/bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm

runmqakm -cert -add -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd
-label Bob_Cert -file bob_public.arm
```

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 フラグを使用します。 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**」 を選択して、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 *AMS 上の UNIX 用クイック・スタート・ガイド*

このガイドを使用して、 UNIX でメッセージ・セキュリティーを提供するように 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。

- Runtime
- サーバー
- サンプル・プログラム
- IBM Global Security Kit
- 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 Linux システム用の IBM MQ コンポーネント
- AIX AIX システム用の IBM MQ コンポーネント
- Solaris Solaris システム用の IBM 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 は、標準の IBM MQ インターフェースを介してメッセージが IBM MQ インフラストラクチャーに入る時点で、インターセプターを使用してメッセージに対して署名および暗号化を行います。 基本的なセットアップは IBM MQ で行い、以下のステップで構成されます。

IBM 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_INSTALLATION_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
```

重要 : IBM MQ は、 SYSTEM.PROTECTION.POLICY.QUEUE をキューに入れます。

IBM MQ は使用可能なすべてのポリシーを必ずしもキャッシュに入れるわけではありません。 ポリシー数が多い場合、IBM MQ は限られた数のポリシーをキャッシュします。 そのため、キュ

ー・マネージャーに含まれる定義済みポリシー数が少ない場合には、SYSTEM.PROTECTION.POLICY.QUEUE に対する参照オプションを提供する必要はありません。

しかし、定義されているポリシー数が多い場合や古いクライアントを使用している場合には、このキューに対する参照権限を付与してください。 SYSTEM.PROTECTION.ERROR.QUEUE は、AMS コードが生成するエラー・メッセージを入れるときに使用されます。 このキューに対する書き込み権限がチェックされるのは、このキューにエラー・メッセージを書き込もうとする場合のみです。 AMS 保護キューに対するメッセージの書き込みや取得を行うときには、このキューに対する書き込み権限はチェックされません。

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、セクション 580 ページの『8. 暗号化のテスト』 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した Java アプリケーションを使用する予定の場合、JRE の一部である **keytool** コマンドを使用して、JKS 鍵ストアおよび証明書を作成する必要があります (詳しくは、591 ページの『Java クライアントを使用する AMS のクイック・スタート・ガイド』 を参照)。 その他すべての言語、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、このガイドに示されているステップで問題ありません。

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、keystore.conf フォルダー内に別個の .mqs ファイルを持っている必要があります。 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、以下の形式にする必要があります。

```
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。
- CMS (暗号メッセージ構文)、JKS (Java 鍵ストア)、および JCEKS (Java 暗号拡張鍵ストア) という鍵ストア・フォーマットがあります。 詳細については、604 ページの『AMS 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 にデフォルト以外のロケーションを使用する方法については、603 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 フラグを使用します。 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。

z/OS *z/OS での構成例*
このセクションでは、 z/OS での Advanced Message Security キューイング・シナリオのポリシーと証明書の構成例を示します。

Advanced Message Security の構成方法について詳しくは、Advanced Message Security for z/OS の構成 を参照してください。

この例では、必要な Advanced Message Security ポリシー、およびユーザーと鍵リングに関連して存在する必要があるデジタル証明書について扱います。 この例は、Advanced Message Security のためのリソース・アクセス権をユーザーに付与するの説明に従って、シナリオに登場するユーザーがセットアップ済みであることが前提となっています。

V 9.1.3 なお、IBM MQ 9.1.3 以降については、「サーバー間メッセージ・チャネル・インターセプトの例」 を参照してください。

z/OS z/OS での保全性保護されたメッセージのローカル・キューイング

この例では、書き込みと取り出しを行うアプリケーションから見てローカル側にあるキューとの間で、保全性保護されたメッセージを送信/取得するのに必要な Advanced Message Security ポリシーおよび証明書について詳しく説明します。

キュー・マネージャーおよびキューの例を、以下に示します。

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

以下のユーザーが使用されます。

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

ユーザー証明書の作成

この例では、1 つだけユーザー証明書が必要です。 これは、保全性保護されたメッセージに署名するために必要となる送信側ユーザーの証明書です。 送信側ユーザーは「TELLER5」です。

認証局 (CA) 証明書も必要です。 CA 証明書は、ユーザーの証明書を発行した認証局の証明書です。 これは、証明書のチェーンになる場合があります。 その場合、Advanced Message Security タスク・ユーザー (この場合、ユーザー WMQBNK6) の鍵リングに、チェーン内のすべての証明書が含まれている必要があります。

CA 証明書は、RACF RACDCERT コマンドを使用して作成できます。 この証明書は、ユーザー証明書を発行するために使用されます。 以下に例を示します。

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

この RACDCERT コマンドは、ユーザー「TELLER5」のユーザー証明書を発行するために使用できる CA 証明書を作成します。 以下に例を示します。

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

インストール済み環境には、CA 証明書の選択または作成のためのプロシージャー、および証明書の発行のためとそれらを関係するシステムに配布するためのプロシージャーが含まれます。

それらの証明書をエクスポートおよびインポートする際に、Advanced Message Security は以下を必要とします。

- CA 証明書 (チェーン)。
- ユーザー証明書とその秘密鍵。

RACF を使用している場合、RACDCERT EXPORT コマンドを使用して証明書をデータ・セットにエクスポートすることができます。また、RACDCERT ADD コマンドを使用してデータ・セットから証明書をインポートすることができます。 これらのコマンドおよびその他の RACDCERT コマンドについて詳しくは、「*z/OS: Security Server RACF* コマンド言語解説書」を参照してください。

この場合の証明書は、キュー・マネージャー BNK6 を実行する z/OS システム上に配置する必要があります。

BNK6 を実行する z/OS システム上に証明書がインポートされたら、ユーザー証明書には TRUST 属性が必要になります。 証明書に TRUST 属性を追加するときには、RACDCERT ALTER コマンドを使用できます。以下に例を示します。

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

この例では、受信側ユーザーに必要な証明書はありません。

関連する鍵リングへの証明書の接続

必要な証明書を作成するかインポートし、信頼済みとして設定したら、それらの証明書は BNK6 を実行する z/OS システム上の適切なユーザー鍵リングに接続する必要があります。 鍵リングを作成するには、RACDCERT ADDRING コマンドを以下のように使用します。

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

これによって、Advanced Message Security タスク・ユーザー (WMQBNK6) の鍵リング、および送信側ユーザー「TELLER5」の鍵リングが作成されます。 鍵リング名 drq.ams.keyring は必須であり、この名前には大/小文字の区別があります。

鍵リングが作成されたら、関連する証明書を接続することができます。

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

送信側ユーザー証明書は DEFAULT として接続する必要があります。 送信側ユーザーの drq.ams.keyring に複数の証明書が存在する場合、デフォルトの証明書が署名に使用されます。

証明書の作成と変更は、キュー・マネージャーを停止して再始動するか、 z/OS **MODIFY** コマンドを使用して Advanced Message Security 証明書構成をリフレッシュするまで、 Advanced Message Security によって認識されません。 以下に例を示します。

```
F BNK6AMSM,REFRESH KEYRING
```

Advanced Message Security ポリシーの作成

この例では、保全性保護されたメッセージは、ユーザー「TELLER5」として実行されるアプリケーションによってキュー FIN.XFER.Q7 に書き込まれ、ユーザー「FINADM2」として実行するアプリケーションによって同じキューから取り出されます。そのため、ただ 1 つの Advanced Message Security ポリシーが必要になります。

Advanced Message Security ポリシーは、CSQ0UTIL ユーティリティーを使用して作成します。このユーティリティーについては、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) を参照してください。

CSQ0UTIL ユーティリティーを使用して次のコマンドを実行します。

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

このポリシーでは、キュー・マネージャーは BNK6 として識別されます。 ポリシー名および関連付けられたキューは FIN.XFER.Q7 です。 送信側の署名を生成するために使用されるアルゴリズムは MD5 で、送信側ユーザーの識別名 (DN) は「CN=Teller5,O=BCO,C=US」です。

ポリシーを定義した後、BNK6 キュー・マネージャーを再始動するか、または z/OS **MODIFY** コマンドを使用して Advanced Message Security ポリシー構成をリフレッシュします。 以下に例を示します。

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS でのプライバシー保護されたメッセージのローカル・キューイング*

この例では、書き込みと取り出しを行うアプリケーションから見てローカル側にあるキューとの間で、プライバシー保護されたメッセージを送信/取得するのに必要な Advanced Message Security ポリシーおよび証明書について詳しく説明します。 プライバシー保護されたメッセージは、署名され、かつ暗号化されます。

キュー・マネージャーおよびローカル・キューの例を、以下に示します。

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

以下のユーザーが使用されます。

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

このシナリオを構成する手順は、以下のとおりです。

ユーザー証明書の作成

この例では、2 つのユーザー証明書が必要です。 1 つはメッセージに署名するために必要な送信側ユーザーの証明書で、もう 1 つはメッセージ・データを暗号化および暗号化解除するために必要な受信側ユーザーの証明書です。 送信側ユーザーは「TELLER5」で、受信側ユーザーは「FINADM2」です。

認証局 (CA) 証明書も必要です。CA 証明書は、ユーザーの証明書を発行した認証局の証明書です。 これは、証明書のチェーンになる場合があります。 その場合、Advanced Message Security タスク・ユーザー (この場合、ユーザー WMQBNK6) の鍵リングに、チェーン内のすべての証明書が含まれている必要があります。

CA 証明書は、RACF RACDCERT コマンドを使用して作成できます。 この証明書は、ユーザー証明書を発行するために使用されます。 以下に例を示します。

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

この RACDCERT コマンドは、ユーザー「TELLER5」と「FINADM2」のユーザー証明書を発行するために使用できる CA 証明書を作成します。 以下に例を示します。

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

インストール済み環境には、CA 証明書の選択または作成のためのプロシージャー、および証明書の発行のためとそれらを関係するシステムに配布するためのプロシージャーが含まれます。

それらの証明書をエクスポートおよびインポートする際に、Advanced Message Security は以下を必要とします。

- CA 証明書 (チェーン)。
- 送信側ユーザー証明書とその秘密鍵。

- 受信側ユーザー証明書とその秘密鍵。

RACF を使用している場合、RACDCERT EXPORT コマンドを使用して証明書をデータ・セットにエクスポートすることができます。また、RACDCERT ADD コマンドを使用してデータ・セットから証明書をインポートすることができます。 これらのコマンドおよびその他の RACDCERT コマンドについて詳しくは、「*z/OS: Security Server RACF* コマンド言語解説書」の「RACDCERT (RACF デジタル証明書の管理)」を参照してください。

この場合の証明書は、キュー・マネージャー BNK6 を実行する z/OS システム上に配置する必要があります。

BNK6 を実行する z/OS システム上に証明書がインポートされたら、ユーザー証明書には TRUST 属性が必要になります。 証明書に TRUST 属性を追加するときには、RACDCERT ALTER コマンドを使用できます。以下に例を示します。

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

関連する鍵リングへの証明書の接続

必要な証明書を作成するかインポートし、信頼済みとして設定したら、それらの証明書は BNK6 を実行する z/OS システム上の適切なユーザー鍵リングに接続する必要があります。 鍵リングを作成するには、RACDCERT ADDRING コマンドを以下のように使用します。

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

これによって、Advanced Message Security タスク・ユーザーの鍵リング、および送信側と受信側のユーザーの鍵リングが作成されます。 鍵リング名 drq.ams.keyring は必須であり、この名前には大/小文字の区別があります。

鍵リングが作成されたら、関連する証明書を接続することができます。

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))

RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

送信側および受信側のユーザー証明書は DEFAULT として接続する必要があります。 いずれかのユーザーの drq.ams.keyring に複数の証明書が存在する場合、デフォルトの証明書が署名および暗号化解除に使用されます。

受信側ユーザーの証明書は、Advanced Message Security タスク・ユーザーの鍵リングに USAGE(SITE) を使用して接続する必要もあります。 これは、メッセージ・データを暗号化する際に、Advanced Message Security のタスクが受信側の公開鍵を必要とするためです。 USAGE(SITE) により、秘密鍵は鍵リング内でアクセスされることを防ぎます。

証明書の作成と変更は、キュー・マネージャーを停止して再始動するか、 z/OS **MODIFY** コマンドを使用して Advanced Message Security 証明書構成をリフレッシュするまで、 Advanced Message Security によって認識されません。 以下に例を示します。

```
F BNK6AMSM,REFRESH KEYRING
```

Advanced Message Security ポリシーの作成

この例では、プライバシー保護されたメッセージは、ユーザー「TELLER5」として実行されるアプリケーションによってキュー FIN.XFER.Q8 に書き込まれ、ユーザー「FINADM2」として実行するアプリケーションによって同じキューから取り出されます。そのため、ただ 1 つの Advanced Message Security ポリシーが必要になります。

Advanced Message Security ポリシーは、CSQ0UTIL ユーティリティーを使用して作成します。このユーティリティーについては、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) を参照してください。

CSQ0UTIL ユーティリティーを使用して次のコマンドを実行します。

```
setmqspl -m BNK6 -p FIN.XFER.Q8 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

このポリシーでは、キュー・マネージャーは BNK6 として識別されます。 ポリシー名および関連付けられたキューは FIN.XFER.Q8 です。 送信側の署名を生成するために使用されるアルゴリズムは SHA1、送信側ユーザーの識別名 (DN) は「CN=Teller5,O=BCO,C=US」、受信側ユーザーは「CN=FinAdm2,O=BCO,C=US」です。 メッセージ・データを暗号化するために使用されるアルゴリズムは 3DES です。

ポリシーを定義した後、BNK6 キュー・マネージャーを再始動するか、または z/OS **MODIFY** コマンドを使用して Advanced Message Security ポリシー構成をリフレッシュします。 以下に例を示します。

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS での保全性保護されたメッセージのリモート・キューイング*

この例では、2 つの異なるキュー・マネージャーによって管理されたキューとの間で、保全性保護されたメッセージを送信/取得するのに必要な Advanced Message Security ポリシーおよび証明書について詳しく説明します。 2 つのキュー・マネージャーは、同じ z/OS システム上で実行することも、異なる z/OS システム上で実行することも可能です。 さらに、1 つのキュー・マネージャーを Advanced Message Security を実行する分散システムに配置することもできます。

キュー・マネージャーおよびキューの例を、以下に示します。

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

注: この例では、BNK6 と BNK7 は、異なる z/OS システム上で実行されるキュー・マネージャーです。

以下のユーザーが使用されます。

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

このシナリオを構成する手順は、以下のとおりです。

ユーザー証明書の作成

この例では、1 つだけユーザー証明書が必要です。 これは、保全性保護されたメッセージに署名するために必要となる送信側ユーザーの証明書です。 送信側ユーザーは「TELLER5」です。

認証局 (CA) 証明書も必要です。CA 証明書は、ユーザーの証明書を発行した認証局の証明書です。 これは、証明書のチェーンになる場合があります。 その場合、Advanced Message Security タスク・ユーザー (この場合、ユーザー WMQBNK7) の鍵リングに、チェーン内のすべての証明書が含まれている必要があります。

CA 証明書は、RACF RACDCERT コマンドを使用して作成できます。 この証明書は、ユーザー証明書を発行するために使用されます。 以下に例を示します。

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

この RACDCERT コマンドは、ユーザー「TELLER5」のユーザー証明書を発行するために使用できる CA 証明書を作成します。 以下に例を示します。

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

インストール済み環境には、CA 証明書の選択または作成のためのプロシージャー、および証明書の発行のためとそれらを関係するシステムに配布するためのプロシージャーが含まれます。

それらの証明書をエクスポートおよびインポートする際に、Advanced Message Security は以下を必要とします。

- CA 証明書 (チェーン)。
- 送信側ユーザー証明書とその秘密鍵。

RACF を使用している場合、RACDCERT EXPORT コマンドを使用して証明書をデータ・セットにエクスポートすることができます。また、RACDCERT ADD コマンドを使用してデータ・セットから証明書をインポートすることができます。 これらのコマンドおよびその他の RACDCERT コマンドについて詳しくは、「*z/OS: Security Server RACF* コマンド言語解説書」の「RACDCERT (RACF デジタル証明書の管理)」を参照してください。

この場合の証明書は、キュー・マネージャー BNK6 と BNK7 を実行する z/OS システム上に配置する必要があります。

この例では、送信側の証明書は BNK6 を実行する z/OS システムにインポートする必要があります。また、CA 証明書は BNK7 を実行する z/OS システムにインポートする必要があります。 証明書がインポートされたら、ユーザー証明書には TRUST 属性が必要になります。 証明書に TRUST 属性を追加するときには、RACDCERT ALTER コマンドを使用できます。 例えば、BNK6 の場合、以下を使用します。

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

関連する鍵リングへの証明書の接続

必要な証明書を作成するかインポートし、信頼済みとして設定したら、それらの証明書は BNK6 と BNK7 を実行する z/OS システム上の適切なユーザー鍵リングに接続する必要があります。

鍵リングを作成するには、RACDCERT ADDRING コマンドを BNK6 で以下のように使用します。

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

これによって、BNK6 の送信側ユーザーの鍵リングが作成されます。 鍵リング名 drq.ams.keyring は必須であり、この名前には大/小文字の区別があります。

BNK7 で次を実行します。

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

これによって、BNK7 の Advanced Message Security タスク・ユーザーの鍵リングが作成されます。 BNK7 の「TELLER5」には、ユーザーの鍵リングは必要ありません。

鍵リングが作成されたら、関連する証明書を接続することができます。

BNK6 で次を実行します。

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

BNK7 で次を実行します。

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

送信側ユーザー証明書は DEFAULT として接続する必要があります。 送信側ユーザーの drq.ams.keyring に複数の証明書が存在する場合、デフォルトの証明書が署名に使用されます。

証明書の作成と変更は、キュー・マネージャーを停止して再始動するか、 z/OS **MODIFY** コマンドを使用して Advanced Message Security 証明書構成をリフレッシュするまで、 Advanced Message Security によって認識されません。 以下に例を示します。

BNK6 で次を実行します。

```
F BNK6AMSM,REFRESH,KEYRING
```

BNK7 で次を実行します。

```
F BNK7AMSM,REFRESH,KEYRING
```

Advanced Message Security ポリシーの作成

この例では、保全性保護されたメッセージは、ユーザー「TELLER5」として実行されるアプリケーションによって BNK6 上のリモート・キュー FIN.XFER.Q7 に書き込まれ、ユーザー「FINADM2」として実行されるアプリケーションによって BNK7 上のローカル・キュー FIN.RCPT.Q7 から取り出されます。そのため、2 つの Advanced Message Security ポリシーが必要になります。

Advanced Message Security ポリシーは、CSQ0UTIL ユーティリティーを使用して作成します。このユーティリティーについては、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) を参照してください。

BNK6 上のリモート・キューの保全性ポリシーを定義するため、CSQ0UTIL ユーティリティーを使用して以下のコマンドを実行します。

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

このポリシーでは、キュー・マネージャーは BNK6 として識別されます。 ポリシー名および関連付けられたキューは FIN.XFER.Q7 です。 送信側の署名を生成するために使用されるアルゴリズムは MD5 で、送信側ユーザーの識別名 (DN) は「CN=Teller5,O=BCO,C=US」です。

また、BNK7 上のローカル・キューの保全性ポリシーを定義するため、CSQ0UTIL ユーティリティーを使用して以下のコマンドを実行します。

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

このポリシーでは、キュー・マネージャーは BNK7 として識別されます。 ポリシー名および関連付けられたキューは FIN.RCPT.Q7 です。 送信側の署名に期待されるアルゴリズムは MD5 で、送信側ユーザーの識別名 (DN) は「CN=Teller5,O=BCO,C=US」が期待されます。

2 つのポリシーを定義した後、BNK6 と BNK7 のキュー・マネージャーを再始動するか、または z/OS **MODIFY** コマンドを使用して Advanced Message Security ポリシー構成をリフレッシュします。 以下に例を示します。

BNK6 で次を実行します。

```
F BNK6AMSM,REFRESH,POLICY
```

BNK7 で次を実行します。

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *z/OS でのプライバシー保護されたメッセージのリモート・キューイング*

この例では、2 つの異なるキュー・マネージャーによって管理されたキューとの間で、プライバシー保護されたメッセージを送信/取得するのに必要な Advanced Message Security ポリシーおよび証明書について詳しく説明します。 2 つのキュー・マネージャーは、同じ z/OS システム上で実行することも、異なる z/OS システム上で実行することも可能です。 さらに、1 つのキュー・マネージャーを Advanced Message Security を実行する分散システムに配置することもできます。

キュー・マネージャーおよびキューの例を、以下に示します。

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

注: この例では、BNK6 と BNK7 は、同じ名前の異なる z/OS システム上で実行されるキュー・マネージャーです。

以下のユーザーが使用されます。

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

このシナリオを構成する手順は、以下のとおりです。

ユーザー証明書の作成

この例では、2 つのユーザー証明書が必要です。 1 つはメッセージに署名するために必要な送信側ユーザーの証明書で、もう 1 つはメッセージ・データを暗号化および暗号化解除するために必要な受信側ユーザーの証明書です。 送信側ユーザーは「TELLER5」で、受信側ユーザーは「FINADM2」です。

認証局 (CA) 証明書も必要です。CA 証明書は、ユーザーの証明書を発行した認証局の証明書です。 これは、証明書のチェーンになる場合があります。 その場合、Advanced Message Security タスク・ユーザー (この場合、ユーザー WMQBNK7) の鍵リングに、チェーン内のすべての証明書が含まれている必要があります。

CA 証明書は、RACF RACDCERT コマンドを使用して作成できます。 この証明書は、ユーザー証明書を発行するために使用されます。 以下に例を示します。

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

この RACDCERT コマンドは、ユーザー「TELLER5」と「FINADM2」のユーザー証明書を発行するために使用できる CA 証明書を作成します。 以下に例を示します。

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

インストール済み環境には、CA 証明書の選択または作成のためのプロシージャー、および証明書の発行のためとそれらを関係するシステムに配布するためのプロシージャーが含まれます。

それらの証明書をエクスポートおよびインポートする際に、Advanced Message Security は以下を必要とします。

- CA 証明書 (チェーン)。
- 送信側ユーザー証明書とその秘密鍵。
- 受信側ユーザー証明書とその秘密鍵。

RACF を使用している場合、RACDCERT EXPORT コマンドを使用して証明書をデータ・セットにエクスポートすることができます。また、RACDCERT ADD コマンドを使用してデータ・セットから証明書をインポートすることができます。

これらのコマンドおよびその他の RACDCERT コマンドについて詳しくは、「*z/OS: Security Server RACF* コマンド言語解説書」の「RACDCERT (RACF デジタル証明書の管理)」を参照してください。

この場合の証明書は、キュー・マネージャー BNK6 と BNK7 を実行する z/OS システム上に配置する必要があります。

この例では、送信側および受信側の証明書は BNK6 を実行する z/OS システムにインポートする必要があります。また、CA および受信側の証明書は BNK7 を実行する z/OS システムにインポートする必要があります。 証明書がインポートされたら、ユーザー証明書には TRUST 属性が必要になります。 証明書に TRUST 属性を追加するときには、RACDCERT ALTER コマンドを使用できます。 以下に例を示します。

BNK6 で次を実行します。

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

BNK7 で次を実行します。

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

関連する鍵リングへの証明書の接続

必要な証明書を作成するかインポートし、信頼済みとして設定したら、それらの証明書は BNK6 と BNK7 を実行する z/OS システム上の適切なユーザー鍵リングに接続する必要があります。

鍵リングを作成するには、RACDCERT ADDRING コマンドを以下のように使用します。

BNK6 で次を実行します。

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

これによって、BNK6 の Advanced Message Security タスク・ユーザーの鍵リングおよび送信側ユーザーの鍵リングが作成されます。 鍵リング名 drq.ams.keyring は必須であり、この名前には大/小文字の区別があります。

BNK7 で次を実行します。

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

このコマンドによって、BNK7 の Advanced Message Security タスク・ユーザーの鍵リングおよび受信側ユーザーの鍵リングが作成されます。

鍵リングが作成されたら、関連する証明書を接続することができます。

BNK6 で次を実行します。

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

BNK7 で次を実行します。

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

送信側および受信側のユーザー証明書は DEFAULT として接続する必要があります。 いずれかのユーザーの drq.ams.keyring に複数の証明書が存在する場合、デフォルトの証明書が署名および暗号化/暗号化解除に使用されます。

BNK6 では、受信側ユーザーの証明書は、Advanced Message Security タスク・ユーザーの鍵リングに USAGE(SITE) を使用して接続する必要もあります。 これは、メッセージ・データを暗号化する際に、Advanced Message Security のタスクが受信側の公開鍵を必要とするためです。 USAGE(SITE) により、秘密鍵は鍵リング内でアクセスされることを防ぎます。

証明書の作成と変更は、キュー・マネージャーを停止して再始動するか、 z/OS **MODIFY** コマンドを使用して Advanced Message Security 証明書構成をリフレッシュするまで、 Advanced Message Security によって認識されません。 以下に例を示します。

BNK6 で次を実行します。

```
F BNK6AMSM,REFRESH,KEYRING
```

BNK7 で次を実行します。

```
F BNK7AMSM,REFRESH,KEYRING
```

Advanced Message Security ポリシーの作成

この例では、プライバシー保護されたメッセージは、ユーザー「TELLER5」として実行されるアプリケーションによって BNK6 上のリモート・キュー FIN.XFER.Q7 に書き込まれ、ユーザー「FINADM2」として実行されるアプリケーションによって BNK7 上のローカル・キュー FIN.RCPT.Q7 から取り出されます。そのため、2 つの Advanced Message Security ポリシーが必要になります。

Advanced Message Security ポリシーは、CSQ0UTIL ユーティリティーを使用して作成します。このユーティリティーについては、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) を参照してください。

BNK6 上のリモート・キューのプライバシー・ポリシーを定義するため、CSQ0UTIL ユーティリティーを使用して以下のコマンドを実行します。

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

このポリシーでは、キュー・マネージャーは BNK6 として識別されます。 ポリシー名および関連付けられたキューは FIN.XFER.Q7 です。 送信側の署名を生成するために使用されるアルゴリズムは SHA1、送信側ユーザーの識別名 (DN) は「CN=Teller5,O=BCO,C=US」、受信側ユーザーは「CN=FinAdm2,O=BCO,C=US」です。 メッセージ・データを暗号化するために使用されるアルゴリズムは 3DES です。

また、BNK7 上のローカル・キューのプライバシー・ポリシーを定義するため、CSQ0UTIL ユーティリティーを使用して以下のコマンドを実行します。

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

このポリシーでは、キュー・マネージャーは BNK7 として識別されます。 ポリシー名および関連付けられたキューは FIN.RCPT.Q7 です。 送信側の署名として期待されるアルゴリズムは SHA1 であり、送信側ユーザーの識別名 (DN) には「CN=Teller5,O=BCO,C=US」が期待され、受信側ユーザーは「CN=FinAdm2,O=BCO,C=US」になります。 メッセージ・データを暗号化解除するために使用されるアルゴリズムは 3DES です。

2 つのポリシーを定義した後、BNK6 と BNK7 のキュー・マネージャーを再始動するか、または z/OS **MODIFY** コマンドを使用して Advanced Message Security ポリシー構成をリフレッシュします。 以下に例を示します。

BNK6 で次を実行します。

```
F BNK6AMSM,REFRESH,POLICY
```

BNK7 で次を実行します。

```
F BNK7AMSM,REFRESH,POLICY
```

Java クライアントを使用する *AMS* 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 アプリケーションにメッセージ・セキュリティーを提供するように 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 は、標準の IBM MQ インターフェースを介してメッセージが IBM MQ インフラストラクチャーに入る時点で、インターセプターを使用してメッセージに対して署名および暗号化を行います。 基本的なセットアップは IBM 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、このシナリオで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、いくつかのシステム・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. 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 +browse
```

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
```

重要 : IBM MQ は、 SYSTEM.PROTECTION.POLICY.QUEUE をキューに入れます。

IBM MQ は使用可能なすべてのポリシーを必ずしもキャッシュに入れるわけではありません。 ポリシー数が多い場合、IBM MQ は限られた数のポリシーをキャッシュします。 そのため、キュー・マネージャーに含まれる定義済みポリシー数が少ない場合には、SYSTEM.PROTECTION.POLICY.QUEUE に対する参照オプションを提供する必要はありません。

しかし、定義されているポリシー数が多い場合や古いクライアントを使用している場合には、このキューに対する参照権限を付与してください。 SYSTEM.PROTECTION.ERROR.QUEUE は、AMS コードが生成するエラー・メッセージを入れるときに使用されます。 このキューに対する書き込み権限がチェックされるのは、このキューにエラー・メッセージを書き込もうとする場合

のみです。 AMS 保護キューに対するメッセージの書き込みや取得を行うときには、このキューに対する書き込み権限はチェックされません。

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、595 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、`JmsProducer` および `JmsConsumer` 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 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。 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を作成します。例えば、`/home/alice/.mqs` などです。 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 : このディレクトリーは、以下のステップでは *keystore-dir* と記載されています。
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、プレーン・テキスト形式の情報を保持する keystore.conf ファイルを介して行われます。 各ユーザーには、別々の keystore.conf ファイルが必要です。 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) の説明に従ったために keystore.conf ファイルが既にある場合は、既存のファイルを編集してこれらの行を追加できます。
- 詳しくは、604 ページの『AMS 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 **strmqikm** コマンド (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、IBM 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 : IBM 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 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 IBM 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、IBM 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 *IBM Integration Bus* を使用した経路指定

Advanced Message Security は、IBM Integration Bus、または WebSphere Message Broker 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。 IBM Integration Bus 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、IBM Integration Bus を流れるメッセージを保護するために、IBM Integration Bus にコンテンツ (シナリオ 1) を知らずにメッセージを処理させることができることを意味します。または、メッセージ (シナリオ 2) を受信、送信することができる許可ユーザーにすることもできます。

シナリオ *1 - Integration Bus* がメッセージの内容を認識できない

始める前に

IBM Integration Bus 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、IBM Integration Bus が読み取ることができます。 Advanced Message Security を使用するのは、*alice*、*bob* および *cecil* のみです。 これをインストールしたり、IBM Integration Bus 用に構成したりする必要はありません。

IBM Integration Bus 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 IBM Integration Bus 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)。 IBM Integration Bus 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、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。

```
DEFINE QLOCAL(QIN)
```

[1] cecil's

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. IBM Integration Bus 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 IBM Integration Bus 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、IBM Integration Bus 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。 IBM Integration Bus は、routeTo プロパティーを読み取るメッセージのルーティング先を決定します。このプロパティーは、すべてのプロパティーとして、暗号化されていません。 IBM Integration Bus は、メッセージを適切な無保護別名に置き、それ以上の保護を回避します。 *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 - Integration Bus* がメッセージの内容を認識できる

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 IBM Integration Bus 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、IBM Integration Bus 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 IBM Integration Bus の間の伝送は、傍受できません。

IBM Integration Bus 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
```

IBM Integration Bus 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ている許可されたパーティーであると見なされる場合は、 IBM Integration Bus 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、IBM Integration Bus 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 IBM Integration Bus 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 IBM Integration Bus 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 IBM Integration Bus 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. IBM Integration Bus 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、IBM Integration Bus サービス・ユーザーの証明書を必要とします。 IBM Integration Bus 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。 IBM Integration Bus 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 IBM Integration Bus から送信されたことを検証するために、IBM Integration Bus サービス・ユーザーの証明書を必要とします。
4. IN という名前のローカル・キューを定義し、*alice* と *bob* を作成者として、また IBM Integration Bus 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. OUT という名前のローカル・キューを定義し、IBM Integration Bus 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. IBM Integration Bus 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 IBM Integration Bus 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、IBM Integration Bus のみがそのメッセージを読み取ることができます。 IBM Integration Bus は、*alice* と *bob* からのメッセージのみを受け入れ、その他のメッセージはすべて拒否します。 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、IBM Integration Bus によって署名されていないメッセージは拒否されます。

Advanced Message Security と *Managed File Transfer* の使用

このシナリオでは、 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 Managed File Transfer によって使用されるキューをホストする IBM MQ インストール済み環境に Advanced Message Security 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ます。

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 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 IBM 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、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、 シナリオの概要のシナリオで説明されているように、2 つの Managed File Transfer キューと 2 つのエージェント AGENT1 および AGENT2 が単一のキュー・マネージャーを共有する 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、単純なモデルを使用しており、グループ FTAGENTS のユーザー ftagent を使用して Managed File Transfer Agent プロセスを実行します。 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:

- 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。
- 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、591 ページの『Java クライアントを使用する AMS 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=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. `keystore.conf` ファイルを作成して、鍵ストアのロケーションとその中にある証明書を、該当する「クイック・スタート・ガイド」の詳細として識別します。

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 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=<organisation>, L=<location>, ST=<state>, C=<country>"
   -e AES128 -r "CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>"
   ```

3. Managed File Transfer Agent 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、 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。

Advanced Message Security のインストールの概要

各種プラットフォームに 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

インストール手順については、Advanced Message Security のインストール (Multiplatforms) と Advanced Message Security のインストール (z/OS) を参照してください。

関連タスク

Advanced Message Security のアンインストール

z/OS での監査

Advanced Message Security (AMS) for z/OS は、ポリシー保護キューに対してアプリケーションが行う操作へのオプションの監査手段を提供します。 有効にすると、ポリシー保護キューへのそれらの操作の成功または失敗に対して、IBM システム管理機能 (SMF) 監査レコードが生成されます。 監査対象の操作には、MQPUT、MQPUT1、および MQGET があります。

監査はデフォルトでは無効になっていますが、AMS アドレス・スペースの構成済み Language Environment® _CEE_ENVFILE ファイルで _AMS_SMF_TYPE および _AMS_SMF_AUDIT を構成することにより、監査をアクティブにすることができます。 詳しくは、Advanced Message Security 用のプロシージャーを作成するを参照してください。 _AMS_SMF_TYPE 変数は SMF レコード・タイプの指定に使用される 128 から 255 までの数値です。 通常、SMF レコード・タイプは 180 ですが、必須ではありません。 値 0 を指定すると、監査は無効化されます。 _AMS_SMF_AUDIT 変数は、成功した操作、失敗した操作、またはその両方の操作に対して監査レコードが作成されるかどうかを構成します。 監査オプションは、オペレーター・コマンドを使用して、AMS がアクティブであっても動的に変更することができます。 詳しくは、Advanced Message Security の操作を参照してください。

SMF レコードはサブタイプを使用して定義されます。サブタイプ 1 は一般監査イベントです。 SMF レコードには処理中の要求に関連するすべてのデータが格納されます。

SMF レコードは CSQ0KSMF マクロ (マクロ名にはゼロが含まれています) を使用してマップされます。これはターゲット・ライブラリー SCSQMACS で提供されています。 SMF データのデータ削減プログラムを記述している場合、このマッピング・マクロを組み込むと、SMF 後処理ルーチンの開発とカスタマイズで役立ちます。

Advanced Message Security z/OS によって作成される SMF レコードでは、データはセクションに編成されます。 レコードは以下の部分から構成されます。

- 標準 SMF ヘッダー
- Advanced Message Security for z/OS によって定義されたヘッダー拡張
- 製品セクション
- データ・セクション

SMF レコードの製品セクションは、Advanced Message Security for z/OS が生成するレコードでは必ず存在します。 データ・セクションはサブタイプに基づいて異なります。 現在、1 つのサブタイプが定義されているので、単一のデータ・セクションが使用されます。

SMF の詳細については、「z/OS MVS システム管理機能 (SMF)」(SA88-8596) のマニュアルを参照してください。 有効なレコード・タイプは、システム PARMLIB データ・セットの SMFPRMxx メンバーに記述されます。 詳細については SMF の資料を参照してください。

Advanced Message Security 監査レポート生成プログラム (CSQ0USMF)

Advanced Message Security for z/OS には、CSQ0USMF という監査レポート生成ツールが提供されています。 これは、インストール・ライブラリー SCSQAUTH にあります。 CSQ0USMF ユーティリティーを実行する CSQ40RSM というサンプル JCL が、インストール・ライブラリー SCSQPROC にあります。

CSQ0USMF ユーティリティーを実行する前に、SMF タイプ 180 レコードをシステム SMF データ・セットから順次データ・セットにダンプする必要があります。 例として、この JCL は SMF タイプ 180 レコードを SMF データ・セットからダンプし、それをターゲット・データ・セットに転送します。

```
//IFAUDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYSn.MANn.syst,DISP=SHR
//OUTDD1 DD DSN=your.target.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
```

```
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(180))
/*
```

インストール済み環境で使用される実際の SMF データ・セット名を検査する必要があります。 ダンプされたレコードのターゲット・データ・セットのレコード・フォーマットは VBS でなければならず、レコード長は 32760 でなければなりません。

注 : SMF ログ・ストリームが使用されている場合は、プログラム IFASMFDL を使用して、ログ・ストリームを順次データ・セットにダンプする必要があります。 使用する JCL の例については、「処理タイプ 116 SMF レコード」を参照してください。

その後、ターゲット・データ・セットを CSQ0USMF ユーティリティーの入力として使用して、AMS 監査レポートを生成できます。 以下に例を示します。

```
//STEP1 EXEC PGM=CSQ0USMF,
// PARM=('/ -SMFTYPE 180 -M qmgr')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//    DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SMFIN DD DSN=your.target.dataset,DISP=SHR
//
```

CSQ0USMF プログラムは 2 つのオプション・パラメーターを受け入れます。これを 603 ページの表 97 にリストします。

表 *97. CSQ0USMF* オプション・パラメーター

パラメーター	値	説明
SMFTYPE	nnn	監査レポートに適用できる SMF レコード・タイプ。 レポート生成時に、CSQ0USMF プログラムは SMFTYPE 値が一致する SMF レコードのみを使用します。 SMFTYPE を指定しない場合、デフォルト値 180 が使用されます。
M	qmgr	監査レポートに適用可能な IBM MQ キュー・マネージャー名。 -M パラメーターを指定しない場合、監査レポートには SMFIN データ・セット内に示されるすべてのキュー・マネージャーのすべての監査レコードが含められます。

鍵ストアおよび証明書の使用

IBM 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。 z/OS では、鍵ストア・ファイルの代わりに SAF 鍵リングを使用します。

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。

Windows と UNIX では、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`) で指定されます。 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 は、`.kdb`、`.jceks`、および `.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。

- IBM i UNIX UNIX および IBM i の場合: $HOME/.mqs/keystore.conf
- Windows Windows 上: *%HOMEDRIVE%%HOMEPATH%*\.mqs\keystore.conf

  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。

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 -DMQS_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 上: `export MQS_KEYSTORE_CONF=path/filename`
  - Windows 上: `set MQS_KEYSTORE_CONF=path\filename`

関連概念

630 ページの『送信者識別名 (AMS)』
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。 送信側は、メッセージをキューに入れる前に、証明書を使用してメッセージに署名します。

631 ページの『受信者識別名 (AMS)』
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。

AMS の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。

以下の各タイプの構成ファイルには接頭部があります。

CMS
証明書管理システムの場合は、構成エントリの接頭部に cms. が付きます。

PKCS#11
Public Key Cryptography Standard #11 の場合は、構成エントリの接頭部に pkcs11. が付きます。

IBM i **PEM**
Privacy Enhanced Mail 形式の場合は、構成エントリの接頭部に pem. が付きます。

JKS
Java 鍵ストア、構成エントリーには接頭部 jks. が付けられます。

JCEKS
Java 暗号化暗号鍵ストア、構成エントリーには、接頭部 jceks. が付きます。

V 9.1.0 z/OS MQ Adv. VUE **JCERACFKS**
Java Cryptographic Encryption RACF keyring KeyStore、構成エントリーには、接頭部 jceracfks が付きます。

重要: IBM MQ 9.0 以降、JCEKS.provider 値と JKS.provider 値が無視されるようになりました。 使用されている JRE によって提供されるあらゆる JCE/JCE プロビジョンと組み合わせて、Bouncy Castle プロバイダーが使用されます。 詳しくは、 608 ページの『AMS を使用した非 IBM JRE のサポート』を参照してください。

鍵ストアの構造の例:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
```

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
```

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
jks.provider = IBMJCE
```

Java JCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
jceks.provider = IBMJCE
```

V 9.1.0 Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

表 98. 各構成ファイル・タイプの必須パラメーターの要約

パラメーター	必須	構成ファイル・タイプ			
		V 9.1.0 Java (JKS、JCEKS、および JCERACFKS)	IBM i PEM	PKCS#11	CMS
keystore	✓	✓			✓
IBM i private	✓		IBM i ✓		
IBM i public	✓		IBM i ✓		
IBM i password	✓		IBM i ✓		
library	✓			✓	
certificate	✓	✓		✓	✓
token	✓			✓	

表 98. 各構成ファイル・タイプの必須パラメーターの要約 (続き)					
		構成ファイル・タイプ			
パラメーター	必須	V 9.1.0 Java (JKS、JCEKS、および JCERACFKS)	IBM i PEM	PKCS#11	CMS
token_pin	✓			✓	
secondary_keystore	✓			✓	
encrypted		✓			
keystore_pass	✓	✓			
key_pass		✓			
provider		✓			

記号を使用してコメントを追加できることに注意してください。

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
CMS と Java 構成のみ。 CMS、JKS、JCEKS の構成の鍵ストア・ファイルのパス。

V 9.1.0 z/OS MQ Adv. VUE JCERACFKS 構成の RACF 鍵リングの URI。

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。
- V 9.1.0 z/OS MQ Adv. VUE RACF 鍵リングの URI は、以下の形式でなければなりません。

```
safkeyring://user/keyring
```

ここで、

- *user* は鍵リングの所有者のユーザー ID です。
- *keyring* は鍵リング名です。

IBM i **private**
PEM 構成のみ。 秘密鍵と証明書が PEM 形式で含まれているファイルのファイル名。

IBM i **public**
PEM 構成のみ。 信頼できるパブリック証明書が PEM 形式で含まれているファイルのファイル名。

IBM i **password**
PEM 構成のみ。 暗号化された秘密鍵を暗号化解除するために使用するパスワード。

library
PKCS#11 のみ。 PKCS#11 ライブラリーのパス名。

certificate
CMS、PKCS#11、および Java 構成のみ。 証明書ラベル。

token
PKCS#11 のみ。 トークン・ラベル。

token_pin
PKCS#11 のみ。 トークンをアンロックするための PIN。

secondary_keystore
PKCS#11 のみ。 .kdb 拡張子なしで提供される、CMS 鍵ストアのパス名。これには、PKCS #11 トークンに保管されている証明書に必要なアンカー証明書 (ルート証明書) が含まれます。 2 次鍵ストアにも、トラスト・チェーンの中間にある証明書、およびプライバシー・セキュリティー・ポリシーに定義された受信側証明書を含めることができます。 この CMS 鍵ストアの付属の stash ファイルは、2 次鍵ストアと同じディレクトリー内に配置する必要があります。

encrypted
Java 構成のみ。 パスワードの状況。

keystore_pass
Java 構成のみ。 鍵ストア・ファイルのパスワード。

注 :

- CMS 鍵ストアの場合、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。
- **重要 :** パスワードを平文形式で保管することは、セキュリティーにリスクがあります。

注 : jceracfks では無視されます。パスワードによるアクセス制御は行われません。

key_pass
Java 構成のみ。 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保管することは、セキュリティーにリスクがあります。

注 : jceracfks では無視されます。パスワードによるアクセス制御は行われません。

provider
Java 構成のみ。 鍵ストア証明書で必要とされる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。

重要 : 鍵ストアに保管される情報は、IBM MQ を使用して送信されるデータの安全なフローのために不可欠な情報です。 セキュリティー管理者は、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ります。

keystore.conf ファイルの例

```
# Native AMS application configuration
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

# Java AMS application configurationjceks.keystore = c:/Documents and Settings/Alice/
AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

関連タスク
621 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。

AMS を使用した非 IBM JRE のサポート

IBM MQ classes for Java および IBM MQ classes for JMS は、IBM 以外の JRE を使用して実行する場合に Advanced Message Security 操作をサポートします。

Advanced Message Security (AMS) は、Cryptographic Message Syntax (CMS) を実装します。 CMS 構文は、任意のメッセージの内容をデジタルで署名、ダイジェスト、認証、または暗号化するために使用されます。

IBM MQ 9.0 以降、 IBM MQ classes for Java および IBM MQ classes for JMS の Advanced Message Security サポートは、オープン・ソースの Bouncy Castle パッケージを使用して CMS をサポートします。 これは、これらのクラスが、非 IBM JRE で実行されている場合に Advanced Message Security 操作をサポートできることを意味します。

IBM MQ 9.0 より前のバージョンでは、 Advanced Message Security は Java クライアントの非 IBM JRE ではサポートされていませんでした。 IBM MQ classes for Java および IBM MQ classes for JMS の Advanced Message Security サポートは、特に Java Cryptography Extensions (JCE) の IBM 実装によって提供される CMS サポートに依存していました。 この制限のため、この機能は、 Java JCE プロバイダーを含む Java runtime environment (JRE) を使用する場合にのみ使用可能でした。

Solaris 重要な点として、Solaris などのプラットフォームのサポートには、ハイブリッド JRE、つまりプラットフォーム用の標準 JRE に IBM 提供のエレメントを追加したものが必要でした。 具体的には、プラットフォーム用の標準 JRE によって提供される JCE プロバイダーではなく、IBM JCE プロバイダーが必要でした。

Bouncy Castle JAR ファイルの場所とバージョン番号付け

非 IBM JRE のサポートに必要な Bouncy Castle JAR ファイルは、IBM MQ classes for Java および IBM MQ classes for JMS のインストール・パッケージの一部として組み込まれています。

使用される Bouncy Castle JAR ファイルは、以下のファイルです。

プロバイダー JAR ファイル。Bouncy Castle 操作の基礎となるファイルです。
この JAR ファイルは `bcprov-jdk15on.jar` と呼ばれます。

「PKIX」JAR ファイル。Advanced Message Security によって使用される CMS 操作のサポートが含まれています。
この JAR ファイルは `bcpkix-jdk15on.jar` と呼ばれます。

V 9.1.0.9 **他の Bouncy Castle JAR ファイルによって使用されるクラスが含まれる「util」JAR ファイル。**
この JAR ファイルは `bcutil-jdk15on.jar` と呼ばれます。

依存関係

IBM MQ 9.1 以降のクラスは、IBM JRE および Oracle JRE でテストされています。 は、J2SE-compliant JRE でも正常に実行される可能性があります。 ただし、次のような依存関係に留意する必要があります。

- Advanced Message Security 構成に変更はありません。
- Bouncy Castle クラスは CMS 操作でのみ使用されます。 他のセキュリティー関連の操作 (鍵ストアのアクセスや、データの実際の暗号化、署名チェックサムの計算など) はすべて、JRE によって提供される機能を使用します。

  重要 : そのため、使用される JRE には、JCE プロバイダーの実装が含まれる必要があります。
- いくらか強い 暗号化アルゴリズムを使用するには、JRE の JCE 実装に対して無制限の ポリシー・ファイルをインストールしなければならない可能性があります。

  詳細は、JRE の資料を参照してください。
- Java セキュリティーを有効にした場合、以下の操作を行います。
  - `java.security.SecurityPermissioninsertProvider.BC` をアプリケーションに追加し、Bouncy Castle クラスをセキュリティー・プロバイダーとして使用できるようにします。

– java.security.AllPermission を Bouncy Castle JAR ファイルに付与します。これらのファイルは以下のとおりです。

V 9.1.0.9 *mq_install_dir*/java/lib/bcutil-jdk15on.jar
mq_install_dir/java/lib/bcpkix-jdk15on.jar
mq_install_dir/java/lib/bcprov-jdk15on.jar

関連概念
IBM MQ classes for JMS のインストール内容
IBM MQ classes for Java のインストール内容

Multi メッセージ・チャネル・エージェント (MCA) インターセプト

MCA インターセプトを使用することにより、IBM MQ で実行するキュー・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有効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、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 AMS を有効にできない場合に 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 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。 詳しくは、611 ページの『クライアントでの Advanced Message Security の無効化』を参照してください。

注 : MCA インターセプターは、AMQP または MQTT チャネルでサポートされていません。

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。 AMS 鍵ストアの構成について詳しくは、603 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例については、609 ページの『Advanced Message Security 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 i

企業が IBM i を使用しており、証明書に署名するために商用認証局 (CA) を選択した場合、デジタル Certificate Manager は PEM (Privacy-Enhanced Mail) 形式の認証要求を作成します。 対象とする CA に要求を転送する必要があります。

これを行うには、以下のコマンドを使用して、channelname に指定されたチャネルの正しい証明書を選択する必要があります。

```
pem.certificate.channel.channelname
```

Advanced Message Security MCA インターセプトの例

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。

企業が IBM i を使用しており、証明書に署名するために商用認証局 (CA) を選択した場合、デジタル Certificate Manager は PEM (Privacy-Enhanced Mail) 形式の認証要求を作成します。 対象とする CA に要求を転送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 7.5 より前は、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。 IBM WebSphere MQ 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 IBM WebSphere MQ 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/mq90
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 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 MQ 9.0 のインストール環境からキューを参照します。

```
/opt/mq90/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 MQ 9.0 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

591 ページの『Java クライアントを使用する AMS 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 アプリケーションにメッセージ・セキュリティーを提供するように 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

563 ページの『AMS の既知の制限』
サポートされていないか、 Advanced Message Security に制限がある IBM MQ オプションがいくつかあります。

クライアントでの Advanced Message Security の無効化

以前のバージョンの製品からキュー・マネージャーに接続するために IBM WebSphere MQ 7.5 以降のクライアントを使用していて、2085 （MQRC_UNKNOWN_OBJECT_NAME）エラーが報告される場合は、IBM 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

IBM WebSphere MQ 7.5 以降、IBM MQ Advanced Message Security (AMS) は IBM 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、それより前のバージョンの製品のサーバー (IBM WebSphere MQ 7.1 など) では、AMS が有効になっていないため、これによって 2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告されます。

前のバージョンの製品のキュー・マネージャーに接続しようとしてこのエラーが報告された場合は、以下のように 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。
  - 環境変数 AMQ_DISABLE_CLIENT_AMS を設定します。
  - 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。
  - 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。
  - 環境変数 MQS_DISABLE_ALL_INTERCEPT を設定します。
  - 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。

注 : IBM WebSphere MQ 7.5 では、環境変数 AMQ_DISABLE_CLIENT_AMS を使用することもできます。 C クライアントの場合。 IBM MQ 8.0 以降、C クライアントで AMQ_DISABLE_CLIENT_AMS 環境変数を使用することはできなくなりました。 代わりに MQS_DISABLE_ALL_INTERCEPT 環境変数を使用する必要があります。

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。

  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
  - IBM WebSphere MQ 7.5 以降の IBM MQ classes for JMS または IBM MQ classes for Java クライアントを使用している場合。

  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
  ```

  Java システム・プロパティー com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  IBM MQ classes for JMS および IBM MQ classes for Java クライアントの場合は、Java system property com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS を Java アプリケーションの値 「TRUE」 に設定することができます。

  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。

  MQS_DISABLE_ALL_INTERCEPT 環境変数
  ネイティブ・クライアントで IBM MQ 8.0 以降を使用しており、クライアントで AMS を無効化する必要がある場合は、この変数を設定する必要があります。

環境変数 MQS_DISABLE_ALL_INTERCEPT を作成し、クライアント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 MQS_DISABLE_ALL_INTERCEPT =TRUE
```

MQS_DISABLE_ALL_INTERCEPT 環境変数は、C クライアントでのみ使用できます。 Java クライアントでは、代わりに AMQ_DISABLE_CLIENT_AMS 環境変数を使用する必要があります。

mqclient.ini ファイル内の 「DisableClientAMS」 プロパティー

IBM MQ classes for JMS クライアント、IBM MQ classes for Java クライアント、および C クライアントでこのオプションを使用できます。

以下の例に示すように、mqclient.ini ファイルの **Security** スタンザの下にプロパティー名 DisableClientAMS を追加します。

```
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、 AMS を JMS と一緒に使用する場合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

609 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 インターセプトを使用することにより、IBM MQ で実行するキュー・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有効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 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、43 ページの『IBM 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、X.509 v3 証明書の鍵用途は、RFC 5280 仕様に従って設定する必要があります。

保護品質「整合性」のためには、証明書の鍵用途拡張を設定する場合、次の 2 つのうち、少なくとも 1 つをその設定に含める必要があります。

- **nonRepudiation**
- **digitalSignature**

保護品質「プライバシー」のためには、証明書の鍵用途拡張を設定する場合、その設定に次を含める必要があります。

- **keyEncipherment**

保護品質「機密性」のためには、証明書の鍵用途拡張を設定する場合、その設定に次を含める必要があります。

- **dataEncipherment**

拡張鍵用途によって、鍵用途拡張をさらに微調整できます。 すべての保護品質について、証明書の拡張鍵用途を設定する場合、その設定に次を含める必要があります。

- **emailProtection**

関連概念

632 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。

AMS での証明書の検証方式

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、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。

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。

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、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、AMS は CRL 検査を使用します。

OCSP 検査と CRL 検査はどちらもデフォルトで有効になります。

関連概念

614 ページの『Online Certificate Status Protocol (OCSP) (AMS)』
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。 OCSP はデフォルトで有効になります。

616 ページの『証明書失効リスト (CRL) (AMS)』
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) (AMS)

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。 OCSP はデフォルトで有効になります。

OCSP は IBM i システムではサポートされていません。

Advanced Message Security のネイティブ・インターセプターでの *OCSP* 検査の有効化

Advanced Message Security では、使用する証明書の情報に基づいて、Online Certificate Status Protocol (OCSP) 検査がデフォルトで有効になります。

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 : OCSP スタンザはすべてオプションで、個別に指定できます。

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 (AIA) 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。 指定可能な値: on または `off`。
`ocsp.url=`*`responder_URL`*	OCSP 応答側の URL アドレス。 このオプションを省略すると、非 AIA OCSP 検査は無効になります。

オプション	説明
ocsp.http.proxy.host=*OCSP_proxy*	OCSP プロキシー・サーバーの URL アドレス。 このオプションを省略すると、非 AIA オンライン証明書検査にプロキシーは使用されません。
ocsp.http.proxy.port=*port_number*	OCSP プロキシー・サーバーのポート番号。 このオプションを省略すると、デフォルト・ポート 8080 が使用されます。
ocsp.nonce.generation=on/off	OCSP の照会時に nonce を生成します。 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。 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。 デフォルト値は off です。
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。
ocsp.unknown=ACCEPT	タイムアウト期間内に OCSP サーバーに到達しない場合の動作を定義します。 考えられる値： • ACCEPT。証明書を許可します。 • WARN。証明書を許可し、警告をログに記録します。 • REJECT。証明書を使用できないようにして、エラーをログに記録します。

Java の OCSP 検査の有効化 *(AMS)*
Java での Advanced Message Security の OCSP 検査を使用可能にするには、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
証明書に認証局情報アクセス (AIA) 証明書拡張が含ま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、構成を完了するために以下の 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

構成を完了するために、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";
```

証明書失効リスト *(CRL) (AMS)*

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。 AMS は、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、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、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、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、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、AMS は CRL 妥当性検査を実行しません。 CA 証明書ごとに、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。

このタスクについて

IBM i ネイティブ・インターセプターでの証明書検証および証明書失効リスト・サポートの有効化は、 IBM i 上の Advanced Message Security ではサポートされていません。

手順

構成ファイルに、以下のオプションを追加します。

注: CRL スタンザはすべてオプションで、個別に指定できます。

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。 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。 AMS Java インターセプターが LDAP サーバーに正常に接続すると、指定されている残りのサーバーからの CRL のダウンロードは試行しません。
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)。

オプション	説明
crl.http.timeout=30	サーバー応答の待ち時間 (秒)。これを超えると、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)。
crl.unknown=ACCEPT	タイムアウト期間内に CRL サーバーに到達しない場合の動作を定義します。 考えられる値： • ACCEPT。証明書を許可します。 • WARN。証明書を許可し、警告をログに記録します。 • REJECT。証明書を使用できないようにして、エラーをログに記録します。

Java の証明書失効リスト・サポートの有効化 *(AMS)*
Advanced Message Security で CRL サポートを使用可能にするには、鍵ストア構成ファイルを変更して、AMS が Lightweight Directory Access Protocol (LDAP) サーバーから CRL をダウンロードし、java.security ファイルを構成できるようにする必要があります。

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。 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。 AMS Java インターセプターが LDAP サーバーに正常に接続すると、指定されている残りのサーバーからの CRL のダウンロードは試行しません。 Java は、crl.ldap.user および crl.ldaworldp.pass の値を使用しません。 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。 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。 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。
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。 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。 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。

z/OS *z/OS での証明書取り消しリスト (CRL) の有効化*

Advanced Message Security では、データ・メッセージの保護に使用されるデジタル証明書に対する証明書取り消しリスト (CRL) 検査がサポートされています。

このタスクについて

有効にすると、Advanced Message Security は、メッセージがプライバシー保護キューに書き込まれる際に受信者の証明書の妥当性検査を実行し、メッセージが保護キューから取り出される際に送信者の証明書の妥当性検査を実行します (整合性またはプライバシー)。 この場合の妥当性検査には、関連する証明書が関連する CRL に登録されていないことの確認が含まれます。

Advanced Message Security は、IBM System SSL サービスを使用することによって、送信者と受信者の証明書の妥当性検査を実行します。 System SSL 証明書の妥当性検査に関する詳細な資料については、「z/OS Cryptographic Services System SSL (Secure Sockets Layer) プログラミング」(SD88-6252) のマニュアルを参照してください。

CRL 検査を有効にするには、AMS アドレス・スペースの開始済みタスク JCL で CRLFILE DD を使用して CRL 構成ファイルの場所を指定します。 カスタマイズ可能なサンプル CRL 構成ファイルが *thlqual*.SCSQPROC(CSQ40CRL) にあります。 このファイルで使用できる設定は次のとおりです。

表 *99. Advanced Message Security の CRL* 構成変数		
変数	**有効値**	**説明**
crl.ldap.host[.n]	*hostname -or- hostname:port*	発行者証明書の CRL をホストする LDAP サーバーの IP アドレス/ホスト名。 LDAP サーバーのポート番号を指定しない場合、crl.ldap.port で指定されたポート番号が使用されます。
crl.ldap.port	*port*	使用する LDAP サーバーの TCP/IP ポート番号。
crl.ldap.user	*ldap_user*	LDAP サーバーに接続する際に使用する LDAP ユーザー名。
crl.ldap.pass	*ldap_password*	crl.ldap.user に関連付けられた LDAP パスワード。

LDAP サーバーのホスト名とポートは、次のようにして複数指定することができます。

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

最大 10 台のホスト名を指定できます。 LDAP サーバーのポート番号を指定しない場合、crl.ldap.port で指定されたポート番号が使用されます。 各 LDAP サーバーへのアクセスには、同じ組み合わせの crl.ldap.user/password を使用する必要があります。

CRLFILE DD を指定すると、Advanced Message Security アドレス・スペースの初期化中に、CRL 検査が有効である場合に構成が読み込まれます。 CRLFILE DD が指定されていない場合、または CRL 構成ファイルが使用できないか無効である場合、CRL 検査は無効になります。

AMS は、IBM System SSL 証明書妥当性検査サービスを使用して次のように CRL 検査を実行します。

表 *100. Advanced Message Security CRL* 検査		
Operation	**保護品質**	**検査される証明書**
PUT	プライバシー	受信者
GET	整合性/プライバシー	送信者

メッセージ操作で CRL 検査に失敗すると、Advanced Message Security は以下のアクションを実行します。

表 *101. Advanced Message Security の CRL* 検査失敗時の動作	
Operation	**CRL 検査失敗**
PUT	メッセージはターゲット・キューに書き込まれません。 アプリケーションに完了コード MQCC_FAILED と理由コード MQRC_SECURITY_ERROR が返されます。
GET	メッセージはターゲット・キューから削除され、システム保護エラー・キューに移動されます。 アプリケーションに完了コード MQCC_FAILED と理由コード MQRC_SECURITY_ERROR が返されます。

AMS for z/OS は、IBM System SSL サービスを使用することによって、証明書の妥当性検査を実行します。それには、CRL 検査と信頼性検査が含まれます。 IBM System SSL には、CRL 検査の操作を調節するための環境変数 GSK_CRL_SECURITY_LEVEL が提供されています。 以下に例を示します。

```
GSK_CRL_SECURITY_LEVEL=MEDIUM
```

この変数については、「z/OS Cryptographic Services System SSL (Secure Sockets Layer) プログラミング」のマニュアルを参照してください。 有効な代入値は次のとおりです。

- LOW - 証明書の検証は、LDAP サーバーと 通信できない場合でも失敗しません。
- MEDIUM - 証明書の検証には、LDAP サーバーと通信可能であることが必要ですが、CRL が定義されている必要はありません。
- HIGH - 証明書の検証では、LDAP サーバーと通信可能であることと、CRL が定義されていることが必要です。

IBM System SSL のデフォルト値は MEDIUM です。 この変数は、AMS アドレス・スペースの開始済みタスク JCL の中で、ENVARS DD を使用して指定する構成ファイルにおいて設定できます。 サンプルの環境変数構成ファイルは *thlqual*.SCSQPROC(CSQ40ENV) にあります。

注 : 管理者には、関連する LDAP サービスが使用可能であることを保証し、関連する認証局の CRL エントリーを保守する責任があります。

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、603 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
```

z/OS z/OS での証明書の使用

このタスクについて

Advanced Message Security では、保全性、機密性、およびプライバシーという 3 つのレベルの保護が実装されています。

保全性ポリシーでは、発信元 (MQPUT を行うアプリケーション) の秘密鍵を使ってメッセージが署名されます。 保全性ではメッセージの改変が検出されますが、メッセージ・テキスト自体は暗号化されません。

機密性ポリシーにより、メッセージはキューに書き込まれるときに暗号化されます。 メッセージは、関連する Advanced Message Security ポリシーで指定されたアルゴリズムおよび対称鍵を使って暗号化されます。 対称鍵自体は、それぞれの受信側 (MQGET を行うアプリケーション) の公開鍵を使用して暗号化されます。 公開鍵は、鍵リングに保管される証明書に関連付けられます。

プライバシー・ポリシーにより、メッセージは署名され、かつ暗号化されます。

プライバシー機能で保護されたメッセージが、MQGET を行う受信側アプリケーションによってデキューされるとき、メッセージは暗号化解除されなければなりません。 メッセージは受信側の公開鍵を使って暗号化されているため、鍵リングの中にある受信側の秘密鍵を使って暗号化解除する必要があります。

z/OS *SAF* 鍵リングの使用

Advanced Message Security (AMS) は、z/OS SAF 鍵リング・サービスを使用することによって、署名と暗号化で必要な証明書を定義および管理します。 RACF と機能的に同等のセキュリティー製品によって同じレベルのサポートが提供されるなら、RACF の代わりにその製品を使用できます。

鍵リングを効率的に使用すると、証明書の管理に必要な管理作業を削減できます。

証明書の生成後 (またはインポート後)、アクセス可能にするためその証明書を鍵リングに接続する必要があります。 同じ証明書を複数の鍵リングに接続できます。

Advanced Message Security は 2 セットの鍵リングを使用します。 1 つのセットには、メッセージの発信または受信を実行する個々のユーザー ID によって所有される鍵リングが格納されます。 各鍵リングには、所有するユーザー ID の証明書に関連付けられた秘密鍵が格納されます。 各証明書の秘密鍵は、整合性保護キューまたはプライバシー保護キューのメッセージに署名するために使用されます。 また、メッセージ受信時にプライバシー保護キューまたは機密性保護キューからのメッセージを暗号化解除するためにも使用されます。

もう 1 つのセットは、AMS アドレス・スペース・ユーザーが所有する単一の鍵リングです。 これには、メッセージの発信元および受信者の証明書を検証するために必要な署名 CA 証明書チェーンが格納されます。

プライバシー保護または機密性保護を使用する場合、AMS アドレス・スペース・ユーザーが所有する鍵リングには、メッセージ受信者の証明書も格納されます。 これらの証明書の公開鍵は対称鍵の暗号化に使用されます。対称鍵は、メッセージが保護キューに書き込まれるときにメッセージ・データを暗号化するために使用されたものです。 これらのメッセージを取得する際、関連する受信者の秘密鍵が対称鍵の暗号化解除に使用され、その対称鍵がメッセージ・データの暗号化解除に使用されます。

Advanced Message Security は、証明書と秘密鍵を検索するとき、鍵リング名 **drq.ams.keyring** を使用します。 これは、ユーザーの鍵リングと AMS アドレス・スペースの鍵リングの両方に該当します。

証明書と鍵リングの図と説明、およびデータ保護でのそれらの役割の詳細については、証明書関連操作の概要を参照してください。

署名と暗号化解除に使用される秘密鍵には任意のラベルを付けることができますが、デフォルト証明書として接続される必要があります。

デジタル証明書と鍵リングは主に RACDCERT コマンドを使用して RACF で管理されます。

証明書、ラベル、RACDCERT コマンドの詳細については、 *z/OS: Security Server RACF* コマンド言語解説書と *z/OS: Security Server RACF* セキュリティー管理者のガイドを参照してください。

z/OS *RACDCERT* コマンドに対するアクセス権限の設定

RACDCERT コマンドを使用するための権限設定は、z/OS システム・プログラマーが完了することの必要なインストール後タスクです。 このタスクには、関連するアクセス権を Advanced Message Security セキュリティー管理者に付与する操作が含まれます。

要約すると、RACF RACDCERT コマンドへのアクセス権限を許可するため、以下のコマンドが必要です。

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

この例では、*admin* に、セキュリティー管理者、または RACDCERT コマンドを使用する任意のユーザーのユーザー ID を指定します。

z/OS 証明書と鍵リングの作成

このセクションでは、RACF 認証局 (CA) を使用して、Advanced Message Security (AMS) の z/OS ユーザーに必要な証明書と鍵リングを作成するために必要なステップについて説明します。

z/OS で Advanced Message Security を使用する場合の証明書の問題の解決

証明書と鍵ストア内のエントリーの欠落により問題が発生している場合、GSKIT トレースを有効にすることができます。

ENVARS 開始タスク・プロシージャー内の AMS DD によって参照されるファイルに、以下を追加します。

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

詳しくは、環境変数 を参照してください。

鍵ストアにアクセスするたびに、GSK_TRACE_FILE で指定されたトレース・ファイルにデータが書き込まれます。

トレース・ファイルをフォーマット設定するには、次のコマンドを使用します。

```
gsktrace inputtrace file > output_file
```

シナリオ

送信アプリケーションと受信アプリケーションのシナリオを使用して、必要な手順を説明します。

次の例では、user1 がメッセージの発信元、user2 が受信者です。 Advanced Message Security アドレス・スペースのユーザー ID は WMQAMSD です。

ここに示す例に含まれるすべてのコマンドは、ISPF オプション 6 から管理ユーザー ID admin を使用して発行されます。

z/OS ローカル認証局証明書の定義

CA として RACF を使用している場合、認証局証明書がまだ作成されていない場合は、それを作成する必要があります。 ここに示すコマンドは、認証局 (または署名者) 証明書を作成します。 この例では、AMSCA

という証明書が作成され、これがその後 Advanced Message Security のユーザーとアプリケーションの ID を示す証明書の作成に使用されます。

このコマンドは、特に SUBJECTSDN を変更することにより、インストール済み環境で使用するネーミングの構造と命名規則に合わせることができます。

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

注 : このローカル認証局証明書で署名された証明書は、RACDCERT LIST コマンドによるリスト出力において、CN=AMSCA,O=ibm,C=us の発行者を示します。

z/OS 秘密鍵によるデジタル証明書の作成

秘密鍵によるデジタル証明書は、各 Advanced Message Security ユーザーに対して生成する必要があります。 ここに示す例では、RACDCERT コマンドを使用して user1 と user2 の証明書を生成します。 これらはラベル AMSCA で識別されるローカル CA 証明書を使用して署名されます。

```
RACDCERT ID(user1) GENCERT SUBJECTSDN(CN('user1') O('ibm') C('us'))
WITHLABEL('user1') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user2) GENCERT SUBJECTSDN(CN('user2') O('ibm') C('us'))
WITHLABEL('user2') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user1) ALTER (LABEL('user1')) TRUST
RACDCERT ID(user2) ALTER (LABEL('user2')) TRUST
```

RACDCERT ALTER コマンドは、証明書に TRUST 属性を追加するために必要です。 最初にこの手順で証明書を作成すると、署名する証明書とは異なる有効日付範囲が設定されます。 その結果、RACF は、この証明書を使用すべきでないことを意味する NOTRUST のマークを付けます。 RACDCERT ALTER コマンドを使用して、TRUST 属性を設定してください。

Advanced Message Security が使用する証明書には、KEYUSAGE 属性 HANDSHAKE、DATAENCRYPT、および DOCSIGN を指定する必要があります。

表 *102. RACDCERT KEYUSAGE* の値と標識

KEYUSAGE 値	**設定される標識**
HANDSHAKE	digitalSignature および keyEncipherment
DATAENCRYPT	dataEncipherment
DOCSIGN	nonRepudiation
CERTSIGN	keyCertSign および cRLSign

z/OS *RACF* 鍵リングの作成

以下に示すコマンドは、RACF 定義ユーザー ID user1 用、user2 用、および Advanced Message Security アドレス・スペース・タスク・ユーザー WMQAMSD 用の鍵リングを作成します。 鍵リング名は Advanced Message Security によって固定されており、ここに示すとおり、引用符なしでコーディングしなければなりません。 名前は大文字小文字が区別されます。

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS 証明書から鍵リングへの接続

ユーザーと CA 証明書を鍵リングに接続します。

```
RACDCERT ID(WMQAMSD) CONNECT(CERTAUTH LABEL('AMSCA')
RING(drq.ams.keyring))
RACDCERT ID(user1) CONNECT(ID(user1) LABEL('user1')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(user2) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(WMQAMSD) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) USAGE(SITE))
```

暗号化解除に使用する秘密鍵を含む証明書は、ユーザーの鍵リングにデフォルト証明書として接続する必要があります。

RACDCERT USAGE(SITE) 属性により、秘密鍵は鍵リング内でアクセスできないようにされます。一方、RACDCERT USAGE(PERSONAL) 属性が存在すると、それにより秘密鍵の使用が許可されます。 user2 の証明書は Advanced Message Security アドレス・スペース鍵リングに接続する必要があります。 メッセージがキューに書き込まれる際に、そのメッセージを暗号化するために公開鍵が必要になるからです。 USAGE(SITE) により user2 の秘密鍵の公開は制限されます。

AMSCA のラベルが付けられた CERTAUTH 証明書は Advanced Message Security アドレス・スペース鍵リングに接続される必要があります。これは、メッセージの発信元である user1 の証明書に署名するために使用されたものだからです。 これは、user1 の署名証明書の妥当性検査に使用されます。

z/OS 鍵リングの検証

すべてのコマンドを入力すると、鍵リングはここに示すように表示されるはずです。

```
RACDCERT ID(user1) LISTRING(drq.ams.keyring)
Digital ring information for user USER1:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user1                            ID(USER1)    PERSONAL YES

RACDCERT ID(user2) LISTRING(drq.ams.keyring)
Digital ring information for user USER2:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user2                            ID(USER2)    PERSONAL YES

RACDCERT ID(WMQAMSD) LISTRING(drq.ams.keyring)
Digital ring information for user WMQAMSD:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
AMSCA                            CERTAUTH     CERTAUTH NO
user2                            ID(USER2)    SITE     NO
```

個々の証明書のリスト表示にはリングのアソシエーションも示されます。

```
RACDCERT ID(user2) LIST(label('user2'))
Digital certificate information for user USER2:

***
Label: user2
Certificate ID: 2QfH8Pny9/LzpKKFmfFA
Status: TRUST
Start Date: 2010/05/03 22:59:53
End Date:  2011/05/04 22:59:52
Serial Number:>15<:
Issuer's Name:>OU=AMSCA.O=ibm.C=us<:
Subject's Name:>CN=user2.O=ibm.C=us<:
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN
```

```
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: USER2
Ring:>drq.ams.keyring<:
Ring Owner: WMQAMSD
Ring:>drq.ams.keyring<:
```

パフォーマンス向上のため、AMS アドレス・スペースに関連付けられた drq.ams.keyring の内容は、アドレス・スペースの存続期間にわたってキャッシュに入れられます。 鍵リングを変更しても自動的には有効になりません。 管理者は次のいずれかの方法でキャッシュをリフレッシュできます。

- キュー・マネージャーを停止して再始動します。
- z/OS MODIFY コマンドを使用して以下を実行します。

```
F qmgrAMSM,REFRESH KEYRING
```

関連タスク

Advanced Message Security の運用

z/OS 証明書関連操作の概要

626 ページの図 35 に、送信アプリケーションと受信アプリケーション、および関連する証明書の関係を示します。 図示するシナリオには、プライバシーのデータ保護ポリシーを使用した 2 つの z/OS キュー・マネージャー間のリモート・キューイングが関係します。 626 ページの図 35 の「AMS」は「Advanced Message Security」を示します。

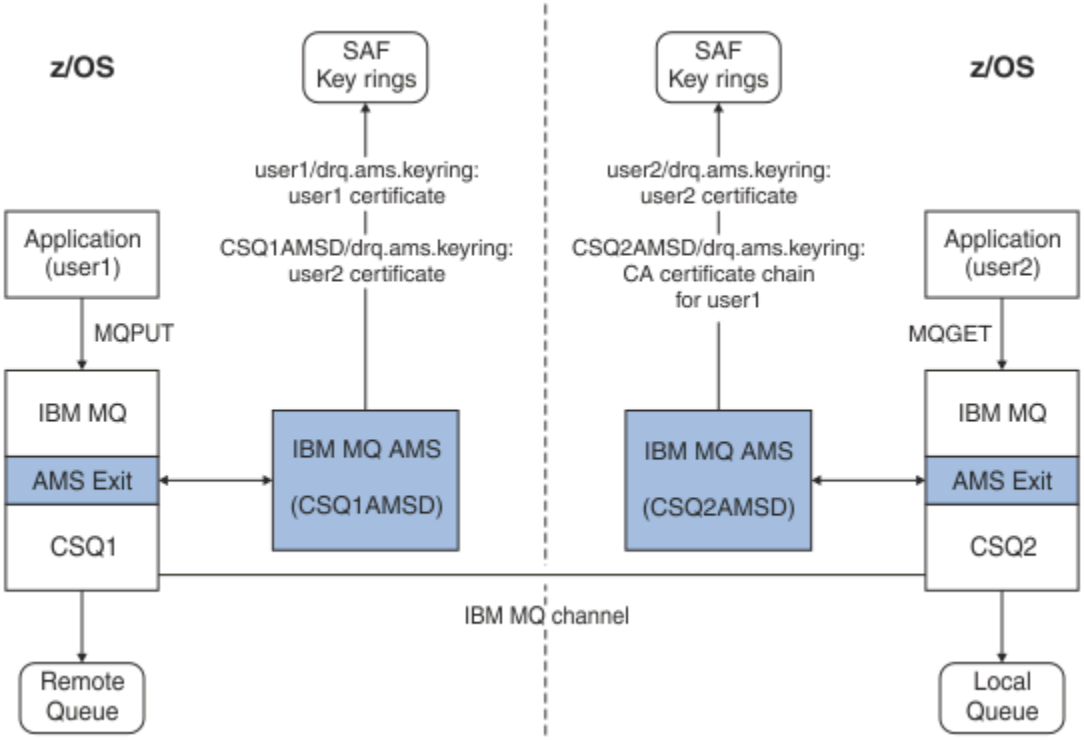


図 *35.* アプリケーションと証明書の関係

この図では、「user1」として実行しているアプリケーションから、キュー・マネージャー CSQ1 が管理するリモート・キューにメッセージを書き込みます。これは、キュー・マネージャー CSQ2 が管理するローカル・キューから「user2」として実行しているアプリケーションが取り出すことを意図しています。 この図では、Advanced Message Security のプライバシーのポリシーがあること、すなわちメッセージに署名と暗号化の両方がなされることを前提としています。

Advanced Message Security は、書き込みが発生するとき、および user2 の証明書 (AMS アドレス・スペース・ユーザーの鍵リングに保管されている) を使用して対称鍵を暗号化するときにメッセージをインターセプトします。対称鍵はメッセージ・データの暗号化に使用されます。

user2 の証明書は AMS アドレス・スペース・ユーザー鍵リングにオプション USAGE(SITE) を使用して接続されることに注意してください。 これは、AMS アドレス・スペース・ユーザーは証明書と公開鍵にアクセスできますが、秘密鍵にはアクセスできないことを意味します。

受信が終了すると、Advanced Message Security は user2 が発行した GET をインターセプトし、user2 の証明書を使用して対称鍵を暗号化解除し、これを使用してメッセージ・データの暗号化解除ができるようにします。 その後、AMS アドレス・スペース・ユーザーの鍵リングに保管された user1 の証明書の CA 証明書チェーンを使用することによって、user1 の署名の妥当性検査が実行されます。

このシナリオで、整合性のデータ保護ポリシーを使用する場合、user2 の証明書は必要ありません。

Advanced Message Security を使用してメッセージをプライバシーまたは整合性のメッセージ保護ポリシーのある IBM MQ 保護キューに入れるには、Advanced Message Security に次のデータ・アイテムへのアクセス権限が付与されている必要があります。

- メッセージをキューに入れるユーザーの X.509 V2 または V3 の証明書と秘密鍵。
- すべてのメッセージ署名者のデジタル証明書に署名するために使用する証明書チェーン。
- データ保護ポリシーがプライバシーである場合、目的の受信者の X.509 V2 または V3 の証明書。 目的の受信者は、キューに関連付けられた Advanced Message Security ポリシーにリストされます。

z/OS で実行されるプロセスとアプリケーションの場合、Advanced Message Security は 2 つの場所に証明書を保持する必要があります。

- 送信アプリケーション (保護メッセージをキューに入れるアプリケーション) または受信アプリケーション (プライバシーを使用する場合) の RACF ID に関連付けられた SAF 管理鍵リング内。

  Advanced Message Security が見つける証明書はデフォルト証明書であり、秘密鍵が含まれていなければなりません。 Advanced Message Security では、送信アプリケーションの z/OS ユーザー ID が前提とされています。 つまり、ユーザーの秘密鍵にアクセス可能にするため、代理として動作します。
- AMS アドレス・スペース・ユーザーに関連付けられた SAF 管理鍵リング内。

  プライバシーで保護されたメッセージを送信する場合、この鍵リングにはメッセージ受信者の公開鍵証明書が含まれています。 メッセージを受信する場合は、メッセージ送信者の署名の妥当性検査のために必要な認証局証明書のチェーンが含まれています。

先に示した例では RACF をローカル CA として使用しました。 しかし、インストール済み環境で別の PKI プロバイダー (認証局) を使用することもできます。 別の PKI 製品を使用する予定の場合は、秘密鍵と証明書を、 Advanced Message Security によって保護されている IBM MQ メッセージを発信する z/OS RACF ユーザー ID に関連付けられた鍵リングにインポートする必要があることに注意してください。

RACF RACDCERT コマンドは、認証要求を生成するためのメカニズムとして使用できます。認証要求はエクスポートして、発行を選択した PKI プロバイダーに送信できます。

以下に、証明書関連の手順を要約します。

1. CA 証明書 (RACF がローカル CA であるもの) の作成を要求します。 別の PKI プロバイダーを使用する場合、このステップは省略します。
2. この CA が署名したユーザー証明書を生成します。
3. ユーザー用および Advanced Message Security AMS アドレス・スペース ID 用に鍵リングを作成します。
4. デフォルト属性を使用してユーザー証明書をユーザー鍵リングに接続します。
5. usage(site) 属性を使用して、受信者証明書を Advanced Message Security AMS アドレス・スペース・ユーザー鍵リングに接続します (この手順は、最終的にプライバシー保護メッセージの受信者になるユーザー証明書のみに必要です)。
6. メッセージ送信者の CA 証明書チェーンを Advanced Message Security AMS アドレス・スペース・ユーザー鍵リングに接続します。 (この手順は、送信者署名を検証する AMS タスクの場合にのみ必要です。)

z/OS 非 *z/OS* 常駐 *PKI* の構成

Advanced Message Security for z/OS は、IBM MQ キューに書き込むメッセージ、またはキューから読み取るメッセージの保護処理に X.509 V3 デジタル証明書を使用します。 Advanced Message Security 自体は、これらの証明書のライフサイクルの作成も管理も実行しません。それらの機能は Public Key Infrastructure (PKI) により提供されます。 この資料の例で証明書の使用を説明する場合、z/OS Security Server RACF を使用して認証要求を満たします。

z/OS または非 z/OS 常駐 PKI を使用するかどうかに関係なく、AMS for z/OS は RACF またはそれと同等の機能で管理される鍵リングのみを使用します。 これらの鍵リングはセキュリティー許可機能 (SAF) に基づいており、AMS for z/OS が使用するリポジトリーになります。これは、IBM MQ キューに書き込まれるメッセージまたは取り出されるメッセージの発信元または受信者の証明書を取得するために使用されます。

z/OS から発信されたメッセージの場合、整合性または暗号化ポリシーのいずれかにより保護されます。発信元ユーザー ID の証明書と秘密鍵は、メッセージ発信元の z/OS ユーザー ID に関連付けられた SAF 管理鍵リングに保管される必要があります。

RACF には、証明書と秘密鍵を RACF 管理鍵リングにインポートする機能が含まれています。 RACF 管理対象鍵リングに証明書をロードする方法の詳細と例については、 z/OS Security Server RACF の資料を参照してください。

サポートされる PKI 製品の 1 つをインストール済み環境で使用している場合、デプロイ方法については、その製品に添付された資料を参照してください。

Advanced Message Security セキュリティー・ポリシーの管理

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 (AMS)

Advanced Message Security のセキュリティー・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

632 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。

631 ページの『AMS でのセキュリティー・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 *(AMS)*

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 (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、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 *(AMS)*

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 *(AMS)*

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。

NONE 以外の暗号化アルゴリズムを指定するポリシーでは、Advanced Message Security 暗号化メッセージも署名されるため、少なくとも 1 つの受信者 DN と署名アルゴリズムも指定する必要があることに注意してください。

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。

容認性 *(AMS)*

容認属性は、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。

送信者識別名 *(AMS)*

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。 送信側は、メッセージをキューに入れる前に、証明書を使用してメッセージに署名します。

Advanced Message Security (AMS) は、メッセージが取り出されるまで、そのメッセージが、有効なユーザーによってデータ保護されたキューに入れられていたかどうかを検査しません。 この時点で、ポリシーに 1 つ以上の有効な送信側が明記されており、キューにメッセージを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い場合、 AMS は受信側アプリケーションにエラーを返し、メッセージを AMS 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、送信側の証明書が信頼されていれば、すべての送信側はデータ保護されたメッセージをキューに書き込むことができます。 送信側の証明書は、受信側アプリケーションで使用可能な鍵ストアにパブリック証明書を追加することによって信頼されます。

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。
- 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 か、**strmqikm** GUI 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 コーディングをオンにして UNIX プラットフォームからポリシーを作成するか、 AMS プラグインを IBM MQ に使用する必要があります。
- 送信者の名前を x.509 形式から DN 形式に変換するために AMS で使用される方式では、常に、都道府県の値に ST= が使用されます。
- 以下の特殊文字にはエスケープ文字が必要です。

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- 組み込みブランクが含まれている識別名は、二重引用符で囲む必要があります。

関連概念

631 ページの『受信者識別名 (AMS)』
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。

受信者識別名 *(AMS)*

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 か、**strmqikm** GUI 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 コーディングをオンにして UNIX プラットフォームからポリシーを作成するか、Advanced Message Security プラグインを IBM MQ に使用する必要があります。

関連概念

630 ページの『送信者識別名 (AMS)』
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。 送信側は、メッセージをキューに入れる前に、証明書を使用してメッセージに署名します。

AMS でのセキュリティー・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。

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。

表 *103. AMS* でのセキュリティー・ポリシー属性	
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 の 3 つの保護品質レベルは、IBM MQ 9.0 以降での 4 つ目のレベルによって補足されます。これらはすべて、メッセージの署名および暗号化に使用される暗号アルゴリズムに依存します。

- 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。
- 機密性 - キューに入れられるメッセージは、暗号化される必要があります。 詳しくは、560 ページの『AMS で使用可能な保護品質』 を参照してください。
- 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。

メッセージがキューに入れられるときに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「機密性」です。 「機密性」の QOP は、ポリシーが暗号化アルゴリズムを規定していることを意味します。

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 は、ポリシーの QOP が「なし」であるキューには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・ポリシーに関連するすべての管理タスクを実行する場所は、使用するプラットフォームによって異なります。

- ULW UNIX および Windows では、DELETE POLICY、DISPLAY POLICY、および SET POLICY (または同等の PCF) コマンドを使用して、セキュリティー・ポリシーを管理します。
  - UNIX UNIX では、管理タスクを *MQ_INSTALLATION_PATH*/bin から実行できます。
  - Windows 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。
- IBM i IBM i では、DSPMQMSPL、SETMQMSPL、および WRKMQMSPL コマンドが、IBM MQ のインストール時のシステムの 1 次言語の QSYS システム・ライブラリーにインストールされます。

  追加の各国語バージョンは、言語機能のロードに従って QSYS29xx ライブラリーにインストールされます。 例えば、1 次言語が米国英語で 2 次言語が韓国語であるマシンでは、米国英語のコマンドが QSYS にインストールされ、2 次言語である韓国語のロードが QSYS2962 にインストールされます (2962 は韓国語の言語ロードです)。
- z/OS z/OS の場合: 管理コマンドはメッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) を使用して実行します。 z/OS でポリシーが作成、変更、または削除された場合、キュー・マネージャーが停止して再始動するか、 z/OS MODIFY コマンドを使用して Advanced Message Security ポリシー構成をリフレッシュするまで、変更は Advanced Message Security によって認識されません。 以下に例を示します。

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

関連タスク

634 ページの『AMS でのセキュリティー・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。

634 ページの『AMS でのセキュリティー・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。

635 ページの『セキュリティー・ポリシーの表示とダンプ (AMS)』
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。

637 ページの『AMS でのセキュリティー・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。

Advanced Message Security の運用

関連資料
メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL)

AMS でのセキュリティー・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。

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「IBM MQ オブジェクトの命名規則」に従う必要があります。
- キュー・マネージャーに接続してセキュリティー・ポリシーを作成するために必要な権限がなければなりません。
  - z/OS z/OS では、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) で説明されている権限を付与します。
  - Multi z/OS 以外のプラットフォームでは、setmqaut コマンドを使用して、必要な +connect、+inq、+chg の各権限を付与する必要があります。

  セキュリティーの構成の詳細については、125 ページの『セキュリティーのセットアップ』を参照してください。
- z/OS z/OS では、必須システム・オブジェクトが CSQ4INSM の定義に従って定義されていることを確認します。

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 SHA256 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 SHA256 -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。 ポリシーは、DN を持つ証明書に対して 3DES アルゴリズムを使用してメッセージを暗号化することを指定しています。CN=john、O=IBM、C=US および CN=jf、O=IBM、C=US、DN を持つ証明書に対して SHA256 アルゴリズムを使用して署名: CN=phil、O=IBM、C=US

```
setmqspl -m QMGR -p MY.OTHER.QUEUE -s SHA256 -e 3DES -r CN=john,O=IBM,C=US -r
CN=jeff,O=IBM,C=US -a CN=phil,O=IBM,C=US
```

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

AMS でのセキュリティー・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。

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。
- キュー・マネージャーに接続してセキュリティー・ポリシーを作成するために必要な権限がなければなりません。
  - z/OS z/OS では、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) で説明されている権限を付与します。
  - Multi z/OS 以外のプラットフォームでは、setmqaut コマンドを使用して、必要な +connect、+inq、+chg の各権限を付与する必要があります。

  セキュリティーの構成の詳細については、125 ページの『セキュリティーのセットアップ』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

以下に、 QMGR という名前のキュー・マネージャーで MYQUEUE という名前のポリシーを作成する例を示します。これは、作成者 (**-a**) の 3DES アルゴリズムを使用してメッセージを暗号化することを指定します。作成者の証明書の識別名 (DN) は CN=alice、O=IBM、C=US であり、 SHA256 アルゴリズムで署名され、受信者 (**-r**) の証明書の DN は CN=j 終了しています。IBM

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -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 SHA256 -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・ポリシーの設定)

セキュリティー・ポリシーの表示とダンプ (*AMS*)

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。
- キュー・マネージャーに接続してセキュリティー・ポリシーを作成するために必要な権限がなければなりません。
  - z/OS z/OS では、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) で説明されている権限を付与します。
  - Multi z/OS 以外のプラットフォームでは、setmqaut コマンドを使用して、必要な +connect、+inq、+chg の各権限を付与する必要があります。

  セキュリティーの構成の詳細については、125 ページの『セキュリティーのセットアップ』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 104. ***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

venus.queue.manager に関する 2 つのセキュリティー・ポリシーを作成する例を以下に示します。

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s sha256 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s sha256 -a "CN=another signer,O=IBM,C=US"
-e NONE
```

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: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

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: SHA256
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 SHA256 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export
```

z/OS z/OS において、エクスポートされたポリシー情報は CSQ0UTIL によって EXPORT DD に書き込まれます。

Multi z/OS 以外のプラットフォームでは、次の例のように、出力をファイルにリダイレクトします。

```
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 Windows で、`policies.bat` を実行します。
- UNIX On UNIX:
  1. `mqm` IBM 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。
- z/OS z/OS では、CSQ0UTIL ユーティリティーを使用して、エクスポートしたポリシー情報を含むデータ・セットを SYSIN に指定します。

関連資料
dspmqspl コマンド属性の完全なリスト

AMS でのセキュリティー・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。

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。
- キュー・マネージャーに接続してセキュリティー・ポリシーを作成するために必要な権限がなければなりません。
  - z/OS z/OS では、メッセージ・セキュリティー・ポリシー・ユーティリティー (CSQ0UTIL) で説明されている権限を付与します。
  - Multi z/OS 以外のプラットフォームでは、setmqaut コマンドを使用して、必要な +connect、+inq、+chg の各権限を付与する必要があります。

  セキュリティーの構成の詳細については、125 ページの『セキュリティーのセットアップ』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て **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

AMS でのシステム・キューの保護

システム・キューは、IBM 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、IBM 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。 634 ページの『AMS でのセキュリティー・ポリシーの作成』を参照。

Windows Windows でシステム・キュー保護を使用するには、`keystore.conf` ファイルを以下のディレクトリーにコピーします。

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS z/OS で `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` にアクセスできる必要があります。これらには、コマンド・サーバーが鍵および証明書にアクセスできるようにするための鍵と構成が含まれています。 `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。 IBM 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、AMS によって保護されません。

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
- z/OS SYSTEM.ADMIN.COMMAND.QUEUE
- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- z/OS SYSTEM.BROKER.CLIENTS.DATA
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- z/OS SYSTEM.BROKER.SUBSCRIPTIONS.DATA
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- z/OS SYSTEM.CHLAUTH.DATA.QUEUE

- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- z/OS SYSTEM.COMMAND.INPUT
- z/OS SYSTEM.DDELAY.LOCAL.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL
- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- z/OS SYSTEM.JMS.PS.STATUS.QUEUE
- z/OS SYSTEM.JMS.REPORT.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- z/OS SYSTEM.QSG.CHANNEL.SYNCQ
- z/OS SYSTEM.QSG.TRANSMIT.QUEUE
- z/OS SYSTEM.QSG.UR.RESOLUTION.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- z/OS SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

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、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 (IBM 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
```

注 : Advanced Message Security 7.0.1 を使用してクライアントをキュー・マネージャーに接続する場合、これらの OAM 権限のみ設定する必要があります。

重要 : すべてのシチュエーションで、SYSTEM.PROTECTION.POLICY.QUEUE に対する参照権限が必要となるわけではありません。 IBM MQ は、 SYSTEM.PROTECTION.POLICY.QUEUE をキューに入れます。

IBM MQ は使用可能なすべてのポリシーを必ずしもキャッシュに入れるわけではありません。 ポリシー数が多い場合、IBM MQ は限られた数のポリシーをキャッシュします。 そのため、キュー・マネージャーに含まれる定義済みポリシー数が少ない場合には、SYSTEM.PROTECTION.POLICY.QUEUE に対する参照オプションを提供する必要はありません。

しかし、定義されているポリシー数が多い場合や古いクライアントを使用している場合には、このキューに対する参照権限を付与してください。 SYSTEM.PROTECTION.ERROR.QUEUE は、AMS コードが生成するエラー・メッセージを入れるときに使用されます。 このキューに対する書き込み権限がチェックされるのは、このキューにエラー・メッセージを書き込もうとする場合のみです。 AMS 保護キューに対するメッセージの書き込みや取得を行うときには、このキューに対する書き込み権限はチェックされません。

セキュリティー権限の付与

コマンド・リソース・セキュリティーを使用する場合、Advanced Message Security が機能することを許可する権限をセットアップする必要があります。 このトピックの例では、RACF コマンドを使用します。 自社で異なる外部セキュリティー・マネージャー (ESM) が使用されている場合、その ESM 用の同等のコマンドを使用する必要があります。

セキュリティー権限の付与には以下の 3 つの側面があります。

- 640 ページの『AMSM アドレス・スペース』
- 640 ページの『CSQ0UTIL』
- 641 ページの『Advanced Message Security ポリシーが定義されたキューの使用』

注 : 例のコマンドでは、以下の変数が使用されています。

1. *QMgrName* - キュー・マネージャーの名前。

   z/OS 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。
2. *username* - これはグループ名にすることができます。
3. 以下の例には、MQQUEUE クラスが示されています。 これは、MXQUEUE、GMQQUEUE、または GMXQUEUE でも構いません。 詳しくは、197 ページの『キュー・セキュリティーのためのプロファイル』を参照してください。

さらに、プロファイルが既に存在している場合、RDEFINE コマンドは必要ありません。

AMSM アドレス・スペース

IBM MQ アドレス・スペースを実行するユーザー名に対して、いくつかの Advanced Message Security セキュリティーを発行する必要があります。

- キュー・マネージャーへのバッチ接続の場合、次を発行します。

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.POLICY.QUEUE へのアクセスの場合、次を発行します。

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

ユーザーに **setmqspl** コマンドおよび **dspmqspl** コマンドの実行を許可するユーティリティーには、以下の権限が必要です。この場合、ユーザー名がジョブ・ユーザー ID になります。

- キュー・マネージャーへのバッチ接続の場合、次を発行します。

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.POLICY.QUEUE へのアクセス (**setmqpol** コマンドに必要) の場合、次を発行します。

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
             PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- SYSTEM.PROTECTION.POLICY.QUEUE へのアクセス (**dspmqpol** コマンドに必要) の場合、次を発行します。

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
               PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

Advanced Message Security ポリシーが定義されたキューの使用

ポリシーが定義されたキューに対してアプリケーションが処理を実行するとき、そのアプリケーションには、メッセージを保護することを Advanced Message Security に許可する追加の権限が必要になります。

アプリケーションには、以下が必要です。

- SYSTEM.PROTECTION.POLICY.QUEUE への読み取り権限。 以下を発行してこれを行います。

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.ERROR.QUEUE への書き込み権限。 以下を発行してこれを行います。

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i 証明書と鍵ストア構成ファイルのセットアップ (IBM i)

Advanced Message Security 保護をセットアップするには、まず証明書を作成し、それを環境に関連付けます。 関連付けは、統合ファイル・システム (IFS) にあるファイルを使用して構成します。

手順

1. IBM i に付属の OpenSSL ツールを使用して自己署名証明書を作成するには、QShell から以下のコマンドを実行します。

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   このコマンドを実行すると、新しい自己署名証明書に関する、以下のようなさまざまな識別名属性を求めるプロンプトが表示されます。

   - 共有名 (CN=)
   - 組織 (O=)
   - 国 (C=)

   これによって、暗号化されていない秘密鍵とそれに対応する証明書が、両方とも PEM (Privacy Enhanced Mail) 形式で作成されます。

   簡単に説明するために、共通名、組織、国の値を入力します。 これらの属性と値は、ポリシーの作成時に重要です。

追加のプロンプトおよび属性は、コマンド行で **-config** パラメーターを使用してカスタム openssl 構成ファイルを指定することでカスタマイズできます。 この構成ファイルの構文について詳しくは、OpenSSL の資料を参照してください。

例えば、次のコマンドは、追加の X.509 v3 証明書拡張を追加します。

```
/QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
-keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
```

ここで、myconfig.cnf は、以下のように設定された ASCII ストリーム・ファイルです。

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = myextensions

[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = GB
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Hants
localityName = Locality Name (eg, city)
localityName_default = Hursley
organizationName = Organization Name (eg, company)
organizationName_default = IBM United Kingdom
organizationalUnitName = Organizational Unit Name (eg, department)
organizationalUnitName_default = IBM MQ Development
commonName = Common Name (eg, Your Name)

[myextensions]
keyUsage = digitalSignature,nonRepudiation,dataEncipherment,keyEncipherment
extendedKeyUsage = emailProtection
```

2. AMS では、証明書と秘密鍵の両方が同じファイル内に含まれている必要があります。 そのためには、以下のコマンドを実行します。

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

これで、$HOME にある private.pem ファイルに、対応する秘密鍵と証明書が入っています。 一方、mycert.pem ファイルには、メッセージの暗号化と署名の検証を行うことのできるすべてのパブリック証明書が入っています。

2 つのファイルは、デフォルト・ロケーションに鍵ストア構成ファイル keystore.conf を作成することによって、ご使用の環境に関連付ける必要があります。

デフォルトでは、AMS は、ホーム・ディレクトリーの .mqs サブディレクトリーにある鍵ストア構成を検索します。

3. QShell では、keystore.conf ファイルを作成します。

```
mkdir -p $HOME/.mqs
echo "pem.private = $HOME/private.pem" > $HOME/.mqs/keystore.conf
echo "pem.public = $HOME/mycert.pem" >> $HOME/.mqs/keystore.conf
echo "pem.password = unused" >> $HOME/.mqs/keystore.conf
```

IBM i ポリシーの作成 *(IBM i)*

ポリシーを作成する前に、保護メッセージを保持するためのキューを作成する必要があります。

手順

1. コマンド行プロンプトで、以下のように入力します。

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

ここで、mqmname は、キュー・マネージャーの名前です。

DSPMQM コマンドを使用して、キュー・マネージャーでセキュリティー・ポリシーを使用できることを確認します。 **Security Policy Capability** に **YES* が表示されていることを確認します。

定義できる最も単純なポリシーは、整合性ポリシーです。これを定義するには、デジタル署名アルゴリズムを持つが暗号化アルゴリズムを持たないポリシーを作成します。

メッセージは署名されますが、暗号化されません。 メッセージを暗号化する場合は、暗号化アルゴリズムと、1 つ以上の所定のメッセージ受信者を指定する必要があります。

所定のメッセージ受信者の公開鍵ストア内の証明書は、識別名で識別されます。

2. QShell で以下のコマンドを使用して、$HOME にある公開鍵ストア mycert.pem 内の証明書の識別名を表示します。

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

所定の受信者としての識別名を入力する必要があります。また、ポリシー名が保護対象のキュー名と一致する必要があります。

3. CL コマンド・プロンプトで、例えば以下のように入力します。

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

ここで、mqmname は、キュー・マネージャーの名前です。

ポリシーが作成されると、そのキュー名で書き込み、参照、または破壊的な除去が行われるメッセージはすべて、AMS ポリシーの対象になります。

関連資料

メッセージ・キュー・マネージャーの表示 (DSPMQM)

MQM セキュリティー・ポリシーの設定 (SETMQMSPL)

IBM i ポリシーのテスト *(IBM i)*

製品と共に提供されているサンプル・アプリケーションを使用して、セキュリティー・ポリシーをテストします。

このタスクについて

IBM MQ と共に提供されているサンプル・アプリケーションの AMQSPUT4、AMQSGET4、AMQSGBR4 や、WRKMQMMSG などのツールを利用して、PROTECTED キュー名を使用したメッセージの書き込み、ブラウズ、取得 を行うことができます。

すべてが適切に構成されている場合、このユーザーに対して無保護のキューとアプリケーションの動作に違いはありません。

ただし、Advanced Message Security で設定されていないユーザー、またはメッセージの暗号化解除に必要な秘密鍵を持っていないユーザーは、メッセージを表示できません。 ユーザーは MQCC_FAILED (2) と同等の 完了コード RCFAIL、および理由コード RC2063 (MQRC_SECURITY_ERROR) を受信します。

AMS の保護が有効になっていることを確認するには、AMQSPUT0 などを使用してテスト・メッセージを PROTECTED キューに書き込みます。 それから、別名キューを作成して、未加工の保護データをブラウズします。

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。

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

AMQSBCG4 や WRKMQMMSG など、ALIAS キュー名を使用してブラウズすると、より大きい scrambled メッセージがわかり、PROTECTED キューのブラウズで平文のメッセージが表示されます。

scrambled メッセージは表示されますが、AMS がこの名前との突き合わせを実施するポリシーがないため、ALIAS キューを使用して元の平文は復号されません。 このため、未加工の保護データが返されます。

関連資料

MQM セキュリティー・ポリシーの設定 (SETMQMSPL)

MQ メッセージの処理 (WRKMQMMSG)

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。

IBM 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、イベントが発生したキュー・マネージャー上の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 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 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。

- 634 ページの『AMS でのセキュリティー・ポリシーの作成』。以下の 2 つの IBM 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
- 634 ページの『AMS でのセキュリティー・ポリシーの変更』。以下の 3 つの IBM 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
- 635 ページの『セキュリティー・ポリシーの表示とダンプ (AMS)』。以下の 1 つの IBM MQ イベント・メッセージを生成します。
  - コマンド・イベント
- 637 ページの『AMS でのセキュリティー・ポリシーの削除』。以下の 2 つの IBM 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

IBM 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。)
Identifier:	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。
Identifier:	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
IBM MQ Explorer・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。)

Identifier: 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)。

Identifier: 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)。

Identifier: 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)。

Identifier: 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)。

Identifier: 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)。

Identifier: 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。

コマンド

説明: コマンド・コード。

Identifier: 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 IBM MQ 8.0 `cmqcfc.h` で定義される

戻り: 常時。

CommandData には、PCF コマンドを構成した PCF エレメントが含まれています。

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 メッセージ記述子の有効な値については、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ています。 有効な 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。)

Identifier: **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。

Identifier: **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。

Identifier: **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
IBM 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。)

Identifier: **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

Identifier: **MQIACF_OBJECT_TYPE**

データ型: MQCFIN

値: **MQOT_PROT_POLICY**
Advanced Message Security 保護ポリシー。 **1019** - IBM MQ 8.0 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。

Identifier: **MQCA_POLICY_NAME**

データ型: MQCFST

値: 「**2112**」 - IBM MQ 8.0 または cmqc.h ファイル内で定義された数値。

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。

Identifier: **MQIA_POLICY_VERSION**

データ型: MQCFIN

値 「**238**」 - IBM MQ 8.0 または cmqc.h ファイル内で定義された数値。

戻り: 常時

TolerateFlag

説明: Advanced Message Security ポリシーの容認フラグ。

Identifier: **MQIA_TOLERATE_UNPROTECTED**

データ型: MQCFIN

値 「**235**」 - IBM MQ 8.0 または cmqc.h ファイル内で定義された数値。

戻り: 常時。

SignatureAlgorithm

説明: Advanced Message Security ポリシーの署名アルゴリズム。

Identifier: **MQIA_SIGNATURE_ALGORITHM**

データ型: MQCFIN

値: 「**236**」 - IBM MQ 8.0 または cmqc.h ファイル内で定義された数値。

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。

Identifier: **MQIA_ENCRYPTION_ALGORITHM**

データ型: MQCFIN

値: 「**237**」 - IBM MQ 8.0 または cmqc.h ファイル内で定義された数値。

戻り: IBM 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。

Identifier: **MQCA_SIGNER_DN**

データ型: MQCFSL

値: 「**2113**」 - IBM MQ 8.0 または `cmqc.h` ファイル内で定義された数値。

最大長: ポリシー内の最長の署名者 DN (MQ_DISTINGUISHED_NAME_LENGTH 以内)

戻り: IBM 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。

Identifier: **MQCA_RECIPIENT_DN**

データ型: MQCFSL

値: 「**2114**」 - IBM MQ 8.0 または `cmqc.h` ファイル内で定義された数値。

最大長: ポリシー内の最長の受信者 DN (MQ_DISTINGUISHED_NAME_LENGTH 以内)

戻り: IBM MQ ポリシーに定義されている場合

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 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: