9.4

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、695 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 4、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

IBM MQ の保護

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。 絶対最小限として、セキュア・ゾーン内のすべてのハードウェアとソフトウェア、およびオペレーター・ワークステーション上のすべてのハードウェアとソフトウェアがサポート・ライフサイクル内にあり、必須ソフトウェア更新を使用して最新状態になっており、セキュリティー更新が迅速に適用されていることを確認する必要があります。

関連資料

IBM セキュリティー脆弱性管理

IBM Z および LinuxOne セキュリティー・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。

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。

- 7 ページの『識別と認証』
- 9 ページの『認証』
- 9 ページの『監査』
- 10 ページの『機密性』
- 10 ページの『データ整合性』

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

- 11 ページの『暗号化方式』
- 13 ページの『メッセージ・ダイジェストとデジタル署名』
- 13 ページの『デジタル証明書』
- 17 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、87 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。

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。

IBM MQ による識別と認証

アプリケーションが IBM MQ に接続すると、ユーザー ID は常に接続に関連付けられます。 ユーザー ID は、最初はアプリケーション・プロセスに関連付けられたオペレーティング・システム・ユーザー ID です。 多くの場合、この ID は、キュー・マネージャーと同じシステムでホストされているローカルにバインドされたアプリケーションには十分です。 ただし、キュー・マネージャーは、いくつかの方法で接続に関

連付けられた ID を認証および変更することもできます。 信頼できるとは限らないクライアント・アプリケーションがネットワークを介してキュー・マネージャーに接続する場合は、接続に関連付けられた ID を認証することが重要です。

IBM MQ キュー・マネージャーへのアプリケーション接続に関連付けられた ID は、以下のいずれかのメカニズムを使用して確立できます。

- アプリケーションは、キュー・マネージャーに接続するときに、ユーザー ID とパスワードを提供することができます。 キュー・マネージャーは、その構成に基づいて資格情報を検証します。 例えば、ユーザー ID とパスワードをキュー・マネージャーのオペレーティング・システム、または認証される LDAP サーバーに渡すことができます。
- V 9.4.0 IBM MQ 9.3.4 以降、アプリケーションは、外部認証サーバーから取得した認証トークンを提供することもできます。 認証トークンについて詳しくは、 325 ページの『認証トークンの処理』を参照してください。
- クライアント・チャネルは、有効なデジタル証明書を使用して構成されている場合、TLS 相互認証を使用するように構成できます。 TLS 認証をチャネル認証 (CHLAUTH) 規則と組み合わせて、適切なユーザー ID を接続に関連付けることができます。 詳しくは、 20 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。
- チャネル認証 (CHLAUTH) 規則は、接続に関する情報に基づいて ID をオーバーライドできます。 例えば、チャネル認証規則は、クライアントの IP アドレスに基づいて、接続に関連付けられたユーザー ID を設定できます。
- カスタム終了コードは、選択した基準に基づいて ID を設定できます。

ID と認証は、2 つのキュー・マネージャー間のチャネルにも適用できます。 これらのチャネルは、メッセージ・チャネルと呼ばれます。 メッセージ・チャネルの開始時に、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) は、そのパートナーを認証できます。 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。 同様に、受信側 MCA は、本物のパートナーからメッセージを受信しようとしていることが保証されます。

ID が確立され、必要に応じて認証されると、 IBM MQ はその ID をいくつかの方法で使用します。

- 重要なことは、デフォルトでは、この ID を使用して後続の 9 ページの『認証』 検査が行われることです。 例えば、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、キュー・マネージャーは、アプリケーションに関連付けられている ID にキュー・オブジェクトに対する「put」権限があることを確認します。
- さらに、すべてのメッセージに メッセージ・コンテキスト 情報を含めることができます。 この情報は、メッセージ記述子 (MQMD) に保持されます。 キュー・マネージャーは、アプリケーションがメッセージをキューに書き込むときに、メッセージ・コンテキストを自動的に生成することができます。 あるいは、アプリケーションに関連付けられているユーザー ID がメッセージ・コンテキストの提供を許可されている場合は、アプリケーションがそのメッセージ・コンテキストを提供することもできます。 メッセージ内のこのコンテキスト情報は、メッセージの発信元に関するメッセージ情報を受け取るアプリケーションに与えます。 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。

否認防止

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。

認証

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。

許可のセットアップ方法について詳しくは、91 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。

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。
- アプリケーションがチャネルを開こうとしましたが、TLS 接続が許可されていないため、失敗しました。 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。

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。

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。
- 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。 for more information.

データ整合性

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

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

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

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

IBM MQ でのデータ保全性

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を受信時に検出します。変更されたメッセージを受信すると、 IBM MQ AMQ9661 エラー・メッセージがエラー・ログに書き込まれ、チャネルは停止します。
- 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が検出された場合、メッセージを受信しようとするアプリケーションは、MQRC_SECURITY_ERROR (2063) 戻りコードを受け取ります。 アプリケーションが MQGET 呼び出しを使用している場合、メッセージは SYSTEM.PROTECTION.ERROR.QUEUE キュー。

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。

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。 詳しくは、22 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。 共有鍵を使用するアルゴリズムは、対称 アルゴリズムと呼ばれます。 12 ページの図 1 は、対称鍵暗号化方式を図示しています。
- 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。 12 ページの図 2 は、公開鍵暗号化方式 とも呼ばれる非対称鍵暗号化方式を示しています。

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

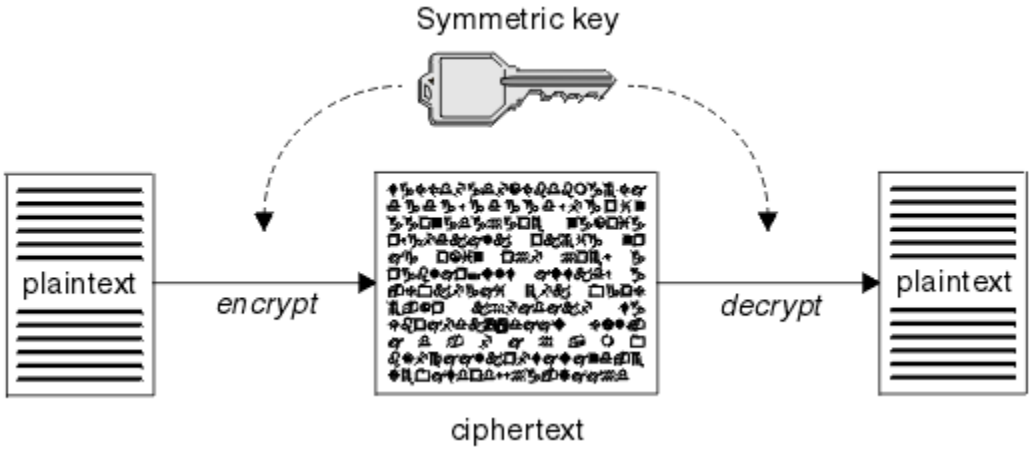


図 *1.* 対称鍵暗号化方式

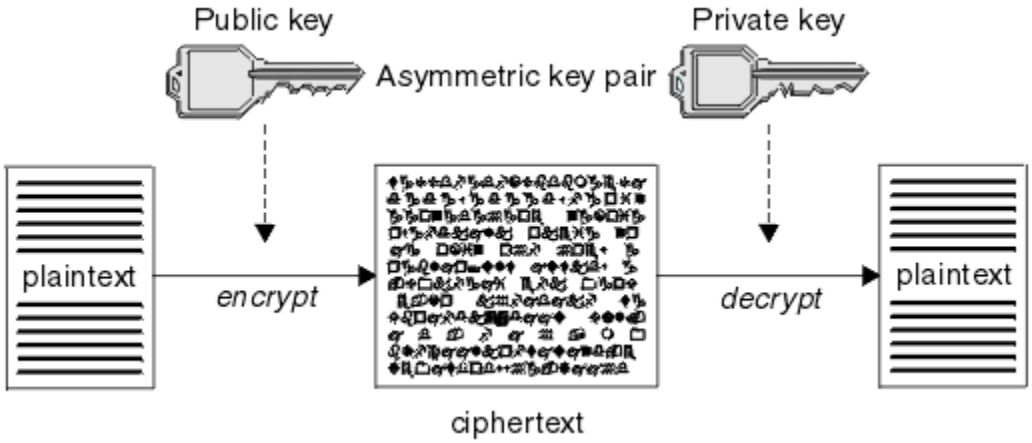


図 *2.* 非対称鍵暗号化方式

12 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
22 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、15 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、45 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、339 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	E メール・アドレス
Deprecated 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	タイトル
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
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

14 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。

- ALW AIX, Linux, and Windows での **runmqakm** および V 9.4.0 V 9.4.0 **runmqktool** コマンド。
- 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。

16 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。

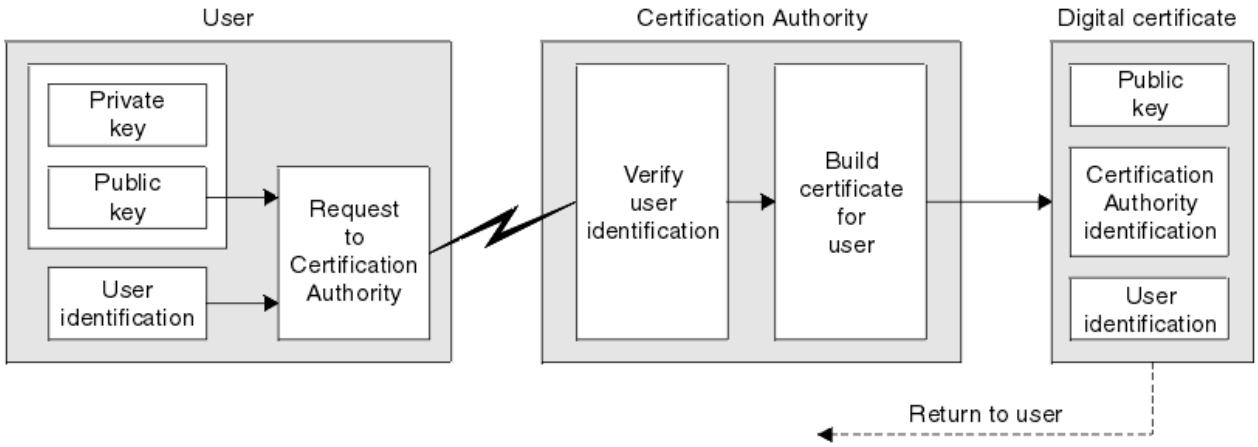


図 3. デジタル証明書の取得

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、14 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。

17 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。

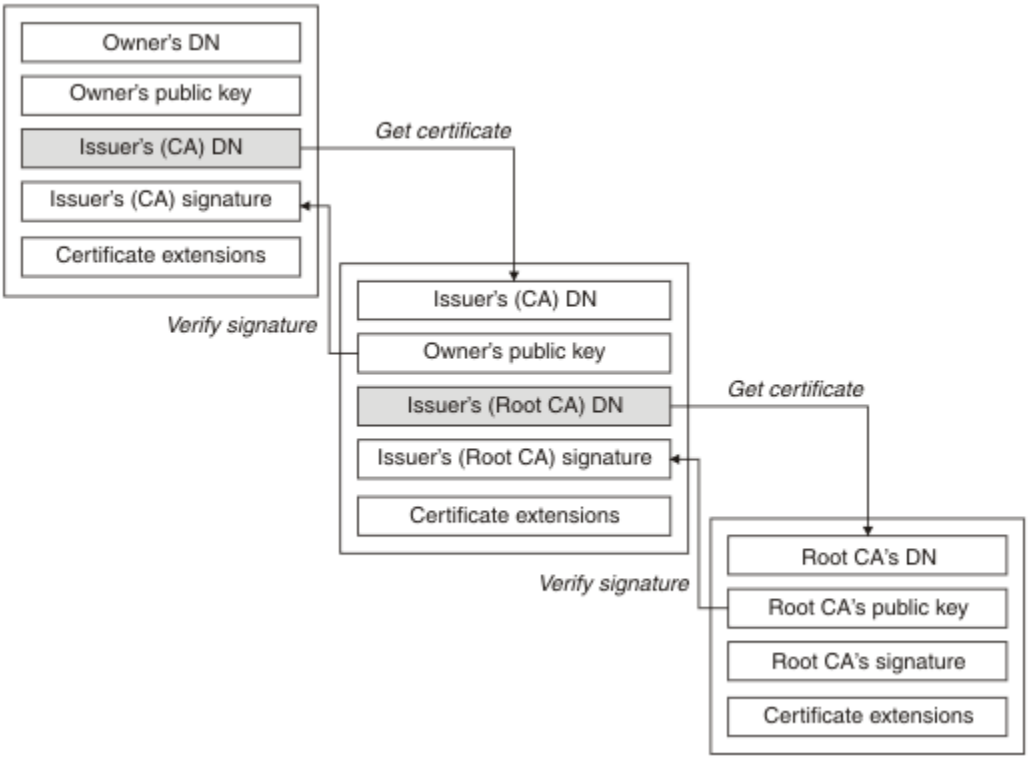


図 *4.* トラストのチェーン

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 (AIX, 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、339 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、13 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

24 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、420 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://developer.mozilla.org/docs/Mozilla/Projects/NSS 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、20 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、20 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。

20 ページの図 5 は、TLS ハンドシェークを示しています。

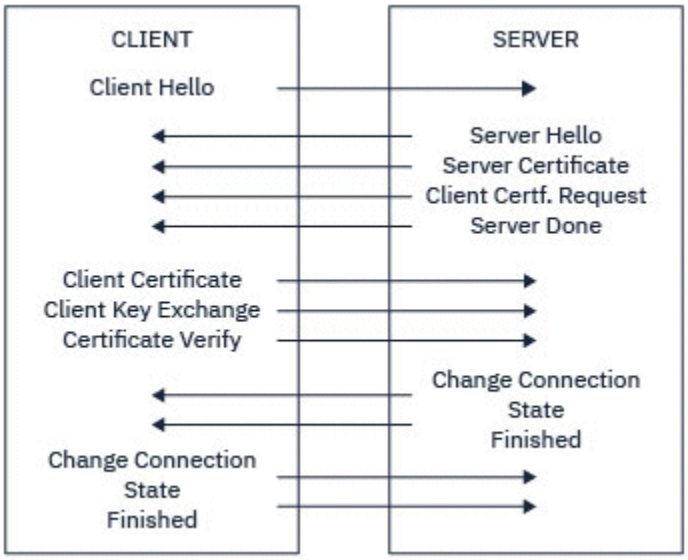


図 *5. TLS* ハンドシェークの概要

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、 18 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。 ランダムなバイト・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 (概要のステップ 19 ページの『4』) 。

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 19 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 19 ページの『7』 と 19 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 19 ページの『3』 および 19 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 (22 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 (16 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 (339 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。 詳しくは、13 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、11 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、476 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、420 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、TLS1.3 と TLS1.2 のプロトコルと CipherSpec 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、420 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、420 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。

23 ページの図 6 には、このプロセスが図示されています。

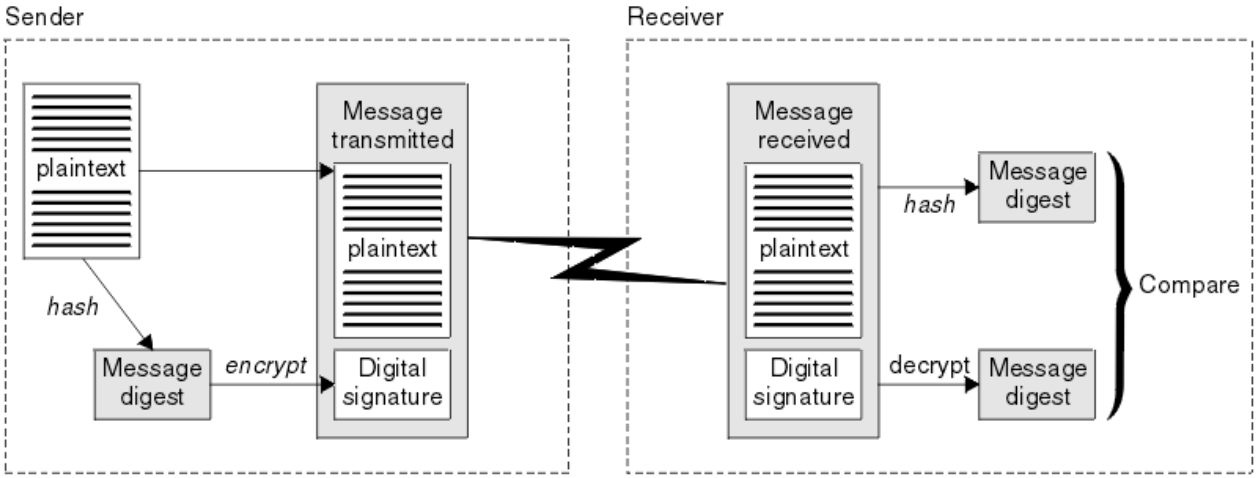


図 6. デジタル署名のプロセス

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。

注 : AIX, Linux, and Windows では、 IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。 このモジュールの証明書は「履歴」ステータスに移動されました。 お客様は、IBM Crypto for C (ICC) 証明書 を表示し、NIST から提供されたアドバイスに注意する必要があります。 交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、 「NIST CMVP modules in process list」 でそのモジュールを検索します。

IBM MQ Operator 3.2.0 およびキュー・マネージャー・コンテナー・イメージ 9.4.0.0 以降は、UBI 9 に基づいています。 FIPS 140-3 準拠は現在保留中であり、その状況を表示するには、 「NIST CMVP modules in process list」 で「Red Hat Enterprise Linux 9- OpenSSL FIPS Provider」を検索します。

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

24 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。

AIX, Linux, and Windows 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、42 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

23 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、さまざまなセキュリティー概念を実装する IBM MQ の特定のメカニズムについて説明します。

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 プロトコルのバージョン 1.2 およびバージョン 1.3 をサポートしています。 以前のバージョンの TLS と SSL は、デフォルトでは有効になっていませんが、必要な場合には有効化できます。 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。

IBM MQ によってサポートされる CipherSpecs のリスト、および非推奨になった CipherSpec の 435 ページの『推奨されない CipherSpec』 のリストについては、 420 ページの『CipherSpecs の有効化』 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、すべてのプラットフォームで TLS 1.2 および TLS 1.3 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。

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。

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

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。

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
18 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
- ALW AIX, 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、13 ページの『デジタル証明書』および 18 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、26 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。

ALW **AIX, 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。 例えば、AIX 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 がデフォルトの場所にインストールされている場合、 Windows での同等のパスは `C:\ProgramData\IBM \MQ\Qmgrs\QM1\ssl\key.kdb` です。

鍵データベース・ファイル IBM MQ にアクセスするには、鍵データベースのパスワードを指定する必要があります。 これは、直接行うことも、パスワード・スタッシュ・ファイルを使用して行うこともできます。 パスワード・スタッシュ・ファイルを使用する場合は、そのファイルが同じディレクトリーにあり、鍵データベースと同じファイル・ステムを持つ必要があり、末尾に接尾部 `.sth` を付ける必要があります (例: `/var/mqm/qmgrs/QM1/ssl/key.sth`)。

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。

AIX, Linux, and 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。

- Server Name Indication (SNI) に対応している (つまりチャネルごとの証明書をサポートしている) Java と JMS のクライアント。
- 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、47 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。 キュー・マネージャーでの複数の証明書の使用について詳しくは、30 ページの『IBM MQ で複数の証明書を使用するための機能』 を参照してください。

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し、 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 2 つの別個の TLS セッションがあります。 MQIPT は、SNI をチャネル名に設定するか、インバウンド接続で受信した SNI を経路にパススルーすることによって、宛先キュー・マネージャーが複数の証明書を使用できるように構成できます。 複数の証明書サポートおよび MQIPT について詳しくは、「IBM MQ による MQIPT 複数の証明書サポート」を参照してください。

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 29 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、310 ページの『Locating the key repository for a queue manager on z/OS』 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。

Linux *IBM MQ* で複数の証明書を使用するための機能

Server Name Indication (SNI) は、必要なサービスをクライアント側で明示するための TLS プロトコルの拡張機能です。 これは、IBM MQ の用語ではチャネルに相当します。

IBM MQ では、複数の証明書を別々のチャネルに指定するために SNI 拡張機能を使用しています。その指定のために、チャネル定義で CERTLABL パラメーターを使用します。

IBM MQ で使用する SNI アドレスは、要求対象のチャネルの名前がベースになってい
て、`.chl.mq.ibm.com` という接尾部が付きます。

IBM MQ のチャネル名は、以下のようにして有効な SNI 名にマップされます。

- 大文字の A から Z は小文字に変換されます
- 数字の 0 から 9 は変更なしです
- その他の文字 (小文字の a から z を含む) はすべて、2 桁の 16 進 ASCII 文字コード (小文字) に変換され、その後にハイフンが続きます。
  - 小文字の a から z は、16 進数の `61-` から `7a-` にマップします
  - パーセント (%) は、16 進数の `25-` にマップします
  - ハイフン (-) は、16 進法の `2d-` にマップします
  - ドット (.) は、16 進数の `2e-` にマップします
  - スラッシュ (/) は、16 進数の `2f-` にマップします
  - 下線 (_) は、16 進数の `5f-` にマップします

EBCDIC プラットフォームでは、チャネル名が ASCII に変換されてから、このマッピングが適用されます。

例えば、チャネル名 `TO.QMGR1` は、SNI アドレス `to2e-qmgr1.chl.mq.ibm.com` にマップします。

一方、小文字のチャネル名 `to.qmgr1` は、SNI アドレス `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com` にマップします。

注 : 生成された SNI URL が URL フォーマット指定に準拠している必要がある環境では、例えば、クライアントが Red Hat® OpenShift® 経路を超えて Red Hat OpenShift で実行されているキュー・マネージャーに接続する場合、チャネル名の末尾には小文字を使用することはできません。

SSL スタンザの **OutboundSNI** プロパティーを使用すれば、TLS 接続の開始時に SNI でリモート・システムのターゲット IBM MQ チャネル名を設定するか、それともホスト名を設定するかを選択できます。**OutboundSNI** プロパティーについて詳しくは、qm.ini ファイルの SSL スタンザおよび 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。

複数の証明書を使用するには、SNI が IBM MQ チャネル名に設定されている必要があります。 証明書ラベルが構成されている IBM MQ チャネルへの接続にホスト名、カスタム、または SNI を使用しない場合、接続しているアプリケーションは MQRC_SSL_INITIALIZATION_ERROR で拒否され、リモート・キュー・マネージャーのエラー・ログに AMQ9673 メッセージが出力されます。

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する場合、MQIPT は、SNI をチャネル名に設定するか、宛先キュー・マネージャーが複数の証明書を使用できるようにするために、経路へのインバウンド接続で受信した SNI をパススルーするように構成する必要があります。 複数の証明書サポートおよび MQIPT について詳しくは、IBM MQ に MQIPT 複数の証明書サポートを参照してください。

このプロパティーを使用する方法について詳しくは、Red Hat OpenShift クラスターにデプロイされたキュー・マネージャーへの接続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リフレッシュ

キー・リポジトリーの内容を変更すると、REFRESH SECURITY TYPE (SSL) コマンドが発行されるか、キュー・マネージャーが再始動されるまで、既存のキュー・マネージャー・プロセスは新しい内容を取得しません。

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。

キュー・マネージャーが鍵ストアの内容を変更した後に (`amqrmppa` または **`runmqchl`** を使用して) 新しいチャネル・プロセスを作成すると、新しいプロセスは新しい証明書の使用を即時に開始し、既存のプロセスは引き続き鍵ストアのキャッシュ・コピーを使用します。 詳細については、306 ページの『AIX, Linux, and Windows 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』を参照してください。

REFRESH SECURITY TYPE (SSL) コマンドを発行するまでは、実行中の複数のチャネルが異なるバージョンのキー・リポジトリーを使用する可能性があることに注意してください。

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

31 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

31 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』
キー・リポジトリーの内容を変更すると、REFRESH SECURITY TYPE (SSL) コマンドが発行されるか、キュー・マネージャーが再始動されるまで、既存のキュー・マネージャー・プロセスは新しい内容を取得しません。

MQCSP パスワード保護

MQCSP 構造で指定された認証資格情報は、 IBM MQ MQCSP パスワード保護機能を使用して保護することも、TLS 暗号化を使用して暗号化することもできます。

IBM MQ client アプリケーションは、キュー・マネージャーに接続するときにユーザー ID とパスワードを提供できます。 V 9.4.0 IBM MQ 9.4.0 以降、アプリケーションは、代替認証方式として認証トークンを提供することもできます。 これらの資格情報は、MQCSP 構造でキュー・マネージャーに送信されます。

チャネルが TLS 暗号化を使用している場合、MQCSP 内の資格情報は TLS 暗号仕様に従って暗号化されます。 チャネルが TLS 暗号化を使用していない場合、 IBM MQ は、これらの資格情報がネットワーク経由で送信される前にそれらの資格情報を保護して、プレーン・テキストでネットワーク経由で資格情報が送信されないようにすることができます。 これらの資格情報を保護する IBM MQ 機能は、MQCSP パスワード保護と呼ばれます。

MQCSP パスワード保護が使用されている場合、MQCSP 構造内の以下のデータが保護されます。

- パスワード (MQCSP.AuthenticationType フィールドが MQCSP_AUTH_USER_ID_AND_PW に設定されている場合)。
- V 9.4.0 認証トークン (MQCSP.AuthenticationType フィールドが MQCSP_AUTH_ID_TOKEN に設定されている場合)。

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。 実動目的の場合、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、TLS 暗号化の方が安全であるため、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。

どの暗号化が使用されているか、およびどの程度の保護が提供されているかについて懸念がある場合は、完全な TLS 暗号化を使用する必要があります。 TLS を使用すると、アルゴリズムが公開され、**SSLCIPH** チャネル属性を使用して企業に適したアルゴリズムを選択できます。

MQCSP 構造の詳細については、MQCSP 構造を参照してください。

MQCSP 構造の資格情報は、以下のすべての条件が満たされている場合に IBM MQ パスワード保護を使用して保護されます。

- 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。 チャネルにブランクの **SSLCIPH** 属性がある場合、または **SSLCIPH** 属性が暗号化を提供しない暗号仕様に設定されている場合、チャネルは TLS 暗号化を使用しません。 NULL_SHA などのヌル暗号は、暗号化を提供しません。
- MQCSP.AuthenticationType フィールドは、 MQCSP_AUTH_USER_ID_AND_PWD または MQCSP_AUTH_ID_TOKEN に設定されます。 MQCSP.AuthenticationType フィールドについて詳しくは、「**AuthenticationType**」を参照してください。
- クライアントが IBM MQ Explorer であり、ユーザー識別互換モードが有効になっていない場合。 このモードは、 IBM MQ Explorer がユーザー ID とパスワードを送信するために使用するデフォルト・モードではありません。 この条件は、 IBM MQ Explorer にのみ適用されます。

これらの条件のいずれかが満たされない場合、資格情報は MQCSP パスワード保護で保護されません。 **PasswordProtection** 属性の値がプレーン・テキストでの資格情報の送信を禁止しており、チャネルが TLS 暗号化を使用していない場合、接続は失敗し、MQRC_PASSWORD_PROTECTION_ERROR (2594) 理由コードが返されます。

PasswordProtection 構成設定

クライアントおよびキュー・マネージャーの構成ファイルの **Channels** スタンザ内の **PasswordProtection** 属性により、資格情報がプレーン・テキストで送信されないようにすることができます。

注 : この属性は、TLS 暗号化を使用しない接続にのみ関係します。 接続で TLS 暗号化が使用されている場合、資格情報は、MQCSP パスワード保護で保護されるのではなく、TLS を使用して暗号化されます。

この属性は、以下のいずれかの値に設定できます。 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、資格情報はプレーン・テキストで送信されます。 つまり、MQCSP パスワード保護をサポートしないバージョンの IBM MQ との互換性のために、資格情報をプレーン・テキストでネットワーク経由で送信できます。

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、資格情報は MQCSP パスワード保護によって保護されます。

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、 MQCSP.AuthenticationType フィールドが MQCSP_AUTH_USER_ID_AND_PW または MQCSP_AUTH_ID_TOKEN に設定されていない場合、資格情報が送信される前に接続が失敗します。

always
資格情報は、ネットワークを介して無保護で送信してはなりません。

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、資格情報は MQCSP パスワード保護によって保護されます。

以下の場合、資格情報が送信される前に接続が失敗します。

- MQCSP.AuthenticationType フィールドが MQCSP_AUTH_USER_ID_AND_PW または MQCSP_AUTH_ID_TOKEN 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 を実行している。

オプション

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、 MQCSP.AuthenticationType フィールドが MQCSP_AUTH_USER_ID_AND_PW または MQCSP_AUTH_ID_TOKEN に設定されている場合、資格情報は MQCSP パスワード保護によって保護されます。 それ以外の場合、資格情報はプレーン・テキストで送信されます。

warn

任意のクライアントがプレーン・テキストの資格情報を送信できます。 プレーン・テキストの資格情報を受信すると、警告メッセージ AMQ9297W がキュー・マネージャーのエラー・ログに書き込まれます。

このオプションは、キュー・マネージャー構成ファイルでのみ指定できます。

Java および JMS クライアントの場合、 **PasswordProtection** 属性の動作は、クライアントが互換モードを使用するか MQCSP モードを使用するかによって異なります。

- Java 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、クライアントの接続時にユーザー ID とパスワードを送信するために MQCSP 構造体は使用されません。 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に対して記述されている動作と同じです。
- Java 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** 属性の動作は説明どおりの動作になります。

Java および JMS クライアントとの接続認証について詳しくは、 84 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。

MQCSP パスワード保護と MQIPT

V 9.4.0

クライアントが IBM MQ Internet Pass-Thru (MQIPT) を介してキュー・マネージャーに接続する場合、TLS 暗号化を追加または削除するように MQIPT 経路が構成されている可能性があります。 つまり、 MQIPT 経路が SSLServer=true と SSLClient=false、または SSLServer=true と SSLClient=false で構成されている可能性があります。 この場合、チャネルの一方が TLS 暗号化を使用しており、もう一方が TLS 暗号化を使用していないため、クライアントとキュー・マネージャーがパスワード保護アルゴリズムに同意しない可能性があります。 これにより、接続は理由コード MQRC_PASSWORD_PROTECTION_ERROR (2594) で失敗します。

IBM MQ 9.4.0 以降、 MQIPT は、TLS 暗号化を追加または削除する MQIPT 経路のクライアントとキュー・マネージャーの間の互換性を維持するために、MQCSP 構造の資格情報の保護を追加または削除できます。 MQIPT での MQCSP パスワード保護は、 **PasswordProtection** 経路プロパティーを使用して構成されます。

PasswordProtection プロパティーのデフォルト値は required です。 この値は、 MQIPT が MQCSP パスワード保護を追加することはできるが、除去することはできないことを意味します。 TLS 暗号化を追加する MQIPT 経路への接続は、この値が **PasswordProtection** の理由コード MQRC_PASSWORD_PROTECTION_ERROR (2594) で失敗する場合があります。 この問題を解決するには、MQIPT 経路構成で **PasswordProtection** プロパティーの値を compatible に設定します。

MQIPT の **PasswordProtection** プロパティーについて詳しくは、 PasswordProtection 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。

注 : AIX, Linux, and Windows では、 IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。 このモジュールの証明書は「履歴」ステータスに移動されました。 お客様は、IBM Crypto for C (ICC) 証明書 を表示し、NIST から提供されたアドバイスに注意する必要があります。 交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、 「NIST CMVP modules in process list」 でそのモジュールを検索します。

IBM MQ Operator 3.2.0 およびキュー・マネージャー・コンテナー・イメージ 9.4.0.0 以降は、UBI 9 に基づいています。 FIPS 140-3 準拠は現在保留中であり、その状況を表示するには、 「NIST CMVP modules in process list」 で「Red Hat Enterprise Linux 9- OpenSSL FIPS Provider」を検索します。

この情報は、次のプラットフォームに当てはまります。

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

ALW AIX, Linux, and Windows での IBM MQ TLS 接続の FIPS 140-2 準拠について詳しくは、35 ページの『AIX, 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、38 ページの『Federal Information Processing Standards (FIPS) for z/OS』 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

269 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。

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

542 ページの『AIX, Linux, and Windows での runmqakm および runmqktool コマンド』
AIX, Linux, and Windows システムでは、 **runmqakm** (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵および証明書を管理します。

23 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。

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

AIX, Linux, and Windows 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)という暗号化パッケージを使用します。 AIX, Linux, and Windows プラットフォームでは、 ICC ソフトウェアは、米国連邦情報・技術局の連邦情報処理標準 (FIPS) 暗号モジュール評価プログラム (レベル 140-2) に合格しています。

注 : AIX, Linux, and Windows では、 IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。 このモジュールの証明書は「履歴」ステータスに移動されました。 お客様は、IBM Crypto for C (ICC) 証明書 を表示し、NIST から提供されたアドバイスに注意する必要があります。 交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、 「NIST CMVP modules in process list」でそのモジュールを検索します。

IBM MQ Operator 3.2.0 およびキュー・マネージャー・コンテナー・イメージ 9.4.0.0 以降は、UBI 9 に基づいています。 FIPS 140-3 準拠は現在保留中であり、その状況を表示するには、 「NIST CMVP modules in process list」で「Red Hat Enterprise Linux 9- OpenSSL FIPS Provider」を検索します。

AIX, Linux, and Windows 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 IBM Global Security Kit (GSKit) ICC のバージョンは、インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーで FIPS 140-2 に準拠していることが認証されています。
  - 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。
  - すべての鍵リポジトリーへのアクセスは、キュー・マネージャーの **KEYRPWD** 属性ではなく、stash ファイルを使用して提供されます。
- すべての IBM MQ MQI client アプリケーションで、以下の条件が満たされている場合、接続は GSKit を使用し、FIPS 準拠です。

  - インストールされている GSKit ICC のバージョンは、インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーで FIPS 140-2 に準拠していることが認証されています。
  - 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。
  - すべての鍵リポジトリーへのアクセスは、鍵リポジトリーのパスワード・メカニズムではなく、stash ファイルを使用して提供されます。
- 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。
- 非管理対象 .NET クライアント・アプリケーションの場合、以下の条件が満たされると、接続は GSKit を使用し、FIPS 準拠になります。
  - インストールされている GSKit ICC のバージョンは、インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーで FIPS 140-2 に準拠していることが認証されています。
  - .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。
  - すべての鍵リポジトリーへのアクセスは、鍵リポジトリーのパスワード・メカニズムではなく、stash ファイルを使用して提供されます。
- 非管理対象 XMS .NET クライアント・アプリケーションの場合、以下の条件が満たされると、接続は GSKit を使用し、FIPS 準拠になります。
  - インストールされている GSKit ICC のバージョンは、インストールされているオペレーティング・システムのバージョンおよびハードウェア・アーキテクチャーで FIPS 140-2 に準拠していることが認証されています。
  - XMS .NET 資料で説明されているように、FIPS 認定の暗号化のみを使用することを指定しました。
  - `-fips` オプションが指定された **runmqakm** などの FIPS 準拠のソフトウェアのみを使用して、すべての鍵リポジトリーが作成および操作されている。
  - すべての鍵リポジトリーへのアクセスは、鍵リポジトリーのパスワード・メカニズムではなく、stash ファイルを使用して提供されます。

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。 任意のプラットフォームで GSKit FIPS 準拠を決定するのは、このコンポーネントのバージョンです。 現在インストールされている ICC のバージョンを確認するには、 **dspmqver -p 64 -v** コマンドを実行します。

以下に、 ICC に関連する **dspmqver -p 64 -v** 出力の抜粋例を示します。

```
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 に含まれている) の NIST 認定ステートメントは、 Cryptographic Module Validation Program のアドレスにあります。

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。

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、465 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

269 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。

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

542 ページの『AIX, Linux, and Windows での runmqakm および runmqktool コマンド』

AIX, Linux, and Windows システムでは、 **runmqakm** (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵および証明書を管理します。

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

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

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

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

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

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

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

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

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

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

Related reference
“連邦情報処理標準” on page 23
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。

クライアント鍵リポジトリーを完全に検証するために、 **mqcertck** コマンドは IBM Global Security Kit (GSKit)を使用してダミー接続を作成します。 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、 AIX, Linux, and Windows システム上の IBM MQ MQI clients に対して 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 AIX, Linux, and Windows 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、465 ページの『SSL および TLS 秘密鍵のリセット』および 35 ページの『AIX, 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、 269 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

IBM MQ MQI client 構成ファイル、 `mqclient.ini`

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

443 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、TLS1.3 と TLS 1.2 の CipherSpec と、RSA と Diffie-Hellman の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、420 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

22 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 TLS 1.2 プロファイルに準拠 するよう IBM MQ for AIX, Linux, and Windows 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。

AIX, Linux, and Windows 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

セキュリティー・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スイート B 標準に準拠しませんが、 IBM MQ は、 AIX, Linux, and Windows システムのいずれかまたは両方のセキュリティー・レベルに準拠するように構成できます。 Suite B を使用するように IBM 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、420 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。

このコマンドの詳細については、「**runmqakm**541 ページの『鍵と証明書の管理 (AIX, Linux, and Windows)』」を参照してください。

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、543 ページの『AIX, 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、545 ページの『AIX, 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。

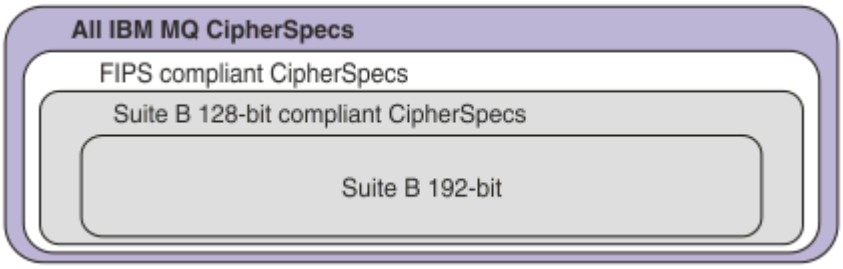
FIPS 140-2 と Suite B

注 : AIX, Linux, and Windows では、 IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。 このモジュールの証明書は「履歴」ステータスに移動されました。 お客様は、IBM Crypto for C (ICC) 証明書 を表示し、NIST から提供されたアドバイスに注意する必要があります。 交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、 「NIST CMVP modules in process list」でそのモジュールを検索します。

IBM MQ Operator 3.2.0 およびキュー・マネージャー・コンテナー・イメージ 9.4.0.0 以降は、UBI 9 に基づいています。 FIPS 140-3 準拠は現在保留中であり、その状況を表示するには、「NIST CMVP modules in process list」で「Red Hat Enterprise Linux 9- OpenSSL FIPS Provider」を検索します。

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。 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

AIX, Linux, and Windows 上で Suite B 準拠操作用に IBM MQ を構成する方法については、44 ページの『Suite B 用 IBM MQ の構成』を参照してください。

IBM MQ は、以下のプラットフォームおよびクライアントでの Suite B 準拠操作をサポートしていません。

- IBM i プラットフォーム
- z/OS プラットフォーム
- Java クライアント
- JMS クライアント

関連概念

269 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。

ALW *Suite B 用 IBM MQ の構成*

IBM MQ は、AIX, Linux, and Windows 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、24 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、42 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

   その他の値を指定して MQ_SUITE_B_NONE を使用することは無効です。

   MQSCO 構造体について詳しくは、 MQSCO-SSL 構成オプションを参照してください。
2. **MQSUITEB** 環境変数を以下の 1 つ以上の値に設定します。
   - 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 を他の値と一緒に使用することは無効です。
3. クライアント構成ファイルの SSL スタンザ の **EncryptionPolicySuiteB** 属性を以下の 1 つ以上の値に設定します。
   - 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。 その他の値を指定して NONE を使用することは無効です。

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ます。これにより、SSL スタンザの属性がオーバーライドされます。

.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。

- ALW AIX, Linux, and Windows システムの場合、証明書妥当性検査ポリシーは IBM Global Security Kit (GSKit) によって提供され、構成することができます。 V 9.4.0 V 9.4.0 3 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。
  - V 9.4.0 V 9.4.0 TLS サーバー証明書を認証しない証明書妥当性検査ポリシー。クライアント・アプリケーションでのみ使用可能です。
- IBM i 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 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。

証明書妥当性検査ポリシーの構成方法については、46 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、AIX, 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・システムから受信したデジタル証明書を妥当性検査するために使用する TLS 証明書妥当性検査ポリシーを指定するには、いくつかの異なる方法があります。

このタスクについて

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。 証明書妥当性検査ポリシーは、プラットフォームと環境によって異なります。 証明書妥当性検査ポリシーの詳細については、45 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。

手順

- キュー・マネージャーで証明書妥当性検査ポリシーを設定するには、キュー・マネージャー属性 **CERTVPOL** を使用します。

  この属性の設定について詳しくは、ALTER QMGR (キュー・マネージャー設定の変更)を参照してください。
- クライアントに証明書妥当性検査ポリシーを設定するには、以下の方法を使用します。

  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 構造体の `CertificateValPolicy` フィールドを使用します。 フィールドを以下のいずれかの値に設定します。

     MQ_CERT_VAL_POLICY_ANY
     セキュア・ソケット・ライブラリーでサポートされる各証明書妥当性検査ポリシーを適用します。 ポリシーのうちのいずれかにおいて証明書チェーンが有効と見なされる場合、その証明書チェーンを受け入れます。

     MQ_CERT_VAL_POLICY_RFC5280
     RFC5280 準拠の証明書妥当性検査ポリシーのみ適用します。 この設定は、ANY 設定よりも厳密に妥当性検査しますが、一部の旧式のデジタル証明書を拒否します。

V 9.4.0 V 9.4.0 **MQ_CERT_VAL_POLICY_NONE**
証明書検証ポリシーを適用しません。 この設定はクライアント・アプリケーション専用であり、トラスト・チェーンを検証せずに TLS サーバー証明書を受け入れます。

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** を使用します。 この環境変数を設定するには、以下のいずれかのコマンドを使用します。
   - Linux AIX AIX and Linux システムの場合:

     ```
     export MQCERTVPOL= value
     ```

   - Windows Windows システムの場合:

     ```
     SET MQCERTVPOL= value
     ```

   - IBM i IBM i システムの場合:

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

3. クライアント構成ファイル内の SSL スタンザの **CertificateValPolicy** 属性を使用します。 この属性は、以下のいずれかの値に設定します。

   ANY
   基礎となるセキュア・ソケット・ライブラリーによってサポートされているいずれかの証明書妥当性検査ポリシーを使用します。 この設定はデフォルト設定です。

   RFC5280
   RFC 5280 標準に準拠する証明書妥当性検査のみを使用します。

   V 9.4.0 V 9.4.0 **NONE**
   証明書検証ポリシーを適用しません。 この設定は、trust チェーンを検証せずに TLS サーバー証明書を受け入れます。

   この属性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。

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。

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

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

表 4. *CipherSpec* とデジタル証明書の関係 (続き)

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
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
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。

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。 つまり、楕円曲線 (EC) 証明書を必要とする CipherSpec を使用してチャネルを構成する場合は、証明書ラベルの RSA 証明書に名前を付けることはできません。 RSA 証明書を必要とする CipherSpec を使用してチャネルを構成する場合は、証明書ラベルの EC 証明書に名前を付けることはできません。

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。

TLS 1.3 CipherSpecs

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、42 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
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

420 ページの『CipherSpecs の有効化』
CipherSpec は、**DEFINE CHANNEL** 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。

269 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。

42 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 TLS 1.2 プロファイルに準拠 するよう IBM MQ for AIX, Linux, and Windows を構成する方法について説明します。

24 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、384 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、386 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、389 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、388 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、389 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、390 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、386 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、387 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、388 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、386 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	Deprecated 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
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

102 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。

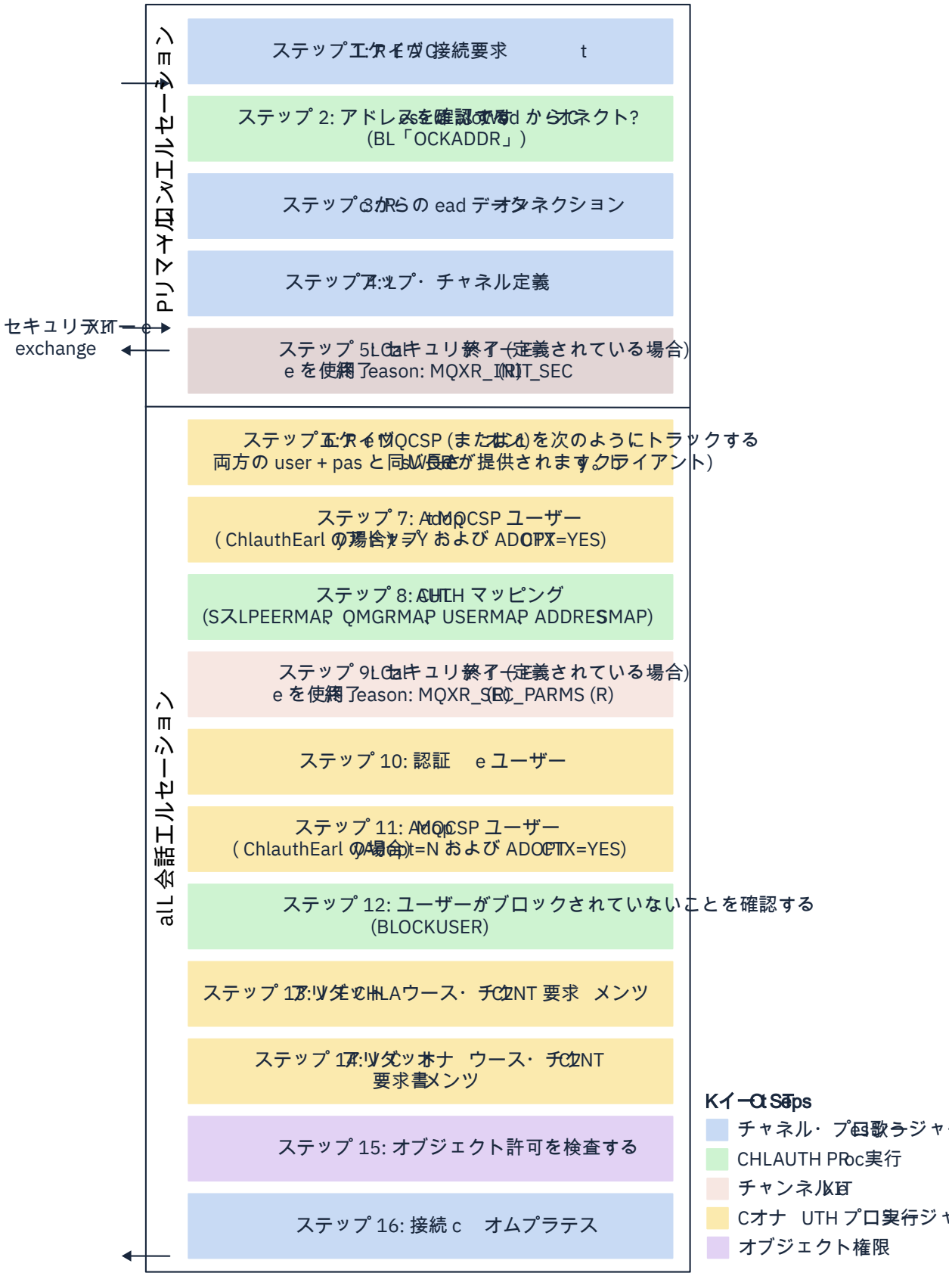
ローカル・バインディング
アプリケーションとキュー・マネージャーが同じオペレーティング・イメージ上にある場合に適用されます。 CHLAUTH は、このタイプのアプリケーション接続には関係しません。

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・フローでは、 IBM MQ は、特に、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
クライアントが認証資格情報を提供した場合は、必要に応じて作成します。

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提供される資格情報は認証されます。

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)*の規則に評価される場合、資格情報が後でステップ 9 で有効な資格情報でオーバーライドされない限り、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。 V 9.4.0 IBM MQ 9.3.4 以降、MQCSP 構造には認証トークン (MQCSP.**TokenPtr**) へのポインターも含まれています。

出口でユーザー ID とパスワード、および認証トークン を変更することができます。 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 マッピング規則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 上で実行されているキュー・マネージャーからの、**AUTHTYPE**(*IDPWOS*) を指定したデフォルトの AUTHINFO 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 上で実行されているキュー・マネージャーからの、 **AUTHTYPE**(*IDPWOS*) を指定したデフォルトの AUTHINFO 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) オブジェクトには、 CHCKCLNT という属性があります。 値を *REQUIRED* に変更すると、すべてのクライアント・アプリケーションが有効な資格情報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、以下の場合を除き、別の認証検査は実行されません。

- MQCXP 構造体の SecurityParms フィールドのユーザー ID とパスワード、または認証トークン が、ステップ 9 でセキュリティー出口によって変更されました。
- クライアント・アプリケーションは、再接続可能な機能を要求するオプションを使用して接続されました。

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。

If the user ID asserted in the MQCSP, or **SecurityParms** field of the MQXCP structure, has been successfully authenticated and **ADOPTCTX** is はい, then the context of the user resulting from steps 7 and 8 is adopted as the context to use for this application, unless the user ID and password、または認証トークン、 in the **SecurityParms** field of the MQCXP structure was changed by a security exit in step 9.

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 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検査されます。 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。

ステップ 13: 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。

ALW 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、161 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
キュー・マネージャーは、アプリケーションが接続時に提供する資格情報を認証するように構成できます。

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) を使用する際の特定のアクセスの問題を解決するための手順と例。

始める前に

注 : このタスクのステップでは、MQSC コマンドを実行する必要があります。 これを行う方法はプラットフォームによって異なります。 見る管理 IBM MQMQSC コマンドの使用。

このタスクについて

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。

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、通常、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。

一時的な手段として、およびリストされているエラーをトラブルシューティングするために、以下のいずれかのステップを実行します。

手順

- **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 規則を無効にするには、次の MQSC コマンドを実行します。

```
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、WARN に対する MQ-admin ユーザーによるすべてのチャネルへのアクセスがブロックされないようにするデフォルト・ルールを変更するには、以下の MQSC コマンドを実行します。

```
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、MQ-admin ユーザーがすべてのチャネルにアクセスできないようにするデフォルト規則を削除するには、次の MQSC コマンドを実行します。

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

- **MATCH (RUNCHECK) を使用したアクセスのテスト**

  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、MQSC コマンドを実行して、デフォルト規則が適用されている CHLAUTH 規則の結果、MQ-admin ユーザー johndoe が CHAN1 という名前のチャネルにアクセスすることを確認します。

```
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、MQSC コマンドを実行して、デフォルト規則が適用されている CHLAUTH 規則の結果として、ユーザー alice が MQ-admin ユーザーではなく、 CHAN1 という名前のチャネルにアクセスすることになることを確認します。

```
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
表示中の承認

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と、これらを実現するための CHLAUTH 規則の例をいくつか示します。

始める前に

注 : このタスクのステップでは、MQSC コマンドを実行する必要があります。 これを行う方法はプラットフォームによって異なります。 見る管理 IBM MQMQSC コマンドの使用。

このタスクについて

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。

ユーザーの新規 CHLAUTH ルールを作成するには、以下のシナリオを 1 つ以上構成します。

手順

- **特定の MQ-admin ユーザーのアクセスの制御**

  a) 管理の観点、つまり IBM MQ Explorer から接続するために排他的に使用され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。

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

  以下の MQSC コマンドを実行します。

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

  c) 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。

d) デフォルトの CHLAUTH 規則 が適用されていることを確認します。

e) 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

これを行うには、以下の MQSC コマンドを実行します。

```
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。

f) オプション: MQSC コマンド MATCH (RUNCHECK) をいつでも実行して、以下の各コマンドの結果を確認できます。

```
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の制御**

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ことを前提として、 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。

a) 次の MQSC コマンドを使用して、SVRCONN チャネル APP1.CHAN。特定のアプリケーションおよび特定のユーザーによって使用されます。

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

b) 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想定すると、これは IBM MQ オブジェクト権限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。

以下の MQSC コマンドを実行します。

```
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)
```

- **特定のユーザーの証明書識別名 (DN) を使用して、そのユーザーのアクセスを制御する**

  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。

  a) 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

  以下の MQSC コマンドを実行します。

```
.
# 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。

  b) オプション: 別の IBM MQ ユーザー ID にマップします。

  USERSRC(CHANNEL)を USERSRC(MAP) MCAUSER('mquser1') に置き換えて、前の MQSC コマンドを再実行します。

- **特定のユーザーを mqm ユーザーにマップする**

  これは、 特定の MQ-admin ユーザーのアクセス制御に対する追加または変更です。

  MQSC コマンドを使用して、 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ために、以下の CHLAUTH ルールを追加します。

```
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

68 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。

関連タスク

63 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) を使用する際の特定のアクセスの問題を解決するための手順と例。

関連資料

SET CHLAUTH

表示中の承認

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。

- 68 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』
- 68 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』
- 69 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

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。

このシナリオでは、チャネル MY.SVRCONN へのアクセスには、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)
```

関連タスク

63 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) を使用する際の特定のアクセスの問題を解決するための手順と例。

65 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と、これらを実現するための CHLAUTH 規則の例をいくつか示します。

関連資料

SET CHLAUTH

表示中の承認

CHLAUTH バック・ストップ・ルールの作成
キュー・マネージャーに入ってくるインバウンド接続を制御する方法については、2 つのオプションがあります。 1 つは、許可されていない接続をすべてリストするという方法であり、もう 1 つは、どの接続も許可されていないと最初に宣言し、それから許可されている接続をすべてリストするという方法です。 ここでは、この 2 番目のオプションを説明します。

このタスクについて

2 番目のオプションを使用する理由は、最初のオプションではセキュリティー・ブリーチが発生するリスクがあるからです。許可されていない接続をすべてリストしようとした場合は、そのリストに含まれていない接続がすべて許可されることになります。その結果、リストから抜け落ちている接続が 1 つでもあると、本来は許可されていない接続が許可されてしまい、そこにセキュリティー・ブリーチが発生します。

その逆に、どの接続も許可されていないと最初に宣言し、それから許可されている接続をリストすると、そのリストから抜け落ちている接続があっても、セキュリティー・ブリーチは発生しません。 自社でさらに接続を追加する必要が生じても、それは比較的シンプルな作業であり、セキュリティー・ブリーチのリスクもありません。

最初に行う作業は安全策 ルールの作成です。つまり、具体的なルールにマッチングしなかった接続をすべてキャッチ (阻止) するためのルールです。 このルールを作成すると、キュー・マネージャーへのリモート接続が完全に停止するという影響があります。

この方式に不安がある場合は、警告モードで安全策 ルールをセットアップできます。ステップ 70 ページの『2』 を参照してください。

手順

1. キュー・マネージャーへのリモート接続を停止する安全策ルールを作成するには、以下のコマンドを発行します。

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

   これですべてのリモート接続のドアを閉じたので、この状態から始めて、特定の接続を許可するための具体的なルールを作成していきます。 以下に例を示します。

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

2. 警告モードで安全策ルールを作成する場合は、以下のコマンドを発行します。

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

   この状態から肯定ルールをすべて作成していきます。 必要な規則をすべて作成したと思われる場合は、以下のコマンドを発行してチャネル・イベントをオンにしてください。

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

   さらに、 **Reason** が MQRC_CHANNEL_BLOCKED_WARNING に設定されているイベントがないか SYSTEM.ADMIN.CHANNEL.EVENT キューをモニターします。

   こうしたイベントには、安全策ルールに一致した接続の詳細情報が記述されていますが、このコマンドを警告モードで実行しているので、現時点では接続がブロックされません。

   これらの各イベントを検討し、その接続を許可する肯定ルールを作成する必要があるか、それとも安全策 ルールに正しく一致しているかを確認してください。 このモードでコマンドを実行し、作成されるイ

ベントを検討していった結果、対象のインバウンド・チャネルがすべて表示され、そのすべてを許可する適切な肯定ルールが存在することを確認できたら、この作業は成功です。

その時点で、安全策 ルールに一致する接続を実際にブロックする処理を開始するように変更できます。そのためには、以下のコマンドを発行します。

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

非特権 *IBM MQ* 管理者の作成
非特権 IBM MQ 管理者を CHLAUTH を使用して作成する方法について取り上げます。

このタスクについて

このタスクのコンテキストでは、以下の用語を使用します。

特権ユーザー
対象の操作を実行するために明示的にアクセス権限を付与されなくてもその操作の実行許可を持つユーザーのことです。 mqm グループ内のユーザーは、こうした特権ユーザーの例となります。

***IBM MQ* 管理者**
IBM MQ に対して管理コマンド (**DEFINE QLOCAL** や **START CHANNEL** など) を発行する必要があるユーザーを意味します。

非特権 IBM MQ 管理者を作成するステップを以下に示します。

手順

1. 自社で使用しているプラットフォームに適切なコマンドを使用して、キュー・マネージャー・マシン上にユーザー ID を作成します。
   この例では、`alice` というユーザー名を使用します。
2. 以下の手順を実行して、この新しいユーザーに、すべての IBM MQ 管理コマンドを実行する権限を付与します。
   a) 特権ユーザーによって IBM MQ Explorer を開始します。
   b) 適切なキュー・マネージャーを選択して「役割ベースのウィザード *(Role Based Wizard)*」にナビゲートし、「オブジェクト権限 (Object Authorities)」、「役割ベースの権限の追加 (Add Role Based Authorities)」と移動します。
   c) ポップアップ表示されるウィザード・パネルで、最初のステップで作成したユーザー ID を入力します。または、グループで作業する場合は、非特権 IBM MQ 管理者にするユーザーまたはユーザーの集合のグループ名を入力します。
   d) 全管理アクセス権限用にウィザードをセットします。
   e) 非特権 IBM MQ 管理者がキュー上のメッセージを参照できるようにする場合、そのチェック・ボックスも選択します。
   f) ウィザード下部に示されるプレビュー・パネルでコマンドを確認します。
   それらのコマンドをカット・アンド・ペーストして、独自のスクリプトを作成できます。
   独自のスクリプトで実行する理由の 1 つは、対象ユーザーに付与するアクセス権限の数を減らすことができることです。 すべてのオブジェクトに対するアクセス権限を付与するのではなく、特定のオブジェクト・グループに対するアクセス権限のみを付与するほうが適切な場合があります。
   ウィザードで「**OK**」押すと、示されているとおりにコマンドが実行されます。
   g) 非特権 IBM MQ 管理者に関する要件がリモート・アクセス向けでもある場合には、このユーザー ID に対してリモート・アクセスを許可する CHLAUTH 規則をいくつかセットアップすることも必要です。
   自社が 70 ページの『CHLAUTH バック・ストップ・ルールの作成』のガイダンスを使用している場合は、行う必要があるのは有効化規則を追加することだけです。
   作成する規則は、リモートの IBM MQ 管理者を認証する方法によって異なります。

Weak TCP/IP authentication を使用する場合、以下のような CHLAUTH 規則をセットアップできます。

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

9. TLS authentication を使用する場合、以下のような CHLAUTH 規則をセットアップできます。

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

これで、ユーザーが `admin-channel-name` に接続する (および CHLAUTH 規則に一致する) ときにキュー・マネージャーでユーザー ID `alice` を使用してコマンドを実行できるようになったので、特権が付与されたリモート・アクセスは不要になります。

接続認証

接続認証により、アプリケーションは、キュー・マネージャーに接続するときに認証資格情報を提供できます。 キュー・マネージャーは資格情報の妥当性検査を行います。 資格情報で提供されるユーザー ID は、アプリケーションがアクセスするリソースの許可検査で使用するためにも採用できます。

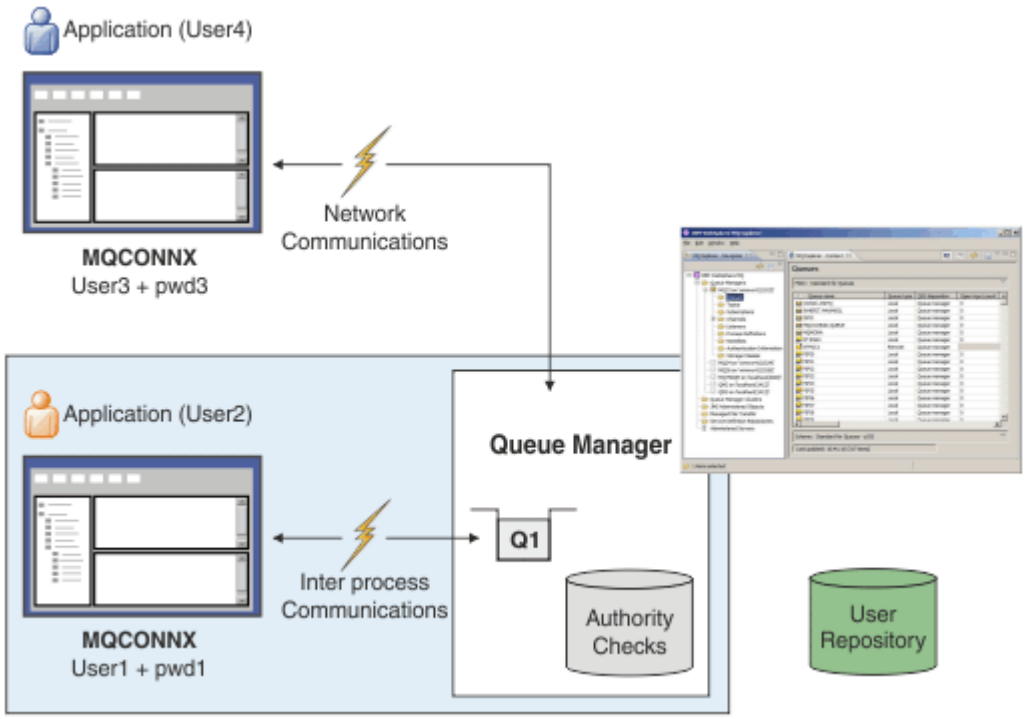
アプリケーションは、キュー・マネージャーに接続するときに、認証用のユーザー ID とパスワードを提供できます。

V 9.4.0 IBM MQ 9.3.4 以降、 IBM MQ client アプリケーションは、認証の代替方法として認証トークンを提供することもできます。

キュー・マネージャーは、アプリケーションから提供される資格情報を検証するように構成できます。

アプリケーションによって提供されるユーザー ID とパスワードは、キュー・マネージャー構成のユーザー・リポジトリーを使用して検査されます。 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について詳しくは、 ユーザー・リポジトリーを参照してください。

V 9.4.0 認証トークンは、トークンの署名を検証するために、キュー・マネージャーのトークン認証鍵ストア内の証明書と対称鍵を使用して検証されます。 認証トークンを使用したユーザーの認証について詳しくは、 325 ページの『認証トークンの処理』を参照してください。



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さまざまな API を使用してキュー・マネージャーに接続できますが、すべてのアプリケーションはユーザー 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。

関連概念

73 ページの『接続認証: 構成』
キュー・マネージャーは、アプリケーションが接続時に提供する資格情報を認証するように構成できます。

78 ページの『接続認証: アプリケーションの変更』

79 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

キュー・マネージャーは、アプリケーションが接続時に提供する資格情報を認証するように構成できます。

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。 AUTHINFO オブジェクトの **AUTHTYPE** 属性は、オブジェクトのタイプを指定します。 接続認証に使用される AUTHINFO オブジェクトは、以下の 2 つのタイプのいずれかです。

IDPWOS
キュー・マネージャーは、ローカル・オペレーティング・システムを使用して、接続アプリケーションから提供されるユーザー ID とパスワードを認証します。

Linux V 9.4.0 AIX IBM MQ 9.3.4 以降、このタイプの AUTHINFO オブジェクトを使用すると、AIX または Linux 上で実行されるキュー・マネージャーでも認証トークンの妥当性検査を行うことができます。 接続認証の構成に使用される AUTHINFO オブジェクトに加えて、 qm.ini ファイルの **AuthInfo** スタンザを使用して認証トークンを受け入れるようにキュー・マネージャーを構成する必要があります。 認証トークンを受け入れるようにキュー・マネージャーを構成する方法について詳しくは、 332 ページの『ローカル鍵ストアを使用して認証トークンを受け入れるためのキュー・マネージャーの構成』を参照してください。

IDPWLDAP
キュー・マネージャーは、LDAP サーバーを使用して、接続アプリケーションによって提供されるユーザー ID とパスワードを認証します。

注 : キュー・マネージャーの **CONNAUTH** 属性に他のタイプの認証情報オブジェクトを指定することはできません。

タイプ IDPWOS および IDPWLDAP の AUTHINFO オブジェクトは、いくつかの属性で類似しています。 ここで説明する属性は、両方のタイプのオブジェクトに共通です。

以下の MQSC コマンド例では、以下の操作を使用して接続認証をオンにします。

1. USE.PW という名前の AUTHINFO オブジェクトを定義します。
2. この AUTHINFO オブジェクトを参照するようにキュー・マネージャーの **CONNAUTH** 属性を変更します。

3. **REFRESH SECURITY** コマンドを発行して、キュー・マネージャーの接続認証構成をリフレッシュします。 **REFRESH SECURITY** コマンドは、キュー・マネージャーが接続認証構成の変更を認識する前に発行する必要があります。

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

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

ローカルにバインドされたアプリケーションによって作成された接続について資格情報を検査するかどうかを制御するには、AUTHINFO 属性 **CHCKLOCL** (ローカル接続の検査) を使用します。 クライアント・アプリケーションによって作成された接続について資格情報を検査するかどうかを制御するには、AUTHINFO 属性 **CHCKCLNT** (クライアント接続の検査) を使用します。

CHCKLOCL は NONE および OPTIONAL の値を受け入れ、 **CHCKCLNT** は認証要件を構成するために NONE の値を許可します。

NONE
アプリケーションによって提供される認証資格情報は検査されません。

OPTIONAL
アプリケーションによって提供されるすべての資格情報が有効であることを確認します。 ただし、アプリケーションが認証資格情報を提供すること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。

以下の場合:

- ユーザー名とパスワードを指定します。これらは認証されます。
- ユーザー名とパスワードを指定しないでください。接続は許可されます。
- エラーを受け取ったパスワードではなく、ユーザー名を指定してください。

重要 : OPTIONAL は、チャネル認証 (CHLAUTH) 規則においてより制限的なオプションを設定する場合に設定できる最小値です。

NONE を選択した場合に、 **CHCKCLNT** が REQUIRED (または z/OS 以外のプラットフォームでは REQDADM) に設定されている CHLAUTH レコードとクライアント接続が一致すると、接続は失敗します。 Multiplatforms 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。

チャネル認証規則を使用して、一部のクライアント接続に対してより制限的な **CHCKCLNT** オプションを設定する方法について詳しくは、 75 ページの『構成の細分度』 を参照してください。

REQUIRED
すべてのアプリケーションが有効な資格情報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資格情報を提供する必要がありますが、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 に設定することは、ユーザーが **runmqsc** コマンドでユーザー ID を指定するために **-u** パラメーターを指定しない限り、 **runmqsc** を使用してキュー・マネージャーをローカルで管理できないことを意味します (エラー AMQ8135: Not authorized)。 このパラメーターを設定すると、 **runmqsc** はコンソールでユーザーのパスワードの入力を求めるプロンプトを出します。

同様に、ローカル・システムで IBM MQ Explorer を実行す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 ID 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 ID とパスワードを入力し、「**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。

キュー・マネージャーの **CONNAUTH** 属性は、 IBM MQ 8.0 より前のバージョンからマイグレーションされたキュー・マネージャーの場合はブランクですが、新しく作成されたキュー・マネージャーの場合は *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* に設定されます。 このデフォルトの **AUTHINFO** 定義では、**CHCKCLNT** はデフォルトで REQDADM に設定されています。

したがって、接続に特権ユーザー ID を使用する既存のクライアントは、有効な資格情報を提供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 構造の資格情報は、プレーン・テキストでネットワークを介して送信されることがあります。 クライアント資格情報が確実に保護されるようにするには、31 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

AUTHINFO オブジェクトの **CHCKLOCL** 属性と **CHCKCLNT** 属性は、キュー・マネージャーへのすべての接続の認証要件を設定します。 これらの属性に加えて、チャネル認証 (CHLAUTH) 規則の **CHCKCLNT** 属性を使用すると、CHLAUTH 規則に一致する特定のクライアント接続に対して、より厳密な認証要件を設定できます。

CHLAUTH 規則で **CHCKCLNT** を REQUIRED または REQDADM に設定することにより、例えば AUTHINFO オブジェクトで全体の **CHCKCLNT** 値を OPTIONAL に設定してから、特定のチャネルに対してより厳格になるようにその値を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)** で定義されているため、この細分度を使用する必要はありません。 例えば、これらの MQSC コマンドは、AUTHINFO オブジェクトの **CHCKCLNT** 属性をオーバーライドする 1 つの CHLAUTH 規則と、以下を行わない 1 つの CHLAUTH 規則を定義します。

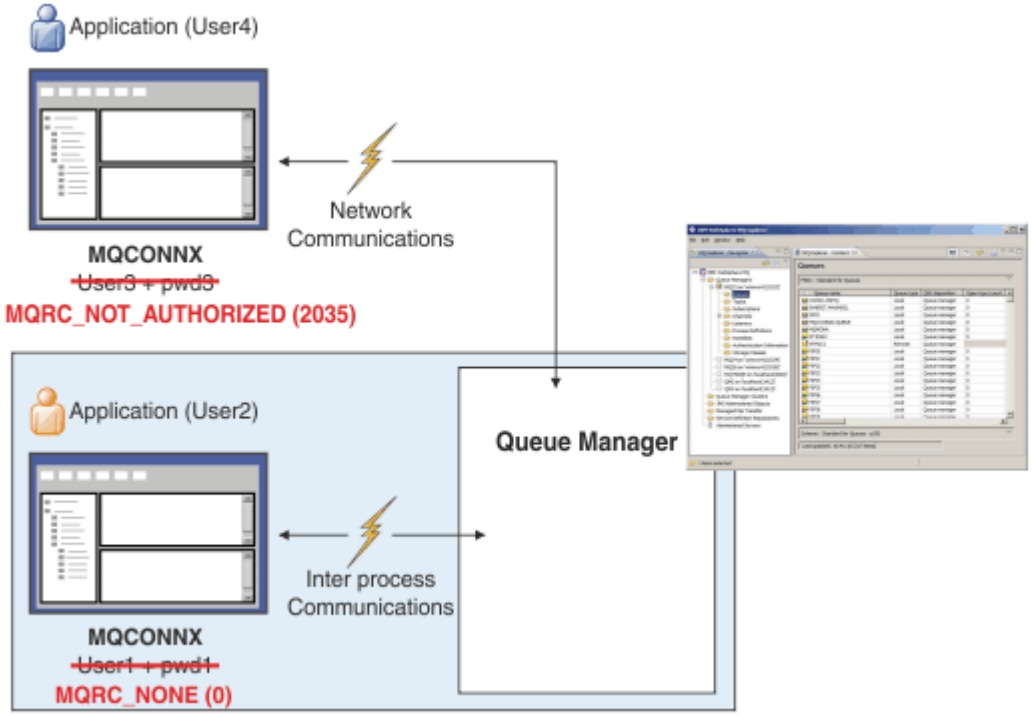
```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)

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

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

CHLAUTH 規則について詳しくは、51 ページの『チャネル認証レコード』を参照してください。

エラー通知



エラーは、以下の状況で記録されます。

- 認証資格情報が必要な場合、アプリケーションは認証資格情報を提供しません。
- アプリケーションが無効な認証資格情報を提供しました。 この状態は、アプリケーションが資格情報を提供することがオプションであることが構成で指定されている場合でも、エラーとして扱われます。

注 : CHCKLOCL または **CHCKCLNT** が NONE に設定されている場合、アプリケーションによって提供される無効な資格情報は検出されません。

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。 この遅延により、接続を繰り返し試行するアプリケーションからの保護が提供されます。

エラーはいくつかの方法で記録されます。

アプリケーション
MQRC_NOT_AUTHORIZED (2035) 理由コードがアプリケーションに戻されます。

管理者
IBM MQ 管理者には、エラー・ログに報告されたイベントが表示されます。 このエラー・メッセージは、ユーザーに接続権限がないなどの理由ではなく、資格情報が無効であるために接続が拒否されたことを示しています。

モニター・ツール
権限イベントをオンにした場合は、 SYSTEM.ADMIN.QMGR.EVENT キュー上のイベント・メッセージによって、障害をモニター・ツールに通知することもできます。 権限イベントをオンにするには、次の MQSC コマンドを発行します。

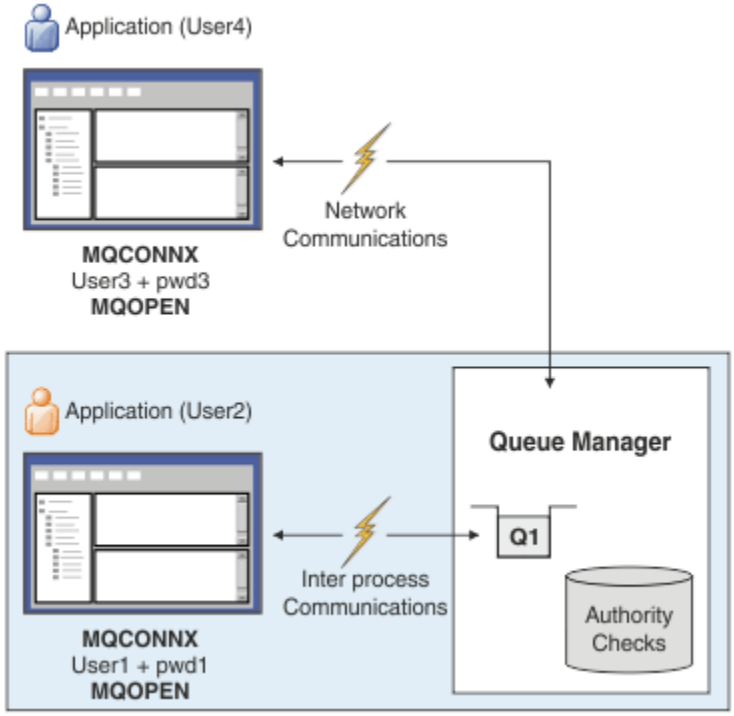
```
ALTER QMGR AUTHOREV(ENABLED)
```

この「許可されない」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を提供します。追加のフィールドとして、提供された MQCSP ユーザー ID を提供します。 アプリケーションがパスワードを提供した場合、そのパスワードはイベント・メッセージに含まれません。 これは、イベント・メッセージに 2 つのユーザー ID があることを意味します。

- アプリケーションが実行されているユーザー ID。
- アプリケーションが提示した資格情報内のユーザー ID。

このイベント・メッセージについて詳しくは、 Not Authorized (type 1)を参照してください。

許可のためのユーザーの採用



アプリケーションによって提示される資格情報を接続のコンテキストとして採用するようにキュー・マネージャーを構成できます。 資格情報を採用するということは、認証資格情報で指定されたユーザー ID が、管理画面に表示される許可検査に使用され、メッセージに表示されることを意味します。 AUTHINFO オブジェクトの **ADOPTCTX** 属性は、資格情報をアプリケーションのコンテキストとして採用するかどうかを制御します。 例えば、以下の MQSC コマンドは、接続認証に使用される USE.PWD という名前の AUTHINFO オブジェクトを定義し、 **ADOPTCTX** 属性を YES に設定します。

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

ALTER QMGR CONNAUTH(USE.PWD)
```

ADOPTCTX 属性には、以下の値を指定できます。

ADOPTCTX(YES)
アプリケーションによって提供される資格情報は、接続の期間中、アプリケーション・コンテキストとして採用されます。 アプリケーションのすべての許可検査は、認証された資格情報内のユーザー ID を使用して行われます。

重要 : ADOPTCTX(YES) およびローカル・オペレーティング・システムのユーザー ID を使用する場合は、採用されるユーザー ID が IBM MQ のユーザー ID の要件を満たしていることを確認する必要があります。 詳しくは、90 ページの『ユーザー ID』を参照してください。

ADOPTCTX(NO)
アプリケーションによって提供される資格情報は、接続時の認証にのみ使用されます。 アプリケーションを実行しているユーザー ID は、今後の許可検査に引き続き使用されます。 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。

チャネル認証との対話

チャネル認証規則を使用すると、クライアントから受け取ったユーザー ID に基づいて、アプリケーション接続のコンテキストとして使用されるユーザー ID を変更できます。 チャネル認証規則を使用して接続に関連付けられたユーザー ID を変更する例については、 387 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。

IBM MQ クライアント・アプリケーション接続のセキュリティー・コンテキストを決定するときに、接続認証規則とチャネル認証規則の処理順序が重要な要素になります。 `qm.ini` ファイルの **channels** スタンザの **ChlauthEarlyAdopt** パラメーターは、キュー・マネージャーがアプリケーションによって提供される資格情報からコンテキストを採用する順序を制御し、チャネル認証規則を適用します。
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。

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場合、アプリケーションによって提供される資格情報から採用されるコンテキストは、**ChlauthEarlyAdopt** パラメーターが Y に設定されている場合にのみ、チャネル認証規則によって変更できます。

接続認証とチャネル認証の相互作用、およびクライアント・アプリケーションがキュー・マネージャーに接続するときに検査が行われる順序について詳しくは、 58 ページの『CHLAUTH および CONNAUTH の相互作用』を参照してください。

関連概念

72 ページの『接続認証』
接続認証により、アプリケーションは、キュー・マネージャーに接続するときに認証資格情報を提供できます。 キュー・マネージャーは資格情報の妥当性検査を行います。 資格情報で提供されるユーザー ID は、アプリケーションがアクセスするリソースの許可検査で使用するためにも採用できます。

78 ページの『接続認証: アプリケーションの変更』

79 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

メッセージ・キュー・インターフェース (MQI) を使用する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) 構造でユーザー ID とパスワードを提供できます。 その他のアプリケーション・プログラミング・インターフェースでは、MQCSP 構造は通常、 IBM MQ ライブラリーによってアプリケーションの代わりに構成されます。

V 9.4.0 IBM MQ 9.3.4 以降、 AIX または Linux システム上で稼働するキュー・マネージャーに接続するクライアント・アプリケーションは、識別の代替手段として MQCSP 構造で認証トークンを送信することもできます。

ユーザー ID とパスワード、または認証トークン は、キュー・マネージャーに付属する オブジェクト権限マネージャー (OAM) 、または z/OS システム上のキュー・マネージャーに付属する許可サービス・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、または認証トークン も、処理のためにクライアント・サイドとサーバー・サイドのセキュリティー出口に渡されます。 また、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 を設定するためにも使用できます。

警告 : クライアント・アプリケーションの MQCSP 構造の資格情報は、プレーン・テキストでネットワークを介して送信されることがあります。 クライアント・アプリケーション資格情報が保護されていることを確認するには、 31 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 とパスワードを提供することにより、アプリケーション・コードを変更する必要がなくなります。

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

72 ページの『接続認証』
接続認証により、アプリケーションは、キュー・マネージャーに接続するときに認証資格情報を提供できます。 キュー・マネージャーは資格情報の妥当性検査を行います。 資格情報で提供されるユーザー ID は、アプリケーションがアクセスするリソースの許可検査で使用するためにも採用できます。

73 ページの『接続認証: 構成』
キュー・マネージャーは、アプリケーションが接続時に提供する資格情報を認証するように構成できます。

79 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。

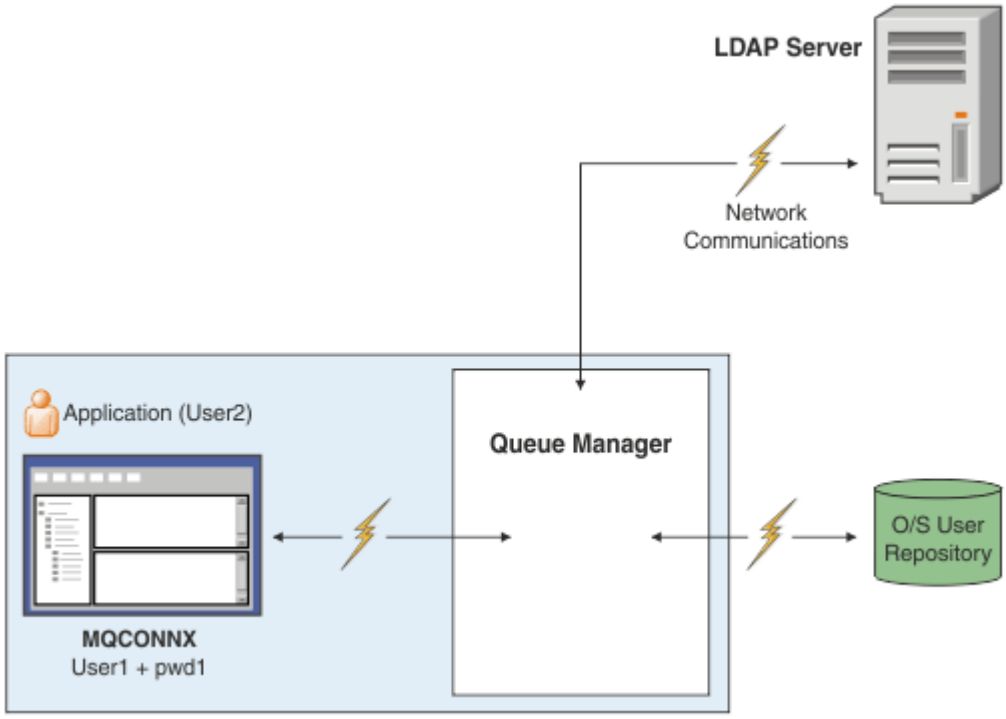


図 7. 認証情報オブジェクトのタイプ

```
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。

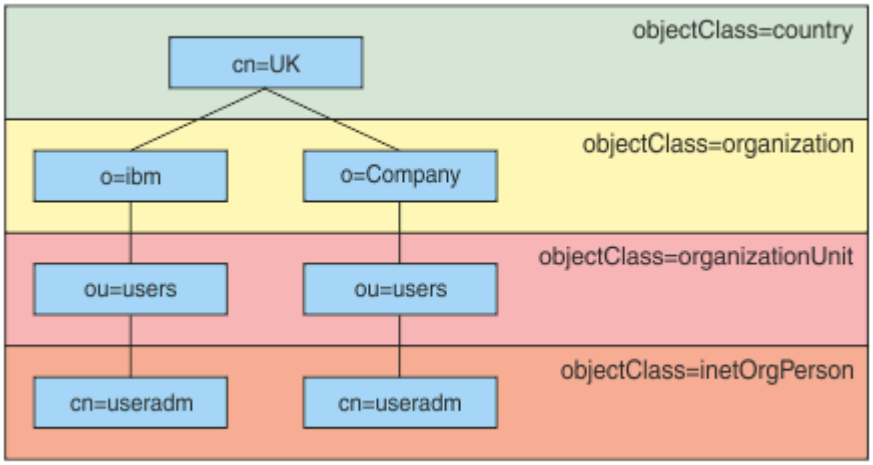


図 *8. LDAP* 階層の例

例えば、81 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

72 ページの『接続認証』
接続認証により、アプリケーションは、キュー・マネージャーに接続するときに認証資格情報を提供できます。 キュー・マネージャーは資格情報の妥当性検査を行います。 資格情報で提供されるユーザー ID は、アプリケーションがアクセスするリソースの許可検査で使用するためにも採用できます。

73 ページの『接続認証: 構成』
キュー・マネージャーは、アプリケーションが接続時に提供する資格情報を認証するように構成できます。

78 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

AIX and Linux 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 AIX または Linux システムでは、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 51 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)などです。

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、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キュー・マネージャーを構成できる、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・トランスポートを使用する 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。

認証対象のユーザー ID とパスワードは、アプリケーションによって以下のいずれかの方法で指定されます。

- IBM MQ classes for Java アプリケーションの場合、 MQEnvironment クラス、または com.ibm.mq.MQQueueManager コンストラクターに渡されるプロパティー Hashtable。
- IBM MQ classes for JMS アプリケーションの場合、createConnection(String username, String Password) メソッドまたは createContext(String username, String password) メソッドへの引数。

MQCSP 認証モード

このモードでは、アプリケーションの実行に使用されるクライアント・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キュー・マネージャーに送信されます。 IBM MQ classes for Java および IBM MQ classes for JMS は、MQCSP 構造体でキュー・マネージャーに対して認証されるユーザー ID とパスワードを送信します。

ユーザー ID とパスワードは、サーバー接続セキュリティー出口が MQCSP 構造体で使用できます。 MQCSP 構造体のアドレスは、チャネルの MQCXP 構造体の **SecurityParms** フィールドにあります。

MQCSP 認証モードには以下の利点があります。

- 認証されるユーザー ID の最大長は 1024 文字です。
- 認証のパスワードの最大長は 256 文字です。
- IBM MQ リソースを使用するためのアクセス権限の許可検査は、キュー・マネージャーで接続権限を制御するために使用される認証情報オブジェクトが ADOPTCTX(NO) で構成されている場合には、アプリケーションの実行に使用されるクライアント・サイドのユーザー ID を使用して行うことができます。

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。

このモードには以下の制限があります。

- ユーザー ID とパスワードの長さは 12 文字以下にする必要があります。 12 文字を超えるユーザー ID は、12 文字に切り捨てられます。 これによって、接続が理由コード MQRC_NOT_AUTHORIZED で失敗する場合があります。
- アプリケーションの実行に使用されるクライアント・サイドのユーザー ID は、キュー・マネージャーに送信されません。 キュー・マネージャーでの接続認証の制御に使用する認証情報オブジェクトで ADOPTCTX(YES) を設定するか、TLS 証明書に基づくチャネル認証規則などの他の方法を使用して、権限のチェック対象となるチャネル MCA ユーザー ID で IBM MQ リソースを使用するように設定する必要があります。

デフォルト認証モード

IBM MQ classes for Java または IBM MQ classes for JMS のクライアント・アプリケーションが使用するデフォルト認証モードは、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かどうかによって異なります。

- ユーザー ID とパスワードが指定されている場合、デフォルトで MQCSP 認証が使用されます。
- ユーザー ID が指定され、パスワードが指定されない場合には、互換モードがデフォルトで使用されます。
- ユーザー ID が指定されない場合には、必ず互換モードが使用されます。

85 ページの『認証モードの選択』で説明されているように、ユーザー ID が指定される場合には、特定の認証モードは、個別の接続に対してアプリケーションで選択することも、アプリケーションの開始前に全体的に設定することもできます。

注 : IBM MQ classes for JMS を使用するアプリケーションは、IBM MQ 9.3.0 におけるデフォルト認証モードの変更によって影響を受ける可能性があります。 IBM MQ classes for JMS を IBM MQ 9.3.0 にアップグレードした後、以前に互換モードをデフォルトで使用していたアプリケーションは、代わりに MQCSP 認証を使用します。 この結果、以前はキュー・マネージャーに正常に接続できていたアプリケーションが、理由コード 2035 (MQRC_NOT_AUTHORIZED) を含む `JMSException` で接続が失敗する可能性があります。 この場合、85 ページの『認証モードの選択』に示されているいずれかの方法で、アプリケーションが互換モードを使用するように指定します。

ローカル・バインディングを使用してキュー・マネージャーに接続する Java アプリケーションは、必ず MQCSP 認証モードを使用します。

認証モードの選択

キュー・マネージャーに接続するときにユーザー ID を指定する Java クライアント・アプリケーションで使用される認証モードは、以下のいずれかの方法で指定できます。 これらの方法については、優先順位の高いものからリストしています。 これらのいずれかの方法で認証モードが指定されない場合、デフォルト認証モードが使用されます。

注 : これらの方式を使用した認証モードの選択については、IBM MQ 9.3.0 で説明されています。 場合によっては、Java クライアント・アプリケーションが使用する認証モードは、IBM MQ classes for Java または IBM MQ classes for JMS の IBM MQ 9.3.0 へのアップグレード時に変更されることがあります。 この結果、以前はキュー・マネージャーに正常に接続できていたアプリケーションが、理由コード 2035 (MQRC_NOT_AUTHORIZED) を含む `JMSException` で接続が失敗する可能性があります。 この場合、以下のいずれかの方法を使用して、必要な認証モードを選択します。

- キュー・マネージャーに接続する前に、アプリケーションの適切なプロパティーを設定して個別の接続の認証モードを指定します。
  - IBM MQ classes for Java を使用する場合は、`com.ibm.mq.MQQueueManager` コンストラクターに渡されるプロパティー Hashtable にプロパティー *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* を設定します。
  - IBM MQ classes for JMS を使用する場合、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* を設定します。

  以下のどちらかの値を使用するようにこれらのプロパティー値を設定します。

  true
  キュー・マネージャーへの認証で MQCSP 認証モードを使用します。

  false
  キュー・マネージャーへの認証で互換モードを使用します。
- アプリケーション開始時に *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java システム・プロパティーを設定し、アプリケーションによるすべてのクライアント接続の認証モードを設定します。 以下のどちらかの値を使用するようにこのプロパティー値を設定します。

  Y
  キュー・マネージャーへの認証で MQCSP 認証モードを使用します。

  N
  キュー・マネージャーへの認証で互換モードを使用します。

  例えば、以下のコマンドは、互換モードを選択し、Java アプリケーションを開始するようにこのプロパティーを設定します。

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```
- アプリケーションが開始される環境で *com.ibm.mq.jmqi.useMQCSPauthentication* 環境変数を設定して、同じ環境で開始されるアプリケーションによるクライアント接続すべての認証モードを指定します。 以下のどちらかの値に環境変数値を設定します。

  Y
  キュー・マネージャーへの認証で MQCSP 認証モードを使用します。

  N
  キュー・マネージャーへの認証で互換モードを使用します。
- クライアント構成ファイルの JMQI スタンザに **useMQCSPauthentication** 属性を指定し、特定の IBM MQ MQI client クライアント構成ファイルを使用するすべてのアプリケーションに対して認証モードを指定します。 以下のどちらかの値にこの属性値を設定します。

  YES
  キュー・マネージャーへの認証で MQCSP 認証モードを使用します。

  NO
  キュー・マネージャーへの認証で互換モードを使用します。

  useMQCSPauthentication 属性について詳しくは、クライアント構成ファイルの JMQI スタンザを参照してください。

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。

MQCSP 認証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。

- 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。

関連概念

72 ページの『接続認証』
接続認証により、アプリケーションは、キュー・マネージャーに接続するときに認証資格情報を提供できます。 キュー・マネージャーは資格情報の妥当性検査を行います。 資格情報で提供されるユーザー ID は、アプリケーションがアクセスするリソースの許可検査で使用するためにも採用できます。

78 ページの『接続認証: アプリケーションの変更』

79 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

599 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、以下を参照してください。

- ALW 401 ページの『AIX, Linux, and Windows 上の IBM MQ を管理する権限』
- IBM i 92 ページの『IBM i 上の IBM MQ を管理する権限』
- z/OS 93 ページの『Authority to administer IBM MQ on z/OS』

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、95 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、116 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。

- 129 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』
- 130 ページの『IBM MQ Publish/Subscribe 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。

- 386 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』
- 387 ページの『MCAUSER ユーザー ID へのユーザー ID のマッピング』

- 388 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』
- 390 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。

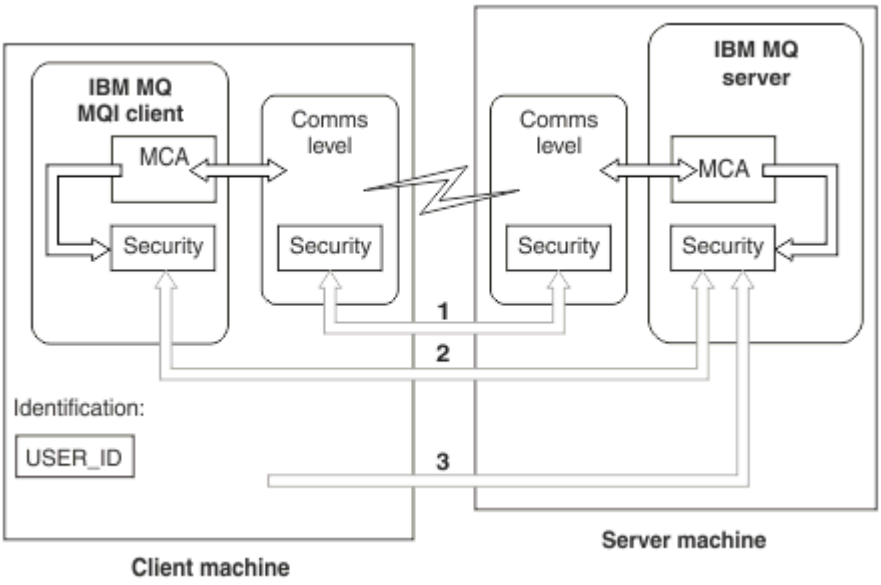


図 9. クライアント/サーバー間接続のセキュリティー

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、18 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、51 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、ユーザー ID とパスワード、または認証トークンを送信します。 詳しくは、73 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。

IBM MQ MQI clients 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。

Linux V 9.4.0 AIX IBM MQ 9.3.4 以降、AIX または Linux システム上で稼働する IBM MQ キュー・マネージャーに接続する IBM MQ MQI clients は、MQCSP 構造で認証トークンを送信することもできます。

警告 : 場合によっては、クライアント・アプリケーションの MQCSP 構造内のパスワード または認証トークン が、プレーン・テキストでネットワーク経由で送信されます。 クライアント・アプリケーション・パスワード および認証トークン が適切に保護されるようにするには、31 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。

- Linux z/OS AIX z/OS、AIX 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、321 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。

AIX 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、353 ページの『AIX 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

@ 文字を含むユーザー ID (例えば、 abc@d) でクライアントが実行されている場合、 IBM MQ for Windows サーバーは IBM MQ MQI client 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。

ALW *AIX, Linux, and 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。

AIX, Linux, and 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、AIX and Linux 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、95 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、93 ページの『Command security and command resource security on z/OS』を参照してください。

IBM MQ for AIX, Linux, and Windows の管理に必要な権限の詳細については、関連情報を参照してください。

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、AIX, Linux, and 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、95 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、162 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 *Authority to administer IBM MQ on z/OS*

This collection of topics describes various aspects of the authority you need to administer IBM MQ for z/OS.

z/OS *Authority checks on z/OS*

IBM MQ for z/OS uses the System Authorization Facility (SAF) to route requests for authority checks to an external security manager (ESM) such as the z/OS Security Server Resource Access Control Facility (RACF). IBM MQ does no authority checks of its own.

It is assumed that you are using RACF as your ESM. If you are using a different ESM, you might need to interpret the information provided for RACF in a way that is relevant to your ESM.

You can specify whether you want authority checks turned on or off for each queue manager individually or for every queue manager in a queue sharing group. This level of control is called *subsystem security*. If you turn subsystem security off for a particular queue manager, no authority checks are carried out for that queue manager.

If you turn subsystem security on for a particular queue manager, authority checks can be performed at two levels:

Queue sharing group level security
Authority checks use RACF profiles that are shared by all queue managers in the queue sharing group. This means that there are fewer profiles to define and maintain, making security administration easier.

Queue manager level security
Authority checks use RACF profiles specific to the queue manager.

You can use a combination of queue sharing group and queue manager level security. For example, you can arrange for profiles specific to a queue manager to override those of the queue sharing group to which it belongs.

Subsystem security, queue sharing group level security, and queue manager level security are turned on or off by defining *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to IBM MQ.

z/OS *Command security and command resource security on z/OS*

Command security relates to the authority to issue a command; command resource authority relates to the authority to perform an operation on a resource. Both are implemented y using RACF classes.

Authority checks are carried out when an IBM MQ administrator issues an MQSC command. This is called *command security*.

To implement command security, you must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. The name of a profile for command security contains the name of an MQSC command.

Some MQSC commands perform an operation on an IBM MQ resource, such as the DEFINE QLOCAL command to create a local queue. When an administrator issues an MQSC command, authority checks are carried out to determine whether the requested operation can be performed on the resource specified in the command. This is called *command resource security*.

To implement command resource security, you must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. The name of a profile for command resource security contains the name of an IBM MQ resource and its type (QUEUE, PROCESS, NAMELIST, TOPIC, AUTHINFO, or CHANNEL).

Command security and command resource security are independent. For example, when an administrator issues the command:

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

the following authority checks are performed:

- Command security checks that the administrator is authorized to issue the DEFINE QLOCAL command.
- Command resource security checks that the administrator is authorized to perform an operation on the local queue called MOON.EUROPA.

Command security and command resource security can be turned on or off by defining switch profiles.

z/OS *MQSC commands and the system command input queue on z/OS*

Use this topic to understand how the command server processes MQSC commands directed to the system command input queue on z/OS.

Command security and command resource security are also used when the command server retrieves a message containing an MQSC command from the system command input queue. The user ID that is used for the authority checks is the one found in the *UserIdentifier* field in the message descriptor of the message containing the MQSC command. This user ID must have the required authorities on the queue manager where the command is processed. For more information about the *UserIdentifier* field and how it is set, see Message context.

Messages containing MQSC commands are sent to the system command input queue in the following circumstances:

- The operations and control panels send MQSC commands to the system command input queue of the target queue manager. The MQSC commands correspond to the actions you choose on the panels. The *UserIdentifier* field in each message is set to the TSO user ID of the administrator.
- The COMMAND function of the IBM MQ utility program, CSQUTIL, sends the MQSC commands in the input data set to the system command input queue of the target queue manager. The COPY and EMPTY functions send DISPLAY QUEUE and DISPLAY STGCLASS commands. The *UserIdentifier* field in each message is set to the job user ID.
- The MQSC commands in the CSQINPX data sets are sent to the system command input queue of the queue manager to which the channel initiator is connected. The *UserIdentifier* field in each message is set to the channel initiator address space user ID.

  No authority checks are performed when MQSC commands are issued from the CSQINP1 and CSQINP2 data sets. You can control who is allowed to update these data sets using RACF data set protection.
- Within a queue sharing group, a channel initiator might send START CHANNEL commands to the system command input queue of the queue manager to which it is connected. A command is sent when an outbound channel that uses a shared transmission queue is started by triggering. The *UserIdentifier* field in each message is set to the channel initiator address space user ID.
- An application can send MQSC commands to a system command input queue. By default, the *UserIdentifier* field in each message is set to the user ID associated with the application.
- On AIX, Linux, and Windows systems, the **runmqsc** control command can be used in indirect mode to send MQSC commands to the system command input queue of a queue manager on z/OS. The *UserIdentifier* field in each message is set to the user ID of the administrator who issued the **runmqsc** command.

z/OS *Access to the queue manager data sets on z/OS*

IBM MQ for z/OS administrators need authority to access the queue manager data sets. Use this topic to understand which data sets need RACF protection.

These data sets include:

- The data sets referred to by CSQINP1, CSQINP2, and CSQINPT in the started task procedure of the queue manager.

- The queue manager's page sets, active log data sets, archive log data sets, and bootstrap data sets (BSDSs)
- The data sets referred to by CSQXLIB and CSQINPX in the channel initiator's started task procedure

You must protect the data sets so that no unauthorized user can start a queue manager or gain access to any queue manager data. To do this, use RACF data set protection.

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。

このコンテキストでは、アプリケーションには、ユーザーおよびベンダーによって作成されたアプリケーション、および IBM MQ for z/OS で提供されたアプリケーションが含まれます。

z/OS 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、116 ページの『チャネル許可』を参照してください。

z/OS 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、93 ページの『Command security and command resource security on z/OS』および 94 ページの『MQSC commands and the system command input queue on z/OS』を参照してください。

IBM i 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、95 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 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、この伝送キューがローカルに定義されているかどうかに関係なく実行されます。
- Multi 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、98 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。

IBM i ALW IBM i *IBM i 、AIX, 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 、 AIX, 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。 AIX 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。

AIX, Linux, and 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、 AIX, 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 ***Authority to work with IBM MQ objects on z/OS***

On z/OS, there are seven categories of authority check associated with calls to the MQI. You must define certain RACF profiles and give appropriate access to these profiles. Use the *RESLEVEL* profile to control how many users IDs are checked.

The seven categories of authority check associated with calls to the MQI:

Connection security
The authority checks that are performed when an application connects to a queue manager

Queue security
The authority checks that are performed when an application opens a queue or deletes a permanent dynamic queue

Process security
The authority checks that are performed when an application opens a process object

Namelist security
The authority checks that are performed when an application opens a namelist object

Alternate user security
The authority checks that are performed when an application requests alternate user authority when opening an object

Context security
The authority checks that are performed when an application opens a queue and specifies that it intends to set or pass the context information in the messages it puts on the queue

Topic security
The authority checks that are performed when an application opens a topic

Each category of authority check is implemented in the same way that command security and command resource security are implemented. You must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. For queue security, the level of access determines the types of operation the application can perform on a queue. For context security, the level of access determines whether the application can:

- Pass all the context fields
- Pass all the context fields and set the identity context fields
- Pass and set all the context fields

Each category of authority check can be turned on or off by defining switch profiles.

All the categories, except connection security, are known collectively as *API-resource security*.

By default, when an API-resource security check is performed as a result of an MQI call from an application using a batch connection, only one user ID is checked. When a check is performed as a result of an MQI call from a CICS or IMS application, or from the channel initiator, two user IDs are checked.

By defining a *RESLEVEL profile*, however, you can control whether zero, one, or two users IDs are checked. The number of user IDs that are checked is determined by the user ID associated with the type of connection when an application connects to the queue manager and the access level that user ID has to the RESLEVEL profile. The user ID associated with each type of connection is:

- The user ID of the connecting task for batch connections
- The CICS address space user ID for CICS connections
- The IMS region address space user ID for IMS connections
- The channel initiator address space user ID for channel initiator connections

For more information about the authority to work with IBM MQ objects on z/OS, see “Authority to administer IBM MQ on z/OS” on page 93.

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 AIX *AIX 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。

AIX 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*

AIX, Linux, and Windows 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。 AIX, Linux, and Windows 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。

AIX, Linux, and Windows 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 AIX, 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 106 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 106 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 (非 z/OS)の場合は、 `inetd.conf` 項目からのユーザー ID、またはリスナーを開始したユーザー ID。
- SNA (非 z/OS)の場合、SNA Server 項目からのユーザー ID、または (存在しない場合は) 着信接続要求からのユーザー ID、またはリスナーを開始したユーザー 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。

ALW AIX, 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 AIX AIX 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

107 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。

112 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。

119 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、107 ページの図 10 です。

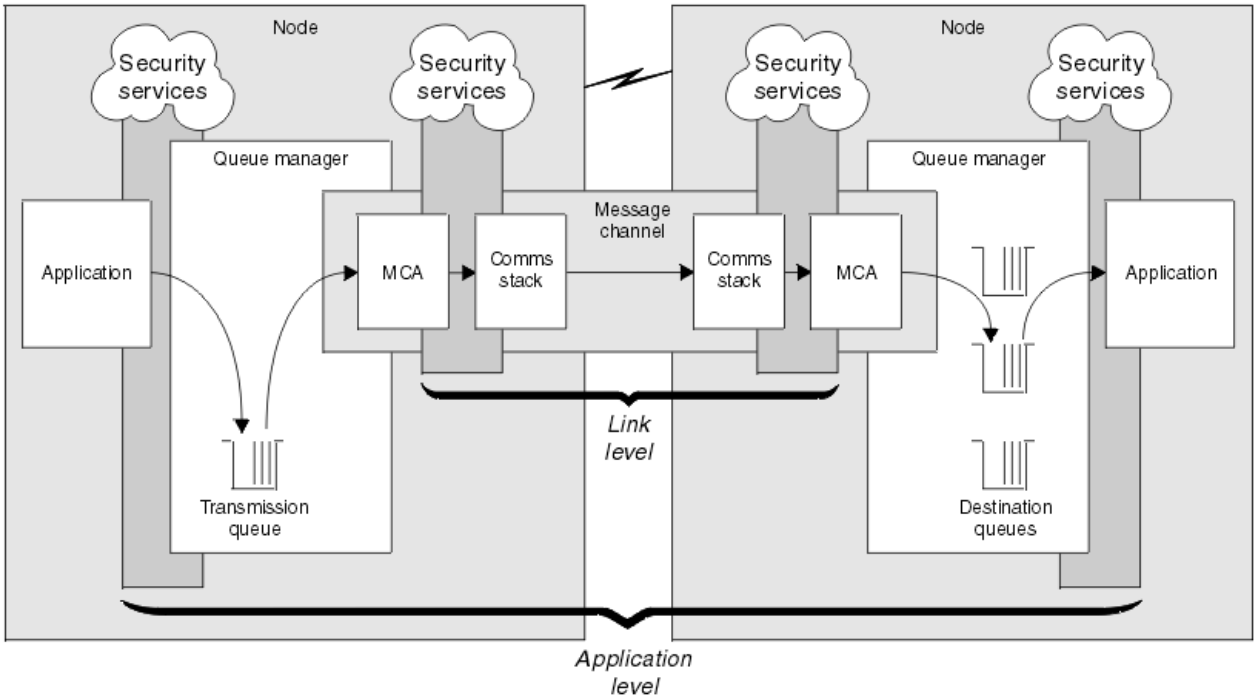


図 *10.* リンク・レベル・セキュリティーとアプリケーション・レベル・セキュリティー

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。

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。 for more information.

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、107 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、107 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、24 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。 詳しくは、51 ページの『チャネル認証レコード』を参照してください。

独自のリンク・レベル・セキュリティーの提供

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。

112 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、107 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。

Advanced Message Security の計画

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、599 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、113 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。

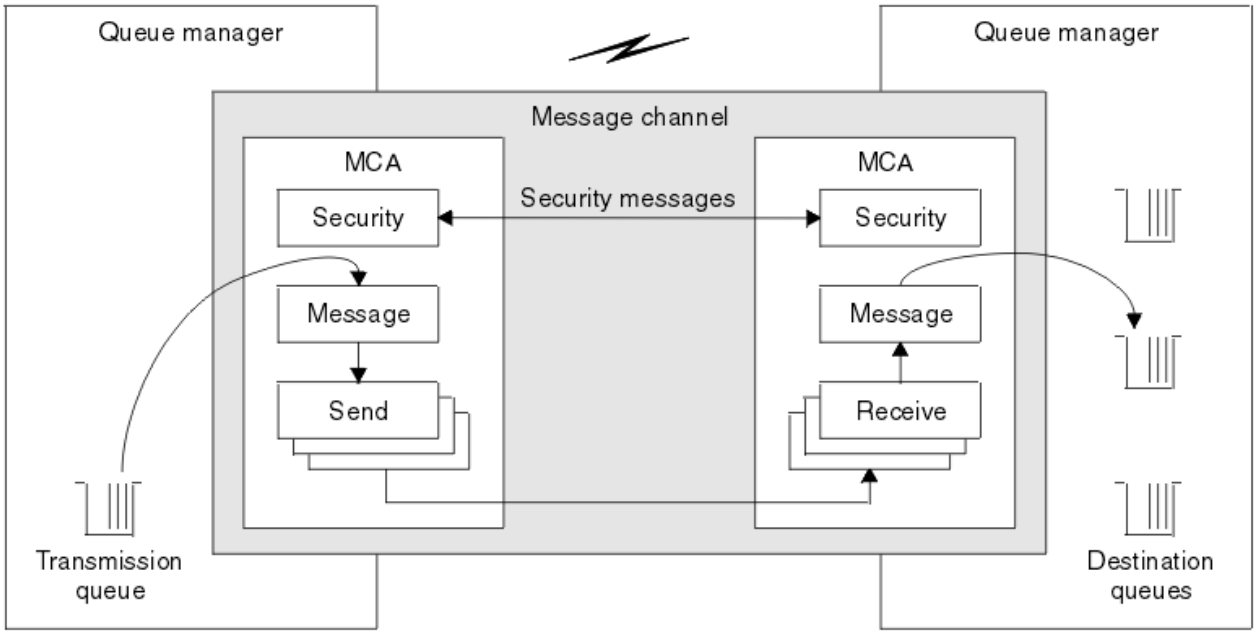


図 *11.* メッセージ・チャネル上のセキュリティー出口、メッセージ出口、送信出口、および受信出口

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、110 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、110 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、110 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

119 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。

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

111 ページの『Advanced Message Security の計画』
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。

関連資料

API 出口参照
チャネル出口呼び出しおよびデータ構造体

監査の計画

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

アクティビティー・モニターには、いくつかの面があります。 考慮しなければならない面はしばしば監査員要件により定義され、これらの要件はしばしば、HIPAA (医療保険の積算と責任に関する法律) または SOX (サーベンス・オクスリー) などの規制基準により主導されます。 IBM 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
51 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。

関連資料
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。136 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、112 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。 鍵リポジトリー内の証明書の数を約 50 個未満に保ち、より大きな鍵リポジトリーでは IBM Global Security Kit (GSKit) のパフォーマンスが大幅に低下するため、100 個を最大にすることを検討してください。

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。

ALW AIX, Linux, and Windows システムでは、自己署名証明書を 使用できます。 説明は、543 ページの『AIX, 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、286 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、311 ページの『Creating a self-signed personal certificate on z/OS』 または 312 ページの『Requesting a personal certificate on z/OS』 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、47 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、45 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。

キー RPWD
AIX, Linux, and Windows システムでは、 IBM MQ が鍵リポジトリーにアクセスするために使用するパスワードを定義します。 このフィールドは、パスワード保護システムを使用して暗号化されます。

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
AIX, Linux, and Windows 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
AIX, 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。 GSKit を使用すると、AIX, Linux, and Windows システムで TLS セキュリティーを使用できます。

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、 AIX, Linux, and Windows で鍵および証明書を管理するために使用できるコマンド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
- 541 ページの『鍵と証明書の管理 (AIX, Linux, and Windows)』

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。

- 24 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 server connection channel*

The IBM MQ for z/OS SVRCONN channel is not secure without implementing channel authentication, or adding a security exit using TLS. SVRCONN channels do not have a security exit defined by default.

Security concerns

SVRCONN channels are not secure as initially defined, SYSTEM.DEF.SVRCONN for example. To secure a SVRCONN channel you must set up channel authentication using the SET CHLAUTH command, or install a security exit and implement TLS.

You must use a publicly available sample security exit, write a security exit yourself, or purchase a security exit.

There are several samples available that you can use as a good starting point for writing your own SVRCONN channel security exit.

In IBM MQ for z/OS, the member CSQ4BCX3 in your hlq.SCSQC37S library is a security exit sample written in the C language. Sample CSQ4BCX3 is also shipped pre-compiled in your hlq.SCSQAUTH library.

You can implement the CSQ4BCX3 sample exit by copying the compiled member hlq.SCSQAUTH(CSQ4BCX3) into a load library that is allocated to the CSQXLIB DD in your CHIN Proc. Note that the CHIN requires the load library to be set as "Program Controlled".

Alter your SVRCONN channel to set CSQ4BCX3 as the security exit.

When a client connects using that SVRCONN channel, CSQ4BCX3 will authenticate using the **RemoteUserIdentifier** and **RemotePassword** pair from MQCD or, from IBM MQ for z/OS 9.1.4, the **CSPUserIdPtr** and **CSPPasswordPtr** pair from the MQCSP. If authentication is successful it will copy **RemoteUserIdentifier** into **MCAUserIdentifier**, changing the identity context of the thread.

For Long Term Support and Continuous Delivery before IBM MQ for z/OS 9.1.4, when a client connects using that SVRCONN channel, CSQ4BCX3 will authenticate using the **RemoteUserIdentifier** and **RemotePassword** pair from MQCD. If authentication is successful it will copy **RemoteUserIdentifier** into **MCAUserIdentifier**, changing the identity context of the thread.

If you are writing an IBM MQ Java client you can use pop-ups to query the user and set MQEnvironment.userID and MQEnvironment.password. These values will be passed when the connection is made.

Now that you have a functional security exit, there is the additional concern that the userid and password are being transmitted in plain text across the network when the connection is made, as are the contents of any subsequent IBM MQ messages. You can use TLS to encrypt this initial connection information as well as the contents of any IBM MQ messages.

Example

To secure the IBM MQ Explorer SVRCONN channel SYSTEM.ADMIN.SVRCONN complete the following steps:

1. Copy hlq.SCSQAUTH(CSQ4BCX3) into a load library that is allocated to the CSQXLIB DD in the CHINIT Proc.
2. Verify that load library is Program Controlled.
3. Alter the SYSTEM ADMIN.SVRCONN to use security exit CSQ4BCX3.
4. In IBM MQ Explorer, right-click the z/OS Queue Manager name, select **Connection Details** > **Properties** > **Userid** and enter your z/OS user ID.
5. Connect to the z/OS Queue Manager by entering a password.

Additional information

For exit CSQ4BCX3 to run in a Program Controlled environment, everything loaded into the CHIN address space must be loaded from a Program Controlled library, for example, all libraries in STEPLIB and any libraries named on CSQXLIB DD. To set a load library as Program Controlled issue RACF commands. In the following example the load library name is MY.TEST.LOADLIB.

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

To alter the SVRCONN channel to implement CSQ4BCX3, issue the following IBM MQ command:

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

In the example above, the SVRCONN channel name being used is SYSTEM ADMIN.SVRCONN.

See “チャネル出口プログラム” on page 112 for more information about channel exits.

Related tasks

Writing channel exit programs on z/OS

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。 z/OS での Triple DES 24 バイト暗号化など、ご使用のプラットフォームで使用できる可能性がある、より強力な形式の暗号化については、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。

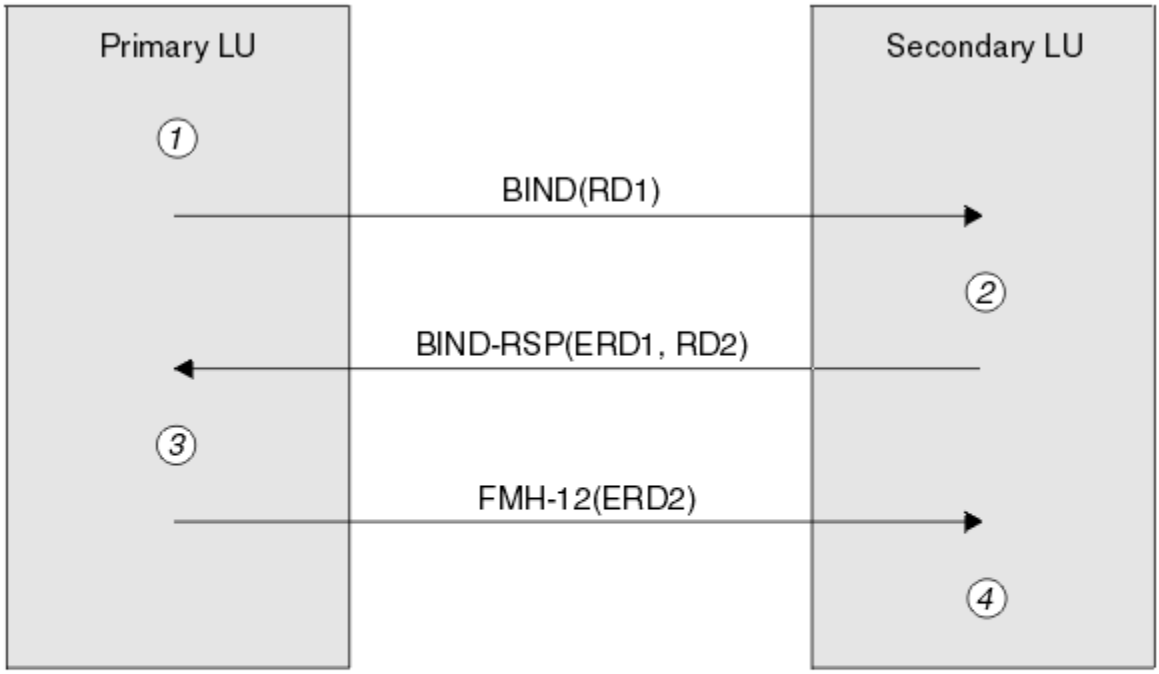
セッション・レベルの認証

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。

125 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、125 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 z/OS に固有の情報については、「z/OS MVS 計画: APPC/MVS 管理」を参照してください。

CPI-C について詳しくは、 CPI コミュニケーションの使用を参照してください。

APPC/MVS TP 会話呼び出し可能サービスの詳細については、 APPC/MVS TP 会話呼び出し可能サービスを参照してください。

Multi *Multiplatforms* での会話レベルの認証に対するサポート

このトピックでは、Multiplatforms 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。

Multiplatforms での会話レベル認証のサポートを 127 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。

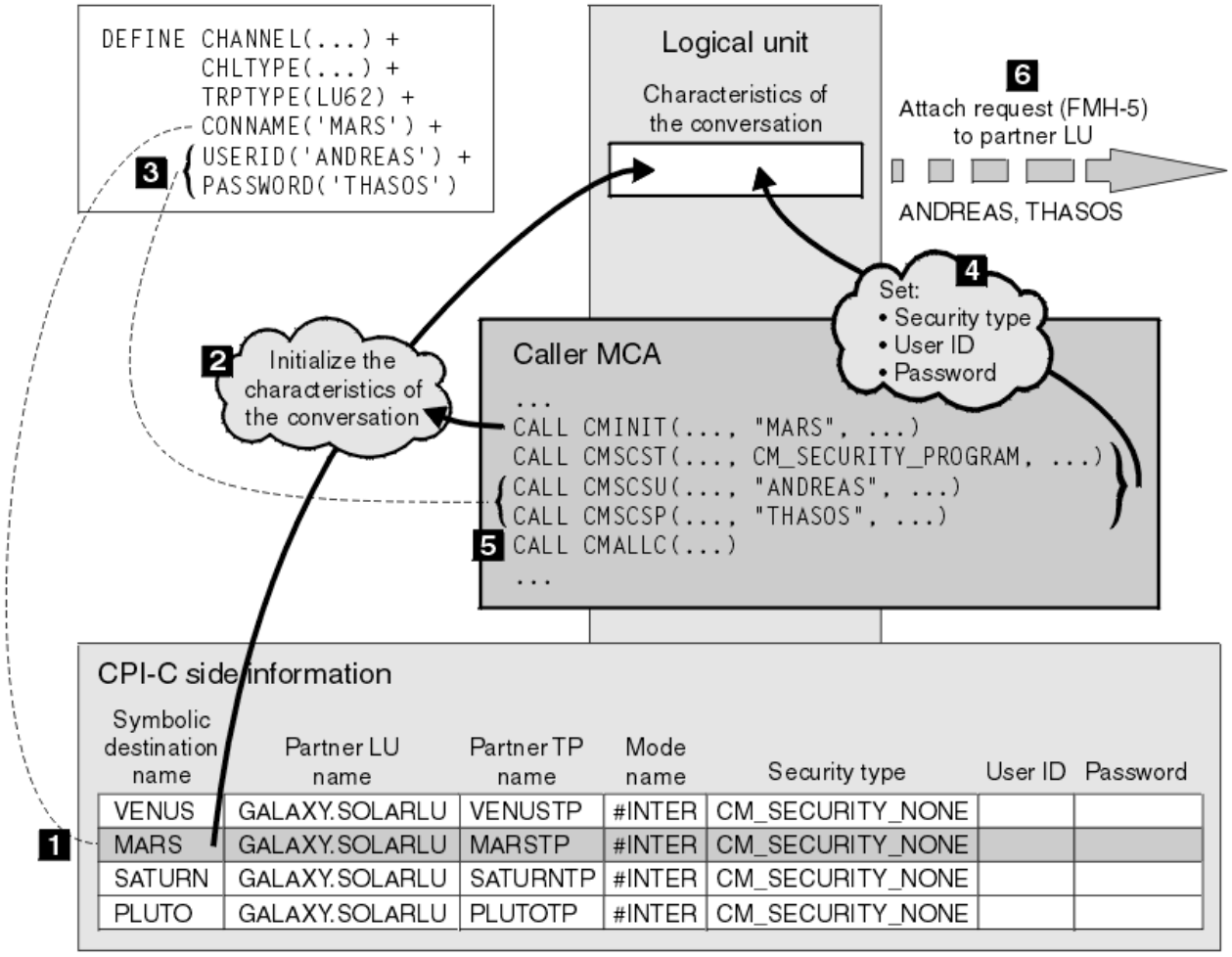


図 *13.* 会話レベルの認証に対する *IBM MQ* のサポート

Multiplatforms 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。

Conversation level authentication and IBM MQ for z/OS
Use this topic to gain an overview of how conversation level authentication works, on z/OS.

On IBM MQ for z/OS, MCAs do not use CPI-C. Instead, they use APPC/MVS TP Conversation Callable Services, an implementation of Advanced Program-to-Program Communication (APPC), which has some CPI-C features. When a caller MCA allocates a conversation, a security type of SAME is specified on the call. Therefore, because an APPC/MVS LU supports persistent verification only for inbound conversations, not for outbound conversations, there are two possibilities:

- If the partner LU trusts the APPC/MVS LU and will accept an already verified user ID, the APPC/MVS LU sends an attach request containing:
  - The channel initiator address space user ID
  - A security profile name, which, if RACF is used, is the name of the current connect group of the channel initiator address space user ID
  - An already verified indicator
- If the partner LU does not trust the APPC/MVS LU and will not accept an already verified user ID, the APPC/MVS LU sends an attach request containing no security information.

On IBM MQ for z/OS, the USERID and PASSWORD parameters on the DEFINE CHANNEL command cannot be used for a message channel and are valid only at the client connection end of an MQI channel. Therefore, an attach request from an APPC/MVS LU never contains values specified by these parameters.

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、263 ページの『Security in queue manager clusters on z/OS』 を参照してください。

関連タスク

481 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、104 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、11 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 IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru は、ファイアウォールを介した通信を簡素化できます。

MQIPT を使用すると、直接 TCP/IP 接続を必要とせずに、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ます。 このアーキテクチャーは、ファイアウォールが 2 つのシステム間の直接 TCP/IP 接続を禁止している場合に役立ちます。 プロキシーとして MQIPT を使用することで、ファイアウォールを介した IBM MQ チャネル・データの通過をより簡単かつ管理しやすくすることができます。 MQIPT は、Transport Layer Security (TLS) を使用してインターネット経由で送信される IBM MQ データ、および HTTP 内のトンネル IBM MQ データを保護することもできます。

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 security implementation checklist

This topic gives a step-by-step procedure you can use to work out and define the security implementation for each of your IBM MQ queue managers.

RACF provides definitions for the IBM MQ security classes in its supplied static Class Descriptor Table (CDT). As you work through the checklist, you can determine which of these classes your setup requires. You must ensure that they are activated as described in “RACF security classes” on page 189.

Refer to other sections for details, in particular “Profiles used to control access to IBM MQ resources” on page 199.

If you require security checking, follow this checklist to implement it:

1. Activate the RACF MQADMIN (uppercase profiles) or MXADMIN (mixed case profiles) class.
   - Do you want security at queue sharing group level, queue manager level, or a combination of both?

     See, “Profiles to control queue sharing group or queue manager level security” on page 194.
2. Do you need connection security?
   - **Yes**: Activate the MQCONN class. Define appropriate connection profiles at either queue manager level or queue sharing group level in the MQCONN class. Then permit the appropriate users or groups access to these profiles.

     Note: Only users of the MQCONN API request or CICS or IMS address space user IDs need to have access to the corresponding connection profile.
   - **No**: Define an hlq.NO.CONNECT.CHECKS profile at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class.
3. Do you need security checking on commands?
   - **Yes**: Activate the MQCMDS class. Define appropriate command profiles at either queue manager level or queue sharing group level in the MQCMDS class. Then permit the appropriate users or groups access to these profiles.

     If you are using a queue sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator. See “Setting up IBM MQ for z/OS resource security” on page 255.

- **No**: Define an hlq.NO.CMD.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.

4. Do you need security on the resources used in commands?
   - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define appropriate profiles for protecting resources on commands at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the appropriate users or groups access to these profiles. Set the CMDUSER parameter in CSQ6SYSP to the default user ID to be used for command security checks.

     If you are using a queue sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator. See "Setting up IBM MQ for z/OS resource security" on page 255.
   - **No**: Define an hlq.NO.CMD.RESC.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
5. Do you need queue security?
   - **Yes**: Activate the MQQUEUE or MXQUEUE class. Define appropriate queue profiles for the required queue manager or queue sharing group in the MQQUEUE or MXQUEUEclass. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.QUEUE.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
6. Do you need process security?
   - **Yes**: Activate the MQPROC or MXPROC class. Define appropriate process profiles at either queue manager or queue sharing group level and permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.PROCESS.CHECKS profile for the appropriate queue manager or queue sharing group in the MQADMIN or MXADMIN class.
7. Do you need namelist security?
   - **Yes**: Activate the MQNLIST or MXNLISTclass. Define appropriate namelist profiles at either queue manager level or queue sharing group level in the MQNLIST or MXNLIST class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.NLIST.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
8. Do you need topic security?
   - **Yes**: Activate the MXTOPIC class. Define appropriate topic profiles at either queue manager level or queue sharing group level in the MXTOPIC class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.TOPIC.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
9. Do any users need to protect the use of the `MQOPEN` or `MQPUT1` options relating to the use of context?
   - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define hlq.CONTEXT.queuename profiles at the queue, queue manager, or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.CONTEXT.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
10. Do you need to protect the use of alternative user IDs?
    - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define the appropriate hlq.ALTERNATE.USER. *alternateuserid* profiles for the required queue manager or queue sharing group and permit the required users or groups access to these profiles.
    - **No**: Define the profile hlq.NO.ALTERNATE.USER.CHECKS for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.

11. Do you need to tailor which user IDs are to be used for resource security checks through RESLEVEL?
    - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define an hlq.RESLEVEL profile at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the required users or groups access to the profile.
    - **No**: Ensure that no generic profiles exist in the MQADMIN or MXADMIN class that can apply to hlq.RESLEVEL. Define an hlq.RESLEVEL profile for the required queue manager or queue sharing group and ensure that no users or groups have access to it.
12. Do you need to 'timeout' unused user IDs from IBM MQ ?
    - **Yes**: Determine what timeout values you would like to use and issue the MQSC ALTER SECURITY command to change the TIMEOUT and INTERVAL parameters.
    - **No**: Issue the MQSC ALTER SECURITY command to set the INTERVAL value to zero.

    Note: Update the CSQINP1 initialization input data set used by your subsystem so that the MQSC ALTER SECURITY command is issued automatically when the queue manager is started.
13. Do you use distributed queuing?
    - **Yes**: Use channel authentication records. For more information, see “チャネル認証レコード” on page 51.
    - You can also determine the appropriate MCAUSER attribute value for each channel, or provide suitable channel security exits.
14. Do you want to use Transport Layer Security (TLS)?
    - **Yes**: To specify that any user presenting an TLS personal certificate containing a specified DN is to use a specific MCAUSER, set a channel authentication record of type SSLPEERMAP. You can specify a single distinguished name or a pattern including wildcards.
    - Plan your TLS infrastructure. Install the System SSL feature of z/OS. In RACF, set up your certificate name filters (CNFs), if you are using them, and your digital certificates. Set up your SSL key ring. Ensure that the SSLKEYR queue manager attribute is nonblank and points to your SSL key ring. Also ensure that the value of SSLTASKS is at least 2.
    - **No**: Ensure that SSLKEYR is blank, and SSLTASKS is zero.

    For further details about TLS, see “IBM MQ での TLS セキュリティー・プロトコル” on page 24.
15. Do you use clients?
    - **Yes**: Use channel authentication records.
    - You can also determine the appropriate MCAUSER attribute value for each server-connection channel, or provide suitable channel security exits if required.
16. Check your switch settings.

    IBM MQ issues messages when the queue manager is started that display your security settings. Use these messages to determine whether your switches are set correctly.
17. Do you send passwords from client applications?
    - **Yes**: Ensure that the z/OS feature is installed and Integrated Cryptographic Service Facility (ICSF) is started for the best protection.
    - **No**: You can ignore the error message reporting that ICSF has not started.

    For further information about ICSF see “Using the Integrated Cryptographic Service Facility (ICSF)” on page 263

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。

ALW AIX, Linux, and Windows でのセキュリティーのセットアップ

AIX, 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。

ALW AIX, 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。

ALW *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。

135 ページの表 10、135 ページの表 11、136 ページの表 12、および 136 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 (137 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 (137 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 **(137 ページの『2』)**	MQZAO_INPUT	適用外	適用外
MQOO_OUTPUT (通常キュー) **(137 ページの『3』)**	MQZAO_OUTPUT	適用外	適用外
MQOO_PASS_IDENTITY_CONTEXT **(137 ページの『4』)**	MQZAO_PASS_IDENTITY_CONTEXT	適用外	検査しない
MQOO_PASS_ALL_CONTEXT **(137 ページの『4』, 137 ページの『5』)**	MQZAO_PASS_ALL_CONTEXT	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 (137 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_SET_IDENTITY_CONTEXT **(137 ページの『4』, 137 ページの『5』)**	MQZAO_SET_IDENTITY_CONTEXT	適用外	MQZAO_SET_IDENTITY_CONTEXT **(137 ページの『6』)**
MQOO_SET_ALL_CONTEXT **(137 ページの『4』, 137 ページの『7』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(137 ページの『6』)**
MQOO_OUTPUT (伝送キュー) **(137 ページの『8』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(137 ページの『6』)**
MQOO_SET	MQZAO_SET	適用外	検査しない
MQOO_ALTERNATE_USER_AUTHORITY	**(137 ページの『9』)**	**(137 ページの『9』)**	MQZAO_ALTERNATE_USER_AUTHORITY **(137 ページの『9』、137 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 (137 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 **(137 ページの『11』)**	適用外	検査しない
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT **(137 ページの『11』)**	適用外	検査しない
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT **(137 ページの『11』)**	適用外	MQZAO_SET_IDENTITY_CONTEXT **(137 ページの『6』)**
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT **(137 ページの『11』)**	適用外	MQZAO_SET_ALL_CONTEXT **(137 ページの『6』)**
(伝送キュー) **(137 ページの『8』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(137 ページの『6』)**
MQPMO_ALTERNATE_USER_AUTHORITY	**(137 ページの『12』)**	適用外	MQZAO_ALTERNATE_USER_AUTHORITY **(137 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 (137 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 **(137 ページの『13』)**	適用外	適用外

表 *13. MQCLOSE* 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 (137 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 _PURGE	MQZAO_DELETE **(137 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 137 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。

ALW エスケープ *PCF* の中の *MQSC* コマンドに関する許可

ここでは、エスケープ PCF に含まれている各 MQSC コマンドで必要な権限をまとめます。

「適用外」は、この操作がこのオブジェクト・タイプには該当しないことを意味します。

コマンドを実行依頼するプログラムを実行させるユーザー ID には、以下の権限も必要になります。

- キュー・マネージャーに対する MQZAO_CONNECT 権限

- PCF コマンドを実行するためのキュー・マネージャー上の MQZAO_DISPLAY 権限
- エスケープ PCF コマンドのテキスト内で MQSC コマンドを発行する権限

ALTER *object*

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

CLEAR *object*

オブジェクト	**必要な権限**
キュー	MQZAO_CLEAR
トピック	MQZAO_CLEAR
プロセス	適用外
キュー・マネージャー	適用外
名前リスト	適用外
認証情報	適用外
チャネル	適用外
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 (142 ページの『1』)

オブジェクト	**必要な権限**
キュー	MQZAO_CREATE (142 ページの『2』)
トピック	MQZAO_CREATE (142 ページの『2』)
プロセス	MQZAO_CREATE (142 ページの『2』)
キュー・マネージャー	適用外
名前リスト	MQZAO_CREATE (142 ページの『2』)
認証情報	MQZAO_CREATE (142 ページの『2』)

オブジェクト	必要な権限
チャネル	MQZAO_CREATE (142 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (142 ページの『2』)
リスナー	MQZAO_CREATE (142 ページの『2』)
サービス	MQZAO_CREATE (142 ページの『2』)
コミュニケーション情報	MQZAO_CREATE (142 ページの『2』)

DEFINE 「オブジェクト」 REPLACE (142 ページの『1』 142 ページの『3』)

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

DELETE *object*

オブジェクト	必要な権限
キュー	MQZAO_DELETE
トピック	MQZAO_DELETE
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

オブジェクト	必要な権限
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

オブジェクト	必要な権限
サービス	MQZAO_CONTROL
コミュニケーション情報	適用外

チャネル・コマンド

コマンド	オブジェクト	必要な権限
PING CHANNEL	チャネル	MQZAO_CONTROL
RESET CHANNEL	チャネル	MQZAO_CONTROL_EXTENDED
RESOLVE CHANNEL	チャネル	MQZAO_CONTROL_EXTENDED

サブスクリプション・コマンド

コマンド	オブジェクト	必要な権限
ALTER SUB	トピック	MQZAO_CONTROL
DEFINE SUB	トピック	MQZAO_CONTROL
DELETE SUB	トピック	MQZAO_CONTROL
DISPLAY SUB	トピック	MQZAO_DISPLAY

Security Commands

コマンド	オブジェクト	必要な権限
SET AUTHREC	キュー・マネージャー	MQZAO_CHANGE
DELETE AUTHREC	キュー・マネージャー	MQZAO_CHANGE
DISPLAY AUTHREC	キュー・マネージャー	MQZAO_DISPLAY
DISPLAY AUTHSERV	キュー・マネージャー	MQZAO_DISPLAY
DISPLAY ENTAUTH	キュー・マネージャー	MQZAO_DISPLAY
SET CHLAUTH	キュー・マネージャー	MQZAO_CHANGE
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

コマンド	オブジェクト	必要な権限
DISPLAY TPSTATUS	キュー・マネージャー	MQZAO_DISPLAY

クラスター・コマンド

コマンド	オブジェクト	必要な権限
DISPLAY CLUSQMGR	キュー・マネージャー	MQZAO_DISPLAY
REFRESH CLUSTER	「mqm」グループ・メンバーシップが必要	
RESET CLUSTER	「mqm」グループ・メンバーシップが必要	
SUSPEND QMGR	「mqm」グループ・メンバーシップが必要	
RESUME QMGR	「mqm」グループ・メンバーシップが必要	

Other Administrative Commands

コマンド	オブジェクト	必要な権限
PING QMGR	キュー・マネージャー	MQZAO_DISPLAY
REFRESH QMGR	キュー・マネージャー	MQZAO_CHANGE
RESET QMGR	キュー・マネージャー	MQZAO_CHANGE
DISPLAY CONN	キュー・マネージャー	MQZAO_DISPLAY
STOP CONN	キュー・マネージャー	MQZAO_CHANGE

注 :

1. DEFINE コマンドでは、LIKE オブジェクトが指定されている場合は LIKE オブジェクトに関する、または LIKE が省略されている場合は適切な SYSTEM.DEFAULT.xxx オブジェクトに関する、MQZAO_DISPLAY 権限も必要です。
2. MQZAO_CREATE 権限は、特定のオブジェクトまたはオブジェクト・タイプに特有のものではありません。 `setmqaut` コマンドで QMGR のオブジェクト・タイプを指定すれば、指定したキュー・マネージャーのすべてのオブジェクトについて作成権限が与えられます。
3. これは、置き換えようとするオブジェクトがすでに存在している場合に適用されます。 存在していない場合は、DEFINE *object* NOREPLACE の検査になります。

関連情報

クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス

ALW *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
Process	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)**
Process	MQZAO_CREATE **(2)**
キュー・マネージャー	適用外
名前リスト	MQZAO_CREATE **(2)**
認証情報	MQZAO_CREATE **(2)**
チャネル	MQZAO_CREATE **(2)**

オブジェクト	必要な権限
クライアント接続チャネル	MQZAO_CREATE **(2)**
リスナー	MQZAO_CREATE **(2)**
サービス	MQZAO_CREATE **(2)**
通信情報	MQZAO_CREATE **(148 ページの『2』)**

「オブジェクト」のコピー (置換を伴う) (1、4)

オブジェクト	必要な権限
キュー	MQZAO_CHANGE
トピック	MQZAO_CHANGE
Process	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

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)**
Process	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 (置換を伴う) (3、4)

オブジェクト	必要な権限
キュー	MQZAO_CHANGE
トピック	MQZAO_CHANGE

オブジェクト	必要な権限
Process	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
Process	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
Process	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

オブジェクト	必要な権限
通信情報	MQZAO_DISPLAY

Inquire *object* names

オブジェクト	必要な権限
キュー	検査しない
トピック	検査しない
プロセス	検査しない
キュー・マネージャー	検査しない
名前リスト	検査しない
認証情報	検査しない
チャネル	検査しない
クライアント接続チャネル	検査しない
リスナー	検査しない
サービス	検査しない
コミュニケーション情報	検査しない

Start *object*

オブジェクト	必要な権限
キュー	適用外
トピック	適用外
プロセス	適用外
キュー・マネージャー	適用外
名前リスト	適用外
認証情報	適用外
チャネル	MQZAO_CONTROL
クライアント接続チャネル	適用外
リスナー	MQZAO_CONTROL
サービス	MQZAO_CONTROL
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

オブジェクト	必要な権限
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
削除権限レコード	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
トピック状況の照会	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
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
キュー・マネージャーのリフレッシュ	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 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。

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。

- 152 ページの『Prepare IBM MQ Wizard での特別なユーザー・アカウントの構成』
- 153 ページの『IBM MQ と Active Directory の併用』
- 153 ページの『IBM MQ Windows サービスに必要なユーザー権限』

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 (Prepare IBM MQ Wizard 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、153 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、153 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。

表 14. *IBM MQ Windows* サービスに必要なユーザー権限 (続き)	
アクセス権	**説明**
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、Prepare IBM MQ Wizard を使用した 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

Prepare IBM MQ Wizard 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、153 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

Prepare IBM MQ Wizard 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 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、113 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、AIX, Linux, and 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

92 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。
*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。

171 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。

171 ページの表 19、171 ページの表 20、172 ページの表 21、および 172 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 (173 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 (173 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 **(173 ページの『2』)**	MQZAO_INQUIRE **(173 ページの『2』)**	MQZAO_INQUIRE **(173 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 **(173 ページの『3』)**	MQZAO_INPUT	適用外	適用外
MQOO_OUTPUT (通常キュー) **(173 ページの『4』)**	MQZAO_OUTPUT	適用外	適用外
MQOO_PASS_ IDENTITY_CONTEXT **(173 ページの『5』)**	MQZAO_PASS_ IDENTITY_CONTEXT	適用外	検査しない
MQOO_PASS_ALL_ CONTEXT **(173 ページの『5』, 173 ページの『6』)**	MQZAO_PASS_ ALL_CONTEXT	適用外	検査しない

表 *20. 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 (173 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_SET_IDENTITY_CONTEXT **(173 ページの『5』, 173 ページの『6』)**	MQZAO_SET_IDENTITY_CONTEXT	適用外	MQZAO_SET_IDENTITY_CONTEXT **(173 ページの『7』)**
MQOO_SET_ALL_CONTEXT **(173 ページの『5』, 173 ページの『8』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(173 ページの『7』)**
MQOO_OUTPUT (伝送キュー) **(173 ページの『9』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(173 ページの『7』)**
MQOO_SET	MQZAO_SET	適用外	検査しない
MQOO_ALTERNATE_USER_AUTHORITY	**(173 ページの『10』)**	**(173 ページの『10』)**	MQZAO_ALTERNATE_USER_AUTHORITY **(173 ページの『10』、173 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 (173 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 **(173 ページの『12』)**	適用外	検査しない
MQPMO_PASS_ALL_CONTEXT	MQZAO_PASS_ALL_CONTEXT **(173 ページの『12』)**	適用外	検査しない
MQPMO_SET_IDENTITY_CONTEXT	MQZAO_SET_IDENTITY_CONTEXT **(173 ページの『12』)**	適用外	MQZAO_SET_IDENTITY_CONTEXT **(173 ページの『7』)**
MQPMO_SET_ALL_CONTEXT	MQZAO_SET_ALL_CONTEXT **(173 ページの『12』)**	適用外	MQZAO_SET_ALL_CONTEXT **(173 ページの『7』)**
(伝送キュー) **(173 ページの『9』)**	MQZAO_SET_ALL_CONTEXT	適用外	MQZAO_SET_ALL_CONTEXT **(173 ページの『7』)**
MQPMO_ALTERNATE_USER_AUTHORITY	**(173 ページの『13』)**	適用外	MQZAO_ALTERNATE_USER_AUTHORITY **(173 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 (173 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 **(173 ページの『14』)**	適用外	適用外

表 22. *MQCLOSE* 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 (173 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 _PURGE	MQZAO_DELETE **(173 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 173 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 173 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
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

オブジェクト	必要な権限
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 (178 ページの『1』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (178 ページの『2』)
トピック	MQZAO_CREATE (178 ページの『2』)
プロセス	MQZAO_CREATE (178 ページの『2』)
キュー・マネージャー	適用外
名前リスト	MQZAO_CREATE (178 ページの『2』)
認証情報	MQZAO_CREATE (178 ページの『2』)
チャネル	MQZAO_CREATE (178 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (178 ページの『2』)
リスナー	MQZAO_CREATE (178 ページの『2』)
サービス	MQZAO_CREATE (178 ページの『2』)

DEFINE 「オブジェクト」 REPLACE (178 ページの『1』 178 ページの『3』)

オブジェクト	必要な権限
キュー	MQZAO_CHANGE
トピック	MQZAO_CHANGE
プロセス	MQZAO_CHANGE
キュー・マネージャー	適用外
名前リスト	MQZAO_CHANGE
認証情報	MQZAO_CHANGE
チャネル	MQZAO_CHANGE
クライアント接続チャネル	MQZAO_CHANGE
リスナー	MQZAO_CHANGE
サービス	MQZAO_CHANGE

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

オブジェクト	必要な権限
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
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

オブジェクト	必要な権限
クライアント接続チャネル	適用外
リスナー	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
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) (184 ページの『1』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (184 ページの『2』)
トピック	MQZAO_CREATE (184 ページの『2』)
プロセス	MQZAO_CREATE (184 ページの『2』)
キュー・マネージャー	適用外
NamelistMQZAO_CREATE	MQZAO_CREATE (184 ページの『2』)
認証情報	MQZAO_CREATE (184 ページの『2』)

オブジェクト	必要な権限
チャネル	MQZAO_CREATE (184 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (184 ページの『2』)
リスナー	MQZAO_CREATE (184 ページの『2』)
サービス	MQZAO_CREATE (184 ページの『2』)

「オブジェクト」 のコピー (置換を伴う) (184 ページの『1』 184 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) (184 ページの『3』)

オブジェクト	必要な権限
キュー	MQZAO_CREATE (184 ページの『2』)
トピック	MQZAO_CREATE (184 ページの『2』)
プロセス	MQZAO_CREATE (184 ページの『2』)
キュー・マネージャー	適用外
名前リスト	MQZAO_CREATE (184 ページの『2』)
認証情報	MQZAO_CREATE (184 ページの『2』)
チャネル	MQZAO_CREATE (184 ページの『2』)
クライアント接続チャネル	MQZAO_CREATE (184 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) (184 ページの『3』 184 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

オブジェクト	必要な権限
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
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

オブジェクト	必要な権限
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
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

オブジェクト	必要な権限
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。

- 184 ページの『ワイルドカード文字の使用』
- 185 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。

186 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、 187 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-メッセージ記述子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、「116 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 Setting up security on z/OS

Security considerations specific to z/OS.

Security in IBM MQ for z/OS is controlled using RACF or an equivalent external security manager (ESM).

The following instructions assume that you are using RACF.

Related concepts
Security scenario: two queue managers on z/OS
Security scenario: queue sharing group on z/OS

z/OS RACF security classes

RACF classes are used to hold the profiles required for IBM MQ security checking. Many of the member classes have equivalent group classes. You must activate the classes and enable them to accept generic profiles.

Each RACF class holds one or more profiles used at some point in the checking sequence, as shown in Table 23 on page 190.

Table 23. RACF classes used by IBM MQ

Member class	Group class	Contents
MQADMIN	GMQADMIN	Profiles that are used mainly for administrative functions. For example: • Profiles for IBM MQ security switches. • The RESLEVEL security profile. • Profiles for alternate user security. • Profiles for context security. • Profiles for command resource security. This class can hold only uppercase RACF profiles.
MXADMIN	GMXADMIN	Profiles that are used mainly for administrative functions. For example: • Profiles for IBM MQ security switches. • The RESLEVEL security profile. • Profiles for alternate user security. • Profiles for context security. • Profiles for command resource security. This class can hold both uppercase and mixed-case RACF profiles.
MQCONN		Profiles used for connection security.
MQCMDS		Profiles used for command security.
MQQUEUE	GMQQUEUE	Uppercase profiles used in queue resource security.
MXQUEUE	GMXQUEUE	Mixed-case and uppercase profiles used in queue resource security.
MQPROC	GMQPROC	Uppercase profiles used in process resource security.
MXPROC	GMXPROC	Mixed-case and uppercase profiles used in process resource security.
MQNLIST	GMQNLIST	Uppercase profiles used in namelist resource security.
MXNLIST	GMXNLIST	Mixed-case and uppercase profiles used in namelist resource security.
MXTOPIC	GMXTOPIC	Mixed-case and uppercase profiles used in topic security.

Some classes have a related *group class* that enables you to put together groups of resources that have similar access requirements. For details about the difference between the member and group classes and when to use a member or group class, see the z/OS Security Server RACF Security Administrator's Guide.

The classes must be activated before security checks can be made. To activate all the IBM MQ classes, you can use this RACF command:

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                  MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

You should also ensure that you set up the classes so that they can accept generic profiles. You also do this with the RACF command **SETROPTS**, for example:

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                 MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

z/OS RACF profiles

All RACF profiles used by IBM MQ contain a prefix, which is either the queue manager name or the queue sharing group name. Be careful when you use the percent sign as a wildcard.

All RACF profiles used by IBM MQ contain a prefix. For queue sharing group level security, this is the queue sharing group name. For queue manager level security, the prefix is the queue manager name. If you are using a mixture of queue manager and queue sharing group level security, you will use profiles with both types of prefix. Queue sharing group and queue manager level security are described in Security controls and options in IBM MQ for z/OS.

For example, if you want to protect a queue called QUEUE_FOR_SUBSCRIBER_LIST in queue sharing group QSG1 at queue sharing group level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

If you want to protect a queue called QUEUE_FOR_LOST_CARD_LIST, that belongs to queue manager STCD at queue manager level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

This means that different queue managers and queue sharing groups can share the same RACF database and yet have different security options.

Do not use generic queue manager names in profiles to avoid unanticipated user access.

IBM MQ allows the use of the percent sign (%) in object names. However, RACF uses the % character as a single-character wildcard. This means that when you define an object name with a % character in its name, you must consider this when you define the corresponding profile.

For example, for the queue CREDIT_CARD_%_RATE_INQUIRY, on queue manager CRDP, the profile would be defined to RACF as follows:

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

This queue cannot be protected by a generic profile, such as, CRDP.**.

IBM MQ allows the use of mixed-case characters in object names. You can protect these objects by defining:

1. Mixed-case profiles in the appropriate mixed-case RACF classes, or
2. Generic profiles in the appropriate uppercase RACF classes.

To use mixed-case profiles and mixed-case RACF classes you must follow the steps described in “Migrating a z/OS queue manager to mixed-case security” on page 268.

There are some profiles, or parts of profiles, that remain uppercase only as the values are provided by IBM MQ. These are:

- Switch profiles.
- All high-level qualifiers (HLQ) including subsystem and queue sharing group identifiers.
- Profiles for SYSTEM objects.
- Profiles for Default objects.

- The **MQCMDS** class, so all command profiles are uppercase only.
- The **MQCONN** class, so all connection profiles are uppercase only.
- **RESLEVEL** profiles.
- The `'object'` qualification in command resource profiles; for example, `hlq.QUEUE.queuename`. The resource name only is mixed case.
- Dynamic queue profiles `hlq.CSQOREXX.*`, `hlq.CSQUTIL.*`, and `CSQXCMD.*`.
- The `'CONTEXT'` part of `hlq.CONTEXT.resourcename`.
- The `'ALTERNATE.USER'` part of `hlq.ALTERNATE.USER.userid`.

For example, you can define a profile to grant access to a queue called `PAYROLL.Dept1` on queue manager `QM01` in one of the following ways.

- If you are using mixed-case profiles, you can define a profile in the IBM MQ RACF class `MXQUEUE` using the following command:

```
RDEFINE MXQUEUE MQ01.PAYROLL.Dept1
```

- If you are using uppercase profiles, you can define a profile in the IBM MQ RACF class `MQQUEUE` using the following command:

```
RDEFINE MQQUEUE MQ01.PAYROLL.*
```

The first example, using mixed-case profiles, gives you more granular control over granting authority to access the resource.

z/OS Switch profiles

To control the security checking performed by IBM MQ, you use *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to IBM MQ. The access list in switch profiles is not used by IBM MQ.

IBM MQ maintains an internal switch for each switch type shown in tables Switch profiles for subsystem level security, Switch profiles for queue sharing group or queue manager level security,and Switch profiles for resource checking. Switch profiles can be maintained at queue sharing group level, or at queue manager level, or at a combination of both. Using a single set of queue sharing group security switch profiles, you can control security on all the queue managers within a queue sharing group.

When a security switch is set on, the security checks associated with the switch are performed. When a security switch is set off, the security checks associated with the switch are bypassed. The default is that all security switches are set on.

z/OS *Switches and classes*

When you start a queue manager or refresh security, IBM MQ sets switches according to the state of various RACF classes.

When a queue manager is started (or when the MQADMIN or MXADMIN class is refreshed by the IBM MQ REFRESH SECURITY command), IBM MQ first checks the status of RACF and the appropriate class:

- The MQADMIN class if you are using uppercase profiles
- The MXADMIN class if you are using mixed case profile.

It sets the subsystem security switch off if any of these conditions is true:

- RACF is inactive or not installed.
- The MQADMIN or MXADMIN class is not defined (these classes are always defined for RACF because they are included in the class descriptor table (CDT)).
- The MQADMIN or MXADMIN class has not been activated.

If both RACF and the MQADMIN or MXADMIN class are active, IBM MQ checks the MQADMIN or MXADMIN class to see whether any of the switch profiles have been defined. It first checks the profiles described in "Profiles to control subsystem security" on page 193. If subsystem security is not required, IBM MQ sets the internal subsystem security switch off, and performs no further checks.

The profiles determine whether the corresponding IBM MQ switch is set on or off.

- If the switch is off, that type of security is deactivated.
- If any IBM MQ switch is set on, IBM MQ checks the status of the RACF class associated with the type of security corresponding to the IBM MQ switch. If the class is not installed or not active, the IBM MQ switch is set off. For example, process security checks are not carried out if the MQPROC or MXPROC class has not been activated. The class not being active is equivalent to defining NO.PROCESS.CHECKS profile for every queue manager and queue sharing group that uses this RACF database.

z/OS *How switches work*

To set off a security switch, define a NO.* switch profile for it. You can override a NO.* profile set at the queue sharing group level by defining a YES.* profile for a queue manager.

To set off a security switch, you need to define a NO.* switch profile for it. The existence of a NO.* profile means that security checks are **not** performed for that type of resource, unless you choose to override a queue sharing group level setting on a particular queue manager. This is described in "Overriding queue sharing group level settings" on page 193.

If your queue manager is not a member of a queue sharing group, you do not need to define any queue sharing group level profiles or any override profiles. However, you must remember to define these profiles if the queue manager joins a queue sharing group at a later date.

Each NO.* switch profile that IBM MQ detects turns off the checking for that type of resource. Switch profiles are activated during startup of the queue manager. If you change the switch profiles while any affected queue managers are running, you can get IBM MQ to recognize the changes by issuing the IBM MQ REFRESH SECURITY command.

The switch profiles must always be defined in the MQADMIN or MXADMIN class. Do not define them in the GMQADMIN or GMXADMIN class. Tables Switch profiles for subsystem level security and Switch profiles for resource checking show the valid switch profiles and the security type they control.

Overriding queue sharing group level settings

You can override queue sharing group level security settings for a particular queue manager that is a member of that group. If you want to perform queue manager checks on an individual queue manager that are not performed on other queue managers in the group, use the (qmgr-name.YES.*) switch profiles.

Conversely, if you do not want to perform a certain check on one particular queue manager within a queue sharing group, define a (qmgr-name.NO.*) profile for that particular resource type on the queue manager, and do not define a profile for the queue sharing group. (IBM MQ only checks for a queue sharing group level profile if it does not find a queue manager level profile.)

z/OS *Profiles to control subsystem security*

IBM MQ checks whether subsystem security checks are required for the subsystem, for the queue manager, and for the queue sharing group.

The first security check made by IBM MQ is used to determine whether security checks are required for the whole IBM MQ subsystem. If you specify that you do not want subsystem security, no further checks are made.

The following switch profiles are checked to determine whether subsystem security is required. Figure 14 on page 194 shows the order in which they are checked.

Table 24. Switch profiles for subsystem level security	
Switch profile name	**Type of resource or checking that is controlled**
qmgr-name.NO.SUBSYS.SECURITY	Subsystem security for this queue manager
qsg-name.NO.SUBSYS.SECURITY	Subsystem security for this queue sharing group
qmgr-name.YES.SUBSYS.SECURITY	Subsystem security override for this queue manager

If your queue manager is not a member of a queue sharing group, IBM MQ checks for the qmgr-name.NO.SUBSYS.SECURITY switch profile only.

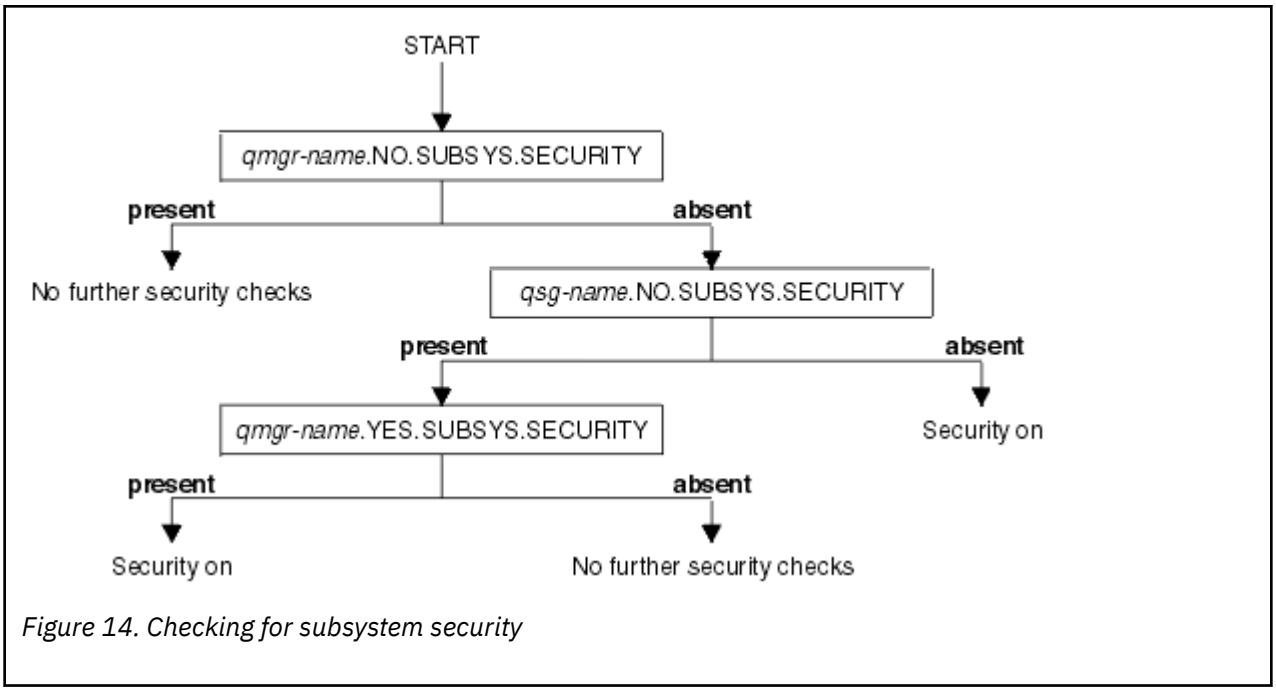


Figure 14. Checking for subsystem security

z/OS *Profiles to control queue sharing group or queue manager level security*

If subsystem security checking is required, IBM MQ checks whether security checking is required at queue sharing group or queue manager level.

When IBM MQ has determined that security checking is required, it then determines whether checking is required at queue sharing group or queue manager level, or both. These checks are not performed if your queue manager is not a member of a queue sharing group.

The following switch profiles are checked to determine the level required. Figure 15 on page 195 and Figure 16 on page 195 show the order in which they are checked.

Table 25. Switch profiles for queue sharing group or queue manager level security	
Switch profile name	**Type of resource or checking that is controlled**
qmgr-name.NO.QMGR.CHECKS	No queue manager level checks for this queue manager
qsg-name.NO.QMGR.CHECKS	No queue manager level checks for this queue sharing group
qmgr-name.YES.QMGR.CHECKS	Queue manager level checks override for this queue manager
qmgr-name.NO.QSG.CHECKS	No queue sharing group level checks for this queue manager
qsg-name.NO.QSG.CHECKS	No queue sharing group level checks for this queue sharing group
qmgr-name.YES.QSG.CHECKS	Queue sharing group level checks override for this queue manager

If subsystem security is active, you cannot switch off both queue sharing group and queue manager level security. If you try to do so, IBM MQ sets security checking on at both levels.

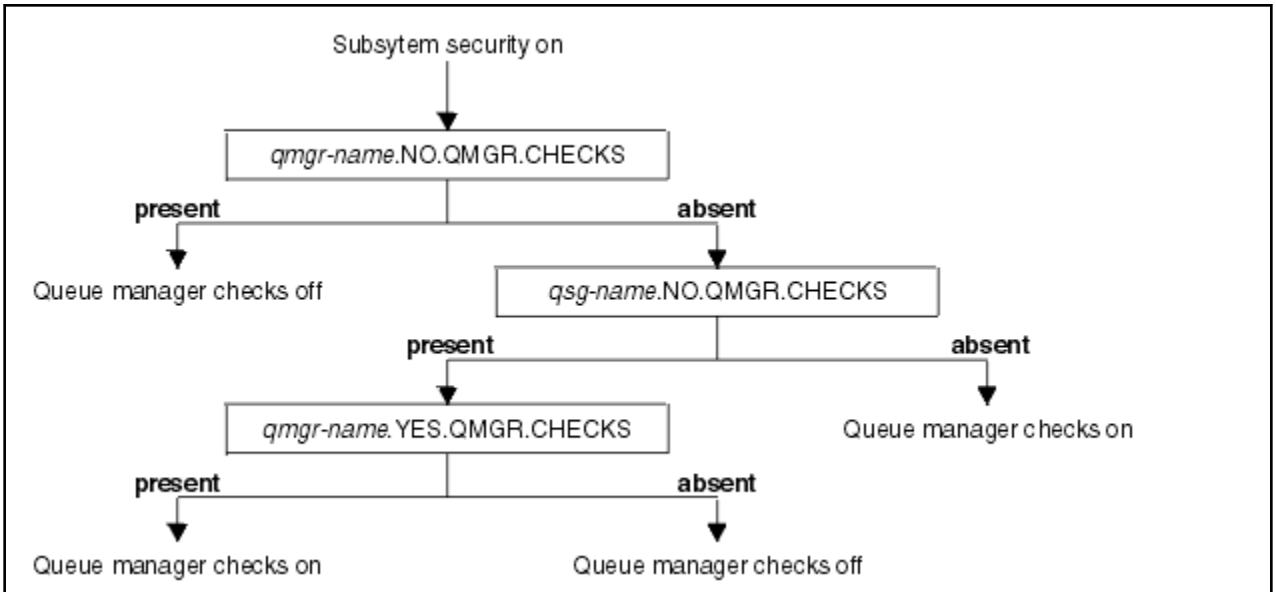


Figure 15. Checking for queue manager level security

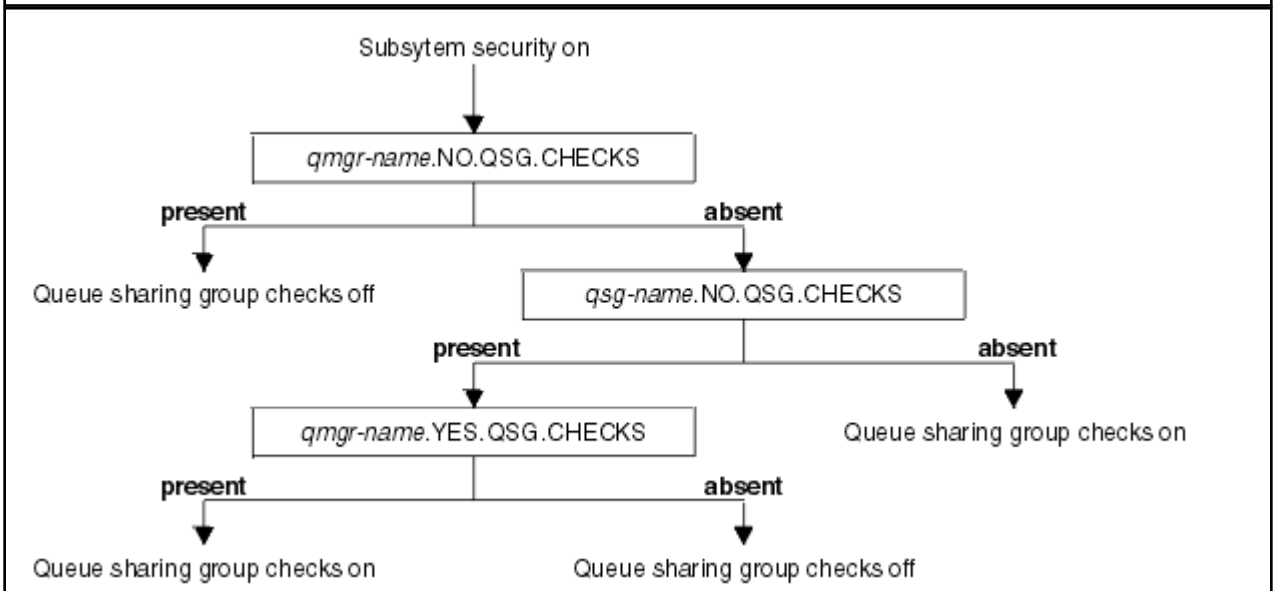


Figure 16. Checking for queue sharing group level security

z/OS *Valid combinations of security switches*

Only certain combinations of switches are valid. If you use a combination of switch settings that is not valid, message CSQH026I is issued and security checking is set on at both queue sharing group and queue manager level.

Table 26 on page 196, Table 27 on page 196, Table 28 on page 196, and Table 29 on page 197 show the sets of combinations of switch settings that are valid for each type of security level.

Table 26. Valid security switch combinations for queue manager level security
Combinations
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QSG.CHECKS 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

Table 27. Valid security switch combinations for queue sharing group level security
Combinations
qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS
qmgr-name.NO.QMGR.CHECKS 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

Table 28. Valid security switch combinations for queue manager and queue sharing group level security
Combinations
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS No QSG.* profiles defined
No QMGR.* profiles defined 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
No profiles for either switch defined

*Table 29. Other valid security switch combinations that switch both levels of checking **on**.*
Combinations
qmgr-name.NO.QMGR.CHECKS 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 *Resource level checks*

A number of switch profiles are used to control access to resources. Some stop checking being performed on either a queue manager or a queue sharing group. These can be overridden by profiles that enable checking for specific queue managers.

Table 30 on page 197 shows the switch profiles used to control access to IBM MQ resources.

If your queue manager is part of a queue sharing group and you have both queue manager and queue sharing group security active, you can use a YES.* switch profile to override queue sharing group level profiles and specifically turn on security for a particular queue manager.

Some profiles apply to both queue managers and queue sharing groups. These are prefixed by the string *hlq* and you should substitute the name of your queue sharing group or queue manager, as applicable. Profile names shown prefixed by *qmgr-name* are queue manager override profiles; you should substitute the name of your queue manager.

Table 30. Switch profiles for resource checking

Type of resource checking that is controlled	**Switch profile name**	**Override profile for a particular queue manager**
Connection security	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
Queue security	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
Process security	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
Namelist security	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
Context security	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
Alternate user security	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS
Command security	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
Command resource security	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
Topic security	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS
Note: Generic switch profiles such as hlq.NO.** are ignored by IBM MQ		

For example, if you want to perform process security checks on queue manager QM01, which is a member of queue sharing group QSG3 but you do not want to perform process security checks on any of the other queue managers in the group, define the following switch profiles:

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

If you want to have queue security checks performed on all the queue managers in the queue sharing group, except QM02, define the following switch profile:

```
QM02.NO.QUEUE.CHECKS
```

(There is no need to define a profile for the queue sharing group because the checks are automatically enabled if there is no profile defined.)

z/OS *An example of defining switches*

Different IBM MQ subsystems have different security requirements, which can be implemented using different switch profiles.

Four IBM MQ subsystems have been defined:

- MQP1 (a production system)
- MQP2 (a production system)
- MQD1 (a development system)
- MQT1 (a test system)

All four queue managers are members of queue sharing group QS01. All IBM MQ RACF classes have been defined and activated.

These subsystems have different security requirements:

- The production systems require full IBM MQ security checking to be active at queue sharing group level on both systems.

  This is done by specifying the following profile:

  ```
  RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
  ```

  This sets queue sharing group level checking for all the queue managers in the queue sharing group. You do not need to define any other switch profiles for the production queue managers because you want to check everything for these systems.
- Test queue manager MQT1 also requires full security checking. However, because you might want to change this later, security can be defined at queue manager level so that you can change the security settings for this queue manager without affecting the other members of the queue sharing group.

  This is done by defining the NO.QSG.CHECKS profile for MQT1 as follows:

  ```
  RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
  ```
- Development queue manager MQD1 has different security requirements from the rest of the queue sharing group. It requires only connection and queue security to be active.

  This is done by defining a `MQD1.YES.QMGR.CHECKS` profile for this queue manager, and then defining the following profiles to switch off security checking for the resources that do not need to be checked:

  ```
  RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
  RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
  RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
  RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
  ```

```
RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
```

When the queue manager is active, you can display the current security settings by issuing the DISPLAY SECURITY MQSC command.

You can also change the switch settings when the queue manager is running by defining or deleting the appropriate switch profile in the MQADMIN class. To make the changes to the switch settings active, you must issue the REFRESH SECURITY command for the MQADMIN class.

See “Refreshing queue manager security on z/OS” on page 250 for more details about using the DISPLAY SECURITY and REFRESH SECURITY commands.

z/OS Profiles used to control access to IBM MQ resources

You must define RACF profiles to control access to IBM MQ resources, in addition to the switch profiles that might have been defined. This collection of topics contains information about the RACF profiles for the different types of IBM MQ resource.

If you do not have a resource profile defined for a particular security check, and a user issues a request that would involve making that check, IBM MQ denies access. You do not have to define profiles for security types relating to any security switches that you have deactivated.

z/OS *Profiles for connection security*

If connection security is active, you must define profiles in the MQCONN class and permit the necessary groups or user IDs access to those profiles, so that they can connect to IBM MQ.

To enable a connection to be made, you must grant users RACF READ access to the appropriate profile. (If no queue manager level profile exists, and your queue manager is a member of a queue sharing group, checks might be made against queue sharing group level profiles, if the security is set up to do this.)

A connection profile qualified with a queue manager name controls access to a specific queue manager and users given access to this profile can connect to that queue manager. A connection profile qualified with queue sharing group name controls access to all queue managers within the queue sharing group for that connection type. For example, a user with access to `QS01.BATCH` can use a batch connection to any queue manager in queue sharing group QS01 that has not got a queue manager level profile defined.

Note:

1. For information about the user IDs checked for different security requests, see “User IDs for security checking on z/OS” on page 239.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see “RESLEVEL セキュリティー・プロファイル” on page 233.

IBM MQ security recognizes the following different types of connection:

- Batch (and batch-type) connections, these include:
  - z/OS batch jobs
  - TSO applications
  - z/OS UNIX System Services sign-ons
  - Db2 stored procedures
- CICS connections
- IMS connections from control and application processing regions
- The IBM MQ channel initiator

z/OS *Connection security profiles for batch connections*

Profiles for checking batch-type connections are composed of the queue manager or queue sharing group name followed by the word *BATCH*. Give the user ID associated with the connecting address space READ access to the connection profile.

Profiles for checking batch and batch-type connections take the form:

```
hlq.BATCH
```

where `hlq` can be either the `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails.

For batch or batch-type connection requests, you must permit the user ID associated with the connecting address space to access the connection profile. For example, the following RACF command allows users in the CONNTQM1 group to connect to the queue manager TQM1; these user IDs will be permitted to use any batch or batch-type connection.

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

z/OS *Using **CHCKLOCL** on locally bound applications*

CHCKLOCL only applies to connections that are made through BATCH connections and does not apply to connections made from CICS or IMS. Connections made through the channel initiator are controlled by **CHCKCLNT**.

Overview

If you want to configure your z/OS queue manager to mandate user ID and password checking for some, but not all, of your locally bound applications, you need to do some additional configuration.

The reason for this is that once **CHCKLOCL** (*REQUIRED*) is configured, legacy batch applications that use the MQCONN API call can no longer connect to the queue manager.

For z/OS only, a more granular mechanism based on the connection security of an address space can be used to downgrade the global CHCKLOCL(REQUIRED) configuration to CHCKLOCL(OPTIONAL) for specifically defined user IDs. The mechanism used, is described in the following text, together with an example.

In order to allow more granularity on **CHCKLOCL** (*REQUIRED*) than just EVERYONE, you modify **CHCKLOCL** in the same manner as you modify the access level of the user ID associated with the connecting address space to the `hlq.batch` connection profiles in the MQCONN class.

If the address space user ID only has READ access, which is the minimum you require to be able to connect at all, the **CHCKLOCL** configuration applies as written.

If the address space user ID has UPDATE access (or above) then the **CHCKLOCL** configuration operates in *OPTIONAL* mode. That is, you do not have to provide a user ID and password, but if you do, the user ID and password must be a valid pair.

Connection security already configured for your z/OS queue manager

If you have connection security configured for your z/OS queue manager and you want **CHCKLOCL** (*REQUIRED*) to apply to WAS locally bound applications, and no others, carry out the following steps:

1. Start with **CHCKLOCL** (*OPTIONAL*) as your configuration. This means that any user ID and passwords that are supplied are checked for validity, but not mandated.
2. List all the users that have access to the connection security profiles by issuing the command:

   ```
   RLIST MQCONN MQ23.BATCH AUTHUSER
   ```

   This command displays, for example:

```
CLASS    NAME
-----    ----
MQCONN   MQ23.BATCH

USER     ACCESS  ACCESS COUNT
----     ------  ------ -----
JOHNDOE  READ    000009
JDOE1    READ    000003
WASUSER  READ    000000
```

3. For each user ID listed as having READ access, change the access to

```
UPDATE:- PERMIT MQ23.BATCH CLASS(MQCONN) ID(JOHNDOE) ACCESS(UPDATE)
```

4. Update the IBM MQ configuration to **CHCKLOCL** (*REQUIRED*).

   The combination of UPDATE access to MQ23.BATCH and the current setting means that you are using **CHCKLOCL** (*OPTIONAL*).

5. Now, apply the **CHCKLOCL** (*REQUIRED*) behavior to one specific user ID, for example WASUSER, so that all the connections coming from that region must provide a user ID and password.

   Do this by reversing the change you made previously, by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

Connection security is not configured for your z/OS queue manager

In this situation, you must:

1. Create connection profiles for hlq.BATCH in the MQCONN class, by issuing the command:

```
RDEFINE MQCONN MQ23.BATCH UACC(NONE)
```

2. Authorize all user IDs that create batch connections to the queue manager, so that they have UPDATE access to this profile. Doing this bypasses the **CHCKLOCL** (*REQUIRED*) requirement for the user ID and password at the time of connection.

   Do this by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN)ID(JOHNDOE) ACCESS(UPDATE)
```

   These include user IDs:

   a. Used for CSQUTIL, ISPF panels, and other locally bound tools.

   b. Associated with batch like connections to the queue manager. Consider for example, Advanced Message Security, IBM Integration Bus, Db2 stored procedures, z/OS UNIX System Services and TSO users, and Java applications

3. Delete the switch profile for the queue manager by issuing the command:

```
hlq.NO.CONNECT.CHECKS
```

4. Now, apply the **CHCKLOCL** (*REQUIRED*) behavior to one specific user ID, for example WASUSER, so that all the connections coming from that region must provide a user ID and password.

   Do this by reversing the change you made previously, by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

z/OS *Connection security profiles for CICS connections*

Profiles for checking CICS connections are composed of the queue manager or queue sharing group name followed by the word *CICS* . Give the user ID associated with the CICS address space READ access to the connection profile.

Profiles for checking connections from CICS take the form:

```
hlq.CICS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by CICS, you need only permit the CICS address space user ID access to the connection profile.

For example, the following RACF commands allow the CICS address space user ID KCBCICS to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

z/OS *Connection security profiles for IMS connections*

Profiles for checking IMS connections are composed of the queue manager or queue sharing group name followed by the word *IMS* . Give the IMS control and dependent region user IDs READ access to the connection profile.

Profiles for checking connections from IMS take the form:

```
hlq.IMS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by IMS, permit access to the connection profile for the IMS control and dependent region user IDs.

For example, the following RACF commands allow:

- The IMS region user ID, IMSREG, to connect to the queue manager TQM1.
- Users in group BMPGRP to submit BMP jobs.

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

z/OS *Connection security profiles for the channel initiator*

Profiles for checking connections from the channel initiator are composed of the queue manager or queue sharing group name followed by the word *CHIN*. Give the user ID used by the channel initiator started task address space READ access to the connection profile.

Profiles for checking connections from the channel initiator take the form:

```
hlq.CHIN
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by the channel initiator, define access to the connection profile for the user ID used by the channel initiator started task address space.

For example, the following RACF commands allow the channel initiator address space running with user ID DQCTRL to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

z/OS ***Profiles for queue security***

If queue security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to these profiles. Queue security profiles are named after the queue manager or queue sharing group, and the queue to be opened.

If queue security is active, you must:

- Define profiles in the **MQQUEUE** or **GMQQUEUE** classes if using uppercase profiles.
- Define profiles in the **MXQUEUE** or **GMXQUEUE** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use queues.

Profiles for queue security take the form:

```
hlq.queuename
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `queuename` is the name of the queue being opened, as specified in the object descriptor on the `MQOPEN` or `MQPUT1` call.

A profile prefixed by the queue manager name controls access to a single queue on that queue manager. A profile prefixed by the queue sharing group name controls access to access to one or more queues with that queue name on all queue managers within the queue sharing group, or access to a shared queue by any queue manager within the group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that queue on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

If you are using shared queues, you are recommended to use queue sharing group level security.

For details of how queue security operates when the queue name is that of an alias or a model queue, see “Considerations for alias queues” on page 205 and “Considerations for model queues” on page 206 .

The RACF access required to open a queue depends on the MQOPEN or MQPUT1 options specified. If more than one of the MQOO_* and MQPMO_* options is coded, the queue security check is performed for the highest RACF authority required.

Table 31. Access levels for queue security using the MQOPEN or MQPUT1 calls

MQOPEN or MQPUT1 option	**RACF access level required to hlq.queuename**
MQOO_BROWSE	READ
MQOO_INQUIRE	READ
MQOO_BIND_*	UPDATE
MQOO_INPUT_*	UPDATE
MQOO_OUTPUT or MQPUT1	UPDATE
MQOO_PASS_ALL_CONTEXT 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

For example, on IBM MQ queue manager QM77, all user IDs in the RACF group PAYGRP are to be given access to get messages from or put messages to all queues with names beginning with 'PAY.'. You can do this using these RACF commands:

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Also, all user IDs in the PAYGRP group must have access to put messages on queues that do not follow the PAY naming convention. For example:

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

You can do this by defining profiles for these queues in the GMQQUEUE class and giving access to that class as follows:

```
RDEFINE GMQQUEUE PAYROLL.EXTRAS UACC(NONE)
        ADDMEM(QM77.REQUEST_QUEUE_FOR_PAYROLL,
               QM77.SALARY.INCREASE.SERVER,
               QM77.REPLIES.FROM.SALARY.MODEL)
PERMIT PAYROLL.EXTRAS CLASS(GMQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Note:

1. If the RACF access level that an application has to a queue security profile is changed, the changes only take effect for any new object handles obtained (that is, new MQOPEN s) for that queue. Those handles already in existence at the time of the change retain their existing access to the queue. If an application is required to use its changed access level to the queue rather than its existing access level, it must close and reopen the queue for each object handle that requires the change.
2. In the example, the queue manager name QM77 could also be the name of a queue sharing group.

Other types of security checks might also occur at the time the queue is opened depending on the open options specified and the types of security that are active. See also "Profiles for context security" on page 219 and "Profiles for alternate user security" on page 217. For a summary table showing the open options and the security authorization needed when queue, context, and alternate user security are all active, see Table 36 on page 211.

If you are using publish/subscribe you must consider the following. When an MQSUB request is processed a security check is performed to ensure that the user ID making the request has the required access to put messages to the target IBM MQ queue as well as the required access to subscribe to the IBM MQ topic.

Table 32. Access levels for queue security using the MQSUB call

MQSUB option	**RACF access level required to hlq.queuename**
MQSO_ALTER, MQSO_CREATE, and MQSO_RESUME	UPDATE

Note:

1. The `hlq.queuename` is the destination queue for publications. When this is a managed queue, you need access to the appropriate model queue to be used for the managed queue and the dynamic queue that are created.
2. You can use a technique like this for the destination queue you provide on an MQSUB API call if you want to distinguish between the users making the subscriptions, and the users retrieving the publications from the destination queue.

z/OS *Considerations for alias queues*

When you issue an MQOPEN or MQPUT1 call for an alias queue, IBM MQ makes a resource check against the queue name specified in the object descriptor (MQOD) on the call. It does not check if the user is allowed access to the target queue name.

For example, an alias queue called PAYROLL.REQUEST resolves to a target queue of PAY.REQUEST. If queue security is active, you need only be authorized to access the queue PAYROLL.REQUEST. No check is made to see if you are authorized to access the queue PAY.REQUEST.

z/OS *Using alias queues to distinguish between MQGET and MQPUT requests*

The range of MQI calls available in one access level can cause a problem if you want to restrict access to a queue to allow only the **MQPUT** call or only the **MQGET** call. A queue can be protected by defining two aliases that resolve to that queue: one that enables applications to get messages from the queue, and one that enable applications to put messages on the queue.

The following text gives you an example of how you can define your queues to IBM MQ:

```
DEFINE QLOCAL(MUST_USE_ALIAS_TO_ACCESS) GET(ENABLED)
       PUT(ENABLED)

DEFINE QALIAS(USE_THIS_ONE_FOR_GETS) GET(ENABLED)
       PUT(DISABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)

DEFINE QALIAS(USE_THIS_ONE_FOR_PUTS) GET(DISABLED)
       PUT(ENABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)
```

You must also make the following RACF definitions:

```
RDEFINE MQQUEUE hlq.MUST_USE_ALIAS_TO_ACCESS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_GETS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_PUTS UACC(NONE)
```

Then you ensure that no users have access to the queue hlq.MUST_USE_ALIAS_TO_ACCESS, and give the appropriate users or groups access to the alias. You can do this using the following RACF commands:

```
PERMIT hlq.USE_THIS_ONE_FOR_GETS CLASS(MQQUEUE)
       ID(GETUSER,GETGRP) ACCESS(UPDATE)
PERMIT hlq.USE_THIS_ONE_FOR_PUTS CLASS(MQQUEUE)
       ID(PUTUSER,PUTGRP) ACCESS(UPDATE)
```

This means user ID GETUSER and user IDs in the group GETGRP are only allowed to get messages on MUST_USE_ALIAS_TO_ACCESS through the alias queue USE_THIS_ONE_FOR_GETS; and user ID PUTUSER and user IDs in the group PUTGRP are only allowed to put messages through the alias queue USE_THIS_ONE_FOR_PUTS.

Note:

1. If you want to use a technique like this, you must inform your application developers, so that they can design their programs appropriately.
2. You can use a technique like this for the destination queue you provide on an MQSUB API request if you want to distinguish between the users making the subscriptions and the users 'getting' the publications from the destination queue.

z/OS *Considerations for model queues*

To open a model queue, you must be able to open both the model queue itself and the dynamic queue to which it resolves. Define generic RACF profiles for dynamic queues, including dynamic queues used by IBM MQ utilities.

When you open a model queue, IBM MQ security makes two queue security checks:

1. Are you authorized to access the model queue?
2. Are you authorized to access the dynamic queue to which the model queue resolves?

If the dynamic queue name contains a trailing asterisk (*) character, this * is replaced by a character string generated by IBM MQ, to create a dynamic queue with a unique name. However, because the whole name, including this generated string, is used for checking authority, you should define generic profiles for these queues.

For example, an MQOPEN call uses a model queue name of CREDIT.CHECK.REPLY.MODEL and a dynamic queue name of CREDIT.REPLY.* on queue manager (or queue sharing group) MQSP.

To do this, you must issue the following RACF commands to define the necessary queue profiles:

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

You must also issue the corresponding RACF PERMIT commands to allow the user access to these profiles.

A typical dynamic queue name created by an MQOPEN is something like CREDIT.REPLY.A346EF00367849A0. The precise value of the last qualifier is unpredictable; this is why you should use generic profiles for such queue names.

A number of IBM MQ utilities put messages on dynamic queues. You should define profiles for the following dynamic queue names, and provide RACF UPDATE access to the relevant user IDs (see "User IDs for security checking on z/OS" on page 239 for the correct user IDs):

```
SYSTEM.CSQUTIL.*  (used by CSQUTIL)
SYSTEM.CSQOREXX.* (used by the operations and control panels)
SYSTEM.CSQXCMD.*  (used by the channel initiator when processing CSQINPX)
CSQ4SAMP.*        (used by the IBM MQ supplied samples)
```

You might also consider defining a profile to control use of the dynamic queue name used by default in the application programming copy members. The IBM MQ-supplied copybooks contain a default *DynamicQName*, which is CSQ.*. This enables an appropriate RACF profile to be established.

Note: Do not allow application programmers to specify a single * for the dynamic queue name. If you do, you must define an hlq.** profile in the MQQUEUE class, and you would have to give it wide-ranging access. This means that this profile could also be used for other non-dynamic queues that do not have a more specific RACF profile. Your users could, therefore, gain access to queues you do not want them to access.

z/OS *Close options on permanent dynamic queues*

If an application opens a permanent dynamic queue that was created by another application and then attempts to delete that queue with an MQCLOSE option, some extra security checks are applied when the attempt is made.

Table 33. Access levels for close options on permanent dynamic queues

MQCLOSE option	RACF access level required to hlq.queuename
MQCO_DELETE	ALTER
MQCO_DELETE_PURGE	ALTER

z/OS *Security and remote queues*

When a message is put on a remote queue, the queue security that is implemented by the local queue manager depends on how the remote queue is specified when it is opened.

The following rules are applied:

1. If the remote queue has been defined on the local queue manager through the IBM MQ DEFINE QREMOTE command, the queue that is checked is the name of the remote queue. For example, if a remote queue is defined on queue manager MQS1 as follows:

   ```
   DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
          RNAME(CREDIT.SCORING.REQUEST)
          RQMNAME(BNK7)
          XMITQ(BANK1.TO.BANK7)
   ```

   In this case, a profile for BANK7.CREDIT.REFERENCE must be defined in the MQQUEUE class.
2. If the *ObjectQMgrName* for the request does not resolve to the local queue manager, a security check is carried out against the resolved (remote) queue manager name except in the case of a cluster queue where the check is made against the cluster queue name.

For example, the transmission queue BANK1.TO.BANK7 is defined on queue manager MQS1. An MQPUT1 request is then issued on MQS1 specifying *ObjectName* as BANK1.INTERBANK.TRANSFERS and an *ObjectQMgrName* of BANK1.TO.BANK7. In this case, the user performing the request must have access to BANK1.TO.BANK7.

3. If you make an MQPUT request to a queue and specify *ObjectQMgrName* as the name of an alias of the local queue manager, only the queue name is checked for security, not that of the queue manager.

When the message gets to the remote queue manager it might be subject to additional security processing. For more information, see “リモート・メッセージングのセキュリティー” on page 102.

z/OS *Dead-letter queue security*

Special considerations apply to the dead-letter queue, because many users must be able to put messages on it, but access to retrieve messages must be tightly restricted. You can achieve this by applying different RACF authorities to the dead-letter queue and an alias queue.

Undelivered messages can be put on a special queue called the dead-letter queue. If you have sensitive data that could possibly end up on this queue, you must consider the security implications of this because you do not want unauthorized users to retrieve this data.

Each of the following must be allowed to put messages onto the dead-letter queue:

- Application programs.
- The channel initiator address space and any MCA user IDs. (If the RESLEVEL profile is not present, or is defined so that channel user IDs are checked, the channel user ID also needs authority to put messages on the dead-letter queue.)
- CKTI, the CICS-supplied CICS task initiator.
- CSQQTRMN, the IBM MQ-supplied IMS trigger monitor.

The only application that can retrieve messages from the dead-letter queue should be a 'special' application that processes these messages. However, a problem arises if you give applications RACF UPDATE authority to the dead-letter queue for MQPUT s because they can then automatically retrieve messages from the queue using MQGET calls. You cannot disable the dead-letter queue for get operations because, if you do, not even the 'special' applications could retrieve the messages.

One solution to this problem is set up a two-level access to the dead-letter queue. CKTI, message channel agent transactions or the channel initiator address space, and 'special' applications have direct access; other applications can only access the dead-letter queue through an alias queue. This alias is defined to allow applications to put messages on the dead-letter queue, but not to get messages from it.

This is how it might work:

1. Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED), as shown in the sample thlqual.SCSQPROC(CSQ4INYG).
2. Give RACF UPDATE authority for the dead-letter queue to the following user IDs:
   - User IDs that the CKTI and the MCAs or channel initiator address space run under.
   - The user IDs associated with the 'special' dead-letter queue processing application.
3. Define an alias queue that resolves to the real dead-letter queue, but give the alias queue these attributes: PUT(ENABLED) and GET(DISABLED). Give the alias queue a name with the same stem as the dead-letter queue name but append the characters ".PUT" to this stem. For example, if the dead-letter queue name is hlq.DEAD.QUEUE, the alias queue name would be hlq.DEAD.QUEUE.PUT.
4. To put a message on the dead-letter queue, an application uses the alias queue. This is what your application must do:
   - Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using MQOPEN and then issues an MQINQ to get the dead-letter queue name.
   - Build the name of the alias queue by appending the characters '.PUT' to this name, in this case, hlq.DEAD.QUEUE.PUT.
   - Open the alias queue, hlq.DEAD.QUEUE.PUT.

- Put the message on the real dead-letter queue by issuing an MQPUT against the alias queue.

5. Give the user ID associated with the application RACF UPDATE authority to the alias, but no access (authority NONE) to the real dead-letter queue. This means that:
   - The application can put messages onto the dead-letter queue using the alias queue.
   - The application cannot get messages from the dead-letter queue using the alias queue because the alias queue is disabled for get operations.

   The application cannot get any messages from the real dead-letter queue either because it does have the correct RACF authority.

Table 34 on page 209 summarizes the RACF authority required for the various participants in this solution.

Table 34. RACF authority to the dead-letter queue and its alias

Associated user IDs	**Real dead-letter queue (hlq.DEAD.QUEUE)**	**Alias dead-letter queue (hlq.DEAD.QUEUE.PUT)**
MCA or channel initiator address space and CKTI	UPDATE	NONE
'Special' application (for dead-letter queue processing)	UPDATE	NONE
User-written application user IDs	NONE	UPDATE

If you use this method, the application cannot determine the maximum message length (MAXMSGL) of the dead-letter queue. This is because the MAXMSGL attribute cannot be retrieved from an alias queue. Therefore, your application should assume that the maximum message length is 100 MB, the maximum size IBM MQ for z/OS supports. The real dead-letter queue should also be defined with a MAXMSGL attribute of 100 MB.

Note: User-written application programs do not normally use alternate user authority to put messages on the dead-letter queue. This reduces the number of user IDs that have access to the dead-letter queue.

z/OS *System queue security*

You must set up RACF access to allow certain user IDs access to particular system queues.

Many of the system queues are accessed by the ancillary parts of IBM MQ:

- The CSQUTIL utility
- The message security policy utility (CSQ0UTIL)
- The operations and control panels
- The channel initiator address space (including the Queued Pub/Sub Daemon)
- The mqweb server, used by the IBM MQ Console and REST API.

The user IDs under which these run must be given RACF access to these queues, as shown in Table 35 on page 209.

Table 35. Access required to the SYSTEM queues by IBM MQ

SYSTEM queue	**CSQUTIL**	**CSQ0UTIL**	**mqweb server**	**Operations and control panels**	**Channel initiator for distributed queuing**
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	UPDATE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	UPDATE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	ALTER

Table 35. Access required to the SYSTEM queues by IBM MQ (continued)

SYSTEM queue	CSQUTIL	CSQ0UTIL	mqweb server	Operations and control panels	Channel initiator for distributed queuing
SYSTEM.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 “1” on page 210	-	-	READ
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	-	-	UPDATE
SYSTEM.REST.REPLY.QUEUE	-	-	UPDATE	-	-
SYSTEM.BLUEMIX.REGISTRATION.QUEUE	-	-	-	-	UPDATE

Notes:

1. The Advanced Message Security address space user also requires READ access to this queue.

z/OS *API-resource security access quick reference*

A summary of the **MQOPEN**, **MQPUT1**, **MQSUB**, and **MQCLOSE** options and the access required by the different resource security types.

Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required. Callouts shown like this **(1)** refer to the notes following this table.

	Minimum RACF access level required			
RACF class:	**MXTOPIC**	**MQQUEUE or MXQUEUE (1)**	**MQADMIN or MXADMIN**	**MQADMIN or MXADMIN**
RACF profile:	**(15 or 16)**	**(2)**	**(3)**	**(4)**
MQOPEN option				
MQOO_INQUIRE		READ **(5)**	No check	No check
MQOO_BROWSE		READ	No check	No check
MQOO_INPUT_*		UPDATE	No check	No check
MQOO_SAVE_ALL_CONTEXT **(6)**		UPDATE	No check	No check
MQOO_OUTPUT (USAGE=NORMAL) **(7)**		UPDATE	No check	No check
MQOO_PASS_IDENTITY_CONTEXT **(8)**		UPDATE	READ	No check
MQOO_PASS_ALL_CONTEXT **(8) (9)**		UPDATE	READ	No check
MQOO_SET_IDENTITY_CONTEXT **(8) (9)**		UPDATE	UPDATE	No check
MQOO_SET_ALL_CONTEXT **(8) (10)**		UPDATE	CONTROL	No check
MQOO_OUTPUT (USAGE (XMITQ) **(11)**		UPDATE	CONTROL	No check
MQOO_OUTPUT (topic object)	UPDATE **(16)**			
MQOO_OUTPUT (alias queue to topic object)	UPDATE **(16)**	UPDATE		
MQOO_SET		ALTER	No check	No check
MQOO_ALTERNATE_USER_AUTHORITY		**(12)**	**(12)**	UPDATE
MQPUT1 option				
Put on a normal queue **(7)**		UPDATE	No check	No check
MQPMO_PASS_IDENTITY_CONTEXT		UPDATE	READ	No check
MQPMO_PASS_ALL_CONTEXT		UPDATE	READ	No check
MQPMO_SET_IDENTITY_CONTEXT		UPDATE	UPDATE	No check
MQPMO_SET_ALL_CONTEXT		UPDATE	CONTROL	No check
MQOO_OUTPUT Put on a transmission queue **(11)**		UPDATE	CONTROL	No check
MQOO_OUTPUT (topic object)	UPDATE **(16)**			
MQOO_OUTPUT (alias queue to topic object)	UPDATE **(16)**	UPDATE		
MQPMO_ALTERNATE_USER_AUTHORITY		**(13)**	**(13)**	UPDATE
MQCLOSE option				
MQCO_DELETE **(14)**		ALTER	No check	No check
MQCO_DELETE_PURGE **(14)**		ALTER	No check	No check
MQCO_REMOVE_SUB	ALTER **(15)**			

Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required. Callouts shown like this **(1)** refer to the notes following this table. *(continued)*

	Minimum RACF access level required			
RACF class:	**MXTOPIC**	**MQQUEUE or MXQUEUE (1)**	**MQADMIN or MXADMIN**	**MQADMIN or MXADMIN**
RACF profile:	**(15 or 16)**	**(2)**	**(3)**	**(4)**
MQSUB option				
MQSO_CREATE	ALTER **(15)**	**(17)**	**(18)**	
MQSO_ALTER	ALTER **(15)**	**(17)**	**(18)**	
MQSO_RESUME	READ **(15)**	**(17)**	No check	
MQSO_ALTERNATE_USER_AUTHORITY				UPDATE
MQSO_SET_IDENTITY_CONTEXT			**(18)**	

Note:

1. This option is not restricted to queues. Use the MQNLIST or MXNLIST class for namelists, and the MQPROC or MXPROC class for processes.
2. Use RACF profile: hlq.resourcename
3. Use RACF profile: hlq.CONTEXT.queuename
4. Use RACF profile: hlq.ALTERNATE.USER. `alternateuserid`

   `alternateuserid` is the user identifier that is specified in the *`AlternateUserId`* field of the object descriptor. Note that up to 12 characters of the *`AlternateUserId`* field are used for this check, unlike other checks where only the first 8 characters of a user identifier are used.
5. No check is made when opening the queue manager for inquiries.
6. MQOO_INPUT_* must be specified as well. This is valid for a local, model or alias queue.
7. This check is done for a local or model queue that has a **Usage** queue attribute of MQUS_NORMAL, and also for an alias or remote queue (that is defined to the connected queue manager.) If the queue is a remote queue that is opened specifying an *`ObjectQMgrName`* (not the name of the connected queue manager) explicitly, the check is carried out against the queue with the same name as *`ObjectQMgrName`* (which must be a local queue with a **Usage** queue attribute of MQUS_TRANSMISSION).
8. MQOO_OUTPUT must be specified as well.
9. MQOO_PASS_IDENTITY_CONTEXT is implied as well by this option.
10. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT and MQOO_SET_IDENTITY_CONTEXT are implied as well by this option.
11. This check is done for a local or model queue that has a **Usage** queue attribute of MQUS_TRANSMISSION, and is being opened directly for output. It does not apply if a remote queue is being opened.
12. At least one of MQOO_INQUIRE, MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT or MQOO_SET must be specified as well. The check carried out is the same as that for the other options specified.
13. The check carried out is the same as that for the other options specified.
14. This applies only for permanent dynamic queues that have been opened directly, that is, not opened through a model queue. No security is required to delete a temporary dynamic queue.
15. Use RACF profile hlq.SUBSCRIBE.topicname.
16. Use RACF profile hlq.PUBLISH.topicname.

17. If on the MQSUB request you specified a destination queue for the publications to be sent to, then a security check is carried out against that queue to ensure that you have put authority to that queue.
18. If on the MQSUB request, with MQSO_CREATE or MQSO_ALTER options specified, you want to set any of the identity context fields in the MQSD structure, you also need to specify the MQSO_SET_IDENTITY_CONTEXT option and you also need the appropriate authority to the context profile for the destination queue.

z/OS *Profiles for topic security*

If topic security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

The concept of topic security within a topic tree is described in Publish/subscribe security.

If topic security is active, you must perform the following actions:

- Define profiles in the **MXTOPIC** or **GMXTOPIC** classes.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use topics.

Profiles for topic security take the form:

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

where

- `hlq` is either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).
- `topicname` is the name of the topic administration node in the topic tree, associated either with the topic being subscribed to through an MQSUB call, or being published to through an MQOPEN call.

A profile prefixed by the queue manager name controls access to a single topic on that queue manager. A profile prefixed by the queue sharing group name controls access to one or more topics with that topic name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that topic on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

Subscribe

To subscribe to a topic, you need access to both the topic you are trying to subscribe to, and the destination queue for the publications.

When you issue an MQSUB request, the following security checks take place:

- Whether you have the appropriate level of access to subscribe to that topic, and also that the destination queue (if specified) is opened for output
- Whether you have the appropriate level of access to that destination queue.

Table 37. Access level required for topic security to subscribe

MQSUB option	**RACF access required to `hlq.SUBSCRIBE.topicname` profile in MXTOPIC class**
MQSO_CREATE and MQSO_ALTER	ALTER
MQSO_RESUME	READ

Table 38. Additional authority required to subscribe using a non-managed destination queue	
MQSUB option	**RACF access required to `hlq.CONTEXT.queuename` profile in MQADMIN or MXADMIN class**
MQSO_CREATE, MQSO_ALTER, and MQSO_RESUME	UPDATE
	RACF access required to `hlq.queuename` profile in MQQUEUE or MXQUEUE class
MQSO_CREATE and MQSO_ALTER	UPDATE
	RACF access required to `hlq.ALTERNATE.USER.alternateuserid` profile in MQADMIN or MXADMIN class
MQSO_ALTERNATE_USER_AUTHORITY	UPDATE

Considerations for managed queues for subscriptions

A security check is carried out to see if you are allowed to subscribe to the topic. However, no security checks are carried out when the managed queue is created, or to determine if you have access to put messages to this destination queue.

You cannot close delete a managed queue.

The model queues used are: `SYSTEM.DURABLE.MODEL.QUEUE` and `SYSTEM.NDURABLE.MODEL.QUEUE`.

The managed queues created from these model queues are of the form `SYSTEM.MANAGED.DURABLE.A346EF00367849A0` and `SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0` where the last qualifier is unpredictable.

Do not give any user access to these queues. The queues can be protected using generic profiles of the form `SYSTEM.MANAGED.DURABLE.*` and `SYSTEM.MANAGED.NDURABLE.*` with no authorities granted.

Messages can be retrieved from these queues using the handle returned on the MQSUB request.

If you explicitly issue an MQCLOSE call for a subscription with the `MQCO_REMOVE_SUB` option specified, and you did not create the subscription you are closing under this handle, a security check is performed at the time of closure to ensure that you have the correct authority to perform the operation.

Table 39. Access level required to profiles for topic security for closure of a subscribe operation	
MQCLOSE option	**RACF access required to `hlq.SUBSCRIBE.topicname` profile in MXTOPIC class**
MQCO_REMOVE_SUB	ALTER

Publish

To publish on a topic you need access to the topic and, if you are using alias queues, to the alias queue as well.

Table 40. Access level required for topic security to publish	
MQOPEN or MQPUT1 option	**RACF access required to `hlq.PUBLISH.topicname` profile in MXTOPIC class**
MQOO_OUTPUT or MQPUT1	UPDATE

Table 41. Access level required to open an alias queue that resolves to a topic	
MQOPEN or MQPUT1 option	**RACF access required to `hlq.queuename` profile in MQQUEUE or MXQUEUE class for the alias queue**
MQOO_OUTPUT or MQPUT1	UPDATE

For details of how topic security operates when an alias queue that resolves to a topic name is opened for publish, see "Considerations for alias queues that resolve to topics for a publish operation" on page 215.

When you consider alias queues used for destination queues for PUT or GET restrictions, see "Considerations for alias queues" on page 205.

If the RACF access level that an application has to a topic security profile is changed, the changes take effect only for any new object handles obtained (that is, a new MQSUB or MQOPEN) for that topic. Those handles already in existence at the time of the change retain their existing access to the topic. Also, existing subscribers retain their access to any subscriptions that they have already made.

Considerations for alias queues that resolve to topics for a publish operation

When you issue an MQOPEN or MQPUT1 call for an alias queue that resolves to a topic, IBM MQ makes two resource checks:

- The first one against the alias queue name specified in the object descriptor (MQOD) on the MQOPEN or MQPUT1 call.
- The second against the topic to which the alias queue resolves

You must be aware that this behavior is different from the behavior you get when alias queues resolve to other queues. You need the correct access to both profiles in order for the publish action to proceed.

System topic security

The following system topics are accessed by the channel initiator address space.

The user IDs under which this runs must be given RACF access to these queues, as shown in Table 42 on page 215.

Table 42. Access required to the SYSTEM topics		
SYSTEM topic	**Profile**	**Channel initiator for distributed queuing**
SYSTEM.BROKER.ADMIN.STREAM	hlq.PUBLISH.topicname	UPDATE
SYSTEM.BROKER.ADMIN.STREAM	hlq.SUBSCRIBE.topicname	ALTER

z/OS *Profiles for processes*

If process security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

If process security is active, you must:

- Define profiles in the **MQPROC** or **GMQPROC** classes if using uppercase profiles.
- Define profiles in the **MXPROC** or **GMXPROC** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use processes.

Profiles for processes take the form:

```
hlq.processname
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue sharing group name), and processname is the name of the process being opened.

A profile prefixed by the queue manager name controls access to a single process definition on that queue manager. A profile prefixed by the queue sharing group name controls access to one or more process definitions with that name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that process definition on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access required for opening a process.

Table 43. Access levels for process security

MQOPEN option	**RACF access level required to hlq.processname**
MQOO_INQUIRE	READ

For example, on queue manager MQS9, the RACF group INQVPRC must be able to inquire (MQINQ) on all processes starting with the letter V. The RACF definitions for this would be:

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

Alternate user security might also be active, depending on the open options specified when a process definition object is opened.

z/OS *Profiles for namelists*

If namelist security is active, you define profiles in the appropriate classes and give the necessary groups or user IDs access to these profiles.

If namelist security is active, you must:

- Define profiles in the **MQNLIST** or **GMQNLIST** classes if using uppercase profiles.
- Define profiles in the **MXNLIST** or **GMXNLIST** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles.

Profiles for namelists take the form:

```
hlq.namelistname
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue sharing group name), and namelistname is the name of the namelist being opened.

A profile prefixed by the queue manager name controls access to a single namelist on that queue manager. A profile prefixed by the queue sharing group name controls access to access to one or more namelists with that name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that namelist on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access required for opening a namelist.

Table 44. Access levels for namelist security

MQOPEN option	RACF access level required to hlq.namelistname
MQOO_INQUIRE	READ

For example, on queue manager (or queue sharing group) PQM3, the RACF group DEPT571 must be able to inquire (`MQINQ`) on these namelists:

- All namelists starting with "DEPT571".
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

The RACF definitions to do this are:

```
RDEFINE MQNLIST PQM3.DEPT571.** UACC(NONE)
PERMIT PQM3.DEPT571.** CLASS(MQNLIST) ID(DEPT571) ACCESS(READ)

RDEFINE GMQNLIST NLISTS.FOR.DEPT571 UACC(NONE)
        ADDMEM(PQM3.PRINTER/DESTINATIONS/DEPT571,
               PQM3.AGENCY/REQUEST/QUEUES,
               PQM3.WAREHOUSE.BROADCAST)
PERMIT NLISTS.FOR.DEPT571 CLASS(GMQNLIST) ID(DEPT571) ACCESS(READ)
```

Alternate user security might be active, depending on the options specified when a namelist object is opened.

System namelist security

Many of the system namelists are accessed by the ancillary parts of IBM MQ:

- The CSQUTIL utility
- The operations and control panels
- The channel initiator address space (including the Queued Publish/Subscribe Daemon)

The user IDs under which these run must be given RACF access to these namelists, as shown in Table 45 on page 217.

Table 45. Access required to the SYSTEM namelists by IBM MQ

SYSTEM namelist	CSQUTIL	Operations and control panels	Channel initiator for distributed queuing
SYSTEM.QPUBSUB.QUEUE.NAMELIST	-	-	READ
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	-	-	READ

z/OS *Profiles for alternate user security*

If alternate user security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

For more information about *AlternateUserId*, see AlternateUserID (MQCHAR12).

If alternate user security is active, you must:

- Define profiles in the MQADMIN or GMQADMIN classes if you are using uppercase profiles.
- Define profiles in the MXADMIN or GMXADMIN classes if you are using mixed case profiles.

Permit the necessary groups or user IDs access to these profiles, so that they can use the ALTERNATE_USER_AUTHORITY options when the object is opened.

Profiles for alternate user security can be specified at subsystem level or at queue sharing group level and take the following form:

```
hlq.ALTERNATE.USER.alternateuserid
```

Where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `alternateuserid` is the value of the *`AlternateUserId`* field in the object descriptor.

A profile prefixed by the queue manager name controls use of an alternative user ID on that queue manager. A profile prefixed by the queue sharing group name controls use of an alternative user ID on all queue managers within the queue sharing group. This alternative user ID can be used on any queue manager within the queue sharing group by a user that has the correct access. This access can be overridden on an individual queue manager by defining a queue manager level profile for that alternative user ID on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access when specifying an alternative user option.

Table 46. Access levels for alternate user security

MQOPEN, MQSUB, or MQPUT1 option	**RACF access level required**
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	UPDATE

In addition to alternate user security checks, other security checks for queue, process, namelist, and context security can also be made. The alternative user ID, if provided, is only used for security checks on queue, process definition, or namelist resources. For alternate user and context security checks, the user ID requesting that the check is used. For details about how user IDs are handled, see “User IDs for security checking on z/OS” on page 239. For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 211.

An alternative user profile gives the requesting user ID access to resources associated with the user ID specified in the alternative user ID. For example, the payroll server running under user ID PAYSERV on queue manager QMPY processes requests from personnel user IDs, all of which start with PS. To cause the work performed by the payroll server to be carried out under the user ID of the requesting user, alternative user authority is used. The payroll server knows which user ID to specify as the alternative user ID because the requesting programs generate messages using the MQPMO_DEFAULT_CONTEXT put message option. See “User IDs for security checking on z/OS” on page 239 for more details about from where alternative user IDs are obtained.

The following example RACF definitions enable the server program to specify alternative user IDs starting with the characters PS:

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

Note:

1. The *AlternateUserId* fields in the object descriptor and subscription descriptor are 12 bytes long. All 12 bytes are used in the profile checks, but only the first 8 bytes are used as the user ID by IBM MQ. If this user ID truncation is not desirable, application programs making the request must translate any alternative user ID over 8 bytes into something more appropriate.
2. If you specify MQOO_ALTERNATE_USER_AUTHORITY, MQSO_ALTERNATE_USER_AUTHORITY, or MQPMO_ALTERNATE_USER_AUTHORITY and you do not specify an *AlternateUserId* field in the object descriptor, a user ID of blanks is used. For the purposes of the alternate user security check the user ID used for the *AlternateUserId* qualifier is -BLANK-. For example `RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-`.

   If the user is allowed to access this profile, all further checks are made with a user ID of blanks. For details of blank user IDs, see "Blank user IDs and UACC levels" on page 247.

The administration of alternative user IDs is easier if you have a naming convention for user IDs that enables you to use generic alternative user profiles. If they do not, you can use the RACF RACVAR feature. For details about using RACVAR, see the z/OS Security Server RACF documentation..

When a message is put to a queue that has been opened with alternative user authority and the context of the message has been generated by the queue manager, the MQMD_USER_IDENTIFIER field is set to the alternative user ID.

z/OS *Profiles for context security*

If context security is active, to control access to the message context information you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles. The message context is contained within the message descriptor (MQMD).

Using profiles for context security

If context security is active, to permit users to access context information for messages on a particular queue, or when publishing to a particular topic, you must define a profile in one of the following classes:

- The `MQADMIN` class if using uppercase profiles.
- The `MXADMIN` class if using mixed-case profiles.

Profiles for context security can be specified at subsystem level or at queue sharing group level and take the following form:

```
hlq.CONTEXT.queuename
hlq.CONTEXT.topicname
```

where *hlq* can be either the queue manager name or the queue sharing group name, and *queuename* and *topicname* can be either the full or generic name of the queue or topic you want to define the context profile for.

A profile prefixed by the queue manager name, and with ** specified as the queue or topic name, allows control for context security on all queues and topics belonging to that queue manager. This can be overridden on an individual queue or topic by defining a specific profile for context on that queue or topic.

A profile prefixed by the queue sharing group name, and with ** specified as the queue or topic name, allows control for context on all queues and topics belonging to the queue managers within the queue sharing group. This can be overridden on an individual queue manager by defining a queue manager level profile for context on that queue manager, by specifying a profile prefixed by the queue manager name. It can also be overridden on an individual queue or topic by specifying a profile suffixed with the queue or topic name.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

You must permit the necessary groups or user IDs access to this profile. The following table shows the access level required, depending on the specification of the context options when the queue is opened.

Table 47. Access levels for context security	
MQOPEN or MQPUT1 option	**RACF access level required to hlq.CONTEXT.queuename or hlq.CONTEXT.topicname**
MQPMO_NO_CONTEXT	No context security check
MQPMO_DEFAULT_CONTEXT	No context security check
MQOO_SAVE_ALL_CONTEXT	No context security check
MQOO_PASS_IDENTITY_CONTEXT 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 or MQPUT1 (USAGE(XMITQ))	CONTROL
MQSUB option	
MQSO_SET_IDENTITY_CONTEXT **(Note 2)**	UPDATE

Note:

1. The user IDs used for distributed queuing require `CONTROL` access to `hlq.CONTEXT.queuename` to put messages on the destination queue. See "User IDs used by the channel initiator" on page 242 for information about the user IDs used.
2. If on the MQSUB request, with `MQSO_CREATE` or `MQSO_ALTER` options specified, you want to set any of the identity context fields in the MQSD structure, you need to specify the `MQSO_SET_IDENTITY_CONTEXT` option. You require also, the appropriate authority to the context profile for the destination queue.

If you put commands on the system-command input queue, use the default context put message option to associate the correct user ID with the command.

For example, the IBM MQ-supplied utility program CSQUTIL can be used to offload and reload messages in queues. When offloaded messages are restored to a queue, the CSQUTIL utility uses the `MQOO_SET_ALL_CONTEXT` option to return the messages to their original state. In addition to the queue security required by this open option, context authority is also required. For example, if this authority is required by the group BACKGRP on queue manager MQS1, this would be defined by:

```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

Depending on the options specified, and the types of security performed, other types of security checks might also occur when the queue is opened. These include queue security (see "Profiles for queue security" on page 203), and alternate user security (see "Profiles for alternate user security" on page 217). For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 211.

System queue context security

Many of the system queues are accessed by the ancillary parts of IBM MQ, for example the channel initiator address space, and the mqweb server used by the IBM MQ Console and REST API.

The user IDs under which these run under must be given RACF access to these queues, as shown in Table 48 on page 221.

Table 48. Access required to the SYSTEM queues for context operations

SYSTEM queue	Channel initiator for distributed queuing	mqweb server
SYSTEM.ADMIN.COMMAND.QUEUE	-	CONTROL
SYSTEM.BROKER.CONTROL.QUEUE	CONTROL	-
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	CONTROL	-
SYSTEM.CHANNEL.SYNCQ	CONTROL	-
SYSTEM.CLUSTER.COMMAND.QUEUE	CONTROL	-
SYSTEM.CLUSTER.TRANSMIT.QUEUE	CONTROL	-

z/OS *Profiles for command security*

To enable security checking for commands, add profiles to the MQCMDS class. The profile names are based on the MQSC commands but control both MQSC and PCF commands. Profiles can apply to a queue manager or a queue sharing group.

If you want security checking for commands (so you have not defined the command security switch profile hlq.NO.CMD.CHECKS) you must add profiles to the MQCMDS class.

The same security profiles control both MQSC and PCF commands. The names of the RACF profiles for command security checking are based on the MQSC command names themselves. These profiles take the form:

```
hlq.verb.pkw
```

Where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), `verb` is the verb part of the command name, for example ALTER, and `pkw` is the object type, for example QLOCAL for a local queue.

Thus, the profile name for the ALTER QLOCAL command in subsystem CSQ1 is:

```
CSQ1.ALTER.QLOCAL
```

You can use generic profiles to protect sets of commands so that you have fewer profiles to maintain and, therefore, fewer access lists. Consider creating a generic profile that applies to all commands not protected by a more specific profile. Define this profile with UACC(NONE) and grant ALTER access only to the RACF groups containing administrators. You might then create a generic profile applicable to all DISPLAY commands and grant widespread access to it. Between these extremes, you might identify groups of users needing access to certain sets of commands, in which case you can create profiles for those sets and grant access to RACF groups representing those classes of user. Avoid giving users access to commands they do not require: Apply the principle of least privilege, so that users only have access to the commands that are required for their jobs.

A profile prefixed by the queue manager name controls the use of the command on that queue manager. A profile prefixed by the queue sharing group name controls the use of the command on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

By setting up command profiles at queue manager level, a user can be restricted from issuing commands on a particular queue manager. Alternatively, you can define one profile for a queue sharing group for each command verb, and all security checks take place against that profile instead of individual queue managers.

If both subsystem security and queue sharing group security are active and a local profile is not found, a command security check is performed to see if the user has access to a queue sharing group profile.

If you use the CMDSCOPE attribute to route a command to other queue managers in a queue sharing group, security is checked on each queue manager where the command is run, but not necessarily on the queue manager where the command is entered.

Table 49 on page 222 shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 50 on page 227 shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 49. MQSC commands, profiles, and their access levels

Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
ALTER AUTHINFO	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
ALTER BUFFPOOL	hlq.ALTER.BUFFPOOL	ALTER	No check	-
ALTER CFSTRUCT	hlq.ALTER.CFSTRUCT	ALTER	No check	-
ALTER CHANNEL	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
ALTER NAMELIST	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
ALTER PROCESS	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
ALTER PSID	hlq.ALTER.PSID	ALTER	No check	-
ALTER QALIAS	hlq.ALTER.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
ALTER QLOCAL“5” on page 227	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	No check	-
ALTER QMODEL“5” on page 227	hlq.ALTER.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QREMOTE	hlq.ALTER.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
ALTER SECURITY	hlq.ALTER.SECURITY	ALTER	No check	-
ALTER SMDS	hlq.ALTER.SMDS	ALTER	No check	-
ALTER STGCLASS	hlq.ALTER.STGCLASS	ALTER	No check	-
ALTER SUB	hlq.ALTER.SUB	ALTER	No check	-
ALTER TOPIC	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
ALTER TRACE	hlq.ALTER.TRACE	ALTER	No check	-
ARCHIVE LOG	hlq.ARCHIVE.LOG	CONTROL	No check	-

Table 49. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
BACKUP CFSTRUCT	hlq.BACKUP.CFSTRUCT	CONTROL	No check	-
CLEAR QLOCAL	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
CLEAR TOPICSTR “3” on page 226	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
DEFINE AUTHINFO	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DEFINE BUFFPOOL	hlq.DEFINE.BUFFPOOL	ALTER	No check	-
DEFINE CFSTRUCT	hlq.DEFINE.CFSTRUCT	ALTER	No check	-
DEFINE CHANNEL	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DEFINE LOG	hlq.DEFINE.LOG	ALTER	No check	-
DEFINE MAXSMSGS	hlq.DEFINE.MAXSMSGS	ALTER	No check	-
DEFINE NAMELIST	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DEFINE PROCESS	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DEFINE PSID	hlq.DEFINE.PSID	ALTER	No check	-
DEFINE QALIAS	hlq.DEFINE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QLOCAL“5” on page 227	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL“5” on page 227	hlq.DEFINE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QREMOTE	hlq.DEFINE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DEFINE STGCLASS	hlq.DEFINE.STGCLASS	ALTER	No check	-
DEFINE SUB	hlq.DEFINE.SUB	ALTER	No check	-
DEFINE TOPIC	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DELETE AUTHINFO	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DELETE BUFFPOOL	hlq.DELETE.BUFFPOOL	ALTER	No check	-
DELETE CFSTRUCT	hlq.DELETE.CFSTRUCT	ALTER	No check	-
DELETE CHANNEL	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DELETE NAMELIST	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DELETE PROCESS	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DELETE PSID	hlq.DELETE.PSID	ALTER	No check	-
DELETE QALIAS	hlq.DELETE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DELETE QLOCAL	hlq.DELETE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER

Table 49. MQSC commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
DELETE QMODEL	hlq.DELETE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QREMOTE	hlq.DELETE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DELETE STGCLASS	hlq.DELETE.STGCLASS	ALTER	No check	-
DELETE SUB	hlq.DELETE.SUB	ALTER	No check	-
DELETE TOPIC	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DISPLAY ARCHIVE “1” on page 226	hlq.DISPLAY.ARCHIVE	READ	No check	-
DISPLAY AUTHINFO	hlq.DISPLAY.AUTHINFO	READ	No check	-
DISPLAY CFSTATUS	hlq.DISPLAY.CFSTATUS	READ	No check	-
DISPLAY CFSTRUCT	hlq.DISPLAY.CFSTRUCT	READ	No check	-
DISPLAY CHANNEL	hlq.DISPLAY.CHANNEL	READ	No check	-
DISPLAY CHINIT	hlq.DISPLAY.CHINIT	READ	No check	-
DISPLAY CHLAUTH	hlq.DISPLAY.CHLAUTH	READ	No check	-
DISPLAY CHSTATUS	hlq.DISPLAY.CHSTATUS	READ	No check	-
DISPLAY CLUSQMGR	hlq.DISPLAY.CLUSQMGR	READ	No check	-
DISPLAY CMDSERV	hlq.DISPLAY.CMDSERV	READ	No check	-
DISPLAY CONN “1” on page 226	hlq.DISPLAY.CONN	READ	No check	-
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	No check	-
DISPLAY LOG “1” on page 226	hlq.DISPLAY.LOG	READ	No check	-
DISPLAY MAXSMSGS	hlq.DISPLAY.MAXSMSGS	READ	No check	-
DISPLAY NAMELIST	hlq.DISPLAY.NAMELIST	READ	No check	-
DISPLAY PROCESS	hlq.DISPLAY.PROCESS	READ	No check	-
DISPLAY PUBSUB	hlq.DISPLAY.PUBSUB	READ	No check	-
DISPLAY QALIAS	hlq.DISPLAY.QALIAS	READ	No check	-
DISPLAY QCLUSTER	hlq.DISPLAY.QCLUSTER	READ	No check	-
DISPLAY QLOCAL	hlq.DISPLAY.QLOCAL	READ	No check	-
DISPLAY QMGR	hlq.DISPLAY.QMGR	READ	No check	-
DISPLAY QMODEL	hlq.DISPLAY.QMODEL	READ	No check	-
DISPLAY QREMOTE	hlq.DISPLAY.QREMOTE	READ	No check	-
DISPLAY QSTATUS	hlq.DISPLAY.QSTATUS	READ	No check	-
DISPLAY QUEUE	hlq.DISPLAY.QUEUE	READ	No check	-

Table 49. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
DISPLAY SBSTATUS	hlq.DISPLAY.SBSTATUS	READ	No check	-
DISPLAY SMDS	hlq.DISPLAY.SMDS	READ	No check	-
DISPLAY SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	No check	-
DISPLAY SUB	hlq.DISPLAY.SUB	READ	No check	-
DISPLAY SECURITY	hlq.DISPLAY.SECURITY	READ	No check	-
DISPLAY STGCLASS	hlq.DISPLAY.STGCLASS	READ	No check	-
DISPLAY SYSTEM “1” on page 226	hlq.DISPLAY.SYSTEM	READ	No check	-
DISPLAY THREAD	hlq.DISPLAY.THREAD	READ	No check	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	No check	-
DISPLAY TOPIC	hlq.DISPLAY.TOPIC	READ	No check	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	No check	-
DISPLAY TRACE	hlq.DISPLAY.TRACE	READ	No check	-
DISPLAY USAGE “1” on page 226	hlq.DISPLAY.USAGE	READ	No check	-
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	No check	-
RECOVER CFSTRUCT	hlq.RECOVER.CFSTRUCT	CONTROL	No check	-
REFRESH CLUSTER	hlq.REFRESH.CLUSTER	ALTER	No check	-
REFRESH QMGR	hlq.REFRESH.QMGR	ALTER	No check	-
REFRESH SECURITY	hlq.REFRESH.SECURITY	ALTER	No check	-
RESET CFSTRUCT	hlq.RESET.CFSTRUCT	CONTROL	No check	-
RESET CHANNEL	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESET CLUSTER	hlq.RESET.CLUSTER	CONTROL	No check	-
RESET QMGR	hlq.RESET.QMGR	CONTROL	No check	-
RESET QSTATS	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
RESET SMDS	hlq.RESET.SMDS	CONTROL	No check	-
RESET TPIPE	hlq.RESET.TPIPE	CONTROL	No check	-
RESOLVE CHANNEL	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESOLVE INDOUBT	hlq.RESOLVE.INDOUBT	CONTROL	No check	-
RESUME QMGR	hlq.RESUME.QMGR	CONTROL	No check	-

Table 49. MQSC commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
RVERIFY SECURITY	hlq.RVERIFY.SECURITY	ALTER	No check	-
SET ARCHIVE	hlq.SET.ARCHIVE	CONTROL	No check	-
SET CHLAUTH	hlq.SET.CHLAUTH	CONTROL	No check	-
SET LOG	hlq.SET.LOG	CONTROL	No check	-
SET SYSTEM	hlq.SET.SYSTEM	CONTROL	No check	-
START CHANNEL	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
START CHINIT “4” on page 226	hlq.START.CHINIT	CONTROL	No check	-
START CMDSERV	hlq.START.CMDSERV	CONTROL	No check	-
START LISTENER	hlq.START.LISTENER	CONTROL	No check	-
START QMGR	None “2” on page 226	-	-	-
START SMDSCONN	hlq.START.SMDSCONN	CONTROL	No check	-
START TRACE	hlq.START.TRACE	CONTROL	No check	-
STOP CHANNEL	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
STOP CHINIT	hlq.STOP.CHINIT	CONTROL	No check	-
STOP CMDSERV	hlq.STOP.CMDSERV	CONTROL	No check	-
STOP LISTENER	hlq.STOP.LISTENER	CONTROL	No check	-
STOP QMGR	hlq.STOP.QMGR	CONTROL	No check	-
STOP SMDSCONN	hlq.STOP.SMDSCONN	CONTROL	No check	-
STOP TRACE	hlq.STOP.TRACE	CONTROL	No check	-
SUSPEND QMGR	hlq.SUSPEND.QMGR	CONTROL	No check	-

Notes:

1. These commands might be issued internally by the queue manager; no authority is checked in these cases.
2. IBM MQ does not check the authority of the user who issues the START QMGR command. However, you can use RACF, or your alternative security facilities to control access to the START xxxxMSTR command that is issued as a result of the START QMGR command.

   This is done by controlling access to the MVS.START.STC.xxxxMSTR profile in the RACF operator commands (OPERCMDS) class. For details of this procedure, see Granting the user access to the RACF OPERCMDS class in *z/OS MVS Planning: Operations*. If you use this technique, and an unauthorized user tries to start the queue manager, it terminates with a reason code of 00F30216.
3. The **hlq.TOPIC.topic** resource refers to the Topic object derived from the TOPICSTR. For more details, see “パブリッシュ/サブスクライブのセキュリティー” on page 484
4. In IBM MQ for z/OS, the resource name MVS.START.STC.CSQ1CHIN has an additional JOBNAME qualifier appended. This can cause problems when starting the channel initiator.

To resolve the problem replace MVS.START.STC. *ssid* CHIN with a profile for a resource named MVS.START.STC. *ssid* CHIN **.*** or MVS.START.STC. *ssid* CHIN. *ssid* CHIN where *ssid* is the subsystem ID for the queue manager. This requires RACF UPDATE authority. For more details, see MVS™ Commands, RACF Access Authorities, and Resource Names in *z/OS MVS Planning: Operations*.

The START for *ssid* MSTR does not include the JOBNAME= parameter. For consistency, you might want to update the profile for MVS.START.STC.ssidMSTR to MVS.START.STC.ssidMSTR.*.

5. Setting the queue attribute STREAMQ to a non blank value also requires ALTER access level to MQADMIN or MXADMIN for `hlq.ALTER.streamQ`.

Table 50. PCF commands, profiles, and their access levels

Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
Backup CF Structure	hlq.BACKUP.CFSTRUCT	CONTROL	No check	-
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change CF Structure	hlq.ALTER.CFSTRUCT	ALTER	No check	-
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Namelist	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Change Process	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Change Queue“2” on page 230	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	No check	-
Change Security	hlq.ALTER.SECURITY	ALTER	No check	-
Change SMDS	hlq.ALTER.SMDS	ALTER	No check	-
Change Storage Class	hlq.ALTER.STGCLASS	ALTER	No check	-
Change Subscription	hlq.ALTER.SUB	ALTER	No check	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Clear Topic String “1” on page 230	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
Copy Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Copy CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	No check	-
Copy Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Copy Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Copy Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Copy Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Copy Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Copy Storage Class	hlq.DEFINE.STGCLASS	ALTER	No check	-
Copy Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER

Table 50. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Create CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	No check	-
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Create Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Create Queue“2” on page 230	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Storage Class	hlq.DEFINE.STGCLASS	ALTER	No check	-
Create Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete CF Structure	hlq.DELETE.CFSTRUCT	ALTER	No check	-
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Namelist	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Delete Process	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Storage Class	hlq.DELETE.STGCLASS	ALTER	No check	-
Delete Subscription	hlq.DELETE.SUB	ALTER	No check	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Archive	hlq.DISPLAY.ARCHIVE	READ	No check	-
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire CF Structure	hlq.DISPLAY.CFSTRUCT	READ	No check	-
Inquire CF Structure Names	hlq.DISPLAY.CFSTRUCT	READ	No check	-
Inquire CF Structure Status	hlq.DISPLAY.CFSTATUS	READ	No check	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	No check	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	No check	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	No check	-
Inquire Cluster Queue Manager	hlq.DISPLAY.CLUSQMGR	READ	No check	-

Table 50. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Inquire Connection	hlq.DISPLAY.CONNPCF	READ	No check	-
Inquire Group	hlq.DISPLAY.GROUP	READ	No check	-
Inquire Log	hlq.DISPLAY.LOG	READ	No check	-
Inquire Namelist	hlq.DISPLAY.NAMELIST	READ	No check	-
Inquire Namelist Names	hlq.DISPLAY.NAMELIST	READ	No check	-
Inquire Process	hlq.DISPLAY.PROCESS	READ	No check	-
Inquire Process Names	hlq.DISPLAY.PROCESS	READ	No check	-
Inquire Pub/Sub Status	hlq.DISPLAY.PUBSUB	READ	No check	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	No check	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	No check	-
Inquire Security	hlq.DISPLAY.SECURITY	READ	No check	-
Inquire SMDS	hlq.DISPLAY.SMDS	READ	No check	-
Inquire SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	No check	-
Inquire Storage Class	hlq.DISPLAY.STGCLASS	READ	No check	-
Inquire Storage Class Names	hlq.DISPLAY.STGCLASS	READ	No check	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	No check	-
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	No check	-
Inquire System	hlq.DISPLAY.SYSTEM	READ	No check	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Names	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Status	hlq.DISPLAY.TPSTATUS	READ	No check	-
Inquire Usage	hlq.DISPLAY.USAGE	READ	No check	-
Move Queue	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Recover CF Structure	hlq.RECOVER.CFSTRUCT	CONTROL	No check	-
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	No check	-
Refresh Queue Manager	hlq.REFRESH.QMGR	ALTER	No check	-
Refresh Security	hlq.REFRESH.SECURITY	ALTER	No check	-
Reset CF Structure	hlq.RESET.CFSTRUCT	CONTROL	No check	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

Table 50. PCF commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
Reset Cluster	hlq.RESET.CLUSTER	CONTROL	No check	-
Reset Queue Manager	hlq.RESET.QMGR	CONTROL	No check	-
Reset Queue Statistics	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
Reset SMDS	hlq.RESET.SMDS	CONTROL	No check	-
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resume Queue Manager	hlq.RESUME.QMGR	CONTROL	No check	-
Resume Queue Manager Cluster	hlq.RESUME.QMGR	CONTROL	No check	-
Reverify Security	hlq.RVERIFY.SECURITY	ALTER	No check	-
Set Archive	hlq.SET.ARCHIVE	CONTROL	No check	-
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	No check	-
Set Log	hlq.SET.LOG	CONTROL	No check	-
Set System	hlq.SET.SYSTEM	CONTROL	No check	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Start Channel Initiator	hlq.START.CHINIT	CONTROL	No check	-
Start Channel Listener	hlq.START.LISTENER	CONTROL	No check	-
Start SMDS Connection	hlq.START.SMDSCONN	CONTROL	No check	-
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel Initiator	hlq.STOP.CHINIT	CONTROL	No check	-
Stop Channel Listener	hlq.STOP.LISTENER	CONTROL	No check	-
Stop SMDS Connection	hlq.STOP.SMDSCONN	CONTROL	No check	-
Suspend Queue Manager	hlq.SUSPEND.QMGR	CONTROL	No check	-
Suspend Queue Manager Cluster	hlq.SUSPEND.QMGR	CONTROL	No check	-

Notes:

1. The **hlq.TOPIC.topic** resource refers to the Topic object derived from the TOPICSTR. For more details, see “パブリッシュ/サブスクライブのセキュリティー” on page 484
2. Setting the queue attribute STREAMQ to a non blank value also requires ALTER access level to MQADMIN or MXADMIN for `hlq.ALTER.streamQ`.

See “IBM MQ Console - required command security profiles” on page 230 for details of the IBM MQ PCF profiles required, when using the IBM MQ Console.

z/OS *IBM MQ Console - required command security profiles*

Operations performed in the IBM MQ Console by a user in the `MQWebAdmin`, or `MQWebAdminRO`, role take place under the security context of the mqweb server started task user ID. If you want to use the IBM MQ Console, the mqweb server started task user ID needs authorization to issue certain PCF commands.

Table 51 on page 231 shows, for each IBM MQ PCF command, the command security profiles required, and the corresponding access level for each profile in the MQCMDS class needed by the IBM MQ Console.

Table 51. IBM MQ Console PCF commands, profiles, and their access levels

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Queue	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	No check	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Subscription	hlq.DELETE.SUB	ALTER	No check	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	No check	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	No check	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	No check	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	No check	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	No check	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	No check	-

Table 51. IBM MQ Console PCF commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	No check	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Names	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Status	hlq.DISPLAY.TPSTATUS	READ	No check	-
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	No check	-
Refresh Security	hlq.REFRESH.SECURITY	ALTER	No check	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	No check	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

z/OS *Profiles for command resource security*

If you have not defined the command resource security switch profile, because you want security checking for resources associated with commands, you must add resource profiles for each resource to the appropriate class. The same security profiles control both MQSC and PCF commands.

If you have not defined the command resource security switch profile, `hlq.NO.CMD.RESC.CHECKS`, because you want security checking for resources associated with commands, you must:

- Add a resource profile in the **MQADMIN** class, if using uppercase profiles, for each resource.
- Add a resource profile in the **MXADMIN** class, if using mixed case profiles, for each resource.

The same security profiles control both MQSC and PCF commands.

Profiles for command resource security checking take the form:

```
hlq.type.resourcename
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).

A profile prefixed by the queue manager name controls access to the resources associated with commands on that queue manager. A profile prefixed by the queue sharing group name controls access to the resources associated with commands on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command resource on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

For example, the RACF profile name for command resource security checking against the model queue CREDIT.WORTHY in subsystem CSQ1 is:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

Because the profiles for all types of command resource are held in the MQADMIN class, the "type" part of the profile name is needed in the profile to distinguish between resources of different types that have the same name. The "type" part of the profile name can be CHANNEL, QUEUE, TOPIC, PROCESS, or NAMELIST. For example, a user might be authorized to define hlq.QUEUE.PAYROLL.ONE, but not authorized to define hlq.PROCESS.PAYROLL.ONE

If the resource type is a queue, and the profile is a queue sharing group level profile, it controls access to one or more local queues within the queue sharing group, or access to a single shared queue from any queue manager in the queue sharing group.

MQSC commands, profiles, and their access levels shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

PCF commands, profiles, and their access levels shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

z/OS *Command resource security checking for alias queues and remote queues*

Alias queue and remote queues both provide indirection to another queue. Additional points apply when you consider security checking for these queues.

Alias queues

When you define an alias queue, command resource security checks are only performed against the name of the alias queue, not against the name of the target queue to which the alias resolves.

Alias queues can resolve to both local and remote queues. If you do not want to permit users access to certain local or remote queues, you must do both of the following:

1. Do not allow the users access to these local and remote queues.
2. Restrict the users from being able to define aliases for these queues. That is, prevent them from being able to issue DEFINE QALIAS and ALTER QALIAS commands.

Remote queues

When you define a remote queue, command resource security checks are performed only against the name of the remote queue. No checks are performed against the names of the queues specified in the RNAME or XMITQ attributes in the remote queue object definition.

z/OS RESLEVEL 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、258 ページの『Auditing considerations on z/OS』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、このプロファイルをキュー・マネージャー・レベルで定義しない場合、検査のレベルに影響す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可能性があることに注意してください。 2 つのユーザー ID の検査をアクティブにするには、UACC (NONE) を使用して RESLEVEL プロファイル (キュー共有グループ名のキュー・マネージャー名のいずれかを接頭部として付ける) を定義し、関連するユーザーにこのプロファイルに対するアクセス権限が付与されていないことを確認します。

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 and batch connections*

By default, when an IBM MQ resource is being accessed through batch and batch-type connections, the user must be authorized to access that resource for the particular operation. You can bypass the security check by setting up an appropriate RESLEVEL definition.

Whether the user is checked or not is based on the user ID used at connect time, the same user ID used for the connection check.

For example, you can set up RESLEVEL so that when a user you trust accesses certain resources through a batch connection, no API-resource security checks are done; but when a user you do not trust tries to access the same resources, security checks are carried out as normal. You should set up RESLEVEL checking to bypass API-resource security checks only when you sufficiently trust the user and the programs run by that user.

The following table shows the checks made for batch connections.

Table 52. Checks made at different RACF access levels for batch connections

RACF access level	**Level of checking**
NONE	Resource checks performed
READ	Resource checks performed
UPDATE	Resource checks performed
CONTROL	No check.
ALTER	No check.

z/OS *RESLEVEL and system functions*

The application of RESLEVEL to the operation and control panels, and to CSQUTIL.

The operation and control panels and the CSQUTIL utility are batch-type applications that make requests to the queue manager's command server, and so they are subject to the considerations described in “RESLEVEL and batch connections” on page 235. You can use RESLEVEL to bypass security checking for the SYSTEM.COMMAND.INPUT and SYSTEM.COMMAND.REPLY.MODEL queues that they use, but not for the dynamic queues SYSTEM.CSQXCMD.*, SYSTEM.CSQOREXX.*, and SYSTEM.CSQUTIL.*.

The command server is an integral part of the queue manager and so does not have connection or RESLEVEL checking associated with it. To maintain security, therefore, the command server must confirm that the user ID of the requesting application has authority to open the queue being used for replies. For the operations and control panels, this is SYSTEM.CSQOREXX.*. For CSQUTIL, it is SYSTEM.CSQUTIL.*. Users must be authorized to use these queues, as described in “System queue security” on page 209, in addition to any RESLEVEL authorization they are given.

For other applications using the command server, it is the queue they name as their reply-to queue. Such other applications might deceive the command server into placing messages on unauthorized queues by passing (in the message context) a more trusted user ID than its own to the command server. To prevent this, use a CONTEXT profile to protect the identity context of messages placed on SYSTEM.COMMAND.INPUT.

z/OS *RESLEVEL and CICS connections*

By default, when an API-resource security check is made on a CICS connection, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

The first user ID checked is that of the CICS address space. This is the user ID on the job card of the CICS job, or the user ID assigned to the CICS started task by the z/OS STARTED class or the started procedures table. (It is not the CICS DFLTUSER.)

The second user ID checked is the user ID associated with the CICS transaction.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED. Both the CICS address space user ID and the user ID of the person running the CICS transaction must have access to the resource at the correct level.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. See Table 53 on page 236 for more information.

The user IDs checked depend on the user ID used at connection time, that is, the CICS address space user ID. This control enables you to bypass API-resource security checking for IBM MQ requests coming from one system (for example, a test system, TESTCICS,) but to implement them for another (for example, a production system, PRODCICS).

Note: If you set up your CICS address space user ID with the "trusted" attribute in the STARTED class or the RACF started procedures table ICHRIN03, this overrides any user ID checks for the CICS address space established by the RESLEVEL profile for your queue manager (that is, the queue manager does not perform the security checks for the CICS address space). For more information, see Securing CICS.

The following table shows the checks made for CICS connections.

Table 53. Checks made at different RACF access levels for CICS connections

RACF access level	Level of checking
NONE	IBM MQ checks the CICS address space user ID and the transaction user ID.
READ	IBM MQ checks the CICS address space user ID only.
UPDATE	If the transaction is defined to CICS with RESSEC(YES), IBM MQ checks the CICS address space user ID and the transaction user ID.
UPDATE	If the transaction is defined to CICS with RESSEC(NO), IBM MQ checks the CICS address space user ID only.
CONTROL or ALTER	IBM MQ does not check any user IDs.

z/OS *RESLEVEL and IMS connections*

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked to see if access is allowed to the resource.

The first user ID checked is that of the address space of the IMS region. This is taken from either the USER field from the job card or the user ID assigned to the region from the z/OS STARTED class or the started procedures table (SPT).

The second user ID checked is associated with the work being done in the dependent region. It is determined according to the type of the dependent region as shown in How the second user ID is determined for the IMS(tm) connection.

If either the first or second IMS user ID does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

The setting of IBM MQ RESLEVEL profiles cannot alter the user ID under which IMS transactions are scheduled from the IBM-supplied MQ-IMS trigger monitor program CSQQTRMN. This user ID is the PSBNAME of that trigger monitor, which by default is CSQQTRMN.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. The possible checks are:

- Check the IMS region address space user ID and the second user ID or alternate user ID.
- Check IMS region address space user ID only.
- Do not check any user IDs.

The following table shows the checks made for IMS connections.

Table 54. Checks made at different RACF access levels for IMS connections

RACF access level	Level of checking
NONE	Check the IMS address space user ID and the IMS second user ID or alternate user ID.
READ	Check the IMS address space user ID.
UPDATE	Check the IMS address space user ID.
CONTROL	No check.
ALTER	No check.

z/OS *RESLEVEL and the channel initiator connection*

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked to see if access is allowed to the resource.

The user IDs checked can be that specified by the MCAUSER channel attribute, that received from the network, that of the channel initiator address space, or the alternate user ID for the message descriptor. Which user IDs are checked depends on the communication protocol you are using and the setting of the PUTAUT channel attribute. See “User IDs used by the channel initiator” on page 242 for more information.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested, and how many are checked.

The following table shows the checks made for the channel initiator's connection, and for all channels since they use this connection.

Table 55. Checks made at different RACF access levels for channel initiator connections

RACF access level	Level of checking
NONE	Check two user IDs.
READ	Check one user ID.

Table 55. Checks made at different RACF access levels for channel initiator connections (continued)	
RACF access level	**Level of checking**
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
Note: See "User IDs used by the channel initiator" on page 242 for a definition of the user IDs checked	

z/OS *RESLEVEL and intra-group queuing*

By default, when an API-resource security check is made by the intra-group queuing agent, two user IDs are checked to see if access is allowed to the resource. You can change which user IDs are checked by setting up an RESLEVEL profile.

The user IDs checked can be the user ID determined by the IGQUSER attribute of the receiving queue manager, the user ID of the queue manager within the queue sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE, or the alternate user ID specified in the *UserIdentifier* field of the message descriptor of the message. See "User IDs used by the intra-group queuing agent" on page 246 for more information.

Because the intra-group queuing agent is an internal queue manager task, it does not issue an explicit connect request and runs under the user ID of the queue manager. The intra-group queuing agent starts at queue manager initialization. During the initialization of the intra-group queuing agent, IBM MQ checks the access that the user ID associated with the queue manager has to a profile in the MQADMIN class called:

```
hlq.RESLEVEL
```

This check is always performed unless the hlq.NO.SUBSYS.SECURITY switch has been set.

If there is no RESLEVEL profile, IBM MQ enables checking for two user IDs. If there is a RESLEVEL profile, the level of checking depends on the access level granted to the user ID of the queue manager for the profile. Checks made at different RACF(r) access levels for the intra-group queuing agent shows the checks made for the intra-group queuing agent.

Table 56. Checks made at different RACF access levels for the intra-group queuing agent	
RACF access level	**Level of checking**
NONE	Check two user IDs.
READ	Check one user ID.
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
Note: See "User IDs used by the intra-group queuing agent" on page 246 for a definition of the user IDs checked	

If the permissions granted to the RESLEVEL profile for the queue manager's user ID are changed, the intra-group queuing agent must be stopped and restarted to pick up the new permissions. Because there is no way to independently stop and restart the intra-group queuing agent, the queue manager must be stopped and restarted to achieve this.

z/OS *RESLEVEL and the user IDs checked*

Example of setting a RESLEVEL profile and granting access to it.

User ID checking against profile name for batch connections through User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels show how RESLEVEL affects which user IDs are checked for different MQI requests.

For example, you have a queue manager called QM66 with the following requirements:

- User WS21B is to be exempt from resource security.
- CICS started task WXNCICS running under address space user ID CICSWXN is to perform full resource checking only for transactions defined with RESSEC(YES).

To define the appropriate RESLEVEL profile, issue the following RACF command:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

Then give the users access to this profile, using the following commands:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

If you make these changes while the user IDs are connected to queue manager QM66, the users must disconnect and connect again before the change takes place.

If subsystem security is not active when a user connects but, while this user is still connected, subsystem security becomes active, full resource security checking is applied to the user. The user must reconnect to get the correct RESLEVEL processing.

z/OS User IDs for security checking on z/OS

IBM MQ initiates security checks based on user IDs associated with users, terminals, applications, and other resources. This collection of topics lists which user IDs are used for each type of security check.

z/OS *User IDs for connection security*

The user ID used for connection security depends on the type of connection.

Connection type	User ID contents
Batch connection	The user ID of the connecting task. For example: • The TSO user ID • The user ID assigned to a batch job by the USER JCL parameter • The user ID assigned to a started task by the STARTED class or the started procedures table
CICS connection	The CICS address space user ID.
IMS connection	The IMS region address space user ID.
Channel initiator connection	The channel initiator address space user ID.

z/OS *User IDs for command and command resource security*

The user ID used for command security or command resource security depends on where the command is issued from.

Issued from...	User ID contents
CSQINP1, CSQINP2, or CSQINPT	No check is made.

Issued from...	User ID contents
System command input queue	The user ID found in the *UserIdentifier* of the message descriptor of the message that contains the command. If the message does not contain a *UserIdentifier*, a user ID of blanks is passed to the security manager.
Console	The user ID signed onto the console. If the console is not signed on, the default user ID set by the CMDUSER system parameter in CSQ6SYSP. To issue commands from a console, the console must have the z/OS SYS AUTHORITY attribute.
SDSF/TSO console	TSO or job user ID.
Operations and control panels	TSO user ID. If you are going to use the operations and control panels, you must have the appropriate authority to issue the commands corresponding to the actions that you choose. In addition, you must have READ access to all the hlq.DISPLAY. *object* profiles in the MQCMDS class because the panels use the various DISPLAY commands to gather the information that they present.
MGCRE	If MGCRE is used with UTOKEN, the user ID in the UTOKEN. If MGCRE is issued without the UTOKEN, the TSO or job user ID is used.
CSQ0UTIL	Job user ID.
CSQUTIL	Job user ID.
CSQINPX	User ID of the channel initiator address space.

z/OS *User IDs for resource security (MQOPEN, MQSUB, and MQPUT1)*

This information shows the contents of the user IDs for normal and alternate user IDs for each type of connection. The number of checks is defined by the RESLEVEL profile. The user ID checked is that used for **MQOPEN**, **MQSUB**, or **MQPUT1** calls.

Note: All user ID fields are checked exactly as they are received. No conversions take place, and, for example, three user ID fields containing "Bob", "BOB", and "bob" are not equivalent.

z/OS *User IDs checked for batch connections*

The user ID checked for a batch connection depends on how the task is run and whether an alternate user ID has been specified.

Table 57. User ID checking against profile name for batch connections

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
No	-	JOB	JOB
Yes	JOB	JOB	ALT

Key:

ALT
Alternate user ID.

JOB

- The user ID of a TSO or z/OS UNIX System Services sign-on.
- The user ID assigned to a batch job.

- The user ID assigned to a started task by the STARTED class or the started procedures table.
- The user ID associated with the executing Db2 stored procedure

A Batch job is performing an MQPUT1 to a queue called Q1 with RESLEVEL set to READ and alternate user ID checking turned off.

Checks made at different RACF(r) access levels for batch connections and User ID checking against profile name for batch connections show that the job user ID is checked against profile hlq.Q1.

z/OS *User IDs checked for CICS connections*

The user IDs checked for CICS connections depend on whether one or two checks are to be carried out, and whether an alternate user ID is specified.

Table 58. User ID checking against profile name for CICS-type user IDs

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
No, 1 check	-	ADS	ADS
No, 2 checks	-	ADS+TXN	ADS+TXN
Yes, 1 check	ADS	ADS	ADS
Yes, 2 checks	ADS+TXN	ADS+TXN	ADS+ALT

Key:

ALT
Alternate user ID

ADS
The user ID associated with the CICS batch job or, if CICS is running as a started task, through the STARTED class or the started procedures table.

TXN
The user ID associated with the CICS transaction. This is normally the user ID of the terminal user who started the transaction. It can be the CICS DFLTUSER, a PRESET security terminal, or a manually signed-on user.

Determine the user IDs checked for the following conditions:

- The RACF access level to the RESLEVEL profile, for a CICS address space user ID, is set to NONE.
- An MQOPEN call is made against a queue with MQOO_OUTPUT and MQOO_PASS_IDENTITY_CONTEXT.

First, see how many CICS user IDs are checked based on the CICS address space user ID access to the RESLEVEL profile. From Table 53 on page 236 in topic “RESLEVEL and CICS connections” on page 236, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 241 on, these checks are carried out:

- The hlq.ALTERNATE.USER.userid profile is not checked.
- The hlq.CONTEXT.queuename profile is checked with both the CICS address space user ID and the CICS transaction user ID.
- The hlq.resourcename profile is checked with both the CICS address space user ID and the CICS transaction user ID.

This means that four security checks are made for this MQOPEN call.

z/OS *User IDs checked for IMS connections*

The user IDs checked for IMS connections depend on whether one or two checks are to be performed, and whether an alternate user ID is specified. If a second user ID is checked, it depends on the type of dependent region and on which user IDs are available.

Table 59. User ID checking against profile name for IMS-type user IDs			
Alternate user ID specified on open?	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
No, 1 check	-	REG	REG
No, 2 checks	-	REG+SEC	REG+SEC
Yes, 1 check	REG	REG	REG
Yes, 2 checks	REG+SEC	REG+SEC	REG+ALT

Key:

ALT
Alternate user ID.

REG
The user ID is normally set through the STARTED class or the started procedures table or, if IMS is running, from a submitted job, by the USER JCL parameter.

SEC
The second user ID is associated with the work being done in a dependent region. It is determined according to Table 60 on page 242.

Table 60. How the second user ID is determined for the IMS connection	
Types of dependent region	**Hierarchy for determining the second user ID**
• BMP message driven and successful GET UNIQUE issued. • IFP and GET UNIQUE issued. • MPP.	User ID associated with the IMS transaction if the user is signed on. LTERM name if available. PSBNAME.
• BMP message driven and successful GET UNIQUE not issued. • BMP not message driven. • IFP and GET UNIQUE not issued.	User ID associated with the IMS dependent region address space if this is not all blanks or all zeros. PSBNAME.

z/OS *User IDs used by the channel initiator*

This collection of topics describes the user IDs used and checked for receiving channels and for client MQI requests issued over server-connection channels. Information is provided for TCP/IP and for LU6.2

You can use the PUTAUT parameter of the receiving channel definition to determine the type of security checking used. To get consistent security checking throughout your IBM MQ network, you can use the ONLYMCA and ALTMCA options.

You can use the DISPLAY CHSTATUS command to determine the user identifier used by the MCA.

z/OS *Receiving channels using TCP/IP*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 61. User IDs checked against profile name for TCP/IP channels			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space of the receiver or requester end is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

Note: The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the receiver or requester is used. This also applies to TCP/IP channels using TLS.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an **MQOPEN** or **MQPUT1** call is issued for the target destination queue.

z/OS *Receiving channels using LU 6.2*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 62. User IDs checked against profile name for LU 6.2 channels			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	CHL	CHL

Table 62. User IDs checked against profile name for LU 6.2 channels (continued)			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)

Requester-server channels
If the channel is started from the requester, there is no opportunity to receive a network user ID (the channel user ID).

If the PUTAUT parameter is set to DEF or CTX on the requester channel, the channel user ID is that of the channel initiator address space of the requester because no user ID is received from the network.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA, the channel user ID is ignored and the MCA user ID of the requester is used.

Other channel types
If the PUTAUT parameter is set to DEF or CTX on the receiver or requester channel, the channel user ID is the user ID received from the communications system when the channel is initiated.

- If the sending channel is on z/OS, the channel user ID received is the channel initiator address space user ID of the sender.
- If the sending channel is on a different platform (for example, AIX), the channel user ID received is typically provided by the USERID parameter of the channel definition.

If the user ID received is blank, or no user ID is received, a channel user ID of blanks is used.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an MQOPEN or MQPUT1 call is issued for the target destination queue.

z/OS *Client MQI requests*

Various user IDs can be used, depending on which user IDs and environment variables have been set. These user IDs are checked against various profiles, depending on the PUTAUT option used and whether an alternate user ID is specified.

This section describes the user IDs checked for client MQI requests issued over server-connection channels for TCP/IP and LU 6.2. The MCA user ID and channel user ID are as for the TCP/IP and LU 6.2 channels described in the previous sections.

For server-connection channels, the user ID received from the client is used if the MCAUSER attribute is blank.

See “クライアントへのアクセス制御” on page 104 for more information.

For client **MQOPEN**, **MQSUB**, and **MQPUT1** requests, use the following rules to determine the profile that is checked:

- If the request specifies alternate-user authority, a check is made against the *hlq*.ALTERNATE.USER. *userid* profile.
- If the request specifies context authority, a check is made against the *hlq*.CONTEXT. *queuename* profile.
- For all **MQOPEN**, **MQSUB**, and **MQPUT1** requests, a check is made against the *hlq.resourcename* profile.

When you have determined which profiles are checked, use the following table to determine which user IDs are checked against these profiles.

Table 63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels

PUTAUT option specified on server-connection channel	Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
DEF, 1 check	No	-	CHL	CHL
DEF, 1 check	Yes	CHL	CHL	CHL
DEF, 2 checks	No	-	CHL + MCA	CHL + MCA
DEF, 2 checks	Yes	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	No	-	MCA	MCA
ONLYMCA, 1 check	Yes	MCA	MCA	MCA
ONLYMCA, 2 checks	No	-	MCA	MCA
ONLYMCA, 2 checks	Yes	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the server-connection; if blank, the channel initiator address space user ID is used.

CHL (Channel user ID)
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

Note: The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the server-connection channel is used. This also applies to TCP/IP channels using TLS.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object or subscription descriptor before an **MQOPEN**, **MQSUB** or **MQPUT1** call is issued on behalf of the client application.

z/OS *Channel initiator example*
An example of how user IDs are checked against RACF profiles.

A user performs an **MQPUT1** operation to a queue on queue manager QM01 that resolves to a queue called QB on queue manager QM02. The message is sent on a TCP/IP channel called QM01.TO.QM02. RESLEVEL is set to NONE, and the open is performed with alternate user ID and context checking. The receiver channel definition has PUTAUT(CTX) and the MCA user ID is set. Which user IDs are used on the receiving channel to put the message to queue QB?

Answer: Table 55 on page 237 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 243 shows that, with PUTAUT set to CTX and 2 checks, the following user IDs are checked:

- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.ALTERNATE.USER.userid profile.
- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.CONTEXT.queuename profile.
- The channel initiator user ID and the alternate user ID specified in the message descriptor (MQMD) are checked against the hlq.Q2 profile.

z/OS *User IDs used by the intra-group queuing agent*
The user IDs that are checked when the intra-group queuing agent opens destination queues are determined by the values of the **IGQAUT** and **IGQUSER** queue manager attributes.

The possible user IDs are:

Intra-group queuing user ID (IGQ)
The user ID determined by the **IGQUSER** attribute of the receiving queue manager. If this is set to blanks, the user ID of the receiving queue manager is used. However, because the receiving queue manager has authority to access all queues defined to it, security checks are not performed for the receiving queue manager's user ID. In this case:

- If only one user ID is to be checked and the user ID is that of the receiving queue manager, no security checks take place. This can occur when **IGQAUT** is set to ONLYIGQ or ALTIGQ.
- If two user IDs are to be checked and one of the user IDs is that of the receiving queue manager, security checks take place for the other user ID only. This can occur when **IGQAUT** is set to DEF, CTX, or ALTIGQ.
- If two user IDs are to be checked and both user IDs are that of the receiving queue manager, no security checks take place. This can occur when **IGQAUT** is set to ONLYIGQ.

Sending queue manager user ID (SND)
The user ID of the queue manager within the queue sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE.

Alternate user ID (ALT)
The user ID specified in the *UserIdentifier* field in the message descriptor of the message.

Table 64. User IDs checked against profile name for intra-group queuing

IGQAUT option specified on receiving queue manager	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
DEF, 1 check	-	SND	SND
DEF, 2 checks	-	SND +IGQ	SND +IGQ
CTX, 1 check	SND	SND	SND
CTX, 2 checks	SND + IGQ	SND +IGQ	SND + ALT
ONLYIGQ, 1 check	-	IGQ	IGQ
ONLYIGQ, 2 checks	-	IGQ	IGQ
ALTIGQ, 1 check	-	IGQ	IGQ
ALTIGQ, 2 checks	IGQ	IGQ	IGQ + ALT

Key:

ALT
Alternate user ID.

IGQ
IGQ user ID.

SND
Sending queue manager user ID.

z/OS *Blank user IDs and UACC levels*

If a blank user ID occurs, a RACF undefined user is signed on. Do not grant wide-ranging access to the undefined user.

Blank user IDs can exist when a user is manipulating messages using context or alternate-user security, or when IBM MQ is passed a blank user ID. For example, a blank user ID is used when a message is written to the system-command input queue without context.

Note: A user ID of " * " (that is, an asterisk character followed by seven spaces) is treated as an undefined user ID.

IBM MQ passes the blank user ID to RACF and a RACF undefined user is signed on. All security checks then use the universal access (UACC) for the relevant profile. Depending on how you have set your access levels, the UACC might give the undefined user a wide-ranging access.

For example, if you issue this RACF command from TSO:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

you define a profile that enables both z/OS-defined user IDs (that have not been put in the access list) and the RACF undefined user ID to put messages on, and get messages from, that queue.

To protect against blank user IDs you must plan your access levels carefully, and limit the number of people who can use context and alternate-user security. You must prevent people using the RACF undefined user ID from getting access to resources that they must not access. However, at the same time, you must allow access to people with defined user IDs. To do this, you can specify a user ID of asterisk (*) in a RACF command PERMIT, giving access to resources for all defined user IDs. Therefore all undefined

user IDs (such as " * ") are denied access. For example, these RACF commands prevent the RACF undefined user ID from gaining access to the queue to put or get messages:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

z/OS z/OS user IDs and Multi-Factor Authentication (MFA)

IBM Multi-Factor Authentication for z/OS allows z/OS security administrators to enhance SAF authentication, by requiring identified users to use multiple authentication factors (for example, both a password and a cryptographic token) to sign on to a z/OS system. IBM MFA also provides support for time-based one time password generation technologies such as RSA SecureId.

For the most part, IBM MQ is unaware of how users have "logged on" to the CICS or batch systems that are driving IBM MQ work, the signed on user ID credential is associated with the z/OS task or address space and IBM MQ uses this for checking authorization to resources. User IDs enabled for MFA can be used for authorization to IBM MQ resources and authentication through pass tickets used with the CICS and IMS bridges.

Important: Special considerations apply however, when using applications, such as the IBM MQ Explorer, which pass a user ID and password credentials on an MQCONNX API call with the *MQCSP_AUTH_USER_ID_AND_PWD* option. IBM MQ has no facility to pass an additional credential on this API request.

Limitations and potential workarounds are described in the following text.

IBM MQ Explorer

The IBM MQ Explorer cannot be used to log on to a z/OS system with a userid for which MFA is enabled because there is no facility for passing a second authentication factor from the IBM MQ Explorer to z/OS.

Additionally, there are two different mechanisms used by the IBM MQ Explorer to re-use a user ID and password credential, that need special attention when one time use passwords are in effect:

1. IBM MQ Explorer has the capability to store passwords in an obfuscated format on the local machine for login at a later time. This capability must be disabled by having explorer prompt for a password each time a connection is made to the z/OS queue manager.

   To do this, use the following procedure:

   a. Select **Queue Managers**.
   b. From the list displayed, choose the queue manager you require and right click that queue manager.
   c. Select **Connection Details** from the menu list that appears.
   d. Select **Properties** from the next menu list and choose the **Userid** tab.

      Ensure that you select the **prompt for password** radio button.
2. Various operations in the IBM MQ Explorer, such as browsing messages on queues, testing subscriptions, and so on, start a new thread which authenticates to IBM MQ using the credential first used at logon. Since the password credential cannot be re-used, you cannot use these operations.

There are two possible workarounds at the MFA configuration level for these issues:

- Use the application ID exclusion of MFA to exclude the IBM MQ tasks from MFA processing altogether.

  To do this, issue the following commands:

  1. `RDEFINE MFADEF MFABYPASS.USERID.`*`chinuser`*

     where *chinuser* is the channel initiator address space level user Id (associated with the channel initiator through the STC class)
  2. `PERMIT MFABYPASS.USERID.`*`chinuser`*` CLASS `*`MFADEF`*` ACCESS(READ) ID(`*`explorer user`*`)`

For more information on this approach, see Bypassing IBM MFA for applications.

- Use Out-of-band support on MFA, which was introduced with IBM MFA 1.2. With this approach, you pre-authenticate to the IBM MFA web server, and in addition to your user ID and password, specify additional authentication as determined through the policy. IBM MFA server generates a cache token credential that you then specify on the IBM MQ Explorer authentication dialogue. The security administrator can allow this credential to be replayed for a reasonable period of time, so enabling normal IBM MQ Explorer use.

For more information on this approach see Introduction to IBM MFA.

z/OS IBM MQ for z/OS security management

IBM MQ uses an in-storage table to hold information relating to each user and the access requests made by each user. To manage this table efficiently and to reduce the number of requests made from IBM MQ to the external security manager (ESM), a number of controls are available.

These controls are available through both the operations and control panels and IBM MQ commands.

z/OS *User ID reverification*

If the RACF definition of a user who is using IBM MQ resources has been changed, for example by connecting the user to a new group, you can tell the queue manager to sign this user on again the next time it tries to access an IBM MQ resource. You can do this by using the IBM MQ command RVERIFY SECURITY.

- User HX0804 is getting and putting messages to the PAYROLL queues on queue manager PRD1. However HX0804 now requires access to some of the PENSION queues on the same queue manager (PRD1).
- The data security administrator connects user HX0804 to the RACF group that allows access to the PENSION queues.
- So that HX0804 can access the PENSION queues immediately (that is, without shutting down queue manager PRD1 or waiting for HX0804 to time out) you must use the IBM MQ command:

```
RVERIFY SECURITY(HX0804)
```

Note: If you turn off user ID timeout for long periods of time (days or even weeks) while the queue manager is running, you must remember to run the RVERIFY SECURITY command for any users that have been revoked or deleted in that time.

z/OS *User ID timeouts*

You can make IBM MQ sign a user off a queue manager after a period of inactivity.

When a user accesses an IBM MQ resource, the queue manager tries to sign this user on to the queue manager (if subsystem security is active). This means that the user is authenticated to the ESM. This user remains signed on to IBM MQ until either the queue manager is shut down, or until the user ID is *timed out* (the authentication lapses) or reverified (reauthenticated).

When a user is timed out, the user ID is *signed off* within the queue manager and any security-related information retained for this user is discarded. The signing on and off of the user within the queue manager is not apparent to the application program or to the user.

Users are eligible for timeout when they have not used any IBM MQ resources for a predetermined amount of time. This time period is set by the MQSC ALTER SECURITY command.

Two values can be specified in the ALTER SECURITY command:

TIMEOUT
The time period in minutes that an unused user ID and its associated resources can remain within the IBM MQ queue manager.

INTERVAL
The time period in minutes between checks for user IDs and their associated resources, to determine whether the *TIMEOUT* has expired.

For example, if the *TIMEOUT* value is 30 and the *INTERVAL* value is 10, every 10 minutes IBM MQ checks user IDs and their associated resources to determine whether any have not been used for 30 minutes. If a timed-out user ID is found, that user ID is signed off within the queue manager. If any timed-out resource information associated with non-timed-out user IDs is found, that resource information is discarded. If you do not want to time out user IDs, set the *INTERVAL* value to zero. However, if the *INTERVAL* value is zero, storage occupied by user IDs and their associated resources is not freed until you issue a **REFRESH SECURITY** or **RVERIFY SECURITY** command.

Tuning this value can be important if you have many one-off users. If you set small interval and timeout values, resources that are no longer required are freed.

Note: If you use values for *INTERVAL* or *TIMEOUT* other than the defaults, you must reenter the command at every queue manager startup. You can do this automatically by putting the **ALTER SECURITY** command in the CSQINP1 data set for that queue manager.

z/OS *Refreshing queue manager security on z/OS*

IBM MQ for z/OS caches RACF data to improve performance. When you change certain security classes, you must refresh this cached information. Refresh security infrequently, for performance reasons. You can also choose to refresh only TLS security information.

When a queue is opened for the first time (or for the first time since a security refresh) IBM MQ performs a RACF check to obtain the user's access rights and places this information in the cache. The cached data includes user IDs and resources on which security checking has been performed. If the queue is opened again by the same user, the presence of the cached data means that IBM MQ does not have to issue RACF checks, which improves performance. The action of a security refresh is to discard any cached security information and so force IBM MQ to make a new check against RACF. Whenever you add, change or delete a RACF resource profile that is held in the MQADMIN, MXADMIN, MQPROC, MXPROC, MQQUEUE, MXQUEUE, MQNLIST, MXNLIST, or MXTOPIC class, you must tell the queue managers that use this class to refresh the security information that they hold. To do this, issue the following commands:

- The RACF SETROPTS RACLIST(classname) REFRESH command to refresh at the RACF level.
- The IBM MQ REFRESH SECURITY command to refresh the security information held by the queue manager. This command needs to be issued by each queue manager that accesses the profiles that have changed. If you have a queue sharing group, you can use the command scope attribute to direct the command to all the queue managers in the group.

  Note: If you have connected a new user to an existing group, you need to run the IBM MQ RVERIFY SECURITY(userid) command. The REFRESH SECURITY(*) command does not let the queue manager sign this user on again, the next time it tries to access an IBM MQ resource.

If you are using generic profiles in any of the IBM MQ classes, you must also issue normal RACF refresh commands if you change, add, or delete any generic profiles. For example, SETROPTS GENERIC(classname) REFRESH.

However, if a RACF resource profile is added, changed or deleted, and the resource to which it applies has not yet been accessed (so no information is cached), IBM MQ uses the new RACF information without a REFRESH SECURITY command being issued.

If RACF auditing is turned on, (for example, by using the RACF RALTER AUDIT(access-attempt (audit_access_level)) command), no caching takes place, and therefore IBM MQ refers directly to the RACF dataspace for every check. Changes are therefore picked up immediately and REFRESH SECURITY is not necessary to access the changes. You can confirm whether RACF auditing is on by using the RACF RLIST command. For example, you could issue the command

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

and receive the results

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

This indicates that auditing is set on. For more information, see the *z/OS Security Server RACF Auditor's Guide* and the *z/OS Security Server RACF Command Language Reference*.

Figure 17 on page 251 summarizes the situations in which security information is cached and in which cached information is used.

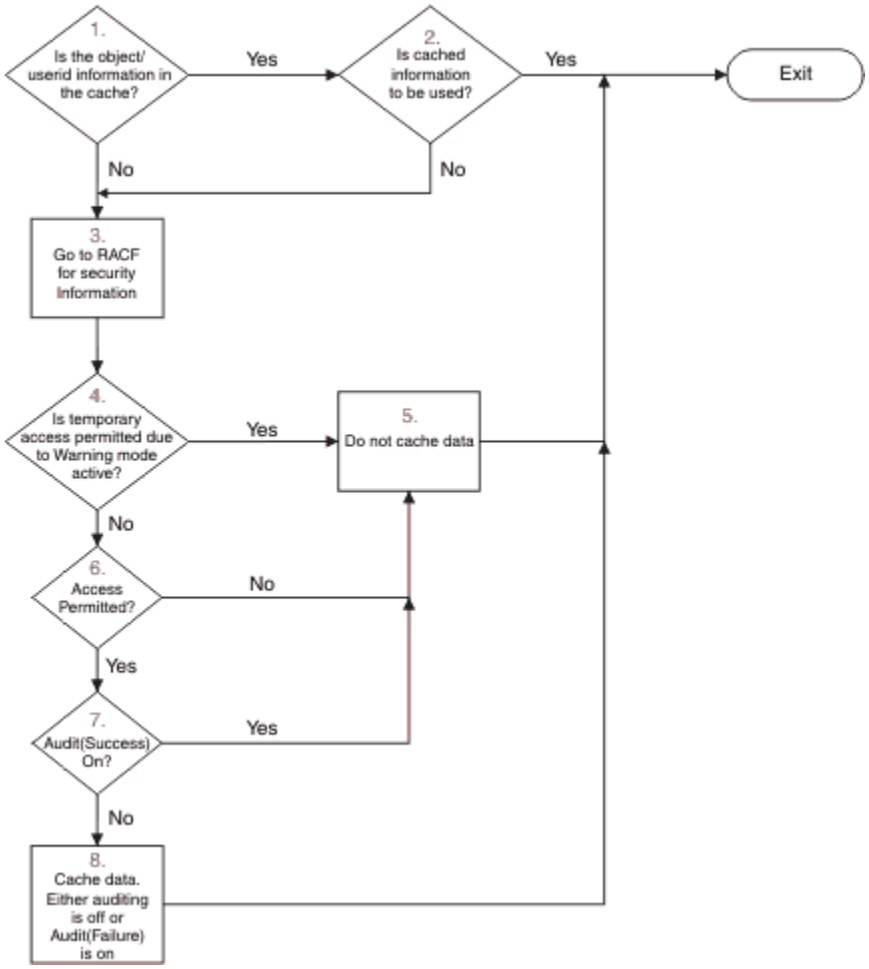


Figure 17. Logic flow for IBM MQ security caching

If you change your security settings by adding or deleting switch profiles in the MQADMIN or MXADMIN classes, use one of these commands to pick up these changes dynamically:

```
REFRESH SECURITY(*)
REFRESH SECURITY(MQADMIN)
REFRESH SECURITY(MXADMIN)
```

This means you can activate new security types, or deactivate them without having to restart the queue manager.

For performance reasons, these are the only classes affected by the REFRESH SECURITY command. You do not need to use REFRESH SECURITY if you change a profile in either the MQCONN or MQCMDS classes.

Note: A refresh of the MQADMIN or MXADMIN class is not required if you change a RESLEVEL security profile.

For performance reasons, use REFRESH SECURITY as infrequently as possible, ideally at off-peak times. You can minimize the number of security refreshes by connecting users to RACF groups that are already in the access list for IBM MQ profiles, rather than putting individual users in the access lists. In this way, you change the user rather than the resource profile. You can also RVERIFY SECURITY the appropriate user instead of refreshing security.

As an example of REFRESH SECURITY, suppose you define the new profiles to protect access to queues starting with INSURANCE.LIFE on queue manager PRMQ. You use these RACF commands:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

You must issue the following command to tell RACF to refresh the security information that it holds, for example:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

Because these profiles are generic, you must tell RACF to refresh the generic profiles for MQQUEUE. For example:

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

Then you must use this command to tell queue manager PRMQ that the queue profiles have changed:

```
REFRESH SECURITY(MQQUEUE)
```

Refreshing SSL/TLS security

To refresh the cached view of the TLS Key Repository, issue the REFRESH SECURITY command with the option TYPE(SSL). This enables you to update some of your TLS settings without having to restart your channel initiator.

z/OS *Displaying security status*

To display the status of the security switches, and other security controls, issue the MQSC DISPLAY SECURITY command.

The following figure shows typical output of the DISPLAY SECURITY ALL command.

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMELIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

Figure 18. Typical output from the DISPLAY SECURITY command

The example shows that the queue manager that replied to the command has subsystem, command, alternate user, process, namelist, and queue security active at queue manager level but not at queue sharing group level. Connection, command resource, and context security are not active. It also shows

that user ID timeouts are active, and that every 12 minutes the queue manager checks for user IDs that have not been used in this queue manager for 54 minutes and removes them.

Note: This command shows the current security status. It does not necessarily reflect the current status of the switch profiles defined to RACF, or the status of the RACF classes. For example, the switch profiles might have been changed since the last restart of this queue manager or REFRESH SECURITY command.

z/OS Security installation tasks for z/OS

After installing and customizing IBM MQ, authorize started task procedures to RACF, authorize access to various resources, and set up RACF definitions. Optionally, configure your system for TLS.

When IBM MQ is first installed and customized, you must perform these security-related tasks:

1. Set up IBM MQ data set and system security by:
   - Authorizing the queue manager started-task procedure xxxxMSTR and the distributed queuing started-task procedure xxxxCHIN to run under RACF.
   - Authorizing access to queue manager data sets.
   - Authorizing access to resources for those user IDs that will use the queue manager and utility programs.
   - Authorizing access for those queue managers that will use the coupling facility list structures.
   - Authorizing access for those queue managers that will use Db2.
2. Set up RACF definitions for IBM MQ security.
3. If you want to use Transport Layer Security (TLS), prepare your system to use certificates and keys.

z/OS *Setting up IBM MQ for z/OS data set security*

There are many types of IBM MQ user. Use RACF to control their access to system data sets.

The possible users of IBM MQ data sets include the following entities:

- The queue manager itself.
- The channel initiator
- IBM MQ administrators, who need to create IBM MQ data sets, run utility programs, and similar tasks.
- Application programmers who need to use the IBM MQ-supplied copybooks, include data sets, macros, and similar resources.
- Applications involving one or more of:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.*

For all these potential users, protect the IBM MQ data sets with RACF.

You must also control access to all your 'CSQINP' data sets.

z/OS *RACF authorization of started-task procedures*

Some IBM MQ data sets are for the exclusive use of the queue manager. If you protect your IBM MQ data sets using RACF, you must also authorize the queue manager started-task procedure xxxxMSTR, and the distributed queuing started-task procedure xxxxCHIN, using RACF. To do this, use the STARTED class. Alternatively, you can use the started procedures table (ICHRIN03), but then you must perform an IPL of your z/OS system before the changes take effect.

For more information, see the *z/OS Security Server RACF System Programmer's Guide*.

The RACF user ID identified must have the required access to the data sets in the started-task procedure. For example, if you associate a queue manager started task procedure called CSQ1MSTR with the RACF user ID QMGRCSQ1, the user ID QMGRCSQ1 must have access to the z/OS resources accessed by the CSQ1 queue manager.

Also, the content of the GROUP field in the user ID of the queue manager must be the same as the content of the GROUP field in the STARTED profile for that queue manager. If the content in each GROUP field does not match then the appropriate user ID is prevented from entering the system. This situation causes IBM MQ to run with an undefined user ID and consequently close due to a security violation.

The RACF user IDs associated with the queue manager and channel initiator started task procedures must not have the TRUSTED attribute set.

z/OS *Authorizing access to data sets*

The IBM MQ data sets should be protected so that no unauthorized user can run a queue manager instance, or gain access to any queue manager data. To do this, use normal z/OS RACF data set protection.

Table 65 on page 254 summarizes the RACF access that the queue manager started task procedure must have to the different data sets.

Table 65. RACF access to data sets associated with a queue manager

RACF access	Data sets
READ	• `thlqual.SCSQAUTH` and `thlqual.SCSQANLx` (where x is the language letter for your national language). • The data sets referred to by CSQINP1, CSQINP2 and CSQXLIB in the queue manager's started task procedure. • SMDS data sets owned by other queue managers in the group. • Log, BSDS and archive log data sets for other queue managers in the group.
UPDATE	• All page sets and log and BSDS data sets. • SMDS data sets owned by a queue manager • SMDS data sets owned by other queue managers in the group, for the structures that the queue manager performs the RECOVER CFSTRUCT command.
ALTER	• All archive log data sets.

Table 66 on page 254 summarizes the RACF access that the started task procedure for distributed queuing must have to the different data sets.

Table 66. RACF access to data sets associated with distributed queuing

RACF access	Data sets
READ	• thlqual.SCSQAUTH, thlqual.SCSQANLx (where x is the language letter for your national language), and thlqual.SCSQMVR1. • LE library data sets. • The data sets referred to by CSQXLIB and CSQINPX in the channel initiator started task procedure.
UPDATE	• Data sets CSQOUTX and CSQSNAP

For more information, see the *z/OS Security Server RACF Security Administrator's Guide*.

z/OS *Encrypting data sets*

The IBM MQ data sets can be encrypted with z/OS data set encryption, so that the data is protected, or for regulatory reasons.

You can protect all page sets, active log, archive log, and bootstrap (BSDS) data sets with z/OS data set encryption.

Attention: You cannot protect shared message data sets (SMDS) with z/OS data set encryption by IBM MQ for z/OS 9.1.4 or earlier.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

z/OS *Setting up IBM MQ for z/OS resource security*

There are many types of IBM MQ user. Use RACF to control their access to IBM MQ resources.

The possible users of IBM MQ resources, such as queues and channels include the following entities:

- The queue manager itself.
- The channel initiator
- IBM MQ administrators, who need to create IBM MQ data sets, run utility programs, and similar tasks
- Application programmers who need to use the IBM MQ-supplied copybooks, include data sets, macros, and similar resources.
- Applications involving one or more of:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.*

For all these potential users, protect the IBM MQ resources with RACF. In particular, note that the channel initiator needs access to various resources, as described in “Security considerations for the channel initiator on z/OS” on page 261, and so the user ID under which it runs must be authorized to access these resources.

If you are using a queue sharing group, the queue manager might issue various commands internally, so the user ID it uses must be authorized to issue such commands. The commands are:

- DEFINE, ALTER, and DELETE for every object that has QSGDISP(GROUP)
- START and STOP CHANNEL for every channel used with CHLDISP(SHARED)

z/OS *Configuring your z/OS system to use TLS*

Use this topic as example of how to configure IBM MQ for z/OS with Transport Layer Security (TLS) using RACF commands.

If you want to use TLS for channel security, there are a number of tasks you need to perform on your system. (For details on using RACF commands for certificates and key repositories (key rings), see Working with TLS on z/OS .)

1. Create a key ring in RACF to hold all the keys and certificates for your system, using the RACF RACDCERT command. For example:

```
RACDCERT ID(CHINUSER) ADDRING(QM1RING)
```

The ID must be either the channel initiator address space user ID or the user ID you want to own the key ring if it is to be a shared key ring.

2. Create a digital certificate for each queue manager, using the RACF RACDCERT command.

   The label of the certificate must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details. In this example it is `ibmWebSphereMQQM1`.

   For example:

```
RACDCERT ID(USERID) GENCERT
SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
WITHLABEL('ibmWebSphereMQQM1')
```

3. Connect the certificate in RACF to the key ring, using the RACF RACDCERT command. For example:

```
RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
CONNECT ID(CHINUSER)
```

   You also need to connect any relevant signer certificates (from a certificate authority) to the key ring. That is, all certificate authorities for the TLS certificate of this queue manager and all certificate authorities for all TLS certificates that this queue manager communicates with. For example:

```
RACDCERT ID(CHINUSER)
CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
```

4. On each of your queue managers, use the IBM MQ ALTER QMGR command to specify the key repository that the queue manager needs to point to. For example, if the key ring is owned by the channel initiator address space:

```
ALTER QMGR SSLKEYR(QM1RING)
```

   or if you are using a shared key ring:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

   where *userid* is the user ID that owns the shared key ring.

5. Certificate Revocation Lists (CRLs) allow the certificate authorities to revoke certificates that can no longer be trusted. CRLs are stored in LDAP servers. To access this list on the LDAP server, you first need to create an AUTHINFO object of AUTHTYPE CRLLDAP, using the IBM MQ DEFINE AUTHINFO command. For example:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

In this example, the certificate revocation list is stored in a public area of the LDAP server, so the LDAPUSER and LDAPPWD fields are not necessary.

Next, put your AUTHINFO object into a namelist, using the IBM MQ DEFINE NAMELIST command. For example:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

Finally, associate the namelist with each queue manager, using the IBM MQ ALTER QMGR command. For example:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. Set up your queue manager to run TLS calls, using the IBM MQ ALTER QMGR command. This defines server subtasks that handle SSL calls only, which leaves the normal dispatchers to continue processing as normal without being affected by any SSL calls. You must have at least two of these subtasks. For example:

   ```
   ALTER QMGR SSLTASKS(8)
   ```

   This change only takes effect when the channel initiator is restarted.

7. Specify the cipher specification to be used for each channel, using the IBM MQ DEFINE CHANNEL or ALTER CHANNEL command. For example:

   ```
   ALTER CHANNEL(LDAPCHL)
   CHLTYPE(SDR)
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
   ```

   Both ends of the channel must specify the same cipher specification.

z/OS Managing channel authentication records in a QSG

Channel authentication records apply to the queue manager that they are created on, they are not shared throughout the queue sharing group (QSG). Therefore if all the queue managers in the queue sharing group are required to have the same rules, some management needs to be carried out to keep all the rules the consistent.

1. Always add the `CMDSCOPE(*)` option to all `SET CHLAUTH` commands. This will send the command to all running queue managers in the queue sharing group
2. Use the `DISPLAY CHLAUTH` command with the `CMDSCOPE(*)` option and then analyze the responses to see if the records are the same from all queue managers. When an inconsistency is found a `SET CHLAUTH` command can be issued containing the same rule with `CMDSCOPE(*)` or `CMDSCOPE(`*qmgr-name*`)`.
3. Add a member to the queue manager's CSQINP2 concatenation (see Initialization commands for details) that has the full set of rules. These will be read as part of the queue manager's initialization process. If the `SET CHLAUTH` command uses `ACTION(ADD)` the rule will only be added if it didn't exist. Using `ACTION(REPLACE)` will replace an existing rule if it already exists or add it if it does not. The same member could then be placed in the CSQINP2 concatenation of all queue managers in the queue sharing group.

4. Use the CSQUTIL utility (see Issuing commands to IBM MQ (COMMAND)) for details) to extract the rules from one queue manager using either the MAKEDEF or MAKEREP option. Then replay the output using CSQUTIL into the target queue manager.

Related concepts

Channel authentication records
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 Auditing considerations on z/OS

The normal RACF auditing controls are available for conducting a security audit of a queue manager. IBM MQ does not gather any security statistics of its own. The only statistics are those that can be created by auditing.

RACF auditing can be based upon:

- User IDs
- Resource classes
- Profiles

For more details, see the *z/OS Security Server RACF Auditor's Guide*.

Note: Auditing degrades performance; the more auditing you implement, the more performance is degraded. This is also a consideration for the use of the RACF WARNING option.

z/OS *Auditing RESLEVEL*

Use the RESAUDIT system parameter to control the production of RESLEVEL audit records. RACF GENERAL audit records are produced.

Produce RESLEVEL audit records by setting the RESAUDIT system parameter to YES. If the RESAUDIT parameter is set to NO, audit records are not produced. For more details about setting this parameter, see Using CSQ6SYSP.

If RESAUDIT is set to YES, no normal RACF audit records are taken when the RESLEVEL check is made to see what access an address space user ID has to the hlq.RESLEVEL profile. Instead, IBM MQ requests that RACF create a GENERAL audit record (event number 27). These checks are only carried out at connect time, so the performance cost is minimal.

Attention: RACFRW is no longer the suggested utility for processing RACF audit records. You should use the RACF SMF data unload utility as this is the preferred reporting method.

You can report the IBM MQ general audit records using the RACF report writer (RACFRW). You could use the following RACFRW commands to report the RESLEVEL access:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

A sample report from RACFRW, excluding the *Date*, *Time*, and *SYSID* fields, is shown in Figure 19 on page 259.

```
       RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE   4
                                E
                                V  Q
                                E  U
*JOB/USER *STEP/  --TERMINAL--  N  A
   NAME    GROUP     ID    LVL  T  L

 WS21B    MQMGRP IGJZM000   0   27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
    TRUSTED USER                      AUTH=(NONE),REASON=(NONE)
                                SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST
PROFILE(QM66.RESLEVEL),
                                CLASS(MQADMIN), ACCESS EQUATES TO
(CONTROL)',RESULT=SUCCESS,MQADMIN
```

Figure 19. Sample output from RACFRW showing RESLEVEL general audit records

From checking the LOGSTR data in this sample output, you can see that TSO user WS21B has CONTROL access to QM66.RESLEVEL. This means that all resource security checks are bypassed when user WS21B access QM66 resources.

For more information about using RACFRW, see The RACF report writer in the *z/OS Security Server RACF Auditor's Guide*.

z/OS Customizing security

If you want to change the way IBM MQ security operates, you must do this through the SAF exit (ICHRFR00), or exits in your external security manager.

To find out more about RACF exits, see the *z/OS Security Server RACROUTE Macro Reference* documentation.

Note: Because IBM MQ optimizes calls to the ESM, RACROUTE requests might not be made on, for example, every open for a particular queue by a particular user.

z/OS Security violation messages on z/OS

A security violation is indicated by the return code MQRC_NOT_AUTHORIZED in an application program or by a message in the job log.

A return code of MQRC_NOT_AUTHORIZED can be returned to an application program for the following reasons:

- A user is not allowed to connect to the queue manager. In this case, you get an ICH408I message in the Batch/TSO, CICS, or IMS job log.
- A user sign-on to the queue manager has failed because, for example, the job user ID is not valid or appropriate, or the task user ID or alternate user ID is not valid. One or more of these user IDs might not be valid because they have been revoked or deleted. In this case, you get an ICHxxxx message and possibly an IRRxxxx message in the queue manager job log giving the reason for the sign-on failure. For example:

  ```
  ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
    LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
  IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
  ```

- An alternate user has been requested, but the job or task user ID does not have access to the alternate user ID. For this failure, you get a violation message in the job log of the relevant queue manager.
- A context option has been used or is implied by opening a transmission queue for output, but the job user ID or, where applicable, the task or alternate user ID does not have access to the context option. In this case, a violation message is put in the job log of the relevant queue manager.

- An unauthorized user has attempted to access a secured queue manager object, for example, a queue. In this case, an ICH408I message for the violation is put in the job log of the relevant queue manager. This violation might be due to the job or, when applicable, the task or alternate user ID.

Violation messages for command security and command resource security can also be found in the job log of the queue manager.

If the ICH408I violation message shows the queue manager jobname rather than a user ID, this is normally the result of a blank alternate user ID being specified. For example:

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

You can find out who is allowed to use blank alternate user IDs by checking the access list of the MQADMIN profile hlq.ALTERNATE.USER.-BLANK-.

An ICH408I violation message can also be generated by:

- A command being sent to the system-command input queue without context. User-written programs that write to the system-command input queue should always use a context option. For more information, see “Profiles for context security” on page 219.
- When the job accessing the IBM MQ resource does not have a user ID associated with it, or when an IBM MQ adapter cannot extract the user ID from the adapter environment.

Violation messages might also be issued if you are using both queue sharing group and queue manager level security. You might get messages indicating that no profile has been found at queue manager level, but still be granted access because of a queue sharing group level profile.

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

See the z/OS for Security Server RACF Messages and Codes documentation for more information on ICH408I messages.

z/OS What to do if access is allowed or disallowed incorrectly

In addition to the information detailed in the z/OS documentation, use this checklist if access to a resource appears to be incorrectly controlled.

See the *z/OS Security Server RACF Security Administrator's Guide* for the detailed steps if access is allowed or disallowed.

- Are the switch profiles correctly set?
  - Is RACF active?
  - Are the IBM MQ RACF classes installed and active?

    Use the RACF command, SETROPTS LIST, to check this.
  - Use the IBM MQ DISPLAY SECURITY command to display the current switch status from the queue manager.
  - Check the switch profiles in the MQADMIN class.

    Use the RACF commands, SEARCH and RLIST, for this.
  - Recheck the RACF switch profiles by issuing the IBM MQ REFRESH SECURITY(MQADMIN) command.

- Has the RACF resource profile changed? For example, has universal access on the profile changed or has the access list of the profile changed?
  - Is the profile generic?

    If it is, issue the RACF command, SETROPTS GENERIC(classname) REFRESH.
  - Have you refreshed the security on this queue manager?

    If required, issue the RACF command SETROPTS RACLIST(classname) REFRESH.

    If required, issue the IBM MQ REFRESH SECURITY(*) command.
- Has the RACF definition of the user changed? For example, has the user been connected to a new group or has the user access authority been revoked?
  - Have you reverified the user by issuing the IBM MQ RVERIFY SECURITY(userid) command?
- Are security checks being bypassed due to RESLEVEL?
  - Check the connecting user ID's access to the RESLEVEL profile. Use the RACF audit records to determine what the RESLEVEL is set to.
  - For channels, remember that the access level that the channel initiator's userid has to RESLEVEL is inherited by all channels, so an access level, such as ALTER, that causes all checks to be bypassed causes security checks to be bypassed for all channels.
  - If you are running from CICS, check the transaction's RESSEC setting.
  - If RESLEVEL has been changed while a user is connected, they must disconnect and reconnect before the new RESLEVEL setting takes effect.
- Are you using queue sharing groups?
  - If you are using both queue sharing group and queue manager level security, check that you have defined all the correct profiles. If queue manager profile is not defined, a message is sent to the log stating that the profile was not found.
  - Have you used a combination of switch settings that is not valid so that full security checking has been set on?
  - Do you need to define security switches to override some of the queue sharing group settings for your queue manager?
  - Is a queue manager level profile taking precedence over a queue sharing group level profile?

z/OS Security considerations for the channel initiator on z/OS

If you are using resource security in a distributed queuing environment, the Channel initiator address space needs appropriate access to various IBM MQ resources. You can use the Integrated Cryptographic Support Facility (ICSF) to seed the password protection algorithm.

See the z/OS Cryptographic Services documentation for more information on ICSF.

Using resource security

If you are using resource security, consider the following points if you are using distributed queuing:

System queues
The channel initiator address space needs RACF UPDATE access to the system queues listed at “System queue security” on page 209, and to all the user destination queues and the dead-letter queue (but see “Dead-letter queue security” on page 208).

Transmission queues
The channel initiator address space needs ALTER access to all the user transmission queues.

Context security
The channel user ID (and the MCA user ID if one has been specified) need RACF CONTROL access to the hlq.CONTEXT.queuename profiles in the MQADMIN class. Depending on the RESLEVEL profile, the channel user ID might also need CONTROL access to these profiles.

All channels need CONTROL access to the MQADMIN hlq.CONTEXT. dead-letter-queue profile. All channels (whether initiating or responding) can generate reports, and consequently they need CONTROL access to the hlq.CONTEXT.reply-q profile.

SENDER, CLUSSDR, and SERVER channels need CONTROL access to the hlq.CONTEXT.xmit-queue-name profiles since messages can be put onto the transmission queue to wake up the channel to end gracefully.

Note: If the channel user ID, or a RACF group to which the channel user ID is connected, has CONTROL or ALTER access to the hlq.RESLEVEL, then there are no resource checks for the channel initiator or any of its channels.

See “Profiles for context security” on page 219 “RESLEVEL and the channel initiator connection” on page 237 and “User IDs for security checking on z/OS” on page 239 for more information.

CSQINPX
If you are using the CSQINPX input data set, the channel initiator also needs READ access to CSQINPX, and UPDATE access to data set CSQOUTX and dynamic queues SYSTEM.CSQXCMD.*.

Connection security
The channel initiator address space connection requests use a connection type of CHIN, for which appropriate access security must be set, see “Connection security profiles for the channel initiator” on page 203.

Data sets
The channel initiator address space needs appropriate access to queue manager data sets, see “Authorizing access to data sets” on page 254.

Commands
The distributed queuing commands (for example, DEFINE CHANNEL, START CHINIT, START LISTENER, and other channel commands) must have appropriate command security set, see Table 49 on page 222.

If you are using a queue sharing group, the channel initiator might issue various commands internally, so the user ID it uses must be authorized to issue such commands. These commands are START and STOP CHANNEL for every channel used with CHLDISP(SHARED).

If the PSMODE of the queue manager is not DISABLED, the channel initiator must have READ access to the DISPLAY PUBSUB command.

Channel security
Channels, particularly receivers and server-connections, need appropriate security to be set up; see “User IDs for security checking on z/OS” on page 239 for more information.

You can also use the Transport Layer Security (TLS) protocol to provide security on channels. See “IBM MQ での TLS セキュリティー・プロトコル” on page 24 for more information about using TLS with IBM MQ.

See also “クライアントへのアクセス制御” on page 104 for information about server-connection security.

User IDs
The user IDs described in “User IDs used by the channel initiator” on page 242 and “User IDs used by the intra-group queuing agent” on page 246 need the following access:

- RACF UPDATE access to the appropriate destination queues and the dead-letter queue
- RACF CONTROL access to the `hlq.CONTEXT.queuename` profile if context checking is performed at the receiver
- Appropriate access to the hlq.ALTERNATE.USER.userid profiles they might need to use.
- For clients, the appropriate RACF access to the resources to be used.

APPC security

Set appropriate APPC security if you are using the LU 6.2 transmission protocol. (Use the APPCLU RACF class for example.) For information about setting up security for APPC, see the following documentation:

- *z/OS MVS Planning: APPC Management*
- *z/OS MVS Programming: Writing Servers for APPC/MVS*

Outbound transmissions use the "SECURITY(SAME)" APPC option. As a result, the user ID of the channel initiator address space and its default profile (RACF GROUP) are flowed across the network to the receiver with an indicator that the user ID has already been verified (ALREADYV).

If the receiving side is also z/OS, the user ID and profile are verified by APPC and the user ID is presented to the receiver channel and used as the channel user ID.

In an environment where the queue manager is using APPC to communicate with another queue manager on the same or another z/OS system, you need to ensure that either:

- The VTAM definition for the communicating LU specifies SETACPT(ALREADYV)
- There is a RACF APPCLU profile for the connection between LUs that specifies CONVSEC(ALREADYV)

Changing security settings

If the RACF access level that either the channel user ID or MCA user ID has to a destination queue is changed, this change takes effect only for new object handles (that is, new MQOPEN s) for the destination queue. The times when MCAs open and close queues is variable; if a channel is already running when such an access change is made, the MCA can continue to put messages on the destination queue using the existing security access of the user IDs rather than the updated security access. Stopping and restarting the channels to enforce the updated access level avoids this scenario.

Automatic restart

If you are using the z/OS Automatic Restart Manager (ARM) to restart the channel initiator, the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ START CHINIT command.

Using the Integrated Cryptographic Service Facility (ICSF)

The channel initiator can use ICSF to generate a random number when seeding the password protection algorithm to obfuscate passwords flowing over client channels if TLS is not being used. The process of generating a random number is called *entropy*.

If you have the z/OS feature installed but have not started ICSF, you see message CSQX213E and the channel initiator uses STCK for entropy.

Message CSQX213E warns you that the password protection algorithm is not as secure as it could be. However, you can continue your process; there is no other impact on runtime.

If you do not have the z/OS feature installed, the channel initiator automatically uses STCK.

Notes:

1. Using ICSF for entropy generates more random sequences than using STCK.
2. If you start ICSF you must restart the channel initiator.
3. ICSF is required for certain CipherSpecs. If you attempt to use one of these CipherSpecs and you do not have ICSF installed, you receive message CSQX629E.

z/OS Security in queue manager clusters on z/OS

Security considerations for clusters are the same for queue managers and channels that are not clustered. The channel initiator needs access to some additional system queues, and some additional commands need appropriate security set.

You can use the MCA user ID, channel authentication records, TLS, and security exits to authenticate cluster channels (as with conventional channels). The channel authentication records or security exit relating to the cluster-receiver channel must check that the remote queue manager is permitted access to the server queue manager's cluster queues. You can start to use IBM MQ cluster support without changing your existing queue access security. You must, however, allow other queue managers in the cluster to write to the SYSTEM.CLUSTER.COMMAND.QUEUE if they are to join the cluster.

IBM MQ cluster support does not provide a mechanism to limit a member of a cluster to the client role only. As a result, you must be sure that you trust any queue managers that you allow into the cluster. If any queue manager in the cluster creates a queue with a particular name, it can receive messages for that queue, regardless of whether the application putting messages to that queue intended this or not.

To restrict the membership of a cluster, take the same action that you would take to prevent queue managers connecting to receiver channels. You restrict the membership of a cluster by using channel authentication records or by writing a security exit program on the receiver channel. You can also write an exit program to prevent unauthorized queue managers from writing to the SYSTEM.CLUSTER.COMMAND.QUEUE.

Note: It is not advisable to permit applications to open the SYSTEM.CLUSTER.TRANSMIT.QUEUE directly. It is also not advisable to permit an application to open any other transmission queue directly.

If you are using resource security, consider the following points in addition to the considerations contained in "Security considerations for the channel initiator on z/OS" on page 261:

System queues
The channel initiator needs RACF ALTER access to the following system queues:

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

and UPDATE access to SYSTEM.CLUSTER.REPOSITORY.QUEUE

It also needs READ access to any namelists used for clustering.

Commands
Set appropriate command security (as described in Table 49 on page 222) for the cluster support commands (REFRESH and RESET CLUSTER, SUSPEND, and RESUME QMGR.

z/OS Security considerations for using IBM MQ with CICS

All the CICS versions supported by IBM MQ 9.0.0, and later, use the CICS supplied version of the adapter and bridge.

For details of security considerations, see:

- Security for the CICS-MQ adapter.
- Security for the CICS-MQ bridge.

z/OS Security considerations for using IBM MQ with IMS

Use this topic to plan your security requirements when you use IBM MQ with IMS.

Using the OPERCMDS class

If you are using RACF to protect resources in the OPERCMDS class, ensure that the userid associated with your IBM MQ queue manager address space has authority to issue the MODIFY command to any IMS system to which it can connect.

Security considerations for the IMS bridge

There are four aspects that you must consider when deciding your security requirements for the IMS bridge, these are:

- What security authorization is needed to connect IBM MQ to IMS
- How much security checking is performed on applications using the bridge to access IMS
- Which IMS resources these applications are allowed to use
- What authority is to be used for messages that are put and got by the bridge

When you define your security requirements for the IMS bridge you must consider the following:

- Messages passing across the bridge might have originated from applications on platforms that do not offer strong security features
- Messages passing across the bridge might have originated from applications that are not controlled by the same enterprise or organization

z/OS *Security considerations for connecting to IMS*

Grant the user ID of the IBM MQ queue manager address space access to the OTMA group.

The IMS bridge is an OTMA client. The connection to IMS operates under the user ID of the IBM MQ queue manager address space. This is normally defined as a member of the started task group. This user ID must be granted access to the OTMA group (unless the /SECURE OTMA setting is NONE).

To do this, define the following profile in the FACILITY class:

```
IMSXCF.xcfgname.mqxcfmname
```

Where `xcfgname` is the XCF group name and `mqxcfmname` is the XCF member name of IBM MQ.

You must give your IBM MQ queue manager user ID read access to this profile.

Note:

1. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
2. If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, no user ID is passed to IMS and the connection fails unless the /SECURE OTMA setting is NONE.

z/OS *Application access control for the IMS bridge*

Define a RACF profile in the FACILITY class for each IMS system. Grant an appropriate level of access to the IBM MQ queue manager user ID.

For each IMS system that the IMS bridge connects to, you can define the following RACF profile in the FACILITY class to determine how much security checking is performed for each message passed to the IMS system.

```
IMSXCF.xcfgname.imsxcfmname
```

Where `xcfgname` is the XCF group name and `imsxcfmname` is the XCF member name for IMS. (You need to define a separate profile for each IMS system.)

The access level you allow for the IBM MQ queue manager user ID in this profile is returned to IBM MQ when the IMS bridge connects to IMS, and indicates the level of security that is required on subsequent transactions. For subsequent transactions, IBM MQ requests the appropriate services from RACF and, where the user ID is authorized, passes the message to IMS.

OTMA does not support the IMS /SIGN command; however, IBM MQ allows you to set the access checking for each message to enable implementation of the necessary level of control.

The following access level information can be returned:

NONE or NO PROFILE FOUND

These values indicate that maximum security is required, that is, authentication is required for every transaction. A check is made to verify that the user ID specified in the *UserIdentifier* field of the MQMD structure, and the password or PassTicket in the *Authenticator* field of the MQIIH structure are known to RACF, and are a valid combination. A UTOKEN is created with a password or PassTicket, and passed to IMS ; the UTOKEN is not cached.

Note: If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, this level of security overrides whatever is defined in the profile.

READ

This value indicates that the same authentication is to be performed as for NONE under the following circumstances:

- The first time that a specific user ID is encountered
- When the user ID has been encountered before but the cached UTOKEN was not created with a password or PassTicket

IBM MQ requests a UTOKEN if required, and passes it to IMS.

Note: If a request to reverify security has been acted on, all cached information is lost and a UTOKEN is requested the first time each user ID is later encountered.

UPDATE

A check is made that the user ID in the *UserIdentifier* field of the MQMD structure is known to RACF.

A UTOKEN is built and passed to IMS ; the UTOKEN is cached.

CONTROL/ALTER

These values indicate that no security UTOKENs need to be provided for any user IDs for this IMS system. (You would probably only use this option for development and test systems.)

Attention: Note that the user ID contained in the *UserIdentifier* field of the MQMD structure is still passed for **CONTROL/ALTER**.

Note:

1. This access is defined when IBM MQ connects to IMS, and lasts for the duration of the connection. To change the security level, the access to the security profile must be changed and then the bridge stopped and restarted (for example, by stopping and restarting OTMA).
2. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
3. You can use a password or a PassTicket, but you must remember that the IMS bridge does not encrypt data. For information about using PassTickets, see “Using RACF PassTickets in the IMS header” on page 267.
4. Some of these results might be affected by security settings in IMS, using the /SECURE OTMA command.
5. Cached UTOKEN information is held for the duration defined by the INTERVAL and TIMEOUT parameters of the IBM MQ ALTER SECURITY command.
6. The RACF WARNING option has no effect on the IMSXCF.xcfgname.imsxcfmname profile. Its use does not affect the level of access granted, and no RACF WARNING messages are produced.

z/OS *Security checking on IMS*

Messages that pass across the bridge contain security information. The security checks made depend on the setting of the IMS command /SECURE OTMA.

Each IBM MQ message that passes across the bridge contains the following security information:

- A user ID contained in the *UserIdentifier* field of the MQMD structure
- The security scope contained in the *SecurityScope* field of the MQIIH structure (if the MQIIH structure is present)

- A UTOKEN (unless the IBM MQ sub system has CONTROL or ALTER access to the relevant `IMSXCF.xcfgname.imsxcfmname` profile)

The security checks made depend on the setting of the IMS command /SECURE OTMA, as follows:

/SECURE OTMA NONE
No security checks are made for the transaction.

/SECURE OTMA CHECK
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE (Accessor Environment Element) is built in the IMS control region.

/SECURE OTMA FULL
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE is built in the IMS dependent region as well as the IMS control region.

/SECURE OTMA PROFILE
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking

The *SecurityScope* field in the MQIIH structure is used to determine whether to build an ACEE in the IMS dependent region as well as the control region.

Note:

1. If you change the authorities in the TIMS or CIMS class, or the associated group classes GIMS or DIMS, you must issue the following IMS commands to activate the changes:
   - /MODIFY PREPARE RACF
   - /MODIFY COMMIT
2. If you do not use /SECURE OTMA PROFILE, any value specified in the **SecurityScope** field of the MQIIH structure is ignored.

z/OS *Security checking done by the IMS bridge*

Different authorities are used depending on the action being performed.

When the bridge puts or gets a message, the following authorities are used:

Getting a message from the bridge queue
No security checks are performed.

Putting an exception, or COA report message
Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure.

Putting a reply message
Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure of the original message

Putting a message to the dead-letter queue
No security checks are performed.

Note:

1. If you change the IBM MQ class profiles, you must issue the IBM MQ REFRESH SECURITY(*) command to activate the changes.
2. If you change the authority of a user, you must issue the MQSC RVERIFY SECURITY command to activate the change.

z/OS *Using RACF PassTickets in the IMS header*

You can use a PassTicket in place of a password in the IMS header.

If you want to use a PassTicket instead of a password in the IMS header (MQIIH), specify the application name against which the PassTicket is validated in the PASSTKTA attribute of the STGCLASS definition of the IMS bridge queue to which the message is to be routed.

If the PASSTKTA value is left blank, you must arrange to have a PassTicket generated. The application name in this case must be of the form MVSxxxx, where xxxx is the SMFID of the z/OS system on which the target queue manager runs.

A PassTicket is built from a user ID, the target application name, and a secret key. It is an 8-byte value containing uppercase alphabetic and numeric characters. It can be used only once, and is valid for a 20 minute period. If a PassTicket is generated by a local RACF system, RACF only checks that the profile exists and not that the user has authority against the profile. If the PassTicket was generated on a remote system, RACF validates the access of the user ID to the profile. For full information about PassTickets, see the *z/OS Security Server RACF Security Administrator's Guide*.

PassTickets in IMS headers are given to RACF by IBM MQ, not IMS.

z/OS Migrating a z/OS queue manager to mixed-case security

Follow these steps to migrate a queue manager to mixed-case security. You review the level of security product you are using and activate the new IBM MQ external security manager classes. Run the **REFRESH SECURITY** command to activate the mixed-case profiles.

Before you begin

1. Ensure all IBM MQ external security manager classes are activated.
2. Ensure your queue manager is started.

About this task

Follow these steps to convert a queue manager to mixed-case security.

Procedure

1. Copy all your existing profiles and access levels from the uppercase classes to the equivalent mixed-case external security manager class.
   a) `MQADMIN` to `MXADMIN`.
   b) `MQPROC` to `MXPROC`.
   c) `MQNLIST` to `MXNLIST`.
   d) `MQQUEUE` to `MXQUEUE`.
2. Change the value of the `SCYCASE` queue manager attribute to `MIXED` by issuing the following command.

```
ALTER QMGR SCYCASE(MIXED)
```

3. Activate the security profiles by issuing the following command.

```
REFRESH SECURITY(*) TYPE(CLASSES)
```

4. Test that your security profiles are working correctly.

What to do next

Review your object definitions and create new mixed-case profiles as appropriate, using the **REFRESH SECURITY** command as required to activate the profiles.

IBM MQ MQI client・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。

  認証は、以下の 2 つのオプションのいずれかによって提供されます。

  – 接続認証フィーチャー。

    接続認証の詳細については、72 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、275 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、352 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、387 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、51 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。

注 : AIX, Linux, and Windows では、 IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。 このモジュールの証明書は「履歴」ステータスに移動されました。 お客様は、IBM Crypto for C (ICC) 証明書 を表示し、NIST から提供されたアドバイスに注意する必要があります。 交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、 「NIST CMVP modules in process list」でそのモジュールを検索します。

IBM MQ Operator 3.2.0 およびキュー・マネージャー・コンテナー・イメージ 9.4.0.0 以降は、UBI 9 に基づいています。 FIPS 140-3 準拠は現在保留中であり、その状況を表示するには、 「NIST CMVP modules in process list」で「Red Hat Enterprise Linux 9- OpenSSL FIPS Provider」を検索します。

実行時に FIPS 準拠にするには、 `-fips` オプションを指定した **runmqakm** などの FIPS 準拠ソフトウェアのみを使用して鍵リポジトリーを作成および管理しておく必要があります。

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 に設定します。
2. 環境変数 **MQSSLFIPS** を YES に設定します。
3. クライアント構成ファイルの SSL スタンザの **SSLFipsRequired** 属性を YES に設定します。

デフォルトでは、FIPS 認定の暗号方式は必要ありません。

これらの値は、 **ALTER QMGR SSLFIPS** 上の同等のパラメーター値と同じ意味を持ちます (**ALTER QMGR** (キュー・マネージャー設定の変更)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。

可能な場合、FIPS のみの CipherSpecs が構成されていると、MQI クライアントは、MQRC_SSL_INITIALIZATION_ERROR を使用して非 FIPS CipherSpec を指定する接続を拒否します。 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

35 ページの『AIX, Linux, and Windows での連邦情報処理標準 (FIPS)』
AIX, Linux, and Windows 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)という暗号化パッケージを使用します。 AIX, Linux, and Windows プラットフォームでは、 ICC ソフトウェアは、米国連邦情報・技術局の連邦情報処理標準 (FIPS) 暗号モジュール評価プログラム (レベル 140-2) に合格しています。

AIX AIX 上の GSKit 8.0 の複数のインストール済み環境での TLS クライアント・アプリケーションの実行

AIX 上の TLS クライアント・アプリケーションを複数の IBM Global Security Kit (GSKit) 8.0 インストール済み環境の AIX システムで実行すると、 MQRC_CHANNEL_CONFIG_ERROR およびエラー AMQ6175 が発生する可能性があります。

複数の GSKit 8.0 インストール済み環境がある AIX システムでクライアント・アプリケーションを実行している場合、クライアント接続呼び出しは、TLS の使用時に MQRC_CHANNEL_CONFIG_ERROR を返すことができます。 `/var/mqm/errors` ログには、失敗したクライアント・アプリケーションのエラー AMQ6175 および AMQ9220 が記録されます。以下に例を示します。

```
09/08/11 11:16:13 - Process(24412.1) User(user) Program(example)
Host(machine.example.ibm.com) Installation(Installation1)
VRMF(7.1.0.0)
AMQ6175: The system could not dynamically load the shared library
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so'. The system returned
error number '8' and error message 'Symbol resolution failed
for /usr/mqm/gskit8/lib64/libgsk8ssl_64.so because:
Symbol VALUE_EC_NamedCurve_secp256r1__9GSKASNOID (number 16) is not
exported from dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_NamedCurve_secp384r1__9GSKASNOID (number 17) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
```

```
Symbol VALUE_EC_NamedCurve_secp521r1__9GSKASNOID (number 18) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecPublicKey__9GSKASNOID (number 19) is not exported from dependent
module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecdsa_with_SHA1__9GSKASNOID (number 20) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecdsa__9GSKASNOID (number 21) is not exported from dependent
module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.'.

EXPLANATION:
This message applies to AIX systems. The shared library
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so' failed
to load correctly due to a problem with the library.
ACTION:
Check the file access permissions and that the file has not been corrupted.
----- amqxufnx.c : 1284 ---------------------------------------------------------
09/08/11 11:16:13 - Process(24412.1) User(user) Program(example)
Host(machine.example.ibm.com) Installation(Installation1)
VRMF(7.1.0.0)
AMQ9220: The GSKit communications program could not be loaded.

EXPLANATION:
The attempt to load the GSKit library or procedure
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so' failed with error code
536895861.
ACTION:
Either the library must be installed on the system or the environment changed
to allow the program to locate it.
----- amqcgska.c : 836 ----------------------------------------------------------
```

このエラーの一般的な原因は、 LIBPATH 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 8.0 インストール済み環境から混合セットのライブラリーをロードしたことです。 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。

MQSC を使用した TLS チャネルの構成

TLS チャネルを構成するには、 **runmqsc** および ALTER CHANNEL コマンドを使用します。 オプションで、チャネルを構成して、指定された値と一致する所有者の識別名の属性を持つ証明書のみを受け入れることができます。 オプションで、キュー・マネージャーのチャネルを構成できます。これにより、開始する相手先が独自の個人証明書を送信しない場合、キュー・マネージャーは接続を拒否できます。

このタスクについて

IBM MQ Explorer でチャネルを構成するには、 IBM MQ Explorer を使用した TLS チャネルの構成を参照してください。

runmqsc を使用してチャネルを構成するには、以下の手順を実行します。

手順

1. ターゲット・キュー・マネージャーに接続する **runmqsc** コマンドを呼び出します。
2. TLS を有効にするチャネルを識別します。
   チャネル名とチャネル・タイプの両方に注意してください。
3. ALTER CHANNEL コマンドを使用して、 IBM MQ チャネルのさまざまなプロパティーを変更します。

コマンドに加えて、チャネル名とチャネル・タイプを指定します。 例えば、 MQ.TEST 次のコマンドを実行します。

```
ALTER CHANNEL(‘MQ.TEST’) CHLTYPE(SDR)
```

IBM MQ チャネル定義で調整できる、TLS に関連したさまざまなチャネル属性があります。

次のタスク

メッセージ・セキュリティーの設定

TLS 対応メッセージングによって、メッセージ・セキュリティーを保証する 2 つの方式が提供されます。

- 暗号化により、メッセージが傍受された場合にそのメッセージが判読不能であることが保証されます。
- ハッシュ機能により、メッセージが変更された場合に検出されることが保証されます。

これらの方式の組み合わせは、暗号仕様または CipherSpec と呼ばれます。 チャネルの両端には同じ CipherSpec を設定する必要があります。設定していない場合、TLS 対応メッセージングは失敗します。 詳細については、7 ページの『IBM MQ の保護』を参照してください。

IBM MQ チャネル使用可能 TLS を変更するには、SSLCIPH 属性に値を指定します。 この属性は、リスト420 ページの『CipherSpecs の有効化』から、キュー・マネージャーのキュー・プラットフォームに対して有効な CipherSpec に設定する必要があります。

TLS を無効にするように IBM MQ チャネルを変更するには、SSLCIPH をブランク値に設定します。 以下に例を示します。

```
ALTER CHANNEL(‘MQ.TEST’) CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

注 : 文字の大/小文字が維持されるようにするには、チャネル名を単一引用符で囲む必要があります。 単一引用符がない場合、 IBM MQ はストリングをすべて大文字に変換します。

所有者の名前での証明書のフィルタリング

証明書には、証明書の所有者の識別名が含まれています。 オプションで、チャネルを構成して、指定された値と一致する所有者の識別名の属性を持つ証明書のみを受け入れることが できます。

IBM MQ がフィルター処理する属性名は、下の表にリストされています。

属性名	意味
SERIALNUMBER	証明書のシリアル番号
MAIL	E メール・アドレス
Deprecated 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	タイトル
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

属性名	意味
UNSTRUCTUREDNAME	ホスト名
UNSTRUCTUREDADDRESS	IP アドレス
DNQ	識別名修飾子

任意の数の文字の代わりに、属性値の先頭または末尾にワイルドカード文字 (*) を使用することができます。 例えば、GB の IBM に勤務する、Smith で終わる名前を持つ人からの証明書のみを受け入れるには、次のように入力します。

```
CN=*Smith, O=IBM, C=GB
```

以下に例を示します。

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

注 : 文字の大/小文字を維持するために、SSLPEER ストリングを単一引用符で囲む必要があります。 単一引用符がない場合、 IBM MQ はストリングをすべて大文字に変換します。

キュー・マネージャーへの接続を開始する相手先の認証

別の通話者がキュー・マネージャーへの TLS 対応接続を開始する場合、キュー・マネージャーは、ID の証明として個人証明書を開始する相手先に送信する必要があります。 オプションで、キュー・マネージャーのチャネルを構成できます。 これにより、開始する相手先が独自の個人証明書を送信しない場合、キュー・マネージャーは接続を拒否できます。

これを行うには、SSLCAUTH 属性を設定します。 この属性はブール属性であり、値 OPTIONAL または REQUIRED を持つことができます。

- OPTIONAL は、接続中のクライアントの証明書が提供されているが、クライアントがそれを送信する必要がない場合に、その証明書を認証します。 クライアントは、無効な証明書を送信すると拒否されます。
- REQUIRED は、有効な TLS 証明書を提供しない接続クライアントをすべて拒否します。

以下に例を示します。

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

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、276 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。

ALW AIX, Linux, and 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。 AIX, Linux, and 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、 47 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、294 ページの『AIX, 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。

AIX, 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 Setting up communications for SSL or TLS on z/OS

Secure communications that use the SSL or TLS cryptographic security protocols involve setting up the communication channels and managing the digital certificates that you will use for authentication.

To set up your SSL or TLS installation you must define your channels to use SSL or TLS. You must also create and manage your digital certificates. On z/OS you can perform the tests with self-signed certificates, or with personal certificates signed by a local certificate authority (CA).

Self-signed certificates cannot be revoked, which could allow an attacker to spoof an identity after a private key has been compromised. CAs can revoke a compromised certificate, which prevents its further use. CA-signed certificates are therefore safer to use in a production environment, though self-signed certificates are more convenient for a test system.

For full information about creating and managing certificates, see “Working with SSL/TLS on z/OS” on page 307.

See the CERTLABL and CERTQSGL parameters of the ALTER QMGR command and the CERLABL parameter of the DEFINE CHANNEL command for more information.

The order of precedence is:

- Channel CERTLABL parameter
- QMGR CERTQSGL parameter if the channel is shared.

  For a sender channel, that means the transmission queue (XMITQ) is shared. For a receiver channel, that means the channel started through the shared listener, that is the listener with INDISP(GROUP).
- QMGR CERTLABL
- The default label of `ibmWebSphereMQ` followed by the name of the queue sharing group for shared channels, or the name of the queue manager.

This collection of topics introduces some of the tasks involved in setting up SSL or TLS communications, and provides step-by-step guidance on completing those tasks.

You might also want to test SSL or TLS client authentication, which are an optional part of the protocols. During the SSL or TLS handshake, the SSL or TLS client always obtains and validates a digital certificate from the server. With the IBM MQ implementation, the SSL or TLS server always requests a certificate from the client.

If the channel is shared, the channel first tries to find a certificate for the queue sharing group. If it does not find a certificate for a queue sharing group, it tries to find a certificate for the queue manager.

On z/OS, IBM MQ uses the `ibmWebSphereMQ` prefix on a label to avoid confusion with certificates for other products.

The SSL or TLS server always validates the client certificate if one is sent. If the SSL or TLS client does not send a certificate, authentication fails only if the end of the channel acting as the SSL or TLS server is defined with either the SSLCAUTH parameter set to REQUIRED or an SSLPEER parameter value set. For more information, see Connecting two queue managers using SSL or TLS.

SSL/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。

- 319 ページの『ユーザーの識別および認証』
- 352 ページの『オブジェクトに対するアクセス権限の設定』
- 420 ページの『メッセージの機密性』
- 476 ページの『メッセージのデータ保全性』
- 477 ページの『クラスターのセキュリティーの確保』

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` 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。
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、25 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、280 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 *IBM i* での鍵リポジトリー・パスワードの暗号化

いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

以下の IBM MQ コンポーネントおよび機能は、鍵リポジトリー・パスワードを保管するための 2 つの異なる方法をサポートしています。

- キュー・マネージャーの TLS キー・リポジトリー。
- TLS を使用する IBM MQ MQI clients 。

これらのコンポーネントで使用する鍵リポジトリー・パスワードは、 IBM MQ パスワード保護システムを使用して保護されます。 パスワードを指定して暗号化するメカニズムは、コンポーネントによって若干異なります。

キュー・マネージャーの TLS キー・リポジトリー

SSLKEYRPWD キュー・マネージャー属性が CHGMQM (メッセージ・キュー・マネージャーの変更) コマンドを使用して設定されている場合、パスワードは暗号化されます。

パスワードは AES-128 アルゴリズムで暗号化されます。 このアルゴリズムの詳細は公開されており、セキュアであると考えられています。

パスワードは、鍵リポジトリーにアクセスする可能性がある他のソフトウェアによって認識されない専有フォーマットで stash ファイルに保管されます。

ある IBM MQ コンポーネントによって暗号化されたパスワードを、別の IBM MQ コンポーネントで使用することはできません。

鍵リポジトリー・パスワードが暗号化されている場合は、固有の暗号鍵を指定できます。 固有の暗号鍵は、暗号鍵にアクセスできないユーザーがパスワードを暗号化解除できないようにします。 この鍵は、 **INITKEY** キュー・マネージャー属性を使用して指定します。この属性は、暗号化するパスワードを指定する前に設定する必要があります。

IBM MQ パスワード保護システムについて詳しくは、565 ページの『IBM MQ コンポーネント構成ファイルでのパスワードの保護』を参照してください。

TLS を使用する IBM MQ MQI clients

291 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)』 は、鍵リポジトリー・パスワードを stash ファイルに保管できます。 IBM i での MQSC コマンドを使用した管理も参照してください。

パスワードは AES-128 アルゴリズムで暗号化されます。 このアルゴリズムの詳細は公開されており、セキュアであると考えられています。

パスワードは、鍵リポジトリーにアクセスする可能性がある他のソフトウェアによって認識されない専有フォーマットで stash ファイルに保管されます。

鍵リポジトリー・パスワードが暗号化されている場合は、固有の暗号鍵を指定できます。 固有の暗号鍵は、暗号鍵にアクセスできないユーザーがパスワードを暗号化解除できないようにします。 このキーは、 **-sf** パラメーターを使用して指定します。

暗号化されたパスワードは、鍵リポジトリー・ファイルと同じディレクトリー内の stash ファイルに保管されます。

IBM MQ MQI clients は、他のメカニズムを介して提供されるパスワードもサポートします。 281 ページの『IBM i 上の IBM MQ MQI client の鍵リポジトリー・パスワードの指定』を参照してください。

鍵リポジトリー・パスワードを暗号化するために選択する方式に関係なく、保管されているパスワードの暗号化の制限を認識していることを確認してください。 572 ページの『パスワード暗号化による保護の制限』 を参照してください。

関連概念

280 ページの『IBM i でのキュー・マネージャーの鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

281 ページの『IBM i 上の IBM MQ MQI client の鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

276 ページの『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 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 (276 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 (279 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

289 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、291 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 293 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、278 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。

パスワード保護

鍵リポジトリー・パスワードが指定されている場合、 IBM MQ は IBM MQ パスワード保護システムを使用してパスワードを暗号化します。 パスワードを暗号化するには、初期鍵が使用されます。これがキュー・マネージャーに提供されない場合は、代わりにデフォルト鍵が使用されます。

鍵リポジトリー・パスワードを指定する前に、キュー・マネージャーの固有の初期鍵を設定する必要があります。 これを行うには、 **ALTER QMGR** MQSC コマンドの **INITKEY** 属性を使用します。

```
ALTER QMGR INITKEY('value')
```

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 は、/QIBM/UserData/ICSS/Cert/Server がパスで、Default はステム名です。

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 の使用: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) ALTER QMGR の使用: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

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 (289 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 (279 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 *IBM i* でのキュー・マネージャーの鍵リポジトリー・パスワードの指定

鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

IBM MQ には、キュー・マネージャーに鍵リポジトリー・パスワードを提供するメカニズムが用意されています。

- **CHGMQM** コマンド上の **SSLKEYRPWD** パラメーター

鍵リポジトリーのパスワードは、 IBM MQ パスワード保護システムを使用して暗号化されます。 鍵リポジトリー・パスワードを保護する方法について詳しくは、 277 ページの『IBM i での鍵リポジトリー・パスワードの暗号化』を参照してください。

IBM i での MQSC コマンドを使用した管理も参照してください。

SSLKEYRPWD 属性

キー・リポジトリー・パスワードをキュー・マネージャーに直接指定するには、以下の **CHGMQM** コマンドを実行します。 *queue_manager* をキュー・マネージャー名に置き換え、 *password* をキー・リポジトリー・パスワードに置き換えます。

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

重要 : キュー・マネージャー名とパスワードは必ず単一引用符で囲んでください。単一引用符で囲まないと、 IBM MQ は文字を大文字に変換します。

この方式を使用して鍵リポジトリー・パスワードを指定すると、パスワードは、保管される前に IBM MQ パスワード保護システムを使用して暗号化されます。

パスワードの暗号化には、初期鍵と呼ばれる暗号鍵が使用されます。 パスワードを安全に保護するために固有の初期鍵を使用するようにキュー・マネージャーを設定します。 初期鍵を指定しない場合は、デフォルト鍵が使用されます。

鍵リポジトリー・パスワードを設定する前に、キュー・マネージャーが固有の初期鍵を使用して構成されていることを確認してください。 **ALTER QMGR** コマンドで **INITKEY** 属性を使用して、初期キーを変更できます。 以下に例を示します。

```
ALTER QMGR INITKEY('mykey')
```

警告 : 鍵リポジトリー・パスワードの設定後に初期鍵を変更した場合、鍵リポジトリー・パスワードは新しい初期鍵で暗号化されません。 初期鍵を変更する場合は、鍵リポジトリー・パスワードもリセットする必要があります。 そうしないと、 IBM MQ は鍵リポジトリー・パスワードを暗号化解除できないため、鍵リポジトリーにアクセスできません。

SSLKEYRPWD 属性について詳しくは、 **CHGMQM** コマンドの **SSLKEYRPWD** パラメーターを参照してください。

関連概念

277 ページの『IBM i での鍵リポジトリー・パスワードの暗号化』
いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

281 ページの『IBM i 上の IBM MQ MQI client の鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

IBM i IBM i 上の IBM MQ MQI client の鍵リポジトリー・パスワードの指定

鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

IBM MQ には、鍵リポジトリー・パスワードを IBM MQ MQI client に提供するための以下の 4 つのメカニズムが用意されています。

- 282 ページの『MQSCO の KeyRepoPassword フィールド』
- 282 ページの『MQKEYRPWD 環境変数』

- 283 ページの『クライアント構成ファイルの SSLKeyRepositoryPassword 属性。』
- 283 ページの『鍵リポジトリー stash ファイル』

鍵リポジトリー stash ファイルを使用しない場合は、鍵リポジトリー・パスワードをプレーン・テキスト・ストリングとして指定することも、 IBM MQ パスワード保護システムを使用して暗号化されたストリングとして指定することもできます。 鍵リポジトリー・パスワードを保護する方法について詳しくは、 277 ページの『IBM i での鍵リポジトリー・パスワードの暗号化』を参照してください。

MQSCO の KeyRepoPassword フィールド

MQSCO 構造体を使用して鍵リポジトリー・パスワードを指定するには、以下の 3 つの変数ストリング・フィールドの組み合わせを使用する必要があります。

KeyRepoPasswordLength
パスワードの長さ。

KeyRepoPasswordPtr
パスワードを含むメモリー内の場所を指すポインター。

KeyRepoPasswordOffset
メモリー内のパスワードの位置。MQSCO 構造体の先頭からのバイト数で表されます。

注 : KeyRepoPasswordPtr または **KeyRepoPasswordOffset** のいずれか 1 つのみを指定できます。

以下に例を示します。

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

重要 : この方法を使用してパスワードを指定する場合は、パスワードを暗号化してから IBM MQ client アプリケーションに提供します。 詳しくは、283 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

MQCSO 構造体について詳しくは、 MQSCO-SSL/TLS 構成オプションを参照してください。

MQKEYRPWD 環境変数

キー・リポジトリー・パスワードが MQSCO 構造体を使用してクライアントに提供されない場合は、*MQKEYRPWD* 環境変数を使用してキー・リポジトリー・パスワードを指定できます。 以下に例を示します。

```
export MQKEYRPWD=passw0rd
```

または

```
set MQKEYRPWD=passw0rd
```

ここで、 *passw0rd* はパスワードです。

重要 : この方法を使用してパスワードを指定する場合は、環境変数の値を設定する前にパスワードを暗号化してください。 詳細については、283 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

クライアント構成ファイルの SSLKeyRepositoryPassword 属性。

他のいずれかの方法を使用しても鍵リポジトリー・パスワードがクライアントに提供されない場合は、クライアント構成ファイルの **SSL** スタンザの **SSLKeyRepositoryPassword** 属性を使用して、鍵リポジトリー・パスワードを指定できます。 以下に例を示します。

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

重要 : この方法を使用してパスワードを指定する場合は、**SSLKeyRepositoryPassword** 属性の値を設定する前にパスワードを暗号化してください。 詳しくは、283 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

クライアント構成ファイルの SSL スタンザについて詳しくは、 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。

鍵リポジトリー stash ファイル

他のいずれかの方法を使用しても鍵リポジトリー・パスワードがクライアントに提供されない場合、 IBM MQ は、鍵リポジトリーと同じディレクトリーに stash ファイルが存在すると想定します。 stash ファイルの語幹名はキー・リポジトリーと同じですが、拡張子は .sth です。

鍵リポジトリー stash ファイルは、 **amqrsslc** コマンド・ライン・ツールを使用して作成されます。 stash ファイルを作成するには、以下のコマンドを実行します。

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

このコマンドは、暗号化するパスワードの入力を求めるプロンプトを出します。 パスワードは、 **-sf** パラメーターを使用して指定されていない限り、デフォルトの暗号鍵を使用して IBM MQ パスワード保護システムによって暗号化されます。

詳しくは、291 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)』および283 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

鍵リポジトリー・パスワードの暗号化

stash ファイル以外の方法を使用して鍵リポジトリー・パスワードを指定する場合は、IBM MQ パスワード保護システムを使用してパスワードを暗号化します。 パスワードを暗号化するには、**runmqicred** コマンドを実行します。 プロンプトが出されたら、鍵リポジトリーのパスワードを入力します。 このコマンドは、暗号化されたパスワードを出力します。 説明されているいずれかの方法を使用して、暗号化されたパスワードをプレーン・テキストのパスワードではなく IBM MQ MQI client に提供できます。

パスワードの暗号化には、初期鍵と呼ばれる暗号鍵が使用されます。 パスワードを暗号化するときには、パスワードを安全に保護するために固有の初期鍵を使用してください。 独自の初期キーを指定するには、**runmqicred** コマンドに **-sf** パラメーターを使用します。 初期鍵を指定しない場合は、デフォルト鍵が使用されます。

詳しくは、 runmqicred (IBM MQ クライアントのパスワードの保護)を参照してください。

鍵リポジトリー・パスワードが暗号化されているときに独自の初期鍵を提供し、暗号化されたパスワードを IBM MQ MQI client に提供する場合は、必ず同じ初期鍵を IBM MQ MQI client に提供する必要もあります。 IBM MQ MQI client に初期鍵を提供する方法について詳しくは、 284 ページの『IBM i での IBM MQ MQI client の初期キーの指定』を参照してください。

関連概念

277 ページの『IBM i での鍵リポジトリー・パスワードの暗号化』
いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

280 ページの『IBM i でのキュー・マネージャーの鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

IBM i *IBM i での IBM MQ MQI client の初期キーの指定*

IBM MQ パスワード保護システムを使用して暗号化された変数を IBM MQ MQI client に提供する場合、値の暗号化に使用された対応する初期鍵を提供する必要がある場合があります。

値を暗号化するときに初期鍵を指定しなかった場合は、IBM MQ client に初期鍵の値を指定する必要はありません。 ただし、固有の初期鍵を使用した場合は、以下の方法を使用して IBM MQ client に初期鍵を提供できます。

- 284 ページの『MQCSP 構造体を使用した初期キーの提供』
- 284 ページの『MQS_MQI_KEYFILE 環境変数を使用した初期キーの指定』
- 285 ページの『クライアント構成ファイルを使用した初期鍵の提供』

MQCSP 構造体を使用した初期キーの提供

MQCSP 構造体を使用して初期キーを提供するには、以下の 3 つの変数ストリング・フィールドを組み合わせて使用する必要があります。

InitialKeyLength
初期鍵の長さ

InitialKeyPtr
初期キーを含むメモリー内の位置を指すポインター

InitialKeyOffset
メモリー内の初期キーの位置。MQCSP 構造体の先頭からのバイト数で表されます。

注 : **InitialKeyPtr** または **InitialKeyOffset** のいずれか 1 つのみを指定できます。

以下に例を示します。

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

MQS_MQI_KEYFILE 環境変数を使用した初期キーの指定

MQCSP 構造体を使用して初期キーがクライアントに提供されていない場合、 IBM MQ は *MQS_MQI_KEYFILE* 環境変数を検査します。 この環境変数は、使用する初期キーで構成される単一行のテキストを含むファイルの場所に設定する必要があります。

例えば、mykey.key という名前のファイルがルート・ディレクトリーに存在し、初期鍵が含まれている場合は、環境変数を以下のように設定する必要があります。

```
export MQS_MQI_KEYFILE=/mykey.key
```

または

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

クライアント構成ファイルを使用した初期鍵の提供

以前のメカニズムを使用して初期鍵がクライアントに提供されていない場合、IBM MQ は mqclient.ini ファイルの Security スタンザの **MQIInitialKeyFile** 属性を検査します。 この属性は、使用する初期キーで構成される単一行のテキストを含むファイルの場所に設定する必要があります。

例えば、mykey.key という名前のファイルがルート・ディレクトリーに存在し、初期鍵が含まれている場合、クライアント構成ファイルには以下が含まれている必要があります。

```
Security:
    MQIInitialKeyFile=/mykey.key
```

関連概念

277 ページの『IBM i での鍵リポジトリー・パスワードの暗号化』
いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

276 ページの『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 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、286 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 (276 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. **次へ** 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 (Finish)**」 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。

次のタスク

既存の証明書を更新する必要がある場合は、 IBM i 資料の 既存の証明書の更新 を参照してください。

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 (276 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。 詳細については、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。

リモートシステムのサーバー証明書を要求する *IBM i*

ローカル証明機関 (CA) によって署名された証明書を作成するか、他のプラットフォームのキー リポジトリにインポートするために商用 CA によって署名されたサーバー証明書を申請するには、この手順に従います。

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され、キー リポジトリにインポートされる個人証明書の場合は、Web ブラウザーで次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 (276 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。
   次のようなメッセージが表示されます。 個人証明書がインストールされました。 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 Web ブラウザに応じて、次のいずれかの手順を実行します。

- 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。
- 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. 手順でエクスポートした証明書をインポートします 287 ページの『7』リモート システム上のキー リポジトリに。
   - 証明書が Microsoft エッジの場合は、555 ページの『Microsoft .pfx ファイルからの個人証明書のインポート』ファイル。
   - 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。

   インポート中に、個人証明書と署名者証明書のラベル名が、IBM MQ 期待します。 ラベルは、IBM MQ キューマネージャー **CERTLABL** 属性が設定されている場合、またはデフォルト値 `ibmwebspheremq` キュー マネージャーの名前がすべて小文字で追加されます。 詳細については、デジタル証明書ラベル。

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 (276 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**。 615 ページの『AMS 上の AIX and Linux 用クイック・スタート・ガイド』の「証明書の共有」セクションの「証明書の共有」。 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 (276 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 (276 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 (276 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

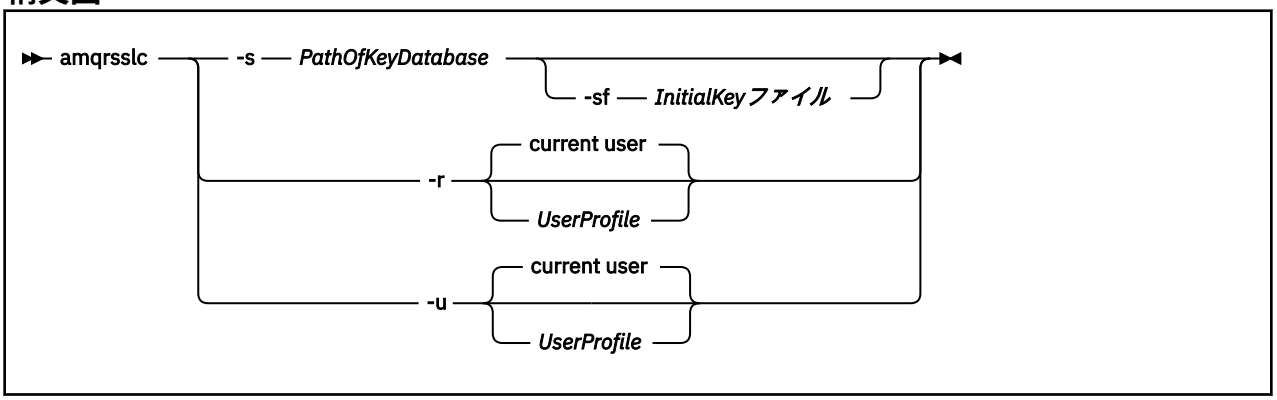
1. サーバーとクライアントのキュー・マネージャーで以下の手順を実行します。
   a) コマンド `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)` を実行して、キュー・マネージャーを変更し、SSLKEYR パラメーターを設定します。
   b) コマンド `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')` を実行して、デフォルトの鍵リポジトリーのパスワードを隠します。
   パスワードは、単一引用符で囲む必要があります。
   c) チャネルを変更して、SSLCIPHER パラメーターに正しい CipherSpec を設定します。
   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 (276 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隠しておくことができるため、実行時にクライアント・アプリケーションで指定する必要はありません。

鍵データベースのパスワードをスタッシュするには、 -s オプションを使用します。 鍵データベースの絶対パスと名前を指定します。 ファイル拡張子が指定されていない場合は、 .kdb であると想定されます。

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、 .sth 拡張子を持つ鍵データベースと同じ名前のファイルにスタッシュされます。

さらに、パスワードを暗号化するための初期鍵を指定できます。 初期キーは、1 行のテキストとしてファイルに保管する必要があります。その後、そのファイルの場所は、 **-sf** フラグを介してプログラムに提供されます。 初期鍵ファイルが指定されていない場合は、デフォルト鍵を使用してパスワードが暗号化されます。

stash ファイルは、鍵データベースと同じパスに保管されます。 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。

ALW AIX, Linux, and Windows での SSL/TLS の取り扱い

AIX, 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。

注 : Deprecated V 9.4.0 IBM MQ 9.4.0 以降、 IBM MQ Java アプリケーションでの CMS 鍵リポジトリーおよび stash ファイルの使用は推奨されなくなりました。 PKCS #12 鍵リポジトリーの使用に移行し、 IBM MQ パスワード保護システムを使用して鍵リポジトリーのパスワードを保護します。

重要 : V 9.4.0 V 9.4.0 IBM MQ 9.4.0 以降、 CMS 鍵リポジトリーおよび stash ファイルは、SSL/TLS を使用する AMQP チャネルおよび MQTT チャネルではサポートされません。 代わりに、PKCS #12 鍵リポジトリーを使用し、 IBM MQ パスワード保護システムを使用して鍵リポジトリーのパスワードを保護します。

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。

AIX, Linux, and Windows で鍵リポジトリーおよび証明書を管理するために使用されるコマンドについて詳しくは、 542 ページの『AIX, Linux, and Windows での runmqakm および runmqktool コマンド』を参照してください。

ALW *AIX, Linux, and Windows* での鍵リポジトリーのセットアップ

新しい鍵リポジトリーを作成するには、以下の手順を実行します。

始める前に

鍵リポジトリーには機密情報が含まれているため、鍵リポジトリーはパスワードで保護されます。 鍵リポジトリーを作成する前に、鍵リポジトリー・パスワードを安全に保管するために IBM MQ が提供するオプションを確認してください。 詳しくは、297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』を参照してください。

注 : Deprecated V 9.4.0 IBM MQ 9.4.0 以降、 IBM MQ Java アプリケーションでの CMS 鍵リポジトリーおよび stash ファイルの使用は推奨されなくなりました。 PKCS #12 鍵リポジトリーの使用に移行し、 IBM MQ パスワード保護システムを使用して鍵リポジトリーのパスワードを保護します。

重要 : V 9.4.0 V 9.4.0 IBM MQ 9.4.0 以降、 CMS 鍵リポジトリーおよび stash ファイルは、SSL/TLS を使用する AMQP チャネルおよび MQTT チャネルではサポートされません。 代わりに、PKCS #12 鍵リポジトリーを使用し、 IBM MQ パスワード保護システムを使用して鍵リポジトリーのパスワードを保護します。 以下のコマンドを使用して、PKCS #12 鍵リポジトリーを作成できます。

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

このコマンドは、指定されたパスワードで保護された *filename*.p12 という名前の PKCS #12 キー・リポジトリー・ファイルを作成します。

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、 25 ページの『SSL/TLS 鍵リポジトリー』を参照してください。

デジタル証明書は鍵リポジトリーに保管されます。 これらのデジタル証明書には、ラベルがあります。 証明書ラベルは、個人証明書を特定のキュー・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。 AIX, Linux, and Windows システムでは、 IBM MQ は証明書ラベルに以下のいずれかの値を使用します。

- **CERTLABL** キュー・マネージャーまたはチャネル属性の値 (設定されている場合)。
- デフォルト値の `ibmwebspheremq` には、キュー・マネージャーの名前または IBM MQ MQI client ユーザー・ログオン ID がすべて小文字で付加されています。

詳しくは、 デジタル証明書ラベルを参照してください。

キー・リポジトリー・ファイル名は、パス名と語幹名で構成されます。

- AIX 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_DATA_PATH`*`\qmgrs\`*`queue_manager_name`*`\ssl` です。ここで、 *`MQ_DATA_PATH`* は、 IBM MQ のインストール時に選択されたデータ・パスです。 例えば、`C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl` などです。

  デフォルトのファイル名は `key.kdb` です。 あるいは、独自のパスとファイル名を使用することもできます。

  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、デフォルトのパスまたはファイル名はありません。 このファイルへのアクセスを厳密に制御してください。

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、300 ページの『AIX, 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・ファイル名は、鍵リポジトリーの作成前でも作成後でも指定できます。

runmqakm (GSKCapiCmd) コマンドまたは V 9.4.0 V 9.4.0 **runmqktool** (keytool) コマンドを使用して、 IBM MQ で使用される鍵リポジトリーを管理できます。 詳しくは、 542 ページの『AIX, Linux, and Windows での runmqakm および runmqktool コマンド』を参照してください。

鍵リポジトリーを管理するためのコマンドを実行するユーザー ID には、鍵リポジトリー・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。 デフォルトの `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、**runmqakm** または **runmqktool** コマンド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 では、クライアントを実行するユーザー ID とは異なるユーザー ID から **runmqakm** または **runmqktool** を実行する場合、ファイル許可を変更して、 IBM MQ MQI client がキー・リポジトリー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、 298 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 299 ページの『AIX 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。

runmqakm コマンドを使用して、新しい空の鍵リポジトリーを作成できます。 V 9.4.0 V 9.4.0 代わりに **runmqktool** コマンドを使用すると、証明書を作成またはインポートするコマンドが発行されたときに鍵リポジトリーが作成されます。

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。

手順

1. 以下のコマンドを発行し、 **runmqakm** コマンドを使用して鍵リポジトリーを作成します。

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

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-type 型
V 9.4.0 V 9.4.0 鍵リポジトリーのタイプを指定します。 IBM MQ によって使用される鍵リポジトリーの場合、可能な値は以下のとおりです。

- pkcs12
- Deprecated cms

注 : IBM MQ 9.4.0 以降、 CMS 鍵リポジトリーおよび stash ファイルの使用は IBM MQ Java アプリケーションでは推奨されなくなり、SSL/TLS を使用する AMQP および MQTT チャネルではサポートされません。

-stash
オプション。 鍵リポジトリー・パスワードを stash ファイルに保管するには、このオプションを指定します。 代わりに IBM MQ パスワード保護システムを使用してパスワードを暗号化する場合は、パスワードを stash ファイルに保管する必要はありません。

-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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

2. 298 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 299 ページの『AIX 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』の説明に従って、鍵リポジトリー・ファイルのアクセス権を設定します。

   Windows では、デフォルトで、鍵リポジトリーを作成するコマンドを実行したユーザー ID にのみ、stash (`.sth`) ファイルを読み取る権限が付与されます。 **runmqakm** コマンドを使用して stash ファイルを作成した後、ファイル許可を確認し、キュー・マネージャーを実行しているサービス・アカウント、またはローカル `mqm` などのグループに許可を付与します。

3. stash ファイルを使用しない場合は、 301 ページの『キュー・マネージャーの鍵リポジトリー・パスワードの指定 (AIX, Linux, and Windows)』 または 302 ページの『AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定』の手順に従って、キュー・マネージャーまたはクライアント・アプリケーションに鍵ストア・パスワードを提供します。

次のタスク

必要に応じて、デフォルトの認証局 (CA) 証明書を空の鍵リポジトリーに追加します。 詳細については、299 ページの『AIX, Linux, and Windows での空の鍵リポジトリーへのデフォルト CA 証明書の追加』を参照してください。

ALW 鍵リポジトリー保護のための強力なパスワードの生成 *(AIX, 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 password_length -strong -fips
```

ここで、 *password_length* は、生成するパスワードの長さです。 指定できるパスワードの最小長は 14 です。

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。 AIX 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。

```
! \ " '
```

runmqakm または V 9.4.0 V 9.4.0 **runmqktool** コマンドからのプロンプトに応答してキー・リポジトリー・パスワードを入力する場合、オペレーティング・システム・シェルはこれらの場合にデータ入力に影響を与えないため、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。

ALW *AIX, Linux, and Windows* での鍵リポジトリー・パスワードの暗号化

いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、鍵リポジトリーはパスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

以下の IBM MQ コンポーネントおよびフィーチャーは、鍵リポジトリー・パスワードを保管するための 2 つの異なる方式をサポートします。

- キュー・マネージャーの TLS 鍵リポジトリー。
- TLS を使用する IBM MQ MQI clients 。
- V 9.4.0 qm.ini ファイルの **NativeHALocalInstance** スタンザにあるネイティブ HA 構成。
- V 9.4.0 qm.ini ファイルの **AuthToken** スタンザでのトークン認証の構成。

これらのコンポーネントで使用する鍵リポジトリー・パスワードは、以下のいずれかの方法を使用して暗号化および保管することができます。

IBM MQ パスワード保護システム。

各 IBM MQ コンポーネントは、鍵リポジトリー・パスワードを暗号化するコマンドを提供します。 コマンドが出力する暗号化されたコマンドは、ファイルに保管されます。

キュー・マネージャーの TLS 鍵リポジトリーの場合、パスワードは、 **SSLKEYRPWD** キュー・マネージャー属性が設定されているときに暗号化されます。

パスワードは AES-128 アルゴリズムで暗号化されます。 このアルゴリズムの詳細は公開されており、セキュアであると見なされます。

パスワードは、鍵リポジトリーにアクセスする可能性がある他のソフトウェアによって認識されない専有形式で保管されます。

ある IBM MQ コンポーネントによって暗号化されたパスワードを、別の IBM MQ コンポーネントで使用することはできません。

鍵リポジトリー・パスワードが暗号化されている場合は、固有の暗号鍵を指定できます。 固有の暗号鍵は、暗号鍵にアクセスできないユーザーがパスワードを暗号化解除できないようにします。

鍵リポジトリーにある証明書を管理するには、プレーン・テキストの鍵リポジトリー・パスワードが必要です。 IBM MQ パスワード保護システムを使用して鍵リポジトリー・パスワードを暗号化するほかに、この目的で鍵リポジトリー・パスワードにアクセスできる安全な場所に鍵リポジトリー・パスワードを保管する必要もあります。

IBM MQ パスワード保護システムについて詳しくは、565 ページの『IBM MQ コンポーネント構成ファイルでのパスワードの保護』を参照してください。

鍵リポジトリー stash ファイル。

runmqakm コマンドは、鍵リポジトリー・パスワードを stash ファイルに保管できます。

パスワードは、IBM MQ の暗号プロバイダー IBM Global Security Kit (GSKit)に固有のプロプラエタリー方式で暗号化されます。

固有の暗号鍵を指定することはできません。

暗号化されたパスワードは、鍵リポジトリー・ファイルと同じディレクトリー内の stash ファイルに保管されます。

鍵リポジトリーと stash ファイルの両方に対する読み取り権限を持つユーザーは、鍵リポジトリーの内容にアクセスして管理することができます。

注 : Deprecated V 9.4.0 IBM MQ 9.4.0 以降、 IBM MQ Java アプリケーションでの stash ファイルの使用は推奨されなくなりました。

重要 : V 9.4.0 V 9.4.0 IBM MQ 9.4.0 以降、stash ファイルは、TLS を使用する AMQP チャネルおよび MQTT チャネルではサポートされません。

鍵リポジトリー・パスワードを暗号化するために選択した方法に関係なく、保管されているパスワードの暗号化の制限に注意してください。 詳細については、572 ページの『パスワード暗号化による保護の制限』を参照してください。

関連概念

301 ページの『キュー・マネージャーの鍵リポジトリー・パスワードの指定 (AIX, Linux, and Windows)』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

302 ページの『AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

294 ページの『AIX, Linux, and Windows での SSL/TLS の取り扱い』
AIX, 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。

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。

ファイル *key*.p12、 *key*.kdb、 *key*.sth、 *key*.crl、および *key*.rdb(*key* は鍵データベースの語幹名) にアクセス制御を設定して、制限されたユーザー・セットに権限を付与します。

.p12 または .kdb 以外の別のキー・リポジトリー拡張を使用した場合は、このファイルのアクセス権が設定されていることも確認する必要があります。

アクセス権の付与の際には、次の事柄を考慮します。

全権限
BUILTIN¥Administrators、NT AUTHORITY¥SYSTEM、およびデータベース・ファイルを作成したユーザー。

読み取り権限
キュー・マネージャーの場合、ローカル mqm グループのみ。 これは、MCA が mqm グループ内のユーザー ID 下で稼働していると想定しています。

クライアントの場合、クライアント・プロセスが実行されているユーザー ID。

Linux AIX *AIX 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*.p12、 *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、*key* は鍵データベースのステム名、ファイル所有者の場合は read および write 、mqm またはクライアント・ユーザー・グループの場合は read (-rw-r -----) に設定します。

.p12 または .kdb 以外の別のキー・リポジトリー拡張を使用した場合は、このファイルのアクセス権が設定されていることも確認する必要があります。

ALW *AIX, Linux, and Windows* での空の鍵リポジトリーへのデフォルト *CA* 証明書の追加
以下の手順に従って、1 つ以上のデフォルトの認証局 (CA) 証明書を空の鍵リポジトリーに追加します。

新規キー・リポジトリーを作成すると、そのキー・リポジトリーは空になります。 **runmqakm** コマンドを使用して、デフォルトの CA 証明書を鍵リポジトリーに追加できます。

runmqakm の使用

以下のコマンドを発行し、 **runmqakm** コマンドを使用してデフォルトの CA 証明書を鍵リポジトリーに追加します。

```
runmqakm -cert -populate -db filename -pw password
```

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

注 : IBM MQ は、鍵リポジトリー内の CA 証明書によって署名されたすべての証明書を信頼します。 どの認証局を信頼するかを慎重に検討し、クライアントおよびキュー・マネージャーの認証に必要な CA 証明書のみを追加してください。 デフォルト CA 証明書の完全セットを鍵リポジトリーに追加することはお勧めしません。

ALW AIX, 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. AIX 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。

   注 : IBM MQ 9.3.0 以降、SSLKEYR フィールドは、完全なファイル名 (拡張子を含む) と語幹名 (拡張子を含まない) の両方をサポートします。 語幹名が設定されている場合、 IBM MQ は自動的に `.kdb` を追加し、そのキー・リポジトリーを使用します。

ALW AIX, 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、AIX and Linux では以下のとおりです。

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

On Windows:

```
ALTER QMGR SSLKEYR('C:&#xa5;Program Files&#xa5;IBM&#xa5;MQ\Qmgrs\QM1\ssl\Mykey.kdb')
```

重要 : Windows および Linux では、TLS AMQP チャネルを使用する場合、鍵リポジトリー・ファイルの接尾部は以下のいずれかでなければなりません。

- `.kdb`(CMS キー・リポジトリーの場合)
- `.p12` または `.pkcs12`(PKCS #12 鍵リポジトリーの場合)。

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、554 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 (AIX, 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。

キュー・マネージャーの鍵リポジトリー・パスワードの指定 (*AIX, Linux, and Windows*)

鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

IBM MQ は、キュー・マネージャーに鍵リポジトリー・パスワードを提供するために、以下の 2 つのメカニズムを提供します。

- 301 ページの『KEYRPWD 属性』
- 301 ページの『鍵リポジトリー stash ファイル』

鍵リポジトリー stash ファイルを使用しない場合、鍵リポジトリー・パスワードは IBM MQ パスワード保護システムを使用して暗号化されます。 鍵リポジトリー・パスワードを保護する方法について詳しくは、297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』を参照してください。

KEYRPWD 属性

キー・リポジトリー・パスワードをキュー・マネージャーに直接提供するには、 *password* をキー・リポジトリー・パスワードに置き換えて、以下の MQSC コマンドを実行します。

```
ALTER QMGR KEYRPWD('password')
```

重要 : パスワードは必ず単一引用符で囲んでください。そうしないと、 IBM MQ によって文字が大文字に変換されます。

この方法を使用して鍵リポジトリー・パスワードを指定すると、パスワードは、保管される前に IBM MQ パスワード保護システムを使用して暗号化されます。

パスワードの暗号化には、初期鍵と呼ばれる暗号鍵が使用されます。 パスワードを安全に保護するために、固有の初期鍵を使用するようにキュー・マネージャーを設定します。 初期鍵を指定しない場合は、デフォルトの鍵が使用されます。

鍵リポジトリー・パスワードを設定する前に、キュー・マネージャーが固有の初期鍵を使用して構成されていることを確認してください。 初期鍵を変更するには、 **ALTER QMGR** コマンドで **INITKEY** 属性を使用します。 以下に例を示します。

```
ALTER QMGR INITKEY('mykey')
```

警告 : 鍵リポジトリー・パスワードの設定後に初期鍵を変更しても、鍵リポジトリー・パスワードが新しい初期鍵で暗号化されることはありません。 鍵リポジトリー・パスワードを再設定せずに初期鍵を変更すると、 IBM MQ は鍵リポジトリー・パスワードを暗号化解除できないため、鍵リポジトリーにアクセスできません。

KEYRPWD 属性について詳しくは、 KEYRPWD を参照してください。

鍵リポジトリー stash ファイル

KEYRPWD 属性を使用してキュー・マネージャーに鍵リポジトリー・パスワードが指定されていない場合、IBM MQ は、鍵リポジトリーと同じディレクトリーに stash ファイルが存在すると想定します。 stash ファイルの語幹名はキー・リポジトリーと同じですが、拡張子は `.sth` です。

鍵リポジトリー stash ファイルは、鍵リポジトリーと同時に作成されるか、後で別個の **runmqakm** コマンドとして作成されます。

重要 : stash ファイルのフォーマットは IBM MQ 暗号プロバイダー IBM Global Security Kit (GSKit) に固有のものであり、別の暗号プロバイダーを使用するプラットフォームでは使用できません。

鍵リポジトリーの作成時に stash ファイルを作成するには、 **-stash** パラメーターを指定します。 以下に例を示します。

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

ここで、 *passw0rd* は、鍵リポジトリーのパスワードです。

後で stash ファイルを作成するには、以下のコマンドを実行します。

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

ここで、 *passw0rd* は、鍵リポジトリーのパスワードです。

関連概念

297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』
いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、鍵リポジトリーはパスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

302 ページの『AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

ALW *IBM MQ MQI client の鍵リポジトリーの位置確認 (AIX, 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 (302 ページの『AIX, 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。

ALW *AIX, 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、AIX and Linux では以下のとおりです。

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

  On Windows:

  ```
  set MQSSLKEYR=C:&#xa5;Program Files&#xa5;IBM&#xa5;MQ\ssl\key.kdb
  ```

- 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。

AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定

鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

IBM MQ は、鍵リポジトリー・パスワードを IBM MQ MQI client に提供するための以下の 4 つのメカニズムを提供します。

- 303 ページの『MQSCO の KeyRepoPassword フィールド』
- 303 ページの『MQKEYRPWD 環境変数』
- 304 ページの『クライアント構成ファイルの SSLKeyRepositoryPassword 属性。』
- 304 ページの『鍵リポジトリー stash ファイル』

鍵リポジトリー stash ファイルを使用しない場合は、鍵リポジトリー・パスワードをプレーン・テキスト・ストリングとして指定することも、 IBM MQ パスワード保護システムを使用して暗号化されたストリングとして指定することもできます。 鍵リポジトリー・パスワードを保護する方法について詳しくは、 297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』を参照してください。

MQSCO の KeyRepoPassword フィールド

MQSCO 構造体を使用してキー・リポジトリー・パスワードを提供するには、以下の 3 つの可変ストリング・フィールドの組み合わせを使用する必要があります。

KeyRepoPasswordLength
パスワードの長さ。

KeyRepoPasswordPtr
パスワードが格納されているメモリー内の場所へのポインター。

KeyRepoPasswordOffset
メモリー内のパスワードの場所。MQSCO 構造体の先頭からのバイト数で表されます。

注 : KeyRepoPasswordPtr または **KeyRepoPasswordOffset** のいずれか 1 つのみを指定できます。

以下に例を示します。

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

重要 : この方法を使用してパスワードを指定する場合は、パスワードを IBM MQ client アプリケーションに提供する前に暗号化してください。 詳しくは、304 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

MQCSO 構造体について詳しくは、 MQSCO-SSL/TLS 構成オプションを参照してください。

MQKEYRPWD 環境変数

MQSCO 構造体を使用して鍵リポジトリー・パスワードがクライアントに提供されない場合は、*MQKEYRPWD* 環境変数を使用して鍵リポジトリー・パスワードを指定できます。 以下に例を示します。

```
export MQKEYRPWD=passw0rd
```

または

```
set MQKEYRPWD=passw0rd
```

ここで、 passw0rd はご使用のパスワードです。

重要 : この方法を使用してパスワードを指定する場合は、環境変数の値を設定する前にパスワードを暗号化してください。 詳細については、304 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

クライアント構成ファイルの SSLKeyRepositoryPassword 属性。

鍵リポジトリー・パスワードが他の方法のいずれかを使用してクライアントに提供されない場合は、クライアント構成ファイルの **SSL** スタンザの **SSLKeyRepositoryPassword** 属性を使用して、鍵リポジトリー・パスワードを指定できます。 以下に例を示します。

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

重要 : この方法を使用してパスワードを指定する場合は、**SSLKeyRepositoryPassword** 属性の値を設定する前にパスワードを暗号化してください。 詳しくは、304 ページの『鍵リポジトリー・パスワードの暗号化』を参照してください。

クライアント構成ファイルの SSL スタンザについて詳しくは、 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。

鍵リポジトリー stash ファイル

鍵リポジトリー・パスワードが他のいずれかの方法でクライアントに提供されない場合、 IBM MQ は、鍵リポジトリーと同じディレクトリーに stash ファイルが存在すると想定します。 stash ファイルの語幹名はキー・リポジトリーと同じですが、拡張子は `.sth` です。

キー・リポジトリー stash ファイルは、別個の **runmqakm** コマンドを使用して、キー・リポジトリーと同時に作成されます。

重要 : stash ファイルのフォーマットは IBM MQ 暗号プロバイダー IBM Global Security Kit (GSKit) に固有のものであり、別の暗号プロバイダーを使用するプラットフォームでは使用できません。

鍵リポジトリーの作成時に stash ファイルを作成するには、 **-stash** パラメーターを指定します。 以下に例を示します。

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

ここで、 *passw0rd* は、鍵リポジトリーのパスワードです。

後で stash ファイルを作成するには、以下のコマンドを実行します。

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

ここで、 *passw0rd* は、鍵リポジトリーのパスワードです。

鍵リポジトリー・パスワードの暗号化

stash ファイル以外の方法を使用して鍵リポジトリー・パスワードを指定する場合は、IBM MQ パスワード保護システムを使用してパスワードを暗号化します。 パスワードを暗号化するには、**runmqicred** コマンドを実行します。 プロンプトが出されたら、鍵リポジトリー・パスワードを入力します。 このコマンドは、暗号化されたパスワードを出力します。 暗号化されたパスワードは、説明されているいずれかの方法を使用して、プレーン・テキスト・パスワードの代わりに IBM MQ MQI client に提供できます。

パスワードの暗号化には、初期鍵と呼ばれる暗号鍵が使用されます。 パスワードを暗号化する場合は、パスワードを安全に保護するために固有の初期鍵を使用してください。 独自の初期鍵を指定するには、**runmqicred** コマンドに **-sf** パラメーターを使用します。 初期鍵を指定しない場合は、デフォルトの鍵が使用されます。

詳しくは、 runmqicred (IBM MQ クライアント・パスワードの保護)を参照してください。

鍵リポジトリー・パスワードが暗号化されているときに独自の初期鍵を提供し、暗号化されたパスワードを IBM MQ MQI client に提供する場合は、 IBM MQ MQI client にも同じ初期鍵を提供する必要があります。 IBM MQ MQI client に初期鍵を提供する方法について詳しくは、 305 ページの『AIX, Linux, and Windows 上の IBM MQ MQI client の初期鍵の提供』を参照してください。

関連概念

297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』

いくつかの IBM MQ コンポーネントは、デジタル証明書または対称鍵を含む鍵リポジトリーにアクセスする必要があります。 鍵リポジトリーには機密情報が含まれているため、鍵リポジトリーはパスワードで保護されます。 鍵リポジトリー・パスワードは、鍵リポジトリーへのアクセス時に IBM MQ が読み取ることができる場所に保管する必要があります。 鍵リポジトリーへの無許可アクセスの可能性を減らすために、パスワードも暗号化する必要があります。

301 ページの『キュー・マネージャーの鍵リポジトリー・パスワードの指定 (AIX, Linux, and Windows)』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

ALW *AIX, Linux, and Windows 上の IBM MQ MQI client の初期鍵の提供*

IBM MQ パスワード保護システムを使用して暗号化された変数を IBM MQ MQI client に提供する場合、値の暗号化に使用された対応する初期鍵の提供が必要になることがあります。

値の暗号化時に初期鍵を指定しなかった場合は、 IBM MQ client に初期鍵値を指定する必要はありません。ただし、固有の初期鍵を使用した場合は、以下の方法を使用して、その初期鍵を IBM MQ client に提供することができます。

- 305 ページの『MQCSP 構造体を使用した初期鍵の提供』
- 305 ページの『MQS_MQI_KEYFILE 環境変数を使用して初期キーを指定する』
- 306 ページの『クライアント構成ファイルを使用した初期鍵の提供』

MQCSP 構造体を使用した初期鍵の提供

MQCSP 構造を使用して初期鍵を提供するには、以下の 3 つの可変ストリング・フィールドの組み合わせを使用する必要があります。

InitialKeyLength
初期鍵の長さ

InitialKeyPtr
初期キーを含むメモリー内の位置へのポインター

InitialKeyOffset
メモリー内の初期キーの位置。MQCSP 構造体の先頭からのバイト数で表されます。

注 : InitialKeyPtr または **InitialKeyOffset** のいずれか 1 つのみを指定できます。

以下に例を示します。

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

MQS_MQI_KEYFILE 環境変数を使用して初期キーを指定する

MQCSP 構造体を使用して初期キーがクライアントに提供されていない場合、 IBM MQ は *MQS_MQI_KEYFILE* 環境変数を検査します。 この環境変数は、使用する初期キーで構成される単一行のテキストを含むファイルの場所に設定する必要があります。

例えば、mykey.key という名前のファイルがルート・ディレクトリーに存在し、初期キーが含まれている場合は、環境変数を以下のように設定する必要があります。

```
export MQS_MQI_KEYFILE=/mykey.key
```

または

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

クライアント構成ファイルを使用した初期鍵の提供

前のメカニズムを使用して初期鍵がクライアントに提供されない場合、IBM MQ は mqclient.ini ファイルのセキュリティー・スタンザの **MQIInitialKeyFile** 属性を検査します。 この属性は、使用する初期キーで構成される単一行のテキストを含むファイルの場所に設定する必要があります。

例えば、mykey.key という名前のファイルがルート・ディレクトリーに存在し、初期鍵が含まれている場合、クライアント構成ファイルには以下が含まれている必要があります。

```
Security:
    MQIInitialKeyFile=/mykey.key
```

関連概念

302 ページの『AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定』
鍵リポジトリーには機密情報が含まれているため、パスワードで保護されます。 TLS 操作を実行するために鍵リポジトリーの内容にアクセスできるようにするには、 IBM MQ が鍵リポジトリーのパスワードを取得できる必要があります。

275 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。

ALW *AIX, Linux, and Windows* 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

鍵リポジトリー内の証明書または鍵リポジトリーの場所を変更すると、その変更は、チャネルのタイプおよびチャネルの実行方法に応じて、一度に有効になります。

鍵リポジトリー内の証明書に対する変更、または鍵リポジトリーの場所に対する変更は、以下の状況で有効になります。

- 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。
- TLS 環境をリフレッシュするために MQSC コマンド **REFRESH SECURITY TYPE(SSL)** が発行されたとき。
- 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・プーリング・プロセスが既に TLS チャネルを実行しており、変更をすぐに有効にする場合は、MQSC コマンド **REFRESH SECURITY TYPE(SSL)** を実行します。
- 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おり、変更をすぐに有効にする場合は、MQSC コマンド **REFRESH SECURITY TYPE(SSL)** を実行します。
- 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おり、変更をすぐに有効にする場合は、MQSC コマンド **REFRESH SECURITY TYPE(SSL)** を実行します。

IBM MQ Explorer コマンドまたは PCF コマンドを使用して、 IBM MQ TLS 環境をリフレッシュすることもできます。

重要 : 鍵ストア構成ファイルへの変更、または Advanced Message Security (AMS) MCA インターセプターまたは AMS クライアントによって使用される鍵ストアへの変更は、キュー・マネージャーまたはアプリケーションの再始動時に有効になります。

ALW 暗号ハードウェア *(AIX, 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、AIX, 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 で説明されているように、 SSLCRYP パラメーターを指定して **ALTER QMGR** MQSC コマンドを使用します。
- IBM MQ Explorer を使用して、 AIX, Linux, and Windows システム上で暗号ハードウェアを構成します。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、AIX, 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できます。

- **MQSSLCRYP** 環境変数を設定します。 **MQSSLCRYP** に許可される値は、ALTER QMGR で説明されているように、**SSLCRYP** パラメーターの場合と同じです。 この環境変数を設定するには、以下のいずれかのコマンドを使用します。
  - Linux AIX AIX and Linux システムの場合:

    ```
    export MQSSLCRYP=string
    ```

  - Windows Windows システムの場合:

    ```
    SET MQSSLCRYP=string
    ```

  ここで、 *string* は、システム上に存在する暗号ハードウェアを構成するために使用されるパラメーター・ストリングを表します。

  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、ハードウェアを構成したラベルと一致している必要があります。
- IBM MQ client 構成ファイルの SSL スタンザに **SSLCryptoHardware** 属性を設定します。 許可される値は、「**ALTER QMGR**」で説明されているように、 **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、ハードウェアを構成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。

重要 : > MQSSLCRYP 環境変数または **SSLCryptoHardware** 属性を使用して暗号ハードウェアの構成を指定する場合は、保管する前にパスワードを保護する必要があります。 詳しくは、569 ページの『暗号ハードウェアを使用する IBM MQ clients』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、563 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。

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 Working with SSL/TLS on z/OS

This information describes how you set up and work with Transport Layer Security (TLS) on z/OS.

Each topic includes examples of performing each task using RACF. You can perform similar tasks using the other external security managers.

On z/OS, you must also set the number of server subtasks that each queue manager uses for processing TLS calls, as described in “Setting the SSLTASKS parameter on z/OS” on page 308.

z/OS TLS support is integral to the operating system, and is known as *System SSL*. System SSL is part of the Cryptographic Services Base element of z/OS. The Cryptographic Services Base members are installed in the *pdsname*. SIEALNKE partitioned data set (PDS). When you install System SSL, ensure that you choose the appropriate options to provide the CipherSpecs that you require.

If you need to renew a self-signed certificate, see Steps for renewing a self-signed certificate in RACF for more information.

z/OS *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。

鍵リポジトリーが CHINIT ユーザー ID によって所有されている場合、このユーザー ID には IRR.DIGTCERT.LISTRING プロファイル、それ以外の場合は更新アクセス、および IRR.DIGTCERT.LIST プロファイル。 必要に応じて、ACCESS (UPDATE) または ACCESS (READ) を指定した PERMIT コマンドを使用してアクセス権限を付与します。

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適切な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。
- チャネル・イニシエーターを実行する権限を持つユーザー ID は、「z/OS UNIX System Services 計画」資料の説明に従って、 z/OS UNIX System Services (z/OS UNIX) を使用するように構成されています。

  チャネル・イニシエーターがゲスト / デフォルト UID および OMVS セグメントを使用して z/OS UNIX を呼び出すことを望まないユーザーは、チャネル・イニシエーターが特別な許可を必要とせず、スーパーユーザーとして UNIX 内で実行されないため、デフォルト・セグメントに基づいて新規 OMVS セグメントをモデリングするだけで済みます。

  チャネル・イニシエーターに正しいアクセス権限を与える方法の例については、310 ページの『Giving the channel initiator the correct access rights on z/OS』 の PERMIT コマンドを参照してください。

z/OS *Setting the SSLTASKS parameter on z/OS*

Use the ALTER QMGR command to set the number of server subtasks for processing TLS calls

To use TLS channels, ensure that there are at least two server subtasks by setting the SSLTASKS parameter, using the ALTER QMGR command. For example:

```
ALTER QMGR SSLTASKS(5)
```

To avoid problems with storage allocation, do not set the SSLTASKS attribute to a value greater than eight in an environment where there is no Certificate Revocation List (CRL) checking.

If CRL checking is used, an SSLTASK is held by the channel concerned for the duration of that check. This could be for a significant elapsed time while the relevant LDAP server is contacted, because each SSLTASK is a z/OS task control block.

You must restart the channel initiator if you change the value of the SSLTASKS attribute.

z/OS *Setting up a key repository on z/OS*

Set up a key repository at both ends of the connection. Associate each key repository with its queue manager.

A TLS connection requires a *key repository* at each end of the connection. Each queue manager must have access to a key repository. Use the SSLKEYR parameter on the ALTER QMGR command to associate a key repository with a queue manager. See “SSL/TLS 鍵リポジトリー” on page 25 for more information.

On z/OS, digital certificates are stored in a *key ring* that is managed by your External Security Manager (ESM) . These digital certificates have labels, which associate the certificate with a queue manager. TLS uses these certificates for authentication purposes. All the examples that follow use RACF commands. Equivalent commands exist for other ESM programs.

On z/OS, IBM MQ uses either the value of the **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager appended. See Digital certificate labels for details.

The key repository name for a queue manager is the name of a key ring in your RACF database. You can specify the key ring name either before or after creating the key ring.

Use the following procedure to create a new key ring for a queue manager:

1. Ensure that you have the appropriate authority to issue the RACDCERT command (see Controlling the use of the RACDCERT command for more details).
2. Issue the following command:

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

   where:

   - *userid1* is the user ID of the channel initiator address space, or the user ID that is going to own the key ring (if the key ring is shared).
   - *ring-name* is the name you want to give to your key ring. The length of this name can be up to 237 characters. This name is case-sensitive. Specify *ring-name* in uppercase characters to avoid problems.

z/OS *Making CA certificates available to a queue manager on z/OS*

After you have created your key ring, connect any relevant CA certificates to it.

If you have the CA certificate in a data set, you must first add the certificate to the RACF database by using the following command:

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

Then to connect a CA certificate for My CA to your key ring, use the following command:

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

where *userid1* is either the channel initiator user ID or the owner of a shared key ring.

For more information about CA certificates, refer to “デジタル証明書” on page 13.

z/OS *Locating the key repository for a queue manager on z/OS*

Use this procedure to obtain the location of your queue manager's key ring.

1. Display your queue manager's attributes, using either of the following MQSC commands:

```
DISPLAY QMGR ALL
DISPLAY QMGR SSLKEYR
```

2. Examine the command output for the location of the key ring.

z/OS *Specifying the key repository location for a queue manager on z/OS*

To specify the location of your queue manager's key ring, use the ALTER QMGR MQSC command to set your queue manager's key repository attribute.

For example:

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

if the key ring is owned by the channel initiator address space, or:

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

if it is a shared key ring, where *userid1* is the user ID that owns the key ring.

z/OS *Giving the channel initiator the correct access rights on z/OS*

The channel initiator (CHINIT) needs access to the key repository and to certain security profiles.

Granting the CHINIT access to read the key repository

If the key repository is owned by the CHINIT user ID, this user ID needs read access to the IRR.DIGTCERT.LISTRING profile in the FACILITY class, and update access otherwise, and read access to the IRR.DIGTCERT.LIST profile. Grant access by using the PERMIT command with ACCESS(UPDATE) or ACCESS(READ) as appropriate:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID( userid ) ACCESS(READ)
```

where *userid* is the user ID of the channel initiator address space.

Granting the CHINIT read access to the appropriate CSF* profiles

For hardware support provided through the Integrated Cryptographic Service Facility (ICSF) to be used, ensure your CHINIT user ID has read access to the appropriate CSF* profiles in the CSFSERV class by using the following command:

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

where *csf-resource* is the name of the CSF* profile and *userid* is the user ID of the channel initiator address space.

Repeat this command for each of the following CSF* profiles:

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

Your CHINIT user ID might also need read access to other CSF* profiles. For example, if you are using the ECDHE_RSA_AES_256_GCM_SHA384 Cipher Spec, your CHINIT user ID also needs read access to the following CSF* profiles:

- CSF1DVK
- CSF1GAV
- CSF1GKP
- CSF1SKE
- CSF1TRC
- CSF1TRD

For more information, see RACF CSFSERV resource requirements.

If your certificate keys are stored in ICSF and your installation has established access control over keys stored in ICSF, ensure your CHINIT user ID has read access to the profile in the CSFKEYS class by using the following command:

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

where *userid* is the user ID of the channel initiator address space.

Using the Integrated Cryptographic Service Facility (ICSF)

The channel initiator can use ICSF to generate a random number when seeding the password protection algorithm to obfuscate passwords flowing over client channels if TLS is not being used.

For further information, see "Using the Integrated Cryptographic Service Facility (ICSF)" on page 263

z/OS When changes to certificates or the key repository become effective on z/OS

Changes become effective when the channel initiator starts or the repository is refreshed.

Specifically, changes to the certificates in the key ring and to the key repository attribute become effective on either of the following occasions:

- When the channel initiator is started or restarted.
- When the REFRESH SECURITY TYPE(SSL) command is issued to refresh the contents of the key repository.

z/OS Creating a self-signed personal certificate on z/OS

Use this procedure to create a self-signed personal certificate.

1. Generate a certificate and a public and private key pair using the following command:

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.

- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.

  userid1 and *userid2* can be the same ID.
- *ring-name* is the name you gave the key ring in “Setting up a key repository on z/OS” on page 309.
- *label-name* must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager appended. See Digital certificate labels for details.

z/OS *Requesting a personal certificate on z/OS*

Apply for a personal certificate using RACF.

To apply for a personal certificate, use RACF as follows:

1. Create a self-signed personal certificate, as in “Creating a self-signed personal certificate on z/OS” on page 311. This certificate provides the request with the attribute values for the Distinguished Name.
2. Create a PKCS #10 Base64-encoded certificate request written to a data set, using the following command:

   ```
   RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
   ```

   where

   - *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space
   - *label_name* is the label used when creating the self-signed certificate

   See “デジタル証明書ラベルの要件に関する説明” on page 27 for details.
3. Send the data set to a Certificate Authority (CA) to request a new personal certificate.
4. When the signed certificate is returned to you by the Certificate Authority, add the certificate back into the RACF database, using the original label, as described in “Adding personal certificates to a key repository on z/OS” on page 313.

z/OS *Creating a RACF signed personal certificate*

RACF can function as a certificate authority and issue its own CA certificate.

This section uses the term *signer certificate* to denote a CA certificate issued by RACF.

The private key for the signer certificate must be in the RACF database before you carry out the following procedure:

1. Use the following command to generate a personal certificate signed by RACF, using the signer certificate contained in your RACF database:

   ```
   RACDCERT ID(userid2) GENCERT
   SUBJECTSDN(CN('common-name')
              T('title')
              OU('organizational-unit')
              O('organization')
              L('locality')
              SP('state-or-province')
              C('country'))
   WITHLABEL('label-name')
   SIGNWITH(CERTAUTH LABEL('signer-label'))
   ```

2. Connect the certificate to your key ring using the following command:

   ```
   RACDCERT ID(userid1)
   CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
   ```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.

  userid1 and *userid2* can be the same ID.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 309.
- *label-name* must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details.
- *signer-label* is the label of your own signer certificate.

z/OS *Adding personal certificates to a key repository on z/OS*

Use this procedure to add or import a personal certificate to a key ring.

After the certificate authority sends you a new personal certificate, add it to the key ring using the following procedure:

1. Add the certificate to the RACF database using the following command:

```
RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID( userid1 )
CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 309.
- *input-data-set-name* is the name of the data set containing the CA signed certificate. The data set must be cataloged and must not be a PDS or a member of a PDS. The record format (RECFM) expected by RACDCERT is VB. RACDCERT dynamically allocates and opens the data set, and reads the certificate from it as binary data.
- *label-name* is the label name that was used when you created the original request. It must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details.

z/OS *Exporting a personal certificate from a key repository on z/OS*

Export the certificate using the RACDCERT command.

On the system from which you want to export the certificate, use the following command:

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the label of the certificate you want to extract.
- *output-data-set-name* is the data set into which the certificate is placed.
- CERTB64 is a DER encoded X.509 certificate that is in Base64 format. You can choose an alternative format, for example:

CERTDER
DER encoded X.509 certificate in binary format

PKCS12B64
PKCS #12 certificate in Base64 format

PKCS12DER
PKCS #12 certificate in binary format

z/OS *Deleting a personal certificate from a key repository on z/OS*

Delete a personal certificate using the RACDCERT command.

Before deleting a personal certificate, you might want to save a copy of it. To copy your personal certificate to a data set before deleting it, follow the procedure in "Exporting a personal certificate from a key repository on z/OS" on page 313. Then use the following command to delete your personal certificate:

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the name of the certificate you want to delete.

z/OS *Renaming a personal certificate in a key repository on z/OS*

Rename a certificate using the RACDCERT command.

If you do not want a certificate with a specific label to be found, but do not want to delete it, you can rename it temporarily using the following command:

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the name of the certificate you want to rename.
- *new-label-name* is the new name of the certificate.

This can be useful when testing TLS client authentication.

z/OS *Associating a user ID with a digital certificate on z/OS*

IBM MQ can use a user ID associated with a RACF certificate as a channel user ID. Associate a user ID with a certificate by installing it under that user ID, or using a Certificate Name Filter.

The method described in this topic is an alternative to the platform-independent method for associating a user ID with a digital certificate, which uses channel authentication records. For more information about channel authentication records, see "チャネル認証レコード" on page 51.

When an entity at one end of a TLS channel receives a certificate from a remote connection, the entity asks RACF if there is a user ID associated with that certificate. The entity uses that user ID as the channel user ID. If there is no user ID associated with the certificate, the entity uses the user ID under which the channel initiator is running.

Associate a user ID with a certificate in either of the following ways:

- Install that certificate into the RACF database under the user ID with which you want to associate it, as described in "Adding personal certificates to a key repository on z/OS" on page 313.
- Use a Certificate Name Filter (CNF) to map the Distinguished Name of the subject or issuer of the certificate to the user ID, as described in "Setting up a certificate name filter on z/OS" on page 315.

z/OS *Setting up a certificate name filter on z/OS*

Use the RACDCERT command to define a certificate name filter (CNF), which maps a Distinguished Name to a user ID.

Perform the following steps to set up a CNF.

1. Enable CNF functions using the following command. You require update authority on the class DIGTNMAP to do this.

```
SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
```

2. Define the CNF. For example:

```
RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
```

   where USER1 is the user ID to be used when:

   - The DN of the subject has an Organization of `IBM` and a Country of UK.
   - The DN of the issuer has an Organization of `ExampleCA` and a Locality of `Internet`.

3. Refresh the CNF mappings:

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

Note:

1. If the actual certificate is stored in the RACF database, the user ID under which it is installed is used in preference to the user ID associated with any CNF. If the certificate is not stored in the RACF database, the user ID associated with the most specific matching CNF is used. Matches of the subject DN are considered more specific than matches of the issuer DN.
2. Changes to CNFs do not apply until you refresh the CNF mappings.
3. A DN matches the DN filter in a CNF only if the DN filter is identical to the *least significant portion* of the DN. The least significant portion of the DN comprises the attributes that are usually listed at the right-most end of the DN, but which appear at the beginning of the certificate.

   For example, consider the SDNFILTER `'O=IBM.C=UK'`. A subject DN of `'CN=QM1.O=IBM.C=UK'` matches that filter, but a subject DN of `'CN=QM1.O=IBM.L=Hursley.C=UK'` does not match that filter.

   The least significant portion of some certificates can contain fields that do not match the DN filter. Consider excluding these certificates by specifying a DN pattern in the SSLPEER pattern on the DEFINE CHANNEL command.
4. If the most specific matching CNF is defined to RACF as NOTRUST, the entity uses the user ID under which the channel initiator is running.
5. RACF uses the `'.'` character as a separator. IBM MQ uses either a comma or a semicolon.

You can define CNFs to ensure that the entity never sets the channel user ID to the default, which is the user ID under which the channel initiator is running. For each CA certificate in the key ring associated with the entity, define a CNF with an IDNFILTER that exactly matches the subject DN of that CA certificate. This ensures that all certificates that the entity might use match at least one of these CNFs. This is because all such certificates must either be connected to the key ring associated with the entity, or must be issued by a CA for which a certificate is connected to the key ring associated with the entity.

Refer to the *z/OS Security Server RACF Security Administrator's Guide* for more information about the commands you use to manipulate CNFs.

z/OS *Defining a sender channel and transmission queue on QMA on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

Procedure

On QMA, issue commands like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

Results

A sender channel, TO.QMB, and a transmission queue, QMB, are created.

z/OS *Defining a receiver channel on QMB on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

Procedure

On QMB, issue a command like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

Results

A receiver channel, TO.QMB, is created.

z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
2. Optional: If any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
3. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

Results

The sender channel is started.

z/OS *Exchanging self-signed certificates on z/OS*

Exchange the certificates you previously extracted. If you use FTP, use the correct format.

Procedure

Transfer the CA part of the QM1 certificate to the QM2 system and vice versa, for example, by FTP.

If you transfer the certificates using FTP, you must do so in the correct format.

Transfer the following certificate types in *binary* format:

- DER encoded binary X.509
- PKCS #7 (CA certificates)

- PKCS #12 (personal certificates)

Transfer the following certificate types in ASCII format:

- PEM (privacy-enhanced mail)
- Base64 encoded X.509

z/OS *Defining a sender channel and transmission queue on QM1 on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

Procedure

On QM1, issue commands like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

The CipherSpecs at each end of the channel must be the same.

Only the SSLCIPH parameter is mandatory if you want your channel to use TLS. See “IBM MQ でのCipherSpec と CipherSuite” on page 42 for information about the permitted values for the SSLCIPH parameter.

Results

A sender channel, QM1.TO.QM2, and a transmission queue, QM2, are created.

z/OS *Defining a receiver channel on QM2 on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

Procedure

On QM2, issue a command like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

The channel must have the same name as the sender channel you defined in “Defining a sender channel and transmission queue on QM1 on z/OS” on page 317, and use the same CipherSpec.

z/OS *Starting the sender channel on QM1 on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start a listener program on QM2.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener
2. Optional: If any SSL/TLS channels have run previously, issue the command REFRESH SECURITY TYPE(SSL).

   This ensures that all the changes made to the key repository are available.
3. On QM1, start the channel, using the command `START CHANNEL(QM1.TO.QM2)`.

Results

The sender channel is started.

z/OS *Refreshing the SSL or TLS environment on z/OS*

Refresh the TLS environment on queue manager QMA using the **REFRESH SECURITY** command.

Procedure

On QMA, enter the following command:

```
REFRESH SECURITY TYPE(SSL)
```

This ensures that all the changes made to the key repository are available.

z/OS *Allowing anonymous connections on a receiver channel on z/OS*

Use the **ALTER CHANNEL** command to make SSL or TLS client authentication optional.

Procedure

On QMB, enter the following command:

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

z/OS *Starting the sender channel on QM1 on z/OS*

If necessary, start the channel initiator, start a listener program, and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: if you have not already done so, start the channel initiator.
2. Optional: If you have not already done so, start a listener program on QM2.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener
3. Optional: If the channel initiator was already running or any SSL/TLS channels have run previously, issue the command REFRESH SECURITY TYPE(SSL).

   This ensures that all the changes made to the key repository are available.
4. On QM1, start the channel, using the command `START CHANNEL(QM1.TO.QM2)`.

Results

The sender channel is started.

z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start the channel initiator, start a listener program, and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start the channel initiator.
2. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
3. Optional: If the channel initiator was already running or if any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
4. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

Results
The sender channel is started.

z/OS *Modifying elliptic curve key length on z/OS*
How you modify the GSK_CLIENT_ECURVE_LIST environment variable, to set the list of elliptic curves or supported groups that are specified by the client, as a string consisting of one or more 4-character values in order of preference for use.

Important: You must apply the fix in z/OS APAR OA61783 to permit certain elliptic curves to be made effective by the operating system, when using TLS 1.0, TLS 1.1 and/or TLS 1.2 negotiated connections.

You can set this TLS environment variable in the channel initiator startup JCL, using the CEEOPTS DD statement:

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

In the dataset referenced above, specify the list that you want to use, for example:

```
ENVAR("GSK_CLIENT_ECURVE_LIST=002300240025")
```

Important: Do not use this CEEOPTS statement with in-stream data, as this prevents the environment variable from being set for all TLS tasks using that statement.

Ensure you reference a sequential dataset, or partitioned dataset member, to allow this to work when using an SSLTASKS value greater than one.

You can also use the server analogue equivalent of GSK_CLIENT_ECURVE_LIST, which is GSK_SERVER_ALLOWED_KEX_ECURVES. See Limiting key exchange elliptic curves for more information.

In addition, see Table 5 in Cipher suite definitions for a list of valid 4-character elliptic curve and supported groups specifications.

The default specification is `00210023002400250019`. If TLS V1.3 is enabled, `0029` （x25519）is appended to the end of the default list.

ユーザーの識別および認証

X.509 証明書、MQCSP 構造、またはいくつかのタイプのユーザー出口プログラムを使用して、ユーザーを識別および認証できます。

X.509 証明書の使用

SET CHLAUTH コマンドおよび **SSLPEER** パラメーターを指定した X.509 証明書を使用して、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。

SET 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。

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。 詳細については、339 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。

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 呼び出しで指定されます。 この構造体には、アプリケーションによって提供される資格情報を含めることができます。 アプリケーションは、MQCSP 構造体にユーザー ID とパスワードを指定できます。 IBM MQ 9.3.4 以降、アプリケーションは認証トークンを提供することもできます。 必要に応じて、セキュリティー出口で MQCSP を変更できます。

警告 : MQCSP 構造体の資格情報は、プレーン・テキストでネットワークを介して送信されることがあります。 クライアント・アプリケーション資格情報が保護されていることを確認するには、 31 ページの『MQCSP パスワード保護』を参照してください。

詳しくは、321 ページの『MQCSP 構造を使用したユーザーの識別および認証』および 325 ページの『認証トークンの処理』を参照してください。

Linux AIX AIX および Linux では、MQCSP 構造で指定されたユーザー ID とパスワードは、オペレーティング・システムまたはプラグ可能認証方式 (PAM) のいずれかを使用して認証できます。 PAM には、詳細をサービスから隠すユーザー認証のための一般的なメカニズムが用意されています。 詳しくは、351 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。

出口での識別と認証の実装

いくつかのタイプのユーザー出口プログラムを使用して、ユーザーを識別および認証することができます。詳しくは、322 ページの『セキュリティー出口による識別と認証の実装』、323 ページの『メッセージ出口による識別マッピング』、および 324 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。

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
AIX and Linux システム	• mqm グループのメンバー
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、および拡張メッセージ・セキュリティー・アドレス・スペースが実行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 構造は、認証に使用される資格情報を制御するためにメッセージ・キュー・インターフェース (MQI) を使用するアプリケーションの主要な方法です。

MQCSP 構造には、許可サービスがユーザーを識別および認証するために使用できる資格情報が含まれています。

MQCSP 構造体は、アプリケーションが MQCSP 構造体を明示的に提供していない場合でも、クライアント・サイドまたはサーバー・サイドのセキュリティー出口によって変更できます。 MQCSP 構造を明示的に提供しないアプリケーションの例としては、IBM MQ classes for JMS を使用するアプリケーションがあります。 MQCSP 構造にユーザー ID とパスワードを挿入するクライアント・サイド・セキュリティー出口の例については、 81 ページの『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)』 を参照してください。

V 9.4.0 MQCSP 構造体には、ユーザー ID とパスワード、または認証トークンが含まれます。 MQCSP 構造体で提供される資格情報には、以下の制約事項が適用されます。

- アプリケーションまたは出口は、ユーザー ID とパスワード、または認証トークンのいずれかを提供する必要がありますが、両方を提供することはできません。
- IBM MQ にアクセスするために使用できるのは、特定の形式と要件を満たす認証トークンのみです。 IBM MQ での認証トークンの要件について詳しくは、328 ページの『認証トークンの要件』を参照してください。
- 認証トークン内の ID がアプリケーションのコンテキストとして採用される場合、トークンは適切なユーザー・クレームを提供する必要があり、クレーム値は有効な IBM MQ ユーザー ID でなければなりません。 例えば、ユーザー名は最大長と特殊文字の制限に従う必要があります。 ユーザー ID の採用について詳しくは、 322 ページの『MQCSP と AdoptCTX 設定の関係』 を参照してください。

MQCSP 構造体について詳しくは、 MQCSP-セキュリティー・パラメーター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 構造の資格情報は、プレーン・テキストでネットワークを介して送信されることがあります。 クライアント・アプリケーション資格情報が保護されていることを確認するには、 31 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

接続認証機能が有効になっている場合、 IBM MQ は常に、MQCSP 構造で渡される資格情報を認証します。資格情報が正常に認証された後、 IBM MQ は、接続されたアプリケーションによって実行される操作に対する後続の許可検査のためにユーザー ID を採用できます。 キュー・マネージャーの **CONNAUTH** 属性によって参照される認証情報 (AUTHINFO) オブジェクトが **ADOPTCTX(YES)** で定義されている場合は、MQCSP 資格情報のユーザー ID が採用されます。

IBM MQ では、許可検査に使用できるユーザー ID の長さに制限があります。 これらの制限について詳しくは、 90 ページの『ユーザー ID』 を参照してください。 MQCSP 構造で渡されたユーザー ID が採用されると、 IBM MQ の動作は、他の構成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、ユーザーの LDAP レコードの短いユーザー名属性にあるユーザー ID を採用します。 短いユーザー名属性は、AUTHINFO オブジェクトの **SHORTUSR** 属性を使用して設定されます。

  例えば、**SHORTUSR** が `'CN'` に設定され、LDAP レコードにユーザーが `'CN=Test,SN=MQ,O=IBM,C=UK'` としてリストされている場合、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 構造に渡されるユーザー 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。 詳しくは、158 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、18 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、467 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、124 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。 詳しくは、324 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。

Linux V 9.4.0 AIX 認証トークンの処理

IBM MQ 9.4.0 以降、クライアント・アプリケーションは、 AIX または Linux で実行されているキュー・マネージャーで認証するためのトークンを提供できるようになりました。 トークン内のユーザー ID は、IBM MQ リソースにアクセスするための許可にも使用できます。

JWT (JSON Web トークン) は、クレーム・ベースの ID モデルを採用しています。 ID およびアクセス制御は、クレームおよびトークン発行者のアイデアに抽象化されます。

- クレームは、ユーザーに関する情報を含む名前と値のペアであり、ユーザーが実行できることではなく、ユーザーが誰であるかを設定します。
- トークン発行者は、信頼のおける第三者機関、またはユーザーの ID のみに基づいてユーザーにトークンを発行するサーバーです。 トークン発行者は、ユーザーが何を行うことができるかについては関係ありません。

トークンは、クレームを含む単純な構造であり、インターネットを介してパーティー間で簡単に転送できます。 認証のためにトークンを使用することには、一元化された ID 管理の利点があります。 1 つのトラステッド・トークン発行者を使用して、アプリケーションが各サービスに個別に登録せずに多数のサービスで認証できるようにすることができます。 トークンを使用すると、各サービスに資格情報が送信されず、信頼できる発行者にのみ資格情報が送信されるため、セキュリティーが強化されます。

JWT は、提案されたインターネット標準 RFC7519 によって定義されます。

IBM MQ でのトークンの処理方法

IBM MQ で使用されるトークンは、 IBM MQ がサポートするアルゴリズムで署名された有効な JWT でなければなりません。 JWT は、JSON Web Signature (JWS) 標準に従って署名されている必要があります。 JSON Web 暗号化 (JWE) および JSON Web Key (JWK) JOSE テクノロジーを使用するトークンは、IBM MQ では使用できません。 詳しくは、 328 ページの『認証トークンの要件』 を参照してください。

認証トークンを提供するアプリケーションは、 IBM MQ clients をサポートする任意のプラットフォームで実行できます。 アプリケーションは、C または Java で作成し、クライアント・バインディングを使用してキュー・マネージャーに接続する必要があります。 ただし、キュー・マネージャーは AIX または Linux で実行する必要があります。

キュー・マネージャーは、鍵リポジトリー内のトラステッド発行者公開鍵または対称鍵に対してトークン・シグニチャーを検証します。 キュー・マネージャーをセットアップするには、 331 ページの『JWKS エンドポイントを使用して認証トークンを受け入れるためのキュー・マネージャーの構成』 または ローカル鍵ストアを使用して認証トークンを受け入れるためのキュー・マネージャーの構成の手順に従います。

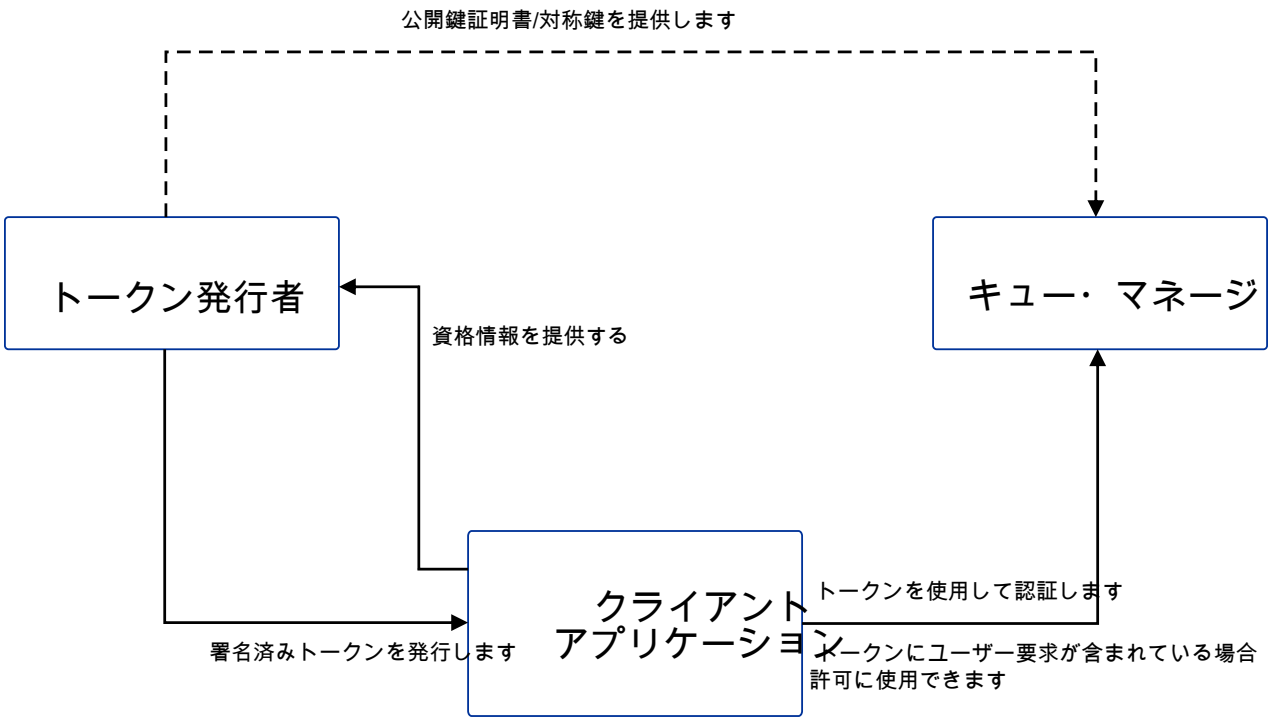
トークン発行者は、委任されたセキュリティー・アクセス権限を持つトラステッド・パーティーです。つまり、アプリケーション・ユーザーの ID を検証します。 キュー・マネージャーは、認証トークンが有効であること、および認証済みユーザーが IBM MQ オブジェクトへのアクセスを許可されていることを検査します。 キュー・マネージャーは、最初にトークンを使用して接続する前に、ユーザーを認識する必要はありませんが、認識する必要はありません。 IBM MQ 管理者は、キュー・マネージャーに接続するアプリケーションの認証と許可をセットアップし、トークンに含める必要があるものの要件を設定する必要があります。

クライアント・アプリケーションは、 IBM MQ に接続するときに、認証に使用する発行者にトークンを動的に要求できます。 その後、アプリケーションは、MQCSP 構造体または選択された API の同等のもの を使用して、接続時にトークンをキュー・マネージャーに渡します。

アプリケーションを変更して認証トークンを要求し、接続時にそのトークンをキュー・マネージャーに提示できない場合は、代わりにセキュリティー出口を使用して、MQCSP 構造にトークンを提供することができます。

トークンが認証トークンの要件を満たしており、トークン・シグニチャーが有効である場合、接続が確立されます。 オプションのユーザー要求がトークンに含まれている場合、キュー・マネージャーは、IBM MQ リソースにアクセスするための許可検査に、トークンに含まれているユーザー ID を使用することもできます。 ユーザー・クレームは、キュー・マネージャーが許可検査のために採用するユーザー ID を含むトークン内のクレームです。 ユーザー・クレームのこの名前は、 `qm.ini` ファイルの **AuthToken** スタンザの **UserClaim** 属性で指定されます。

詳しくは、 336 ページの『アプリケーションでの認証トークンの使用』 および MQCSP-セキュリティー・パラメーターを参照してください。



この図は、 IBM MQ でトークンを使用する場合に予期されるフローの基本的な例を示しています。 予期されるライフサイクルは以下のとおりです。

- トークンは、トラステッド発行者によってアプリケーションに発行されます。 詳しくは、 認証トークンの要件を参照してください。
- アプリケーションは、接続時にトークンをキュー・マネージャーに渡します。 詳しくは、 アプリケーションでの認証トークンの使用を参照してください。
- キュー・マネージャーは、鍵リポジトリー内のトラステッド発行者公開鍵または対称鍵に対してトークン・シグニチャーを検証します。 キュー・マネージャーをセットアップするには、331 ページの『JWKS エンドポイントを使用して認証トークンを受け入れるためのキュー・マネージャーの構成』の手順に従います。
- 認証トークンに有効なユーザー要求が含まれている場合、トークン内のユーザーを、 IBM MQ リソースにアクセスするための許可検査に採用することができます。 詳しくは、 許可のためのユーザーの採用を参照してください。
- IBM MQ 管理者は、トラステッド・トークン発行者証明書を管理します。 証明書の有効期限が切れると、トークン発行者から新しい証明書を取得し、鍵リポジトリーに追加する必要があります。
- キュー・マネージャーを構成していて、アプリケーションが接続しているにもかかわらず、トークンに関する問題が発生する場合は、 認証トークンの問題のトラブルシューティング および トークン認証エラー・コードを参照してください。

IBM MQ は、JWT および JWS 標準に準拠するトークンを提供するすべてのトークン発行者と連携します。

まだトークンを使用していないが、トークン・サーバーの立ち上げに何が関係しているのかを理解したい場合は、無料でオープン・ソースの Keycloak プロジェクトの「 入門ガイド 」を参照してください。

関連資料

qm.ini ファイルの AuthToken スタンザ

Linux V 9.4.0 AIX 認証トークンの要件

IBM MQ で使用される認証トークンの検証要件、構造、およびアルゴリズム。

要件

IBM MQ で使用される認証トークンは、以下の要件を満たしている必要があります。

- トークンの長さは、最大長の 8192 文字を超えてはなりません。 詳しくは、 MQCSP の TokenLength (MQLONG)を参照してください。
- トークン構造とエンコードは、RFC7519 の JSON Web Token (JWT) 仕様、および RFC7515 の JSON Web Signature (JWS) 仕様で定義されているとおりに有効です。
- 329 ページの表 68 で指定されている必須のトークン・ヘッダー・パラメーターが存在し、それらのパラメーターの値が有効である。
- 330 ページの表 69 で指定された必須のペイロード・クレームが存在し、クレームの値が有効である。
- トークンは、 IBM MQ がサポートする 330 ページの表 70 のアルゴリズムで署名されます。
- 有効期限 (**exp**) クレームの値が現在時刻より後になっています。
- not before (**nbf**) クレームが存在する場合、値は現在時刻より前になります。
- ユーザー請求が存在する場合、値は 331 ページの『認証トークン内のユーザー ID』の要件を満たしている必要があります。

トークン構造

IBM MQ は、 RFC7519 標準に準拠した JWT を受け入れます。 JWT は、 RFC7515 で定義されている JWS 標準に従って署名およびエンコードされている必要があります。

IBM MQ は、JWS 保護トークンに以下の 3 つのコンポーネントが含まれていることを予期しています。

JOSE ヘッダー

トークンのタイプと、その内容を保護するために使用される暗号アルゴリズムを記述するパラメーターを格納する JSON オブジェクト。

以下のヘッダー例では、エンコードされたオブジェクトが JWT であること、およびヘッダーとペイロードが HMAC SHA-256 アルゴリズムを使用して保護されていることを宣言しています。

```
{
 "typ":"JWT",
 "alg":"HS256"
}
```

JWS ペイロード

JWT 標準で指定されたクレームを含む JSON オブジェクト。 JSON オブジェクトの各メンバーはクレームです。 クレームは、トークン発行者の ID、またはベアラーのユーザー ID を表明できます。

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

JWS 署名

トークンが信頼できる発行者によって発行されたことを検証するために使用されます。

これらのコンポーネントは、JWS 保護トークンでは、ピリオド (.) で区切られた base64url-encoded ストリングとして表されます。

JWS 標準に準拠する認証トークンは、トークンの認証性を検証するために署名されますが、暗号化されません。 したがって、トークンにアクセスできるすべてのユーザーが読み取ることができ、場合によっては再利用することもできます。 キュー・マネージャーへの接続を構成して、ネットワークを介して送信されるときに、例えば TLS を使用して、暗号化を使用して認証が保護されるようにします。 アプリケーションによって提供される資格情報を保護するためのオプションについて詳しくは、MQCSP パスワード保護を参照してください。

IBM MQ は、ヘッダー内の以下のパラメーターとクレーム、および認証トークンのペイロードをサポートします。 トークン内の追加のパラメーターまたはクレームはすべて無視されます。 トークンに同じ名前の複数のパラメーターまたはクレームが含まれている場合、重複する名前を持つ最後のパラメーターまたはクレームが使用されます。

表 *68.* トークン・ヘッダー・パラメーターの説明

トークン・パーツ	パラメーター名	データ・タイプ	必須	説明
ヘッダー	**typ**	ストリング	はい	トークン・タイプ。 このパラメーターの値は「JWT」でなければなりません。
	alg	ストリング	はい	ヘッダーとペイロードを保護するために使用されるアルゴリズム。 このパラメーターの値は、 330 ページの表 70 のアルゴリズムのいずれかでなければなりません。

表 69. トークン・ペイロード・クレームの説明				
トークン・パーツ	**パラメーター名**	**データ・タイプ**	**必須**	**説明**
ペイロード	**exp**	整数	はい	トークンの有効期限時刻。1979 年 1 月 1 日 00:00 協定世界時からの秒数で表されます。 この時間が経過すると、トークンは受け入れられません。
	nbf	整数	いいえ	協定世界時の 1979 年 1 月 1 日 00:00 以降トークンが受け入れられなくなるまでの秒数で表される時刻。
	qm.ini ファイル内の **AuthToken** スタンザの **UserClaim** フィールドに指定されているユーザー要求名。	ストリング	トークン内のユーザー・クレームが許可に使用される場合にのみ必要です。	許可検査に採用されるユーザー ID が含まれているクレームの名前。 例えば、トークンにユーザー要求 "AppUser": "MyUserName"がある場合は、 qm.ini ファイルの **AuthToken** スタンザに **UserClaim**=AppUser を指定する必要があります。

エンコードおよびデコードされたトークンの良い例については、jwt.io Web サイトの デバッガー ページを参照してください。

アルゴリズム

IBM MQ は、JWS 保護トークンの JSON Web Algorithms (JWA) 仕様 に含まれているアルゴリズムのサブセットをサポートします。

表 70. *JWS* 保護トークン用に *IBM MQ* によってサポートされる *JSON Web* アルゴリズム *(JWA)*	
alg パラメーター値	**デジタル署名または MAC アルゴリズム**
HS256	SHA-256 を使用した HMAC
HS384	SHA-384 を使用した HMAC
HS512	SHA-512 を使用した HMAC
RS256	RSASSA-PKCS1-v1_5 (SHA-256 を使用)
RS384	RSASSA-PKCS1-v1_5 (SHA-384 を使用)
RS512	RSASSA-PKCS1-v1_5 (SHA-512 を使用)

非対称鍵証明書の要件

トークンが非対称鍵で署名されている場合、トークン発行者からの公開鍵証明書は、キュー・マネージャーがトークン認証に使用する鍵リポジトリー内になければなりません。 認証トークンを受け取る場合、証明書はその有効期間内でなければなりません。 トークン発行者からの証明書が取り消されていないことを確認するための検査は行われません。

認証トークン内のユーザー ID

キュー・マネージャーが、認証トークンのユーザー要求に含まれるユーザー ID をアプリケーションのコンテキストとして採用するように構成されている場合、採用されるユーザー ID は以下の要件を満たしている必要があります。

- 最大 12 文字を使用できます。
- 以下のいずれかの文字で開始する必要があります。

  A-Z、a-z
- 以下のいずれかの文字を含めることができます。

  0 から 9 A から Z、a から z、+、-。 : = _
- 予約済みユーザー ID UNKNOWN および NOBODY のいずれかであってはなりません。

関連タスク

AuthTokens を受け入れるためのキュー・マネージャーの構成

関連資料

qm.ini ファイルの AuthToken スタンザ

Linux V 9.4.0 AIX JWKS エンドポイントを使用して認証トークンを受け入れるためのキュー・マネージャーの構成

JWKS エンドポイントを使用して認証トークンでユーザーおよびアプリケーションを認証するように、AIX または Linux 上で実行される IBM MQ キュー・マネージャーを構成します。

始める前に

トークンが IBM MQ でどのように機能するかについて詳しくは、認証トークンの処理を参照してください。

キュー・マネージャーを構成する前に、キュー・マネージャーの **CONNAUTH** 属性で参照されている AUTHINFO オブジェクトのタイプが IDPWOS であることを確認してください。 トークン認証は、OS ユーザー ID およびパスワード検査用にキュー・マネージャーが構成されている場合にのみ使用可能です。

Service スタンザの **SecurityPolicy** 属性が Group に設定されていないことを確認します。 **SecurityPolicy** が明示的に Group に設定されている場合、トークン認証は使用できません。 もし **SecurityPolicy** に設定されていますグループ、削除 **SecurityPolicy**Service スタンザから属性を削除し、キュー マネージャーを再起動します。

このタスクについて

アプリケーションは、トークンを使用してキュー・マネージャーで認証を行うことができます。 IBM MQ は、提案されたインターネット標準 RFC7519 に準拠する信頼できる発行者から JSON Web トークン (*JWT*) を受け入れます。 トークンを使用して ID を認証することができます。認証された ID は、将来の許可検査に採用される可能性があります。

トークンを受け入れるようにキュー・マネージャーを構成する最も簡単な方法は、以下で説明するように JWKS エンドポイントを指すことです。 ご使用の認証サービスがそのようなものを提供しておらず、エンドポイントまたは JWKS が他の理由で不適切である場合は、 332 ページの『ローカル鍵ストアを使用して認証トークンを受け入れるためのキュー・マネージャーの構成』 を参照してください。

手順

1. 以下の詳細については、認証サーバー管理者にお問い合わせください。
   - 正しい JWKS エンドポイント (URL)。
   - このサーバーが HTTP トラフィックを暗号化するために使用する証明書、またはこの証明書に署名する認証局 (あるいはその両方)。

重要 : TLS/HTTPS を介して常に JWKS 情報を提供する必要があり、キュー・マネージャーが接続を信頼できるようにするためにこの情報が必要になります。

2. qm.ini ファイルに **HTTPSKeyStore** を指定して、発信 https 接続を作成するようにキュー・マネージャーを構成します。

   詳細については、

   - qm.ini ファイル内の HTTPSKeyStore の説明。
   - 338 ページの『TLS トラストストアとして使用する鍵リポジトリーの作成』.

   認証サーバーがカスタム証明書 /CA を使用する場合は、これがこの HTTPSKeyStore に正しく存在することを確認する必要があります。
3. qm.ini 構成ファイルに JWKS スタンザ を定義して、JWKS エンドポイントを構成します。

   追加スタンザは、以下を提供します。

   - **issuername**。 これは、この権限によって署名されたすべてのトークンに存在する「iss」クレームと一致する必要があり、多くの場合、認証サービスの URL に基づいています。
   - **endpoint**。 これは、トークン署名の検証に使用される公開鍵をキュー・マネージャーが照会するアドレスです。
   - **userclaim**。これは、トークンが検証された後の IBM MQ 権限検査に使用するトークン内のカスタム・フィールドを識別するためのオプションです。

   重要 : このような接続に **ADOPTCTX**(YES) を使用する場合は、これを指定する必要があります。
4. .ini ファイルの変更が完了したら、コマンド REFRESH SECURITY TYPE(AUTHINFO) を発行するか、キュー・マネージャーを再始動します。

   構成が正常に完了すると、アプリケーションは署名済みトークンを使用して即時に接続できます。

   例えば、公開鍵を取得するために認証サービスに接続できないなどの問題がある場合、その問題はキュー・マネージャーの AMQERR01 ログ・ファイルに報告されます。

タスクの結果

JWKS エンドポイントを使用して認証トークンを受け入れるようにキュー・マネージャーが正常に構成されました。

注 : 鍵は、認証サーバーから (15 分ごとに) 定期的にリフレッシュされます。また、接続アプリケーションによって不明な鍵 ID が提示された場合は、より頻繁にリフレッシュされます。 これは通常、証明書の有効期限が切れてサーバー・サイドで置き換えられるため、証明書を更新するために追加の IBM MQ 構成アクションは必要ないことを意味します。 即時リフレッシュを強制するには、任意の時点でコマンド REFRESH SECURITY TYPE(AUTHINFO) を発行します。

関連概念

認証トークンの問題のトラブルシューティング

関連タスク

アプリケーションでの認証トークンの使用

関連資料

qm.ini ファイルの AuthToken スタンザ

Linux V 9.4.0 AIX ローカル鍵ストアを使用して認証トークンを受け入れるためのキュー・マネージャーの構成

認証トークンを使用してユーザーおよびアプリケーションを認証するように IBM MQ キュー・マネージャーを構成します。

始める前に

可能な場合は、トークン検証証明書を手動で構成するのではなく、JWKS エンドポイントの使用を検討してください。 331 ページの『JWKS エンドポイントを使用して認証トークンを受け入れるためのキュー・マ

ネージャーの構成』を参照してください。 通常、JWKS を使用すると、初期構成と継続的な保守の両方が簡素化されます。

トークンが IBM MQ でどのように動作するかについては、「認証トークンの処理」を参照してください。

キュー・マネージャーを構成する前に、キュー・マネージャーの **CONNAUTH** 属性で参照される AUTHINFO オブジェクトのタイプが IDPWOS であることを確認してください。 トークン認証は、キュー・マネージャーが OS ユーザー ID およびパスワード検査用に構成されている場合にのみ使用可能です。

サービス・スタンザの **SecurityPolicy** 属性が Group に設定されていないことを確認します。 **SecurityPolicy** が明示的に Group に設定されている場合、トークン認証は使用できません。 **SecurityPolicy** が Group に設定されている場合は、Service スタンザから **SecurityPolicy** 属性を削除してから、キュー・マネージャーを再始動します。

このタスクについて

IBM MQ 9.3.4 以降では、アプリケーションはトークンを使用してキュー・マネージャーで認証できます。 IBM MQ は、提案されたインターネット標準 RFC7519 に従う信頼できる発行者から JSON Web トークン (*JWT*) を受け入れます。 トークンを使用して ID を認証できます。認証された ID は、将来の許可検査に採用できます。

トラステッド発行者の公開鍵証明書または対称鍵をキュー・マネージャーの鍵リポジトリーに保存して、トークンを受け入れるようにキュー・マネージャーを構成します。 AuthToken スタンザを qm.ini ファイルに追加し、セキュリティー構成をリフレッシュして、キュー・マネージャーが新しい構成を取得できるようにします。

テスト環境で JWKS を使用するのではなく、ローカル鍵ストアを構成したい場合や、キュー・マネージャーから認証サーバーに直接接続できない場合などが考えられます。 JWKS エンドポイントに加えて、ローカル鍵ストアを定義することもできます。

注 : JWKS エンドポイントとローカル鍵ストアの両方が、提示されたトークンに一致する発行者と KID を提供する場合、JWKS エンドポイント提供の鍵が優先して使用されます。

このような場合は、ローカル鍵ストアを以下のように構成します。

手順

1. 鍵リポジトリーを作成します。
   a) トラステッド発行者から受け取った公開鍵証明書または対称鍵の鍵リポジトリーを作成します。 ファイル拡張子 .kdb を持つ CMS キー・リポジトリー、またはファイル拡張子 .p12 を持つ PKCS#12 キー・リポジトリーのいずれかを使用できます。

   以下のコマンドを発行して、 CMS キー・リポジトリーを作成します。

   ```
   runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
   -type cms
   ```

   runmqakm コマンドがエラーを返した場合は、runmqakm -keydb を参照してください。 コマンドが正常に完了したら、 ls コマンドを使用してディレクトリーの内容をリストします。

   ```
   ls -l /var/mqm/qmgrs/qm1/tokenissuer
   ```

   以下のファイルが表示されます。

   ```
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.crl
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.kdb
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.rdb
   ```

   b) 必要な場合は、mqm グループに読み取り権限を付与できるように、作成した鍵リポジトリー・ファイルのグループ所有権を変更します。 最初は、コマンドを実行した管理者ユーザーのみが、作成されたファイルにアクセスできます。

   ```
   chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
   ```

c) 鍵リポジトリー・ファイルのモードを変更して、グループ mqm の読み取り権限を追加します。 例えば、以下のコマンドは、ファイル所有者の読み取り/書き込み許可と、グループの読み取り専用許可を追加します。

```
chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

2. **runqmcred** コマンドを使用して鍵リポジトリー・パスワードを暗号化し、暗号化されたストリングをファイルに保存します。

a) 鍵リポジトリー・パスワードの暗号化に使用される初期鍵を含むファイルを作成します。

ファイルには、単一行のテキストとして初期キーが含まれている必要があります。 初期鍵の最大長は 256 バイトです。 **INITKEY** キュー・マネージャー属性を使用してキュー・マネージャーの初期鍵を既に設定している場合は、**INITKEY** 属性の値を新規ファイルにコピーします。 キュー・マネージャーの初期鍵をまだ設定していない場合は、新しい固有の暗号鍵を作成し、それを初期鍵ファイルに追加します。

注: 詳しくは、 INITKEY を参照してください。 初期キーを指定しない場合は、デフォルトのキーが使用されます。 独自の初期鍵を使用する方が安全です。

注: ファイルの内容をセキュアに保つために、初期鍵ファイルに対して必要な最小限の許可を付与します。 初期鍵ファイルは、鍵リポジトリー・パスワードの暗号化にのみ使用されます。 したがって、初期鍵を使用してパスワードを暗号化する管理者のみが、初期鍵ファイルの読み取りにアクセスする必要があります。

b) キュー・マネージャーの初期鍵がまだ設定されていない場合は、キュー・マネージャーの **INITKEY** 属性の値を、ステップ 334 ページの『2.a』で作成した初期鍵に設定します。 **ALTER QMGR** コマンドを使用して、キュー・マネージャーの初期鍵を設定します。 以下に例を示します。

```
ALTER QMGR INITKEY('myEncrypt10nK3y')
```

c) **runqmcred** コマンドを発行して、鍵リポジトリー・パスワードを暗号化します。 **-sf** パラメーターを使用して、初期キーを含むファイルへのパスを指定します。

```
runqmcred -sf initial.key
```

プロンプトが出されたら、鍵リポジトリー・パスワードを入力します。 暗号化されたパスワードは、コマンドによって出力されます。

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
**************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

最後の行のストリングをコピーして、ファイルに保存します。

3. 以下のいずれかの方法を使用して、トークン発行者の公開鍵証明書または対称鍵を鍵リポジトリーに追加します。

- RSA 公開鍵証明書を鍵リポジトリーに追加するには、次のコマンドを発行します。

```
runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile
```

- base64 エンコードの対称鍵を鍵リポジトリーに追加するには、次のコマンドを発行します。

```
runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile -format ascii
```

ここで、*keylabel* は証明書または秘密鍵に付加されるラベルであり、*keyfile* は証明書または base64 エンコード秘密鍵を含むファイルの名前です。

4. **AuthToken** スタンザおよび以下の属性を qm.ini ファイルに追加します。

- **KeyStore** 属性を使用して指定された、鍵リポジトリーへのパス。

- **KeyStorePwdFile** 属性を使用して指定された、鍵リポジトリーのパスワードを含むファイル。
- ステップ 334 ページの『3』で追加した証明書または対称鍵のラベル。 **CertLabel** 属性を使用して指定します。

以下に例を示します。

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
   CertLabel=rsakey
```

ここで、 key.kdb はステップ 333 ページの『1.a』で作成した鍵リポジトリーの名前、 key.pw はステップ 334 ページの『2.c』で作成した鍵リポジトリーの暗号化されたパスワードが含まれているファイルです。

AuthToken スタンザについて詳しくは、qm.ini ファイルの AuthToken スタンザを参照してください。

5. 後続の許可検査で使用するために、トークン・ユーザー要求に含まれるユーザー ID を採用するようにキュー・マネージャーが構成されている場合は、 **UserClaim** 属性を **AuthToken** スタンザに追加します。

   トークン内のユーザー ID を採用するようにキュー・マネージャーが構成されているかどうかを判別するには、次の MQSC コマンドを発行します。

   ```
   DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
   ```

   ここで、 *authinfo_name* は、キュー・マネージャーの **CONNAUTH** 属性の値です。 **ADOPTCTX** 属性の値が YES の場合、キュー・マネージャーはトークン内のユーザー ID を採用するように構成され、 **AuthToken** スタンザで **UserClaim** 属性を指定する必要があります。

   UserClaim 属性の値を、採用するユーザー ID を含むトークン・クレームの名前に設定します。 例えば、トークンにクレーム "AppUser": "MyUserName"が含まれている場合は、以下の行を **AuthToken** スタンザに追加します。

   ```
   UserClaim=AppUser
   ```

6. キュー・マネージャーのセキュリティー構成をリフレッシュして、 qm.ini ファイルからトークン構成が取得されるようにします。 以下のコマンドを発行して、 **runmqsc** コマンドを開始します。

   ```
   runmqsc qm1
   ```

   次に、次の MQSC コマンドを発行します。

   ```
   REFRESH SECURITY TYPE(CONNAUTH)
   ```

次のタスク

開発者と協力して、開発者がキュー・マネージャーで認証するために アプリケーションでトークンを使用する方法を理解できるようにします。

関連概念

認証トークンの問題のトラブルシューティング

関連タスク

アプリケーションでの認証トークンの使用

関連資料

qm.ini ファイルの AuthToken スタンザ

Linux V 9.4.0 AIX 選択したトークン発行者からの認証トークンの取得

IBM MQ キュー・マネージャーに接続するときに、選択したトークン発行者から認証トークンを取得するようにアプリケーションを作成します。

始める前に

336 ページの『アプリケーションでの認証トークンの使用』の情報を参照してください。

手順

- 認証トークンの取得方法、およびトークンの正確な内容は、トークン発行者によって異なります。
  選択したトークン発行者と対話して認証トークンを要求および取得するアプリケーションを作成します。 認証トークンは、認証トークンの IBM MQ 要件に準拠している必要があります。 これらの要件について詳しくは、 328 ページの『認証トークンの要件』を参照してください。
  トークン・クレームに含まれているユーザー ID をアプリケーションのコンテキストとして採用する場合は、認証トークンも以下の要件を満たしている必要があります。
  - 認証トークンには、キュー・マネージャーのトークン認証構成内のユーザー・クレーム名と一致するクレームが含まれている必要があります。
  - ユーザー・クレームの値は、認証トークン内のユーザー ID の要件を満たしている必要があります。詳細については、331 ページの『認証トークン内のユーザー ID』を参照してください。

タスクの結果

これで、正しい形式の JWT が取得されました。この JWT は、検証のために IBM MQ に提示できます。

関連タスク

AuthTokens を受け入れるためのキュー・マネージャーの構成

関連資料

qm.ini ファイルの AuthToken スタンザ

MQCSP - セキュリティー・パラメーター

Linux V 9.4.0 AIX アプリケーションでの認証トークンの使用

IBM MQ キュー・マネージャーに接続するときに認証トークンを提供するアプリケーションを作成します。

始める前に

IBM MQ 9.4.0 以降、アプリケーションはキュー・マネージャーへの接続時に認証トークンを提供できるようになりました。

アプリケーションは、以下の要件を満たしている必要があります。

- C または Java で作成する必要があります (IBM MQ classes for JMS/ Jakarta Messaging を使用)。
- IBM MQ client としてキュー・マネージャーに接続する必要があります。 つまり、アプリケーションは、ローカル・バインディングを使用する代わりに、ネットワークを介してキュー・マネージャーに接続する必要があります。
- AIX または Linux 上で稼働するキュー・マネージャーに接続する必要があります。

アプリケーションがこれらの要件を満たしていない場合、接続は失敗し、理由コード MQRC_FUNCTION_NOT_SUPPORTED (2298) がアプリケーションに戻されます。

認証トークンを提供するアプリケーションは、 IBM MQ MQI clients をサポートする任意のプラットフォームで実行できます。

自動クライアント再接続を使用するクライアントは、接続時に認証トークンを提供できません。 アプリケーションが認証トークンを提供し、MQCNO 構造体に MQCNO_RECONNECT または MQCNO_RECONNECT_Q_MGR オプションを指定すると、接続は失敗し、理由コード MQRC_RECONNECT_COMPATIBLE (2547) がアプリケーションに戻されます。 クライアントの自動再接続について詳しくは、クライアントの自動再接続を参照してください。

これらの要件のために認証トークンを提供するアプリケーションを作成できない場合は、代わりに、クライアント・セキュリティー出口を使用して、認証トークンを使用するようにアプリケーションをマイグレーションすることができます。 クライアント・セキュリティー出口を作成して、MQCSP 構造に認証トーク

ンを設定することができます。 セキュリティー出口について詳しくは、 クライアント接続のセキュリティー出口を参照してください。

IBM MQ 9.4.0 以降、 JMS クライアント・アプリケーションは、接続時にトークンを直接提供できます (335 ページの『選択したトークン発行者からの認証トークンの取得』を参照)。 IBM MQ 9.4.0 より前では、Java アプリケーションは、出口プログラムを介して間接的にトークンを提供することができます。 詳しくは、 Java クラス MQCSP を参照してください。

このタスクについて

注 : JSON Web Signature (JWS) 標準に準拠する認証トークンは、トークンの認証性を検証するために署名されますが、暗号化されません。 したがって、トークンにアクセスできるすべてのユーザーが読み取ることができ、場合によっては再利用することもできます。 キュー・マネージャーへの接続を構成して、認証トークンがネットワークを介して (例えば TLS を使用して) 送信されるときに暗号化を使用して保護されるようにします。 アプリケーションによって提供される資格情報を保護するためのオプションについて詳しくは、 31 ページの『MQCSP パスワード保護』を参照してください。

トークンを使用して接続するようにアプリケーションを変更する前に、以下を確認してください。

- キュー・マネージャーは、 332 ページの『ローカル鍵ストアを使用して認証トークンを受け入れるためのキュー・マネージャーの構成』の手順に従って認証トークンを受け入れるように構成されています。
- アプリケーションは、必要に応じて認証サーバーから有効なトークンを取得できます。 335 ページの『選択したトークン発行者からの認証トークンの取得』を参照してください。

アプリケーションが IBM MQ キュー・マネージャーに接続するときに認証トークンを提供するには、以下のプロセスを組み込みます。

手順

- C (MQI) アプリケーションから認証トークンを提供するには、次のようにします。
  アプリケーションは、(MQCONN ではなく) MQCONNX を使用して接続し、 MQCSP 構造を提供する必要があります。
  - **AuthenticationType** フィールドは MQCSP_AUTH_ID_TOKEN に設定する必要があります。
  - この構造体のバージョンは、 MQCSP_VERSION_3 に設定する必要があります。
  - **TokenPtr** または **TokenOffset** フィールドは、認証トークンを参照する必要があります。
  - **TokenLength** フィールドは、認証トークンの長さに設定する必要があります。

  MQCSP バージョン 3 と認証トークンを使用してキュー・マネージャーに接続する C コードの例:

```
MQCNO cno = {MQCNO_DEFAULT};   /* Connection options  */
MQCSP csp = {MQCSP_DEFAULT};   /* Security parameters */

char  token[MQ_CSP_TOKEN_LENGTH +1] = {0};  /* Authentication token string */

/* Set the connection options */
cno.SecurityParmsPtr = &csp;
cno.Version = MQCNO_VERSION_5;

/* Set the security parameters */
csp.Version = MQCSP_VERSION_3;
csp.AuthenticationType = MQCSP_AUTH_ID_TOKEN;
csp.TokenPtr = token;
csp.TokenLength = (MQLONG) strlen(token);

/* Connect to the queue manager */
MQCONNX(qmName,                 /* Queue manager name  */
        &cno,                   /* Connection options  */
        &hCon,                  /* Connection handle   */
        &compCode,              /* Completion code     */
        &reason);               /* Reason code         */
```

- Java アプリケーションから認証トークンを提供するには、以下のようにします。

IBM MQ classes for JMS/Jakarta Messaging を使用するアプリケーションは、ユーザー名とパスワードを使用する `createContext` メソッドまたは `createConnection` メソッドのいずれかを介してトークンを提供できます。

認証トークンを提供するには、以下のようにします。

- **UserID** は、ヌルまたは空ストリング (つまり、スペースなし) のいずれかに設定する必要があります。 `""`
- トークンは **Password** ストリングとして提供されます。

これは、 `ConnectionFactory` インターフェースのすべての IBM MQ 実装に適用されます。

明示的なパラメーター形式 (例えば、 `createContext(String` **userID**, String **password**) を使用することも、暗黙的なパラメーター・バージョン (例えば、 `createContext()`) を使用することもできます。

後者の場合、最初に空の **userID** およびトークン **Password** が接続ファクトリーのプロパティーとして指定されている必要があります。

認証トークンを使用してキュー・マネージャーに接続するための Java コード例:

```
// Obtain token from authentication provider here:

String myToken = "xxxxxxxxxxxxxxx";

// Acquire instance of an MQ connection Factory:

JmsFactoryFactory ff = JmsFactoryFactory.getInstance(WMQConstants.WMQ_PROVIDER);

JmsConnectionFactory cf = ff.createConnectionFactory();

// Configure any required CF properties here - e.g. MQ Channel details

// Connect to (and authenticate with) the queue manager:

context = cf.createContext(null, myToken); // NOTE - null userID indicates token being
provided
```

接続が理由コード MQRC_NOT_AUTHORIZED (2035) または MQRC_SECURITY_ERROR (2063) で失敗した場合は、キュー・マネージャーのエラー・ログで、失敗の原因に関する詳細情報を含むエラー・メッセージを確認してください。 認証トークンの問題の診断について詳しくは、認証トークンの問題のトラブルシューティングを参照してください。

タスクの結果

これで、アプリケーションはキュー・マネージャーに接続されました。 認証に使用されたトークンの有効期限が切れた場合でも、切断されるまで接続されたままになります。 アプリケーションがキュー・マネージャーから切断され、再接続する必要がある場合は、再接続できるようになる前に、有効期限が後の新しい認証トークンを取得する必要がある場合があります。

関連タスク

AuthTokens を受け入れるためのキュー・マネージャーの構成

関連資料

qm.ini ファイルの AuthToken スタンザ

MQCSP - セキュリティー・パラメーター

Linux V 9.4.0 AIX TLS トラストストアとして使用する鍵リポジトリーの作成

発信 TLS 接続を作成する際には、認証局 (CA) の共通セットによって署名された証明書を検証できる単純な「トラストストア」を作成する必要があります。 TLS 接続の例としては、IBM MQ の一部のコンポーネントの構成時に使用される IBM MQ クライアント・チャネルまたは HTTPS 接続があります。

このタスクについて

重要 : ご使用の環境でどの証明書および認証局を信頼するかを決定することは、エンドツーエンド構成のセキュリティーに影響を与える重要なステップです。 このトピックでは、ご使用のオペレーティング・システム用に既に構成されているのと同じ証明書のセットを IBM MQ コンポーネントが信頼できるようにするための共通ステップについて説明します。ただし、疑問点がある場合は、このプロセスについてセキュリティー管理者と相談する必要があります。

ほとんどの UNIX および Linux ベースのオペレーティング・システムには、「信頼できる」CA のセットを含むファイル・システム・ロケーションがあります。 このファイル・システムは、オペレーティング・システムのインストール済み環境で構成されているか、システム管理者によってカスタマイズされている可能性があります (例えば、組織に属する内部 CA を組み込むため)。 これらのファイルの場所はさまざまですが、一般的なオペレーティング・システムでよく使用される値には、以下のようなものがあります。

- AIX: `/var/ssl/cert.pem and/or /var/ssl/certs/*.crt`
- RHEL: `/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem`
- Ubuntu: `/etc/ssl/certs/*.pem`

IBM MQ 鍵ストアを作成して構成すると、`/etc/ssl/certs` などのディレクトリー内のすべての証明書ファイルを 1 つのコマンドで IBM MQ 鍵データベースに簡単に追加できます。

手順

1. 以下のコマンドを使用して、 `/etc/ssl/certs` ディレクトリーから証明書ファイルを追加します。

```
runmqakm -cert -add -file /etc/ssl/certs/*.pem -db mykdb.p12 -stashed
```

2. オプション: 場合によっては、トラストストア用の「デフォルト」の証明書セットを生成すると役立つことがあります。

   製品とともに提供される IBM MQ セキュリティー・コンポーネントには、一連の「デフォルト」CA 証明書が用意されています。

   注 : これらの証明書は、頻繁に更新されたり、比較的短い存続時間になったりすることはありません。

   事前構成された CA 証明書を使用する場合は、 **runmqakm** コマンドで **populate** パラメーターと **ibmcloudtrust** パラメーターを使用してトラストストアを生成できます。

```
runmqakm -keydb -create -db mqauto.p12 -genpw -stash -type pkcs12 -populate -ibmcloudtrust
```

関連概念
認証トークンの問題のトラブルシューティング
関連タスク
アプリケーションでの認証トークンの使用
関連資料
qm.ini ファイルの AuthToken スタンザ

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。

ALW AIX, Linux, and Windows 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。

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)。

IBM i IBM i では、 IBM MQ SSL サポートは、LDAP サーバー上でのみ CRL および ARL を使用して、取り消された証明書を検査します。

z/OS z/OS では、 IBM MQ SSL サポートは、LDAP サーバー上でのみ CRL および ARL を使用して、取り消された証明書を検査します。

認証局について詳しくは、13 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

企業で openSSL を使用して OCSP を検証してから IBM Global Security Kit (GSKit) TLS 接続を使用しようとすると、UNKNOWN 状況警告が表示されます。

これは、ルートを除くチェーン内のすべての証明書が、 GSKit によって失効状況について検査されるためです。 GSKit 操作は RFC 5280 に準拠しており、これについては GSKit トラスト・ポリシーで説明されています。 GSKit アルゴリズムは、RFC 5280 および GSKit トラスト・ポリシーで説明されているように、取り消し情報のために使用可能なすべてのソースを試行します。

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 によって使用される 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。

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。

ALW 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、 AIX, 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 AIX `qm.ini` 上の AIX 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 検査を実行す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、OCSP 応答側証明書のコピーを追加する必要があります。 551 ページの『AIX, Linux, and Windows での鍵リポジトリーへの CA 証明書またはトラステッド証明書の公開部分の追加』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。

プロパティー名	説明
ocsp.responderCertIssuerName	このプロパティーの値は、OCSP 応答側の証明書の発行者名です。 例えば、`ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"` です。 デフォルトでは、OCSP 応答側の証明書は、検証される証明書の発行者の証明書です。 このプロパティーは、デフォルトが適用されない場合に OCSP 応答側の証明書を特定します。 その値はストリングでの識別名 (RFC 2253 で定義) で、証明書パスの検証時に提供される一連の証明書の中から 1 つの証明書を特定します。 このプロパティーを設定する場合は、ocsp.responderCertSerialNumber プロパティーも設定する必要があります。 ocsp.responderCertSubjectName プロパティーが設定されている場合、このプロパティーは無視されます。
ocsp.responderCertSerialNumber	このプロパティーの値は、OCSP 応答側の証明書のシリアル番号です。 例えば、`ocsp.responderCertSerialNumber=2A:FF:00` です。 デフォルトでは、OCSP 応答側の証明書は、検証される証明書の発行者の証明書です。 このプロパティーは、デフォルトが適用されない場合に OCSP 応答側の証明書を特定します。 この値は 16 進数字のストリング (分離文字のコロンまたはスペースが存在することがあります) であり、証明書パスの検証時に提供される一連の証明書の中から、1 つの証明書を特定します。 このプロパティーを設定する場合は、ocsp.responderCertIssuerName プロパティーも設定する必要があります。 ocsp.responderCertSubjectName プロパティーが設定されている場合、このプロパティーは無視されます。

この方法で OCSP を有効にする前に、以下に示す、いくつかの点を考慮してください。

- OCSP 構成を設定すると、JVM プロセス内のすべてのセキュア・ソケットに影響します。 場合によっては、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。

- Multi Multiplatforms では、CRL および ARL のサポートは、PKIX X.509 V2 CRL プロファイルの推奨事項に準拠しています。
- z/OS 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。

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。

345 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。

346 ページの図 21 は、345 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。

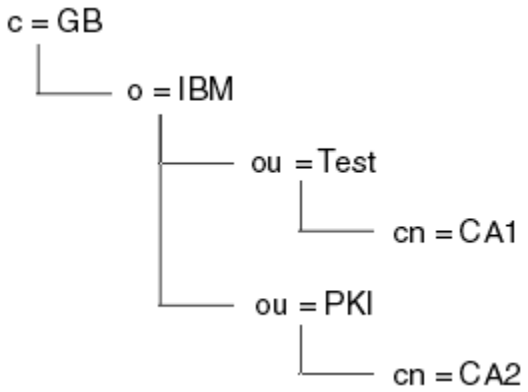


図 *21. LDAP Directory Information Tree* 構造例

IBM MQ は、CRL と ARL 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。 IBM 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 346 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、 347 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 (276 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、278 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 348 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 348 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) の認証情報オブジェクトは IBM i または z/OS キュー・マネージャーでの使用には適用されませんが、クライアントでの使用のためにクライアント・チャネル定義テーブル (CCDT) にコピーされるように、これらのプラットフォーム上で指定することはできます。

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 、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・プラットフォーム上では、AIX 、Linux、IBM i 、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 AIX プラグ可能認証方式 (PAM) の使用法

PAM は AIX and Linux プラットフォームのみで使用できます。 典型的な AIX システムまたは Linux 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。

ALW AIX, 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。

V 9.4.0 IBM MQ 9.3.4 以降では、認証のためにトークンを指定できます。 **ADOPTCTX** が YES に設定されている場合、トークンに含まれるクレームからユーザーが採用されます。

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、 352 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 71. 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
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。

ALW OAM によるオブジェクトへのアクセスの制御 (AIX, 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。

401 ページの『AIX, 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 AIX AIX and Linux での OAM ユーザーに基づく許可

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。 から IBM MQ 8.0ACL はユーザー ID とグループの両方に基づいており、ユーザーベースモデルまたはグループベースモデルのどちらかを認証に使用することができます。**SecurityPolicy** 属性を適切な値に設定するのサービススタンザ qm.ini ファイル。

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。

移行に関する考慮事項

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)

406 ページの『プリンシパルとグループ (AIX, 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。

関連資料

qm.ini ファイルの Service スタンザ

crtmqm (キュー・マネージャーの作成) コマンド

ALW AIX, 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、356 ページの『AIX, 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。 AIX 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。

ALW AIX, 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 : AIX and Linux システムでワイルドカード文字を使用しているとき、プロファイル名を単一引用符で囲む**必要があります**。

プロファイルの優先順位

汎用プロファイルの使用を理解する上で重要な点は、作成するオブジェクトに適用する権限を決定するときにプロファイルに与えられる優先順位です。 例えば、次のコマンドを発行するとします。

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

1 番目は、プロファイル AB. *; と一致する名前を持つプリンシパルのすべてのキューに対する書き込み権限を付与します。 2 番目のコマンドは、プロファイル AB.C*。

AB.CD と呼ばれるキューを作成するとします。 ワイルドカード突き合わせのルールによると、いずれかの setmqaut がそのキューに適用されます。 その場合、書き込み権限と読み取り権限のどちらが付与されるのでしょうか。

答えを見つけるために、複数のプロファイルを特定のオブジェクトに適用できるときには、必ず**最も特定されたプロファイルだけを適用する**というルールを適用します。 この規則を適用する方法として、プロファイル名は左から右に比較します。 違いを見つけた箇所では、必ず非総称文字が総称文字よりも限定的と

いうことになります。 このため、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 : AIX and Linux 上のユーザーは、 **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
```

ALW OAM プロファイルでのワイルドカード文字の使用 (AIX, 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 : AIX 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**。

ALW プロファイルの優先順位 (AIX, 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。

ALW プロファイル設定のダンプ *(AIX, 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 : AIX 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
```

ALW アクセス設定の表示 (AIX, 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
```

ALW IBM MQ オブジェクトへのアクセス権の変更と取り消し (AIX, 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 150 ページの『Windows でのグループの作成と管理』
- AIX 149 ページの『AIX でのグループの作成と管理』
- Linux 149 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。

ALW AIX, 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

注: このトピックでは、使用可能にすることが推奨されていない機能について説明します。 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) を使用不可にすることができます。 テスト環境では、その設定が適している場合もあります。 無効にすると、キュー・マネージャーは許可または接続の認証検査を実行できなくなります。 TLS、チャネル認証レコード、およびセキュリティー出口は引き続き使用できます。 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・システム環境変数 **MQSNOAUT** を設定します。

  MQSNOAUT 環境変数を設定した場合の影響、および AIX, Linux, and Windows 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。

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。

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。
- 認証検査のために OAM に渡される資格情報は検証されません。

関連概念

AIX, Linux, and 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

このトピックを使用して、ご使用の IBM MQ システムにセキュリティーを適用するために、どのタスクを実行すべきかを判別してください。

このタスクについて

このタスクでは、ご使用の 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: 365 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: 374 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: 379 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: 381 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: 382 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: 383 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 72. 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	366 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	367 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	368 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	369 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

表 72. 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 (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**
Processes	370 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	371 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	373 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

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

- ALW
  AIX, 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。

- ALW AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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。

- ALW AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
On AIX, 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。

- ALW AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

  - ALW AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

- ALW AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

  - ALW On AIX, 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。」

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

  - ALW AIX, 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 73. 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	374 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	375 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	376 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	376 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	377 ページの『いくつかのプロセスへの全管理アクセス権の付与』の説明に従い、必要なプロセスへの全管理アクセス権を付与する
名前リスト	378 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	379 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW

  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW

  On AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

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。

- ALW

AIX, 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 ** -t queue -g GroupName +browse +dsp
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get
+put
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

SYSTEM.ADMIN.COMMAND.QUEUE 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

注 : ALW

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 AIX
  AIX 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、AIX 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。

マルチプラットフォーム では、DELETE AUTHREC コマンドも使用できます。

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

- ALW

  AIX, 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 (390 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 (383 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、390 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	391 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	397 ページの『トピックへのユーザー・アクセスの制御』を参照。
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	399 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。
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	400 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	400 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
384 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、320 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  On AIX, 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

203 ページの『Connection security profiles for the channel initiator』
Profiles for checking connections from the channel initiator are composed of the queue manager or queue sharing group name followed by the word *CHIN*. Give the user ID used by the channel initiator started task address space READ access to the connection profile.

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	392 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。

記述	アクション
アプリケーションがコンテキストを設定する	393 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	394 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	478 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	395 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	395 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	396 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- Windows
  AIX, 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
  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
  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にすることもできます。

  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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
  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
  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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
  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
  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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

  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

  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。 395 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW

  AIX, 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

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

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)
```

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 74. 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	397 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	398 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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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
  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
  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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
  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
  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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
  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
  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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
  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
  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。

Multi Multiplatforms では、 SET AUTHREC コマンドを使用することもできます。

手順

- ALW
  AIX, 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

  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

  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。

ALW AIX, 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 Script (MQSC) コマンドを発行する制御コマンド **runmqsc** を使用するために、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。

AIX 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、406 ページの『プリンシパルとグループ (AIX, 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 118 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。

ALW mqm グループの管理 (AIX, 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
- 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。

ALW IBM MQ 上で AIX, 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

401 ページの『AIX, 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。

ALW セキュリティー検査が行われるタイミング (AIX, 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 137 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。

ALW AIX, 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 (364 ページの『AIX, 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。 353 ページの『OAM によるオブジェクトへのアクセスの制御 (AIX, 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、353 ページの『OAM によるオブジェクトへのアクセスの制御 (AIX, 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。

ALW ユーザー ID の識別 (AIX, 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。)

AIX 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、408 ページの『コンテキスト権限 (AIX, 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。

ALW プリンシパルとグループ (AIX, 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。

- AIX AIX 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 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 AIX AIX and Linux システム

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、設定することでどちらかを認証に使用できます。**SecurityPolicy** 属性を適切な値に設定するのサービススタンザ qm.ini ファイル。

許可に ユーザー・ベースのモデル を使用できます。これにより、ユーザーとグループの両方を使用できます。 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、353 ページの『AIX and Linux 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。

から IBM MQ 9.3.0、あなたは UserExternal オプションの **SecurityPolicy** 非オペレーティングシステム ユーザー名を作成するための属性。 非オペレーティング・システム・ユーザー名を作成すると、ユーザーは nobody グループ以外のグループには属していないと見なされます。 このオプションについて詳しくは、crtmqm と qm.ini ファイルの Service スタンザを参照してください。

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。

LDAP の使用方法については、414 ページの『許可の設定』を参照してください。

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、AIX and Linux 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。

設定の詳細については、**SecurityPolicy** 属性オン Windows、 見るのサービススタンザ qm.ini ファイル。

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。

ALW 代替ユーザー権限 (AIX, 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。

Linux Linux での特定のグループ・メンバーシップ問題の解決

一部のシステムでは、通常の一連の **getgrent** オペレーティング・システム API 呼び出しを使用してグループ情報を戻すのに時間がかかる場合があります。また、mqm ユーザーがどのグループを対象としているかを検索するために数千ものグループが存在する場合、遅い応答によって内部キュー・マネージャーのタイムアウトが発生する可能性があります。 この問題を回避するに、代替オペレーティング・システム API があります。

より高速な代替 API を使用し、1 回の呼び出しからすべてのグループを戻すには、環境変数 MQS_GETGROUPLIST_API を設定します。

ユーザーの 2 次グループへの接続アクセス権限を付与し、MQS_GETGROUPLIST_API 変数を使用可能にすると、問題が緩和される可能性があるため、RC2035 エラーを受け取った可能性があります。

その後、 IBM MQ は **getgrent** API の代わりに **getgrouplist** API を使用します。

を使用可能にする場合 **getgrouplist**：

1. キュー・マネージャーを停止する。
2. コマンド・エクスポートの発行 MQS_GETGROUPLIST_API=1
3. キュー・マネージャーを再始動します。

失敗したシナリオを再試行してください。問題が解決した場合は、ユーザー mqm の .bashrc / .profile ファイルを変更してこの環境変数を追加するか、キュー・マネージャーの開始に使用するスクリプトに環境変数を追加することを検討してください。

ご使用のシステムが、NIS または LDAP などの複数のリポジトリーからオペレーティング・システムのユーザー情報またはグループ情報をマージする場合は、グループまたはユーザー ID が、オペレーティング・システム・レベルのアクセス権をインストールおよび設定するために使用されるものとして、ローカルのリポジトリーを含むすべてのリポジトリーで一貫していることを確認します。

ALW コンテキスト権限 (AIX, 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 を参照してください。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-メッセージ記述子 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 の場合は 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、323 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。

Multi ストリーミング・キューのセキュリティー

ストリーミング・キューの機能を使用すると、管理者は 2 次キューを持つローカル (またはモデル) キューを構成できます。2 次キューは、元のキューにメッセージが入れらると複製したメッセージが配置される場所です。 キューのストリーミング権限に関しては、考慮すべき点が 2 つあります。

複製メッセージのストリーミング用のキューを構成する権限

メッセージを 1 つのキューから 2 次キューに複製するメッセージのストリーミングを有効にするには、そのための許可が必要です。 キューの **STREAMQ** 属性を構成するための許可を得るには、以下の権限が必要です。

1. **STREAMQ** 属性を変更するキューの CHG 権限
2. 複製メッセージを入れるキューの CHG 権限

構成時にこれら 2 つの権限の組み合わせがチェックされることで、元のキューの CHG 権限のみを持つユーザーが、権限を持たない他方のキューにメッセージを入れられないようにすることができます。

キューをオープンしてメッセージを入れる権限

2 次キューで構成されたキューをアプリケーションがオープンすると、アプリケーション・ユーザーに元のキューに対する PUT 権限があることを確認する権限検査が **STREAMQ** 属性を通して行われます。

注 : 2 次キューではアプリケーション・ユーザーに対する追加の権限検査は行われません。このことは、別名キューで使用される権限モデルと類似しています。

元のキューまたは 2 次キューのいずれかからメッセージを取り込むアプリケーションには、取り込み元のキューでのみ、GET 権限または BROWSE 権限が必要です。

メッセージを入れるときや取得する時に追加の権限検査は行われません。

例

以下の例は、ユーザー admin が元のキュー INQUIRIES.QUEUE から複製したメッセージをローカル・キュー ANALYTICS.QUEUE にストリーミングできる一方で、admin がメッセージを PURCHASES.QUEUE には複製できないように設定する適切な権限を示しています。

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(ANALYTICS.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(PURCHASES.QUEUE) PRINCIPAL('admin') AUTHRMV(CHG)
```

これにより、ユーザー admin は以下のコマンドを発行できます。

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

ただし、同じユーザーが以下のコマンドを発行すると、

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

重複メッセージを PURCHASES.QUEUE に書き込むように INQUIRIES.QUEUE を構成するには、以下のエラーを受け取ります。

```
AMQ8135E 許可されていません
```

INQUIRIES.QUEUE が ANALYTICS.QUEUE にメッセージを複製するように構成されていると、ユーザー appuser として実行しているアプリケーションが、INQUIRIES.QUEUE にメッセージを入れ、ANALYTICS.QUEUE にメッセージを複製できるように、以下の権限レコードが使用されます。

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

注 : appuser は、ANALYTICS.QUEUE に対する権限レコードを必要としません。 複数メッセージはキュー・マネージャーによってキューに入れられます。

関連概念

ストリーミング・キュー

z/OS Streaming queues security on z/OS

The streaming queues feature allows an administrator to configure a local (or model) queue with a secondary queue, where duplicate messages are placed, whenever a message is put to the original queue. There are two aspects to consider regarding queue streaming authorities.

Authority to configure a queue for streaming duplicate messages

If you want to enable message streaming of duplicate messages from one queue to a secondary queue, you must have permission to do so. Permission to configure the **STREAMQ** attribute of a queue requires that you have the following profiles setup:

1. ALTER access level to MQADMIN or MXADMIN for the queue they are altering the **STREAMQ** attribute for
2. ALTER access level to MQADMIN or MXADMIN for the queue you want to stream messages to

The combination of these security checks at configuration time ensures that a user, who only has ALTER access on the original queue, cannot cause messages to be put to another queue on which they have no permissions.

Authority to open the queue or queues and put messages

When an application opens a queue that has been configured with a secondary queue, through its **STREAMQ** attribute, an authority check is made that the application user has UPDATE authority on the original queue.

Note: No additional authority check is made for the application user on the secondary queue, which is similar to the authority model used for alias queues.

Applications consuming messages from either the original or the secondary queue require UPDATE or READ authority, only on the queue they are consuming from.

No additional authority checks are made at put or get time.

Example

The following example shows the correct profiles being set to allow user ADMIN to configure an original queue, INQUIRIES.QUEUE, to stream messages to local queue ANALYTICS.QUEUE using RACF:

```
RDEFINE MQCMDS <QMGR>.ALTER.QLOCAL UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.ALTER.QLOCAL CLASS(MQCMDS) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.INQUIRIES.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.INQUIRIES.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.ANALYTICS.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.ANALYTICS.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)
```

User ADMIN is then able to issue the following command:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

but if the same user issues the following command without setting up the correct security profiles:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

to configure INQUIRIES.QUEUE to put duplicate messages to PURCHASES.QUEUE, they receive the following error:

`CSQM166I <QMGR> CSQMAQLC QLOCAL(INQUIRIES.QUEUE) NOT AUTHORIZED`

Related concepts

Streaming queues

Multi LDAP 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 Multiplatforms で使用可能です。

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。

Linux AIX 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 AIX and Linux 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。

Multi 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。

413 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 75. 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。

SET AUTHREC MQSC コマンドは、 **setmqaut** コマンドの代わりに使用できます。

```
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、415 ページの表 76 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 76. 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

表 76. グループ許可設定 (続き)	
コマンド	**注記**
`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

SET AUTHREC MQSC コマンドは、上記の **setmqaut** コマンドの代わりに使用できます。

```
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。

Multi 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、それに相当する MQSC または PCF コマンドは、414 ページの『許可の設定』で説明されている **setmqaut** 表のように、サポートされている任意の形式で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)
```

Multi 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。

Multi 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。

IBM i, AIX and Linux

Linux IBM i AIX

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。

Multi 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。

AIX and Linux

Linux AIX

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。

Windows

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 AIX and Linux 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。

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。

MQADMIN 特権を付与するためのサンプル・スクリプト

Linux AIX

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、AIX and Linux 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 -g
      groupname
```

メッセージの機密性

メッセージを暗号化すると、メッセージの内容の機密性が保たれます。 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。

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。

最も安全なソリューションは、アプリケーションによってメッセージが書き込まれる時点からコンシューム側アプリケーションがそのメッセージを入手する時点まで、メッセージを暗号化することにより、エンドツーエンドの暗号化を提供することです。 これは、111 ページの『Advanced Message Security の計画』(AMS) を使用して行うか、独自の API 出口または API 交差出口を作成することによって行うことができます。詳しくは、467 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。

ネットワークでのトランスポート中のみメッセージを暗号化する必要がある場合は、TLS を使用できます。詳しくは、24 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。あるいは、独自のセキュリティー出口、メッセージ出口、または送信出口プログラムを作成して暗号化を実行することもできます。

z/OS キュー・マネージャーで保存メッセージを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できます。詳しくは、468 ページの『Confidentiality for data at rest on IBM MQ for z/OS with data set encryption』 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

CipherSpecs の有効化

CipherSpec は、**DEFINE CHANNEL** 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。

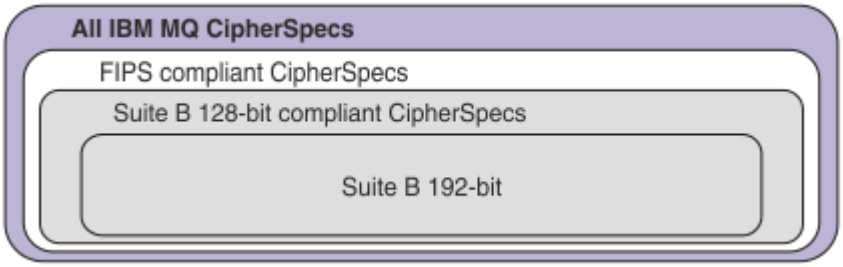
注 : AIX, Linux, and Windows では、 IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。 このモジュールの証明書は「履歴」ステータスに移動されました。 お客様は、IBM Crypto for C (ICC) 証明書 を表示し、NIST から提供されたアドバイスに注意する必要があります。 交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、 「NIST CMVP modules in process list」でそのモジュールを検索します。

IBM MQ Operator 3.2.0 およびキュー・マネージャー・コンテナー・イメージ 9.4.0.0 以降は、UBI 9 に基づいています。 FIPS 140-3 準拠は現在保留中であり、その状況を表示するには、 「NIST CMVP modules in process list」で「Red Hat Enterprise Linux 9- OpenSSL FIPS Provider」を検索します。

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。



この製品は、すべてのプラットフォームで TLS 1.3 セキュリティー・プロトコルをサポートしています。

これらのそれぞれのプラットフォームで使用できる CipherSpec は、421 ページの表 77 にリストされています。 これらの CipherSpecs の使用方法については、424 ページの『IBM MQ での TLS 1.3 の使用』 および 425 ページの『IBM MQ MQI client および TLS 1.3』 を参照してください。

構成や将来のマイグレーションを容易にするために、IBM MQ では、別名 CipherSpec のセットも提供されます。 既存のセキュリティー構成を、別名 CipherSpec を使用するように移行することは、今後さらに侵略的な構成変更を行わなくても、暗号化の追加と非推奨に適合できることを意味します。 これらの別名 CipherSpec は、421 ページの表 77 の『別名 CipherSpec』セクションにリストされています。 別名 CipherSpec を使用するためのマイグレーションについて詳しくは、別名 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を参照してください。

デフォルトの CipherSpec は、425 ページの『IBM MQ で有効化されているデフォルト CipherSpec 値』 の説明に従って構成できます。 以下の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。

- Multi IBM MQ for Multiplatforms (434 ページの『順序付けされた有効な CipherSpec のカスタム・リストの提供 (IBM MQ for Multiplatforms)』 を参照)。
- z/OS IBM MQ for z/OS (435 ページの『順序付けされた有効な CipherSpec のカスタム・リストの提供 (IBM MQ for z/OS)』 を参照)。

必要に応じて、IBM MQ で使用するために再有効化できる非推奨の CipherSpec のリストについては、435 ページの『推奨されない CipherSpec』 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 CipherSpec 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 77. *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 424 ページの『1』	CipherSpec 名	16 進コード	使用されるプロトコル	MAC アルゴリズム	暗号化アルゴリズム (暗号化ビット)	FIPS 424 ページの『2』	Suite B
別名 CipherSpecs							

表 77. *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 424 ページの『1』	CipherSpec 名	16 進コード	使用されるプロトコル	MAC アルゴリズム	暗号化アルゴリズム (暗号化ビット)	FIPS 424 ページの『2』	Suite B
すべて	ANY_TLS13_OR_HIGHER 424 ページの『3』 424 ページの『4』	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 424 ページの『4』 424 ページの『5』	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 424 ページの『4』 424 ページの『6』	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 424 ページの『7』	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 424 ページの『8』	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
TLS 1.3 の CipherSpec							
すべて	TLS_AES_128_GCM_SHA256	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	1302	TLS 1.3	GCM	AES-256 と GCM (256)	はい	いいえ
すべて	TLS_CHACHA20_POLY1305_SHA256	1303	TLS 1.3	POLY1305	CHACHA20 (256)	いいえ	いいえ
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 (CTR (128) を使用)	はい	いいえ
ALW	TLS_AES_128_CCM_8_SHA256 424 ページの『10』	1305	TLS 1.3	CBC-MAC	AES-128 (CTR (128) を使用)	はい	いいえ
TLS 1.2 の CipherSpec							

表 77. *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 424 ページの『1』	CipherSpec 名	16 進コード	使用されるプロトコル	MAC アルゴリズム	暗号化アルゴリズム (暗号化ビット)	FIPS 424 ページの『2』	Suite B
すべて	TLS_RSA_WITH_AES_128_CBC_SHA256 424 ページの『9』	003C	TLS 1.2	SHA-256	AES (128)	はい	いいえ
すべて	TLS_RSA_WITH_AES_256_CBC_SHA256 424 ページの『9』 424 ページの『11』	003D	TLS 1.2	SHA-256	AES (256)	はい	いいえ
すべて	TLS_RSA_WITH_AES_128_GCM_SHA256 424 ページの『9』 424 ページの『12』	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 424 ページの『9』 424 ページの『11』 424 ページの『12』	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 424 ページの『9』	C023	TLS 1.2	SHA-256	AES (128)	はい	いいえ
すべて	ECDHE_ECDSA_AES_256_CBC_SHA384 424 ページの『9』 424 ページの『11』	C024	TLS 1.2	SHA-384	AES (256)	はい	いいえ
すべて	ECDHE_RSA_AES_128_CBC_SHA256 424 ページの『9』	C027	TLS 1.2	SHA-256	AES (128)	はい	いいえ
すべて	ECDHE_RSA_AES_256_CBC_SHA384 424 ページの『9』 424 ページの『11』	C028	TLS 1.2	SHA-384	AES (256)	はい	いいえ
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 424 ページの『11』 424 ページの『12』	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 424 ページの『11』 424 ページの『12』	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 424 ページの『12』	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 424 ページの『11』 424 ページの『12』	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	はい	いいえ

表 77. *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 424 ページの『1』	CipherSpec 名	16 進コード	使用されるプロトコル	MAC アルゴリズム	暗号化アルゴリズム (暗号化ビット)	FIPS 424 ページの『2』	Suite B

注 :

1. 各プラットフォーム・アイコンでカバーされるプラットフォームのリストについては、 製品資料で使用される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. ALW 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. 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。
5. ALW 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。
6. ALW 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。
7. 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。
8. ALW 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。
9. 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。
10. ALW 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。
11. 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。
12. ALW 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。

    z/OS 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

この製品は、すべてのプラットフォームで TLS 1.3 をサポートします。

IBM MQ 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 をサポートします。 以前のバージョンの IBM MQ からマイグレーションしたキュー・マネージャーでは、TLS 1.3 を使用可能にする必要があります。 **AllowTLSV13**=TRUE プロパティーを以下のように設定することにより、マイグレーションしたキュー・マネージャーで TLS 1.3 を有効化できます。

- Multi IBM MQ for Multiplatforms キュー・マネージャーの場合は、qm.ini ファイルを編集し、SSL スタンザの下に **AllowTLSV13**=TRUE プロパティーを追加します (リンク先)

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS IBM MQ for z/OS のキュー・マネージャーの場合は、キュー・マネージャーの始動 JCL で指定されている QMINI データ・セットを編集し、TransportSecurity スタンザの下に **AllowTLSV13**=TRUE プロパティーを追加します

```
TransportSecurity:
    AllowTLSV13=TRUE
```

TLS 1.3 が有効にされている場合は、TLS 1.3 の仕様に従って、脆弱な CipherSpec を使用した通信の試みは、それらが IBM MQ で有効になっている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 [3] 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、次のようにして TLS 1.3 モードを無効にする必要があります。

- ALW 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
```

- z/OS キュー・マネージャーの QMINI データ・セットを編集し、**AllowTLSV13** プロパティーの設定を次のように変更します。

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client および TLS 1.3

ALW

IBM MQ MQI client 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

新しい IBM MQ キュー・マネージャーのデフォルト構成では、IBM MQ は、TLS 1.2 プロトコルと TLS 1.3 プロトコル、および 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になっている CipherSpecs のリストは、保守を適用することで変更される可能性があります。

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。
- ALW SUITEB を使用して、NSA Suite B 準拠 CipherSpec のみを許可します。
- Multi **AllowedCipherSpecs** を使用して、CipherSpec のカスタム・リストを許可します。
- ALW **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。
- ALW **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。

関連概念

47 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。

22 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。

44 ページの『Suite B 用 IBM MQ の構成』
IBM MQ は、AIX, Linux, and Windows 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。

34 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

別名 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

ALW 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 $2^{24.5}$ 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 IBM Global Security Kit (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。

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、 420 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 $^{24.5}$ 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。

TLS ハンドシェークでの CipherSpec の順序

CipherSpec の順序は、ANY* CipherSpec の 1 つを使用する場合など、複数の使用可能な CipherSpec の中から選択する場合に使用されます。

TLS ハンドシェーク時に、クライアントとサーバーは、サポートする CipherSpec とプロトコルを、優先順位の順に交換します。 両側で優先されている共通の CipherSpec が選択され、TLS 通信に使用されます。CipherSpec プロトコルを選択するときには、バージョンも考慮されます。例えば、サーバーで TLS 1.3 CipherSpec の前に TLS 1.2 CipherSpec がリストされている場合でも、クライアントが TLS 1.3 をサポートでき、使用可能な共通の TLS 1.3 CipherSpec を持っているならば、TLS 1.3 が優先されます。

IBM MQ が TLS 用に構成されている場合、 CipherSpecs は、以下の表に示す順序 (最も優先されるものから最も優先されないものの順) に設定されます。

注 : CipherSpec は、**AllowedCipherSpecs** 属性を介して有効にされていない場合、TLS ハンドシェーク中に使用されるように構成されません。

AllowedCipherSpecs 属性が指定されていない場合は、以下の表で示されている有効な暗号のデフォルト・リストが使用されます。

表 *78. IBM MQ 9.2.0* 以降の *CipherSpec* の順序

プラットフォーム	CipherSpec	プロトコル	16 進コード	デフォルトで有効
すべて	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	はい
すべて	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	はい
すべて	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	はい
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	はい
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	はい
すべて	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	はい
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	はい
すべて	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	はい
すべて	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	はい
すべて	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	はい

表 78. *IBM MQ 9.2.0* 以降の *CipherSpec* の順序 (続き)

プラットフォーム	CipherSpec	プロトコル	16 進コード	デフォルトで有効
すべて	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	はい
すべて	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	はい
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	はい
すべて	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	はい
すべて	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	はい
すべて	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	はい
すべて	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	はい
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	いいえ
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	いいえ
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	いいえ
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	いいえ
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	いいえ
すべて	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	いいえ
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	いいえ
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	いいえ
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	0000	いいえ
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	いいえ

表 78. IBM MQ 9.2.0 以降の CipherSpec の順序 (続き)				
プラットフォーム	**CipherSpec**	**プロトコル**	**16 進コード**	**デフォルトで有効**
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	いいえ
IBM i	AES_SHA_US	TLS 1.0	002E	いいえ
すべて	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	いいえ
すべて	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	いいえ
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	いいえ
すべて	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	いいえ
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	いいえ
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	いいえ
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	いいえ
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	いいえ
すべて	TRIPLE_DES_SHA_US	SSL v3	000A	いいえ
すべて	RC4_SHA_US	SSL v3	0005	いいえ
すべて	RC4_MD5_US	SSL v3	0004	いいえ
すべて	DES_SHA_EXPORT	SSL v3	0009	いいえ
すべて	RC4_MD5_EXPORT	SSL v3	0003	いいえ
すべて	RC2_MD5_EXPORT	SSL v3	0006	いいえ
すべて	NULL_SHA	SSL v3	0002	いいえ
すべて	NULL_MD5	SSL v3	0001	いいえ
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	FEFF	いいえ
ALW	RC4_56_SHA_EXPORT1024	SSL v3	0064	いいえ
ALW	DES_SHA_EXPORT1024	SSL v3	0062	いいえ
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	FEFE	いいえ

このリストは、IBM MQ で z/OS が使用する暗号ライブラリーによって提供されるデフォルト・リストを使用してプロトコルをオーダーすることによって構成され、z/OS および分散プラットフォーム間で一貫性があります。

順序の変更

別の順序が必要な場合は、 IBM MQ for Multiplatforms z/OS、または IBM MQ for z/OS の TransportSecurity スタンザ、 の SSL スタンザの **AllowedCipherSpecs** 属性を使用して、以下の規則で CipherSpecs の新しい順序を指定できます。

- リスト内での位置に関係なく、常に、より高いプロトコル・バージョンが使用されます。
- 無効にされている CipherSpec は、リストに指定されると再有効化されます。
- TLS サーバーのリスト順序の優先順位が、TLS クライアントよりも高くなっています。
- TLS 1.3 が有効になっている場合、特定の CipherSpec はサポートされません。

例えば、IBM MQ for Multiplatforms のキュー・マネージャー上で以下が構成されている場合:

```
SSL:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

z/OS また、IBM MQ for z/OS のキュー・マネージャー上で以下が構成されている場合:

```
TransportSecurity:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

次に以下を行います。

- ANY_TLS12 で接続するクライアントは、TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256 を使用する可能性が高くなります。
- ANY_TLS12_OR_HIGHER で接続するクライアントは、TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256 を使用する可能性が高くなります (クライアントが TLS 1.3 をサポートする場合)。
- TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA で接続するクライアントは、この CipherSpec を使用します。

IBM MQ の以前のバージョン

IBM MQ 9.2.0 より前では、以下の CipherSpec の順序が使用されていました。

表 *79. IBM MQ 9.2.0* より前の *CipherSpec* の順序

プラットフォーム	CipherSpec	プロトコル	デフォルトで有効
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	いいえ
IBM i	AES_SHA_US	TLS 1.0	いいえ
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	いいえ
すべて	RC4_SHA_US	SSL v3	いいえ
すべて	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	いいえ

表 79. IBM MQ 9.2.0 より前の CipherSpec の順序 (続き)			
プラットフォーム	**CipherSpec**	**プロトコル**	**デフォルトで有効**
すべて	RC4_MD5_US	SSL v3	いいえ
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	いいえ
すべて	TRIPLE_DES_SHA_US	SSL v3	いいえ
すべて	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	いいえ
ALW	DES_SHA_EXPORT1024	SSL v3	いいえ
すべて	RC4_56_SHA_EXPORT1024	SSL v3	いいえ
すべて	RC4_MD5_EXPORT	SSL v3	いいえ
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	いいえ
すべて	RC2_MD5_EXPORT	SSL v3	いいえ
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	いいえ
すべて	DES_SHA_EXPORT	SSL v3	いいえ
すべて	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	いいえ
すべて	NULL_SHA	SSL v3	いいえ
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	いいえ
すべて	NULL_MD5	SSL v3	いいえ
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	いいえ
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	いいえ
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	いいえ
すべて	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	はい
すべて	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	はい
すべて	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	いいえ
すべて	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	はい
すべて	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	はい

表 79. *IBM MQ 9.2.0 より前の CipherSpec の順序 (続き)*

プラットフォーム	CipherSpec	プロトコル	デフォルトで有効
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	いいえ
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	いいえ
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	いいえ
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	いいえ
すべて	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	はい
すべて	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	はい
すべて	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	はい
すべて	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	はい
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	はい
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	はい
すべて	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	はい
すべて	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	はい
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	いいえ
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	いいえ
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	いいえ
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	いいえ
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	はい
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	はい
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	はい
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	はい
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	はい

重要 : 2020 年 7 月 23 日の時点で、以下の AllowedCipherSpecs 属性は、現在デフォルトで有効になっている CipherSpec のみを有効にします。 ただし、この日付以降に非推奨になった CipherSpec が、不注意で再有効化されないように、以下の AllowedCipherSpecs 属性によって有効になっている CipherSpec を現在のデータで確認する必要があります。

この CipherSpec の順序に戻す必要がある場合は、次に示す SSL/TransportSecurity スタンザの **AllowedCipherSpecs** 属性値を使用します。

```
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,ECDHE_ECDSA_AES_128_CBC_SHA256,
ECDHE_ECDSA_AES_256_CBC_SHA384,ECDHE_RSA_AES_128_CBC_SHA256,ECDHE_RSA_AES_256_CBC_SHA384,
ECDHE_ECDSA_AES_128_GCM_SHA256,ECDHE_ECDSA_AES_256_GCM_SHA384,ECDHE_RSA_AES_128_GCM_SHA256,
ECDHE_RSA_AES_256_GCM_SHA384
```

順序付けされた有効な CipherSpec のカスタム・リストの提供 (IBM MQ for Multiplatforms)

Multi

ALW **AMQ_ALLOWED_CIPHERS** 環境変数 または .ini ファイルの **AllowedCipherSpecs** SSL スタンザ属性のいずれかを使用して、 IBM MQ チャネルで使用するために、有効化された CipherSpecs の代替セットを希望の順序で提供することができます。 この設定は、以下のいずれかの理由で使用できます。

- 指定された CipherSpec のいずれかを使用しない限り、IBM MQ リスナーが着信チャネル開始要求を受け入れないように制限するため。
- TLS ハンドシェークで使用される CipherSpec の優先順位を変更するため。

この機能を使用して、ANY* CipherSpecs に含まれている CipherSpecs を制御できます。

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 CipherSpec 名のコンマ区切りリスト。
- すべての CipherSpec を表わす特殊値 ALL

注 : ALL CipherSpecs を有効化すると、SSL 3.0 プロトコルと TLS 1.0 プロトコルに加えて多数の脆弱な暗号アルゴリズムが有効化されるため、これは有効化しないでください。

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
```

さらに、このリスト内の CipherSpec は、TLS ハンドシェーク中に使用される CipherSpec の優先順位を決定するために使用されます。 例えば、TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 というリストを指定した場合は、クライアントがこれらの CipherSpec の両方を指定して接続すると (つまり、クライアントが ANY_TLS12 で接続すると)、ハンドシェーク中に TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec ではなく TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec が選択されることが予想されます。

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。

順序付けされた有効な CipherSpec のカスタム・リストの提供 (IBM MQ for z/OS)

z/OS

QMINI データ・セットの **AllowedCipherSpecs** TransportSecurity スタンザ属性を使用して、IBM MQ チャネルで使用する CipherSpecs の代替セットを好みの順序で提供することができます。 これは、以下のいずれかの目的で行うことができます。

- 指定された CipherSpec のいずれかを使用しない限り、IBM MQ リスナーが着信チャネル開始要求を受け入れないように制限するため。
- TLS ハンドシェークで使用される CipherSpec の優先順位を変更するため。

この機能を使用して、ANY* CipherSpec に含まれる CipherSpec を制御できます。 **AllowedCipherSpecs** 属性に指定できるものは、以下のとおりです。

- 単一の CipherSpec 名。
- 再有効化する CipherSpec 名のコンマ区切りリスト。
- すべての CipherSpec を表わす特殊値 ALL

  注 : ALL CipherSpecs を有効化すると、SSL 3.0 プロトコルと TLS 1.0 プロトコルに加えて多数の脆弱な暗号アルゴリズムが有効化されるため、これは有効化しないでください。 この設定を構成すると、デフォルトの CipherSpec リストがオーバーライドされ、IBM MQ は弱い暗号化の非推奨設定を無視するようになります。439 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 リスナーは、指定されたいずれかの CipherSpec を使用する SSL/TLS プロポーザルのみを受け入れます。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。

例えば、チャネルの定義/変更とリスナーによる ECDHE_RSA_AES_128_GCM_SHA256 または ECDHE_RSA_AES_256_GCM_SHA384 の受け入れだけを許可する場合は、次のように設定できます。

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

さらに、このリスト内の CipherSpec は、TLS ハンドシェーク中に使用される CipherSpec の優先順位を決定するために使用されます。 例えば、TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 というリストを指定した場合は、クライアントがこれらの CipherSpec の両方を指定して接続すると (つまり、クライアントが ANY_TLS12 で接続すると)、ハンドシェーク中に TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec ではなく TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec が選択されることが予想されます。

Deprecated 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。

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 *80. 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438 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)	FIPS438 ページの『2』	Suite B	非推奨時の更新
SSL 3.0 の CipherSpec								

表 80. 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 438 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 438 ページの『2』	Suite B	非推奨時の更新
IBM i	AES_SHA_US 438 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 438 ページの『3』 438 ページの『4』 438 ページの『5』	0009	SSL 3.0	SHA-1	DES (56)	いいえ	いいえ	9.0.0.0
ALW	DES_SHA_EXPORT1024 438 ページの『3』 438 ページの『6』	0062	SSL 3.0	SHA-1	DES (56)	いいえ	いいえ	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA 438 ページの『3』	FEFE	SSL 3.0	SHA-1	DES (56)	いいえ 438 ページの『7』	いいえ	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA 438 ページの『3』	FEFF	SSL 3.0	SHA-1	3DES (168)	いいえ 438 ページの『8』	いいえ	9.0.0.1 および 9.0.1
すべて	NULL_MD5 438 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 438 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 438 ページの『3』 438 ページの『4』 438 ページの『5』	0006	SSL 3.0	MD5	RC2 (40)	いいえ	いいえ	9.0.0.0
すべて	RC4_MD5_EXPORT 438 ページの『4』 438 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 438 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 438 ページの『3』 438 ページの『5』	0005	SSL 3.0	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 438 ページの『3』 438 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 438 ページの『3』 438 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 438 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 438 ページの『3』 438 ページの『4』	0003	TLS 1.0	MD5	RC4 (40)	いいえ	いいえ	9.0.0.0

表 80. 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 438 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 438 ページの『2』	Suite B	非推奨時の更新
すべて	TLS_RSA_WITH_DES_CBC_SHA 438 ページの『3』	0009	TLS 1.0	SHA-1	DES (56)	いいえ 438 ページの『9』	いいえ	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 438 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 438 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 438 ページの『3』	0004	TLS 1.0	MD5	RC4 (128)	いいえ	いいえ	9.0.0.0
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA 438 ページの『10』	002F	TLS 1.0	SHA-1	AES (128)	はい	いいえ	9.0.5
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA 438 ページの『6』 438 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								
ALW	ECDHE_ECDSA_NULL_SHA256 438 ページの『3』	C006	TLS 1.2	SHA-1	なし	いいえ	いいえ	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 438 ページの『3』	C007	TLS 1.2	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 438 ページの『3』	C010	TLS 1.2	SHA-1	なし	いいえ	いいえ	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 438 ページの『3』	C011	TLS 1.2	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL 438 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 438 ページの『3』	003B	TLS 1.2	SHA-256	なし	いいえ	いいえ	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 438 ページの『3』	0005	TLS 1.2	SHA-1	RC4 (128)	いいえ	いいえ	9.0.0.0
ALW	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

表 80. *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 438 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 438 ページの『2』	Suite B	非推奨時の更新
ALW 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

注:

1. 各プラットフォーム・アイコンでカバーされるプラットフォームのリストについては、 製品資料で使用される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. ALW 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・マネージャーでは、デフォルトで TLS 1.3 が有効です。これによってこれらの CipherSpecs は無効になります。 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。 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. Deprecated 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. Deprecated `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. 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。

非推奨の CipherSpec の有効化 (IBM MQ for 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。 IBM MQ for Multiplatforms 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 CipherSpec 名のコンマ区切りリスト
- すべての CipherSpec を表わす特殊値 ALL

重要 : ALL は有効なオプションですが、ALL CipherSpec を再有効化すると SSL 3.0 および TLS 1.0 のプロトコルに加えて多くの脆弱な暗号アルゴリズムが有効化されるため、このオプションは、自社で必要とされる特定の状況で**のみ**使用してください。

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、CSQX616E、または CSQX674E が表示されます。

こうしたメッセージのいずれかが表示される状況で、自社の会社が脆弱な CipherSpec の使用を再び有効にする必要がある場合は、このセクションにリストされている手順に従ってください。

重要 : 以下の手順では、ダミー定義 (DD) ステートメントを有効にするために、SSLTASKS はゼロ以外の値である必要があります。 これに伴い SSLTASKS を変更する必要がある場合は、チャネル・イニシエーターをリサイクルする必要があります。

IBM MQ for z/OS では、脆弱であるか、または壊れている CipherSpec を制御する方法は、現時点で以下のとおりです。

- 脆弱な CipherSpec の使用を再度有効にする場合は、CSQXWEAK という名前のダミーのデータ定義 (DD) ステートメントをチャネル・イニシエーター JCL に追加します。 単独で指定した場合は、TLS 1.2 プロトコルに関連した脆弱な CipherSpec だけが有効になります。以下に例を示します。

  ```
  //CSQXWEAK DD DUMMY
  ```

  注 : すべての非推奨の CipherSpec でこの DD ステートメントを使用する必要はありません。前出の表の注 10 を参照してください。
- SSLv3 CipherSpec の使用を再度有効にする場合は、CSQXSSL3 という名前のダミーの DD ステートメントをチャネル・イニシエーター JCL に追加します。 SSLv3 CipherSpec はすべて**脆弱**と見なされるので、CSQXWEAK も指定する必要があります。

  ```
  //CSQXSSL3 DD DUMMY
  ```

- 非推奨の TLS V1 CipherSpec を再度有効にする場合は、TLS10ON (TLS V1.0 をオンにするという意味) という名前のダミーの DD ステートメントをチャネル・イニシエーター JCL に追加します。 単独で指定した場合は、TLS 1.0 プロトコルに関連した強力な CipherSpec が有効になります。

  ```
  //TLS10ON DD DUMMY
  ```

  CSQXWEAK と一緒に指定した場合は、TLS 1.0 に関連した**脆弱な** CipherSpec も有効になります。
- 非推奨の TLS V1 CipherSpecs を明示的にオフにする場合は、以下のように TLS10OFF (TLS V1.0 をオフにするという意味) という名前のダミーの DD ステートメントを次のようにチャネル・イニシエーター JCL に追加します。

  ```
  //TLS10OFF 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
```

重要 : IBM MQ for z/OS 9.2.0 以降では、チャネル・イニシエーターの始動時に有効なプロトコルと有効ではないプロトコルを示すためにメッセージを表示する際、以前にリストされた DD カードおよび **AllowTLSV13** の値が考慮されます。 したがって、以前にリストされていた DD カードのいずれかが指定されているとしても、それは、これらの設定の組み合わせのために、あるプロトコルを別のプロトコルと一緒に有効にできないことを意味している可能性があります。 例えば、TLS 1.3 が有効になっている場合、プロトコル SSL 3.0 は許可されません。

データ定義の変更が適さない場合、脆弱な CipherSpecs および SSLv3 サポートを強制的に再有効化するために使用できる別のメカニズムがあります。 詳細については、IBM サービスに連絡してください。

関連概念

47 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

ここでは、クライアント構成とサーバー構成での別名 CipherSpec のさまざまな組み合わせで予期される動作について説明します。 ここでは、クライアントは、通信を開始するエンティティー (クライアント・アプリケーションやキュー・マネージャーの送信側チャネルなど) を指し、サーバーは、クライアントから通信を受信するエンティティー (サーバー接続チャネルや受信側チャネルなど) を指します。

最小プロトコルの CipherSpec と固定プロトコルの CipherSpec

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。

最小プロトコル
: 最小プロトコルの CipherSpec は、上限を設定しない CipherSpec です (例えば、ANY、ANY_TLS12_OR_HIGHER、ANY_TLS13_OR_HIGHER)。

固定プロトコル
: 固定プロトコルの CipherSpec は、特定のプロトコルを指定する CipherSpec です (例えば、ANY_TLS12 および ANY_TLS13。または ECDHE_ECDSA_3DES_EDE_CBC_SHA256 などの特定のアルゴリズム)。

すべてのプラットフォームで、最小および固定のプロトコル CipherSpecs がサポートされます。

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**最小プロトコル**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**最小プロトコル**の CipherSpec を使用し、受信側で**固定プロトコル**の CipherSpec を使用すると、接続が拒否され、以下のことが起きる可能性があります。

- Multi メッセージ AMQ9631 および AMQ9641 が発行される。
- z/OS メッセージ CSQX631E および CSQX641E が発行される。

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。 441 ページの表 81 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。 441 ページの表 82 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 : 以下の表の中で「失敗する可能性が高い」というマークが付いているセルがあります。これは、接続の一方の側に**最小プロトコル**の CipherSpec を指定し、もう一方の側に特定の (**固定プロトコル**の) CipherSpec を指定すると、対立が発生する可能性がある、という意味です。

例えば、クライアントとサーバーが ANY CipherSpec を使用するように設定されていて、サーバー・チャネルが特定の CipherSpec を使用するように設定されているとします。

- クライアントとサーバーの両方にとっての最も強力なサポート対象 CipherSpec が、チャネルで構成されているその特定の CipherSpec と一致する場合は、TLS ハンドシェークによる解決が正常に実行されます。
- しかし、それよりも強力な CipherSpec をクライアントとサーバーの両方がサポートしている場合は、チャネルで指定されている CipherSpec と一致しなくても、その CipherSpec が TLS ハンドシェークによる解決で使用されることになり、結果として TLS ハンドシェークが失敗します。

表 *81.* 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 *82.* 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

表 82. 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_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
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

47 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。

22 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。

420 ページの『CipherSpecs の有効化』
CipherSpec は、**DEFINE CHANNEL** 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。

420 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。

z/OS IBM i CipherSpec の代替指定方法

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、420 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。

注 : このセクションは、AIX, Linux, and 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。

IBM i **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 **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。

注 : Deprecated SSL V3.0 または TLS 1.0 といった脆弱な CipherSpec、または推奨されないプロトコルに属する CipherSpec を使用するには、 チャネル・イニシエーターの始動 JCL に関連する DD カードを指定する必要があります。 詳しくは、435 ページの『推奨されない CipherSpec』 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、420 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、481 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 447 ページの『制約事項』 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。

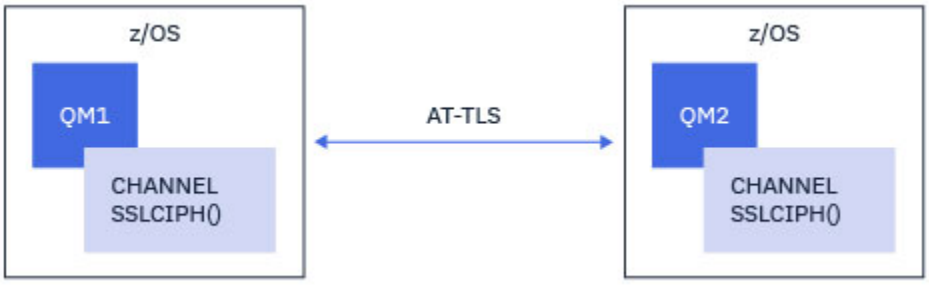
チャネルで TLS を使用可能にすることに決定したら、IBM MQ 管理者は、AT-TLS または IBM MQ TLS を使用するかどうかを決定できます。 多くの場合、この決定は AT-TLS が他のミドルウェアに使用されているか、パフォーマンスへの影響があるかに基づいて行われます。 AT-TLS と IBM MQ TLS のパフォーマンスの基本的な比較については、MP 16: の容量の計画と、IBM MQ for z/OS のチューニングを参照してください。

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。



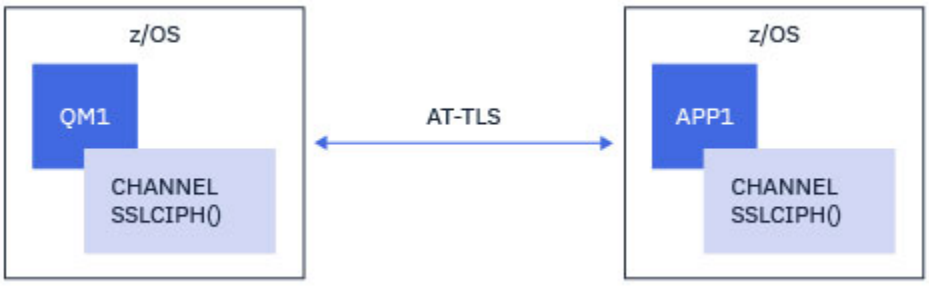
このシナリオの実装は、チャネルの両側に 1 つずつ、2 つの AT-TLS ポリシーを定義することから構成されます。 これらのポリシーは、「シナリオ 3」 または 「シナリオ 4」 のいずれかで使用されるポリシーと同じです。

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 448 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 457 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。

チャネルが別名 CipherSpec を使用して AT-TLS を使用するように変更されていた場合は、アウトバウンド・チャネルは 452 ページの『別名 CipherSpecs を使用した IBM MQ for Multiplatforms キュー・マネージャーへのアウトバウンド・チャネルでの AT-TLS の構成』 からのポリシーを使用し、インバウンド・チャネルは 461 ページの『別名 CipherSpec を使用した 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。



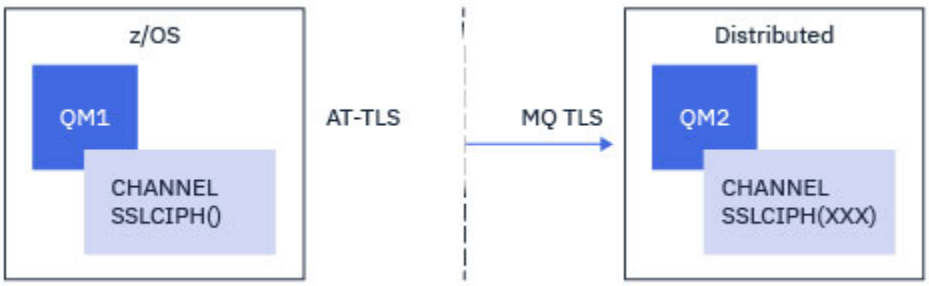
このシナリオの実装は、チャネルの両側に 1 つずつ、2 つの AT-TLS ポリシーを定義することから構成されます。 これらのポリシーは、「シナリオ 3」 または 「シナリオ 4」 のいずれかで使用されるポリシーと同じです。

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 448 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 457 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。

例えば、チャネルが、別名 CipherSpec を使用して AT-TLS を使用するように変更されていた場合は、クライアント接続チャネルは 452 ページの『別名 CipherSpecs を使用した IBM MQ for Multiplatforms キュー・マネージャーへのアウトバウンド・チャネルでの AT-TLS の構成』 からのポリシーを使用し、サーバー接続チャネルは 461 ページの『別名 CipherSpec を使用した 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

間の IBM MQ for z/OS キューマネージャとキューマネージャが IBM MQ for Multiplatforms、 どこ 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 MQTLS では、SSLCIPH 属性を単一の名前付き文字列で指定することで、CipherSpec。 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。

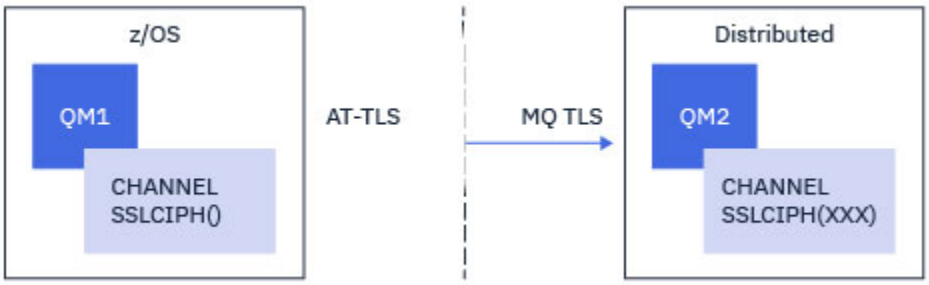


448 ページの『単一の名前付き CipherSpec を使用して、アウトバウンド・チャネル上の AT-TLS を IBM MQ for Multiplatforms キュー・マネージャーに構成する』 キュー・マネージャーから IBM MQ for z/OS キュー・マネージャーへのアウトバウンド・チャネル用の AT-TLS 構成例、および IBM MQ for Multiplatforms キュー・マネージャーから 457 ページの『単一の名前付き CipherSpec を使用して、IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネル上での AT-TLS 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キュー・マネージャー間で、IBM MQ for z/OS キュー・マネージャーが AT-TLS を使用し、IBM MQ for Multiplatforms キュー・マネージャーが IBM MQ TLS を使用する場合は、エイリアス CipherSpec を使用して SSLCIPH 属性を指定します。 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。

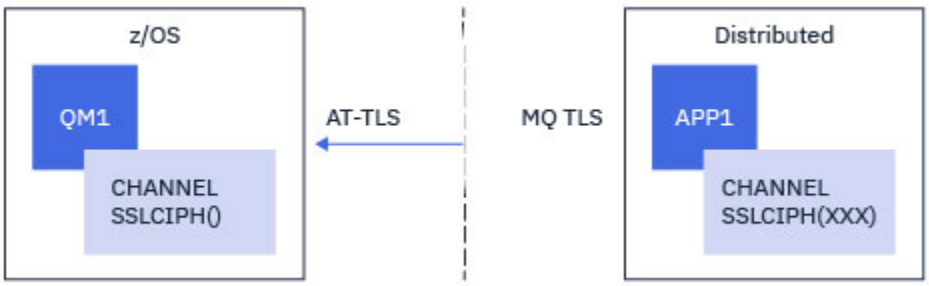


452 ページの『別名 CipherSpecs を使用した IBM MQ for Multiplatforms キュー・マネージャーへのアウトバウンド・チャネルでの AT-TLS の構成』 キュー・マネージャーから IBM MQ for z/OS キュー・マネージャーへのアウト・バウンドチャネルの AT-TLS 設定例については IBM MQ for Multiplatforms を参照し、461 ページの『別名 CipherSpec を使用した IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネルでの AT-TLS の構成』 キュー・マネージャーから 461 ページの『別名 CipherSpec を使用した 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・チャネルの AT-TLS 設定例については、IBM MQ for Multiplatforms および 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 5

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。

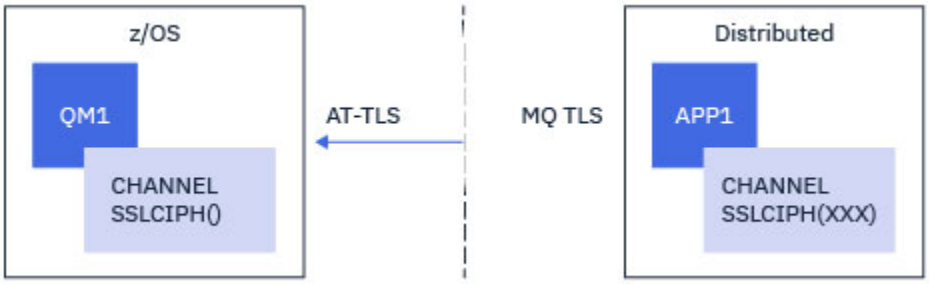


このシナリオでは、インバウンド・メッセージ・チャネルによって使用される要件と同じ要件を満たす単一の AT-TLS ポリシーが必要です。457 ページの『単一の名前付き CipherSpec を使用して、IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネル上での AT-TLS の構成』 を参照してください。

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。

シナリオ 6

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。461 ページの『別名 CipherSpec を使用した 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。

制約事項

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、465 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。

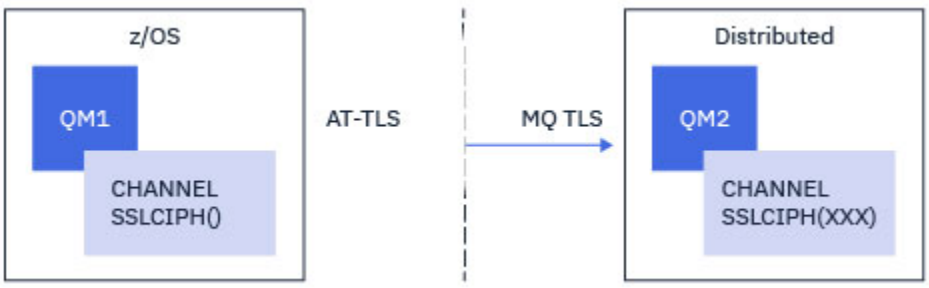
重要 : ここには記載されていない、AT-TLS の他の「AT-TLS ポリシー・ステートメント」があり、必要に応じて IBM MQ で使用することもできます。 ただし、IBM MQ は、このトピックで説明されているポリシーでのみテストされています。

z/OS 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。

別名 CipherSpec を使用した例については、452 ページの『別名 CipherSpecs を使用した IBM MQ for Multiplatforms キュー・マネージャーへのアウトバウンド・チャネルでの AT-TLS の構成』 を参照してください。

この例では、TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec を使用する既存の送信側チャネルと受信側チャネルのペアが調整され、送信側チャネルが IBM MQ TLS の代わりに AT-TLS を使用するようになります。



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。255 ページの『Configuring your z/OS system to use TLS』 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、IBM MQ CipherSpec 名と一致する AT-TLS 暗号スイート名は、以下の表で IBM MQ CipherSpec 名を検出し、「TTLSCipherParms」 ステートメント・トピックの表 2 の拡張文字列と 16 進コード列を相互参照することによって見つけることができます。

表 *83. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs*

CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	はい
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	はい
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	はい
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	はい
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	はい
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	はい
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	はい
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	はい
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	はい
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	はい
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	はい
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	はい
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	はい
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	いいえ
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	いいえ

表 83. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs (続き)			
CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	いいえ
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	いいえ
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	いいえ
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	いいえ
TRIPLE_DES_SHA_US	SSL v3	000A	いいえ
RC4_SHA_US	SSL v3	0005	いいえ
RC4_MD5_US	SSL v3	0004	いいえ
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	いいえ
RC2_MD5_EXPORT	SSL v3	0006	いいえ
NULL_SHA	SSL v3	0002	いいえ
NULL_MD5	SSL v3	0001	いいえ

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_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         OFF
  TLSv1.3         ON
}
```

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_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(SDR) SSLCIPH(' ')
```

ステップ 4: チャネルを開始する

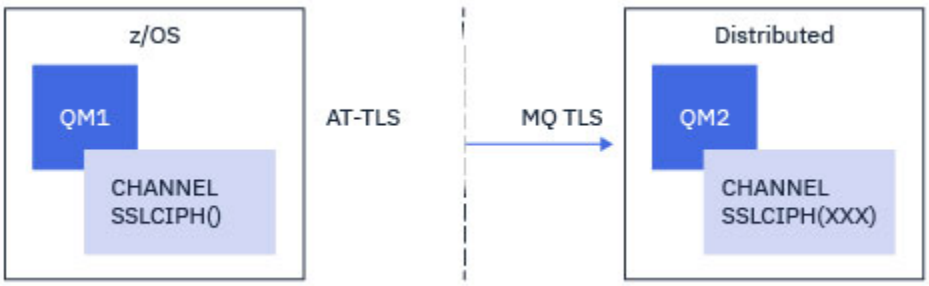
チャネルが開始されると、そのチャネルは AT-TLS と IBM MQ TLS の組み合わせを使用します。

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。

z/OS 別名 *CipherSpecs* を使用した *IBM MQ for Multiplatforms* キュー・マネージャーへのアウトバウンド・チャネルでの *AT-TLS* の構成

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、別名 CipherSpec に設定された SSLCIPH 属性を持つ送信側チャネルです。

この例では、ANY_TLS13 の別名 CipherSpec を使用する既存の送信側と受信側チャネルのペアが調整され、送信側チャネルが IBM MQ TLS の代わりに AT-TLS を使用するようになります。



その他の TLS プロトコルおよび CiperSpec は、構成を微調整することによって使用できます。 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。255 ページの『Configuring your z/OS system to use TLS』 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、1 つ以上の暗号スイート名が含まれている必要があります。少なくとも 1 つは、ターゲットの受信側チャネルで使用される別名 CipherSpec によって暗黙指定される CipherSpec のセットと互換性がなければなりません。

   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、IBM MQ CipherSpec 名と一致する AT-TLS 暗号スイート名は、以下の表で IBM MQ CipherSpec 名を検出し、「TTLSCipherParms」 ステートメント・トピックの表 2 の拡張文字列と 16 進コード列を相互参照することによって見つけることができます。

表 *84. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs*

CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	はい
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	はい
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	はい
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	はい
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	はい
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	はい
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	はい
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	はい
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	はい
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	はい

表 84. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs (続き)			
CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	はい
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	はい
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	はい
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	いいえ
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	いいえ
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	いいえ
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	いいえ
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	いいえ
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	いいえ
TRIPLE_DES_SHA_US	SSL v3	000A	いいえ
RC4_SHA_US	SSL v3	0005	いいえ
RC4_MD5_US	SSL v3	0004	いいえ
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	いいえ
RC2_MD5_EXPORT	SSL v3	0006	いいえ
NULL_SHA	SSL v3	0002	いいえ
NULL_MD5	SSL v3	0001	いいえ

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

重要 : キュー・マネージャーと AT-TLS ポリシーの両方が TLS 1.3 をサポートする場合、少なくとも 1 つの TLS 1.3 CipherSpec を含む別名 CipherSpecs のみがチャネルの開始を許可します。 例えば、 ANY_TLS12 を使用すると、 TTLSCipherParms に TLS 1.2 CipherSpecs が含まれていてもチャネルの開始に失敗しますが、 ANY_TLS12_OR_HIGHER または ANY_TLS13 を使用するとチャネルの開始が許可されます。 説明は 440 ページの『別名 CipherSpec 設定間の関係』 を参照してください。

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は、どの SSL プロトコルおよび TLS プロトコルを有効にするかを指定するために使用でき、 TTLSCipherParms ステートメント内の暗号スイートと整合している必要があります。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

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_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

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。

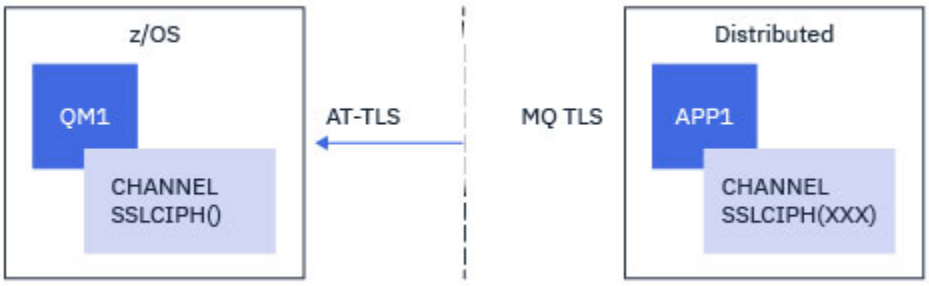
重要 : 上記の AT-TLS ステートメントは、最小限の構成にすぎません。 ここには記載されていない、AT-TLS の他の「AT-TLS ポリシー・ステートメント」があり、必要に応じて IBM MQ で使用することもできます。 ただし、 IBM MQ は、記載されているポリシーでのみテストされています。

z/OS 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。

別名 CipherSpec を使用した例については、461 ページの『別名 CipherSpec を使用した IBM MQ for Multiplatforms キュー・マネージャーからのインバウンド・チャネルでの AT-TLS の構成』 を参照してください。

この例では、TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec を使用する既存の送信側チャネルと受信側チャネルのペアが調整され、送信側チャネルが IBM MQ TLS の代わりに AT-TLS を使用するようになります。



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。255 ページの『Configuring your z/OS system to use TLS』 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、IBM MQ CipherSpec 名と一致する AT-TLS 暗号スイート名は、以下の表で IBM MQ CipherSpec 名を検出し、「TTLSCipherParms」 ステートメント・トピックの表 2 の拡張文字列と 16 進コード列を相互参照することによって見つけることができます。

表 *85. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs*

CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	はい
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	はい
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	はい
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	はい

表 85. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs (続き)

CipherSpec	プロトコル	16 進コード	デフォルトで有効
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	はい
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	はい
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	はい
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	はい
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	はい
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	はい
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	はい
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	はい
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	はい
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	いいえ
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	いいえ
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	いいえ
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	いいえ
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	いいえ
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	いいえ
TRIPLE_DES_SHA_US	SSL v3	000A	いいえ
RC4_SHA_US	SSL v3	0005	いいえ
RC4_MD5_US	SSL v3	0004	いいえ
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	いいえ
RC2_MD5_EXPORT	SSL v3	0006	いいえ
NULL_SHA	SSL v3	0002	いいえ
NULL_MD5	SSL v3	0001	いいえ

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_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        OFF
  TLSv1.3        ON
}
```

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                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

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

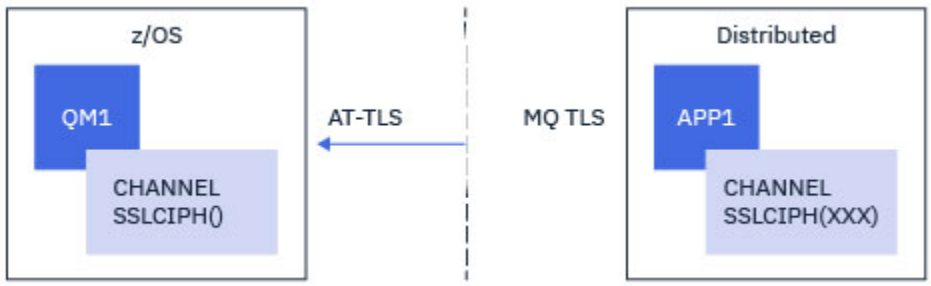
チャネルが開始されると、そのチャネルは AT-TLS と IBM MQ TLS の組み合わせを使用します。

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。

z/OS 別名 *CipherSpec* を使用した *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、別名 CipherSpec に設定された SSLCIPH 属性を持つ送信側チャネルです。

この例では、すべての TLS 1.3 CipherSpec を使用する既存の送信側と受信側のチャネル・ペアが調整され、受信側チャネルが IBM MQ TLS の代わりに AT-TLS を使用するようになります。



その他の TLS プロトコルおよび CiperSpec は、構成を微調整することによって使用できます。 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。255 ページの『Configuring your z/OS system to use TLS』 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、リモート送信側チャネル上の別名 CipherSpec セットに含まれている暗号スイート名が少なくとも 1 つ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、IBM MQ CipherSpec 名と一致する AT-TLS 暗号スイート名は、以下の表で IBM MQ CipherSpec 名を検出し、「TTLSCipherParms」 ステートメント・トピックの表 2 の拡張文字列と 16 進コード列を相互参照することによって見つけることができます。

表 *86. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs*

CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	はい
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	はい

表 86. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs (続き)

CipherSpec	プロトコル	16 進コード	デフォルトで有効
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	はい
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	はい
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	はい
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	はい
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	はい
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	はい
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	はい
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	はい
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	はい
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	はい
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	はい
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	いいえ
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	いいえ
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	いいえ
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	いいえ
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	いいえ
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	いいえ
TRIPLE_DES_SHA_US	SSL v3	000A	いいえ
RC4_SHA_US	SSL v3	0005	いいえ
RC4_MD5_US	SSL v3	0004	いいえ
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	いいえ
RC2_MD5_EXPORT	SSL v3	0006	いいえ

表 86. z/OS から IBM MQ for z/OS 9.2.0 の CipherSpecs (続き)			
CipherSpec	**プロトコル**	**16 進コード**	**デフォルトで有効**
NULL_SHA	SSL v3	0002	いいえ
NULL_MD5	SSL v3	0001	いいえ

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

重要 : キュー・マネージャーと AT-TLS ポリシーの両方が TLS 1.3 をサポートする場合、少なくとも 1 つの TLS 1.3 CipherSpec を含む別名 CipherSpecs のみがチャネルの開始を許可します。 例えば、 ANY_TLS12 を使用すると、 TTLSCipherParms に TLS 1.2 CipherSpecs が含まれていてもチャネルの開始に失敗しますが、 ANY_TLS12_OR_HIGHER または ANY_TLS13 を使用するとチャネルの開始が許可されます。 説明は 440 ページの『別名 CipherSpec 設定間の関係』 を参照してください。

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は、どの SSL プロトコルおよび TLS プロトコルを有効にするかを指定するために使用でき、 TTLSCipherParms ステートメント内の暗号スイートと整合している必要があります。

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

   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                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

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。

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** にパラメーター **SSLRKEYC** を指定して使用し、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 では、 **SSLRSTCNT** パラメーターを指定して **CHGMQM** を使用します。

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。
2. 環境変数 **MQSSLRESET** を使用する。
3. クライアント構成ファイルの SSL スタンザで **SSLKeyResetCount** 属性を設定します。

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。

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 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示します。 IBM MQ classes for .NET でのオブジェクト・プロパティーの使用について詳しくは、 属性値の取得および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 クライアントとの整合性を保つために、 IBM MQ 管理対象 .NET クライアントでは、アプリケーションで **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
表示キュー・マネージャー
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、467 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。

z/OS Confidentiality for data at rest on IBM MQ for z/OS with data set encryption

IBM MQ for z/OS can harden customer and configuration data by writing the data to the active log data sets, the archive log data sets, page sets, boot strap data sets (BSDS), and shared message data sets (SMDS).

z/OS provides efficient, policy-based encryption of data sets. IBM MQ for z/OS supports z/OS data set encryption for:

- Active log data sets; see note “1” on page 469
- Archive log data sets; see note “2” on page 469

- Page sets; see note “1” on page 469
- BSDS; see note “2” on page 469
- CSQINP* data sets; see note “2” on page 469
- SMDS; see note “1” on page 469

This provides confidentiality of data at rest on an individual z/OS queue manager.

Notes:

1. From IBM MQ for z/OS 9.2.0, z/OS data set encryption for active logs. page sets, and SMDS are supported.
2. Data set encryption for archive logs, BSDS and CSQINP* data sets is supported on all versions of IBM MQ for z/OS.
3. IBM MQ Advanced Message Security provides an alternative mechanism of protecting data at rest. In addition AMS also protects data in memory and in flight

See Using the z/OS data set encryption enhancements for more information about z/OS data set encryption.

Configuration of z/OS data set encryption is outside of the control of IBM MQ for z/OS. Encryption settings take effect when the data set is created.

This means that any existing data sets need to be recreated before a new data set encryption policy can be used.

IBM MQ for z/OS can run with a mixture of encrypted and non-encrypted data sets, but a standard configuration would encrypt all, or none, of the data sets used.

z/OS Overview of steps to encrypt an IBM MQ for z/OS data set

How you encrypt an IBM MQ for z/OS data set.

Before you begin

You must ensure that you have configured z/OS data set encryption correctly in your enterprise. If you are setting up data set encryption in a queue sharing group, you must configure z/OS data set encryption for data sharing.

Note: A z/OS encrypted data set must be an extended format data set.

Procedure

1. Set up encryption key and `key-label` in RACF to use to encrypt the data set.
2. Create a profile for `key-label` in the RACF CSFKEYS class.
3. Grant READ access to the user Id of the queue manager, and any other user Ids that need access to the encrypted data.

   This might include user IDs that are used to run print utilities against the data set. For example, the user running CSQUTIL SCOPY would need to decrypt the relevant page set.
4. Associate the encryption `key-label` with the data set name.

   You can do this by using an SMS data class, or a RACF DFP segment, for the data set name or high-level qualifier.

   You can also associate the `key-label` with the data set when the data set is allocated.
5. Rename any existing data set using IDCAMS ALTER.
6. Re-allocate the data set with the appropriate attributes.
7. Copy the contents of the renamed data set to the new data set using IDCAMS REPRO.

   The data is encrypted by the action of copying it into the data set.
8. Repeat steps “4” on page 469 to “6” on page 469 for any other data sets that need to be encrypted.

z/OS Example of how to encrypt queue manager active logs

The following topics guide you through the process of enabling data set encryption on existing active logs.

Note: The process for other data sets is similar to that for active logs.

In this example:

- Queue manager CSQ1 is run under user QMCSQ1, and has active log data sets CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, and so on
- The hardware and software environment is capable of using z/OS data set encryption
- RACF is used as the SAF
- The queue manager has been stopped

Carry out the procedure in the following order:

1. “Configuring the data set encryption key for the queue manager” on page 470
2. “Configuring data set encryption for the log data sets” on page 470

z/OS Configuring the data set encryption key for the queue manager

How you configure a data set encryption key for a queue manager.

About this task

This task is a prerequisite for “Configuring data set encryption for the log data sets” on page 470.

Procedure

1. Set up an AES-256 bit encryption DATA key with a label, for example, CSQ1DSKY, using the z/OS key generator utility program (KGUP).
2. Define the RACF CSFKEYS profile for the CSQ1DSKY encryption key, by issuing the following command:

   ```
   RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
   ```

3. Configure the ICSF segment of the profile to allow the key to be used as a protected key, by issuing the following command:

   ```
   RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
   ```

4. Allow the queue manager to use the encryption key by giving QMCSQ1 READ access to the profile, by issuing the following command:

   ```
   PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
   ```

   Give the same access to any administrative user that needs to read or write the encrypted data set.
5. Refresh the CSFKEYS class by issuing the following command.

   ```
   SETROPTS RACLIST(CSFKEYS) REFRESH
   ```

What to do next

Configure data set encryption for the data sets as described in “Configuring data set encryption for the log data sets” on page 470

z/OS Configuring data set encryption for the log data sets

How you configure the encryption on the log data sets.

Before you begin

Ensure that you have read:

Overview of steps to encrypt an IBM MQ for z/OS data set, and carried out the procedure in “Configuring the data set encryption key for the queue manager” on page 470

About this task

This method uses the DFP segment of a RACF generic profile, so that you can use the encryption key for all new data sets that match the profile.

Alternatively, you can configure and use an SMS data class, or the key label can be specified directly when allocating the data set.

As previously described, in this example, queue manager CSQ1 is run under user QMCSQ1, and has active log data sets CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, and so on.

Procedure

1. Create the generic profile if it does not exist, by issuing the following command:

```
ADDSD 'CSQ1.LOGS.*' UACC(NONE)
```

2. Permit the queue manager user alter access on the profile, by issuing the following command:

```
PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
```

   Also, permit the appropriate access needed for any administrative user.

3. Add the DFP segment with the encryption key label by issuing the following command:

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

   Note: You must use the same encryption key that you used in configuring the data set encryption key for the queue manager.

4. Refresh the generic dataset profiles by issuing the following command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Rename each log data set to a backup, then recreate and restore the data, using IDCAMS. The following JCL fragment converts CSQ1.LOGS.LOGCOPY1.DS001:

   a) Rename the data set to a back-up

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME DATASET TO BACKUP                                    */
  /*-------------------------------------------------------------*/
  ALTER 'CSQ1.LOGS.LOGCOPY1.DS001'                  -
        NEWNAME('CSQ1.BAK.LOGS.LOGCOPY1.DS001')
```

   b) Redefine the data set.

   The new data set will be encrypted due to the RACF profile.

   Note: Replace ++EXTDCLASS++ with the name of the extended format data class you want to use for the data set.

```
//REDEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* REDEFINE THE DATASET                                        */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                    -
         (NAME(CSQ1.LOGS.LOGCOPY1.DS001)            -
         LINEAR                                     -
         SHAREOPTIONS(2 3)                          -
```

```
          MODEL(CSQ1.BAK.LOGS.LOGCOPY1.DS001)         -
          DATACLAS(++EXTDCLASS++))
```

c) Copy the data from the backup into the recreated data set.

This step encrypts the data:

```
//RESTORE  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RESTORE DATA INTO ENCRYPTED LOG                             */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)      -
        OUTDATASET(CSQ1.LOGS.LOGCOPY1.DS001)
```

What to do next

Repeat Step “5” on page 471 for all active log data sets.

Only a single encryption key is required, and all data sets can be associated with the same key label.

Restart queue manager CSQ1. Use the output from the DISPLAY LOG command to verify that the log data sets have been encrypted.

z/OS Considerations for z/OS data set encryption in a queue sharing group

Each queue manager in a queue sharing group (QSG) must be able to read the logs, BSDS, and shared message data sets (SMDS), of every other queue manager in the QSG.

This means that each system on which a member of the QSG can run, must meet the requirements for z/OS data set encryption, and all the key labels and encryption keys used to protect the data sets for each queue manager in the QSG must be available on each system.

A queue manager prior to IBM MQ for z/OS 9.1.4 cannot access an encrypted active log data set.

A queue manager prior to IBM MQ for z/OS 9.1.5 cannot access an encrypted SMDS.

Before making use of z/OS data set encryption, you should migrate all queue managers in a QSG to at least IBM MQ for z/OS 9.1.5.

If a queue manager in a QSG is started with any encrypted active log data set, and any other queue manager in the QSG has been started, but was not last started with a version of IBM MQ for z/OS that supports encrypted active logs, the queue manager with the encrypted active log terminates abnormally with abend code 5C6-00F50033.

You can convert a QSG to use encrypted active logs and SMDS without a full outage, by:

1. Migrating each queue manager to at least IBM MQ for z/OS 9.1.5 in turn.
2. Converting active logs to encrypted data sets for each queue manager in turn. This requires the queue manager to be shut down and then restarted.

   At the same time, it is likely that page sets and archive logs would be enabled for encrypted data sets too, but this does not affect QSG migration.

   The procedure for converting each data set is described in “Example of how to encrypt queue manager active logs” on page 470
3. Converting SMDS to encrypted data sets for each individual CF structure in turn by:

   a. Issuing the command RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name) to suspend queue manager access to the SMDS.

      Note that during this time, the data on the shared queues associated with the SMDS is temporarily unavailable.

b. Converting each data set that makes up the SMDS to encrypted data sets, using the procedure described in “Example of how to encrypt queue manager active logs” on page 470.

c. Issuing the command RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name) to resume queue manager access to the SMDS.

Attention: You should shut the queue manager down cleanly prior to converting the logs, and coupling facility structure recovery might not be possible during the conversion, as the active log data sets will be temporarily unavailable.

z/OS Backwards migration considerations when using z/OS data set encryption

You need to consider the following when backwards migrating a queue manager, which has one or more encrypted data sets.

z/OS data set encryption is supported on the following IBM MQ for z/OS data sets:

- Active log data sets
- Archive log data sets
- Page sets
- BSDS
- SMDS
- CSQINP* data sets

There are no backwards migration considerations for BSDS, archive log, or CSINP* data sets.

However, there are considerations for

- SMDS
- Page set and
- Active log

data sets, as using these with z/OS data set encryption is not supported in IBM MQ for z/OS 9.1.0, and earlier, long term support releases.

Prior to backwards migration, all encryption policies for SMDS, page set, and active log data sets need to be removed and the data decrypted. This process is described in “Removing data set encryption from a data set” on page 473.

Attention: If the queue manager to be backwards migrated is part of a queue sharing group (QSG), read the “Queue sharing group considerations” on page 475 section first.

Removing data set encryption from a data set

This example describes how to remove data set encryption from the log data set CSQ1.LOGS.LOGCOPY1.DS001. You can use an equivalent process for SMDS and page sets.

The example assumes that:

- RACF is the SAF
- The queue manager that uses the data set has been stopped
- The encryption key label has been associated with the generic RACF profile CSQ1.LOGS.*

Carry out the following procedure:

1. Copy the data from the data set to a back-up data set.

   a. Define a backup data set which is not associated with an encryption key label.

Note: Replace ++EXTDCLASS++ with the name of the extended format data class you want to use for the data set.

```
//DEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DEFINE UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                    -
         (NAME(CSQ1.BAK.LOGS.LOGCOPY1.DS001)        -
         LINEAR                                     -
         SHAREOPTIONS(2 3)                          -
         MODEL(CSQ1.LOGS.LOGCOPY1.DS001)            -
         DATACLAS(++EXTDCLASS++))
/*
```

b. Copy the data from the original data set to the backup. This step decrypts the data.

```
//COPY     EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* COPY DATA INTO UNENCRYPTED DATA SET                         */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.LOGS.LOGCOPY1.DS001)          -
        OUTDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)
/*
```

c. Delete the original data set

```
//DELETE   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DELETE ORIGINAL                                             */
  /*-------------------------------------------------------------*/
  DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
/*
```

d. Rename the backup to the original data set name. The data remains unencrypted

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  ALTER CSQ1.BAK.LOGS.LOGCOPY1.DS001'                -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001)
  ALTER 'CSQ1.BAK.LOGS.LOGCOPY1.DS001.*'              -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001.*')
/*
```

2. Optionally, repeat this process for other data sets that have an encryption key label associated with them through the CSQ1.LOGS.* generic profile.
3. Optionally, if all data sets associated with the CSQ1.LOGS.* generic profile have been decrypted, remove the DATAKEY associated with the generic profile by issuing the following command

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

4. Refresh the generic dataset profiles by issuing the following command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Restart the queue manager.
6. If the encryption key is no longer needed, delete it, and delete its associated RACF profile from the CSFKEYS class.

Queue sharing group considerations

If a queue manager that is part of a queue sharing group is going to be backwards migrated to a version of IBM MQ for z/OS that does not support data set encryption then all of the active log data sets and SMDS of all queue managers in the QSG need to have their data set encryption policies removed, and their data decrypted.

This applies regardless of whether a single member of QSG is backwards migrated, or all members of the QSG.

You can achieve removal of encryption policies, and decryption of data, without a full QSG outage by:

1. Shutting down each queue manager in the QSG in turn, removing the encryption policies and decrypting the data from its active logs, using the process described in "Removing data set encryption from a data set" on page 473.

   If the queue manager is to be backwards migrated, its page set should also be decrypted at this time. Then restart the queue manager.
2. Removing the encryption policies and decrypting the data for the SMDS of each individual CF structure in turn by:
   a. Issuing the command

   ```
   RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
   ```

   to suspend queue manager access to the SMDS. During this time the data on the shared queues associated with the SMDS will be temporarily unavailable.
   b. Following the process in "Removing data set encryption from a data set" on page 473 for each data set which makes up the SMDS.
   c. Issuing the command

   ```
   RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
   ```

   to resume queue manager access to the SMDS.

Using z/OS data set encryption with a queue manager that does not support it

If you accidentally backwards migrate a queue manager to a version of IBM MQ for z/OS that does not support data set encryption, and forget to remove the encryption policies and decrypt the data you get an error when the queue manager tries to access the data set.

The error depends on the data set type and is shown in the following table.

Note: If one or more of these errors occur, you need to follow the processes described in "Removing data set encryption from a data set" on page 473 for the affected data set. These can be performed without changing the version of IBM MQ for z/OS.

Data set	Error if queue manager does not support z/OS data set encryption
Page set 0	Abend 5C6-00C91400 at queue manager start
Page sets 1-99	MQRC 2193 "Pageset error" when accessing page set, for example, on MQPUT
Active log	Abend 5C6-00E80084 at queue manager start
SMDS	Message IEC161I-122 logged "The data set has a KEYLABEL, but the user did not specify that the application could handle encryption". SMDS marked AVAIL(ERROR).

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、420 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。 RACF (z/OS の場合) または OAM (Multiplatforms の場合) を使用してメッセージ内のユーザー ID を検査することにより、リモート・キュー・マネージャーを許可します。

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。

   - z/OS 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)
   - Multi 他の Multiplatforms でのオブジェクト権限マネージャー (OAM)。
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 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 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。

z/OS **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。

z/OS **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・キューに書き込むための許可をセットアップします。 Multiplatforms では、キュー・マネージャーに接続し、それらのキュー・マネージャー上のキューに書き込む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
  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)
```

- ALW
  AIX, 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
  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にすることもできます。

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。 396 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
   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。

   AIX, Linux, and Windows 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、479 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、24 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。 例えば、CipherSpec ストリング RC4_SHA_US は、 IBM i では 05 の値を持ちますが、 AIX, Linux, and 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、 47 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。

いずれかの別名 CipherSpecs (ANY_TLS13、 ANY_TLS13_OR_HIGHER、 ANY_TLS12、 ANY_TLS12_OR_HIGHER など) にアップグレードする前に、以下のようにキュー・マネージャーをアップグレードする必要があります。

- Multi クラスター内のすべての IBM MQ for Multiplatforms キュー・マネージャーを IBM MQ 9.1.4 以降にアップグレードします。
- z/OS クラスター内のすべての IBM MQ for z/OS キュー・マネージャーを IBM MQ for z/OS 9.2.0 以降にアップグレードします。

you must

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

420 ページの『CipherSpecs の有効化』
CipherSpec は、**DEFINE CHANNEL** 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。

47 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 485 ページの図 22 に示し、続いて例を挙げて説明します。

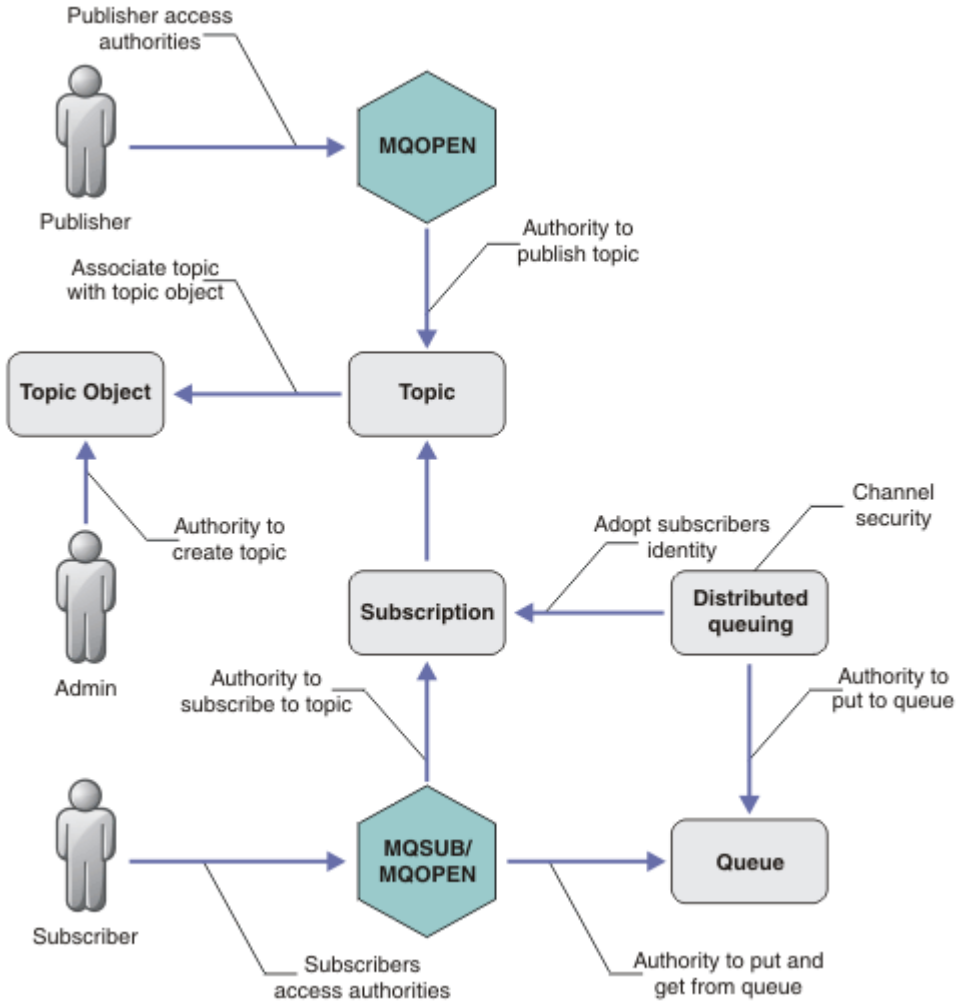


図 22. 各種コンポーネント間におけるパブリッシュ/サブスクライブのセキュリティー関係

「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。 487 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。 492 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 500 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

488 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
489 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
490 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。 506 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。 490 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。

507 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。

507 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。

役割

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。

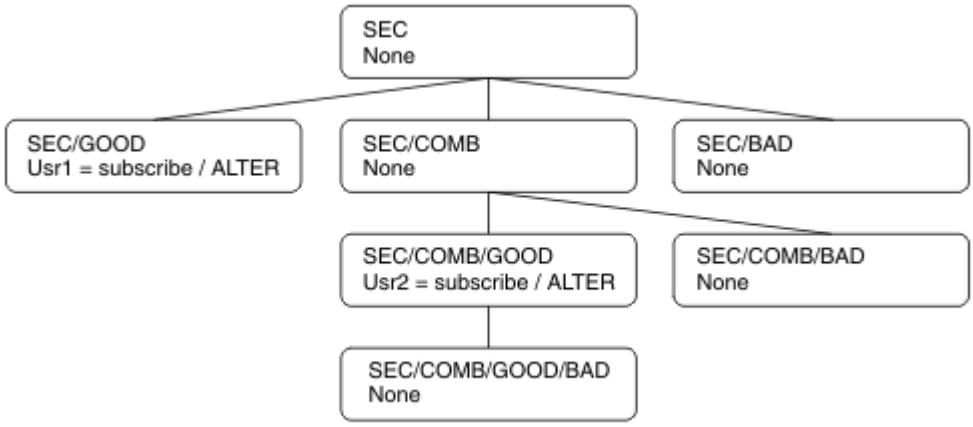
以下のようにセキュリティー属性または権限が定義されたトピック・オブジェクトの例を考えてみます。

表 *87.* 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- Multiplatforms	z/OS 権限
SECROOT	SEC	なし	なし
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD

表 87. 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- Multiplatforms**	**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。

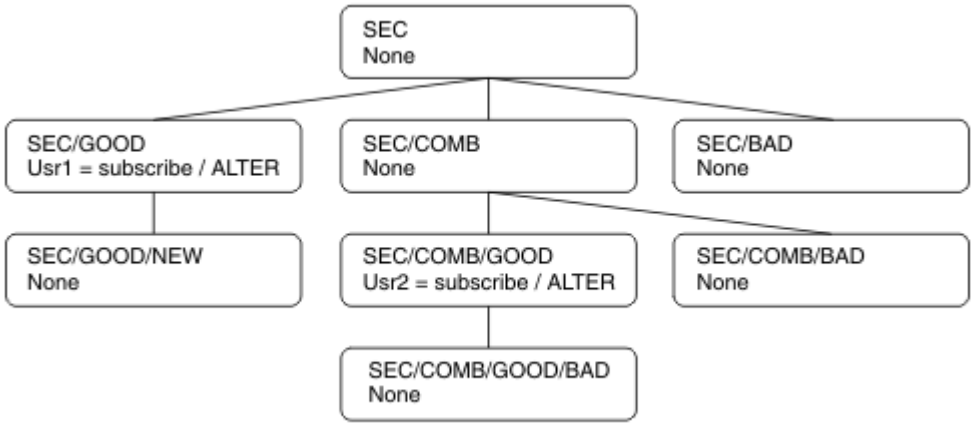
トピック・ストリングを使用したサブスクライブ操作 (トピック・ノードが存在する場合)

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、492 ページの表 88 に示すように、検査の実行方法に影響を与えます。

表 *88.* 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。

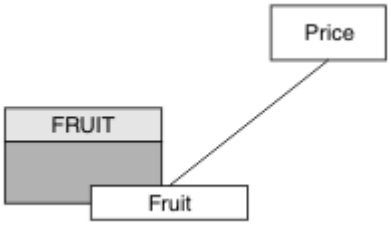


図 *23.* トピック・オブジェクトのアクセス権限の例

表 *89.* 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)
```

- Multi **Multiplatforms**:

  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。

  ALW **AIX, Linux, and Windows 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 ...
```

- ALW AIX, Linux, and Windows 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、492 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。

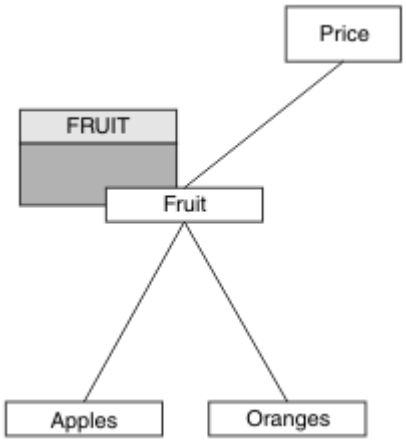


図 *24.* トピック・ツリー内のトピックへのアクセス権限を付与する例

表 *90.* 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	

492 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』では、USER1 に z/OS 上の hlq.SUBSCRIBE.FRUIT プロファイルへのアクセス権限 Multiplatforms での FRUIT プロファイルへのサブスクライブ・アクセスを付与することにより、トピック "Price/Fruit" にサブスクライブするためのアクセス権限が付与されています。 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 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 ...
```

- Multi Multiplatforms 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 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。
- Multi Multiplatforms で受け取るイベント・メッセージは、前のタスクで受け取ったものと似ていますが、実際のトピック・ストリングは異なります。

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、494 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

494 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』では、USER2 はトピック "Price/Fruit/Apples"への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。

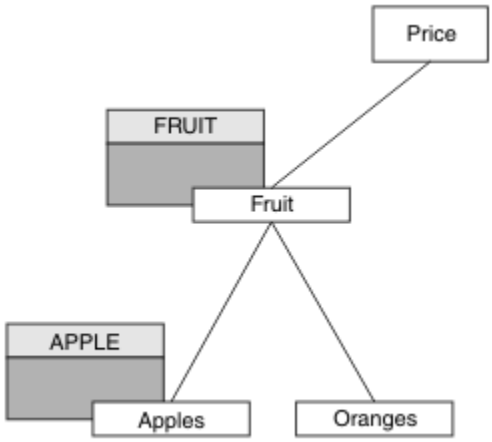


図 *25.* トピック・ツリー内の特定のトピックへのアクセス権限の付与

表 *91.* 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** :

  494 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』
  USER1 では、 hlq.SUBSCRIBE.FRUIT プロファイルへのアクセス権限をユーザーに付与することにより、トピック "Price/Fruit/Apples" にサブスクライブする権限が付与されました。

  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)
  ```

- Multi マルチプラットフォーム:

  494 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』
  USER1 では、ユーザーに FRUIT プロファイルへのサブスクライブ・アクセス権限を付与することにより、トピック "Price/Fruit/Apples" にサブスクライブする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/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。

  ALW **AIX, Linux, and Windows 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 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 ...
```

- ALW AIX, Linux, and Windows 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"
```

z/OS 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 z/OS での追加の RACF ICH408I メッセージの回避が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を説明するタスクのリストの 4 番目のトピックです。

始める前に

このトピックでは、496 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。

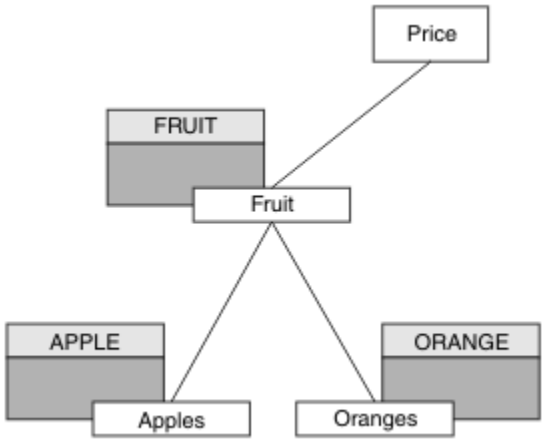


図 *26.* 追加のメッセージを回避するためにアクセス制御を付与する例。

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)
```

- Multi マルチプラットフォーム:

  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。

  ALW **AIX, Linux, and Windows 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 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。

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 ...
```

- ALW AIX, Linux, and Windows 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 IBM i 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。

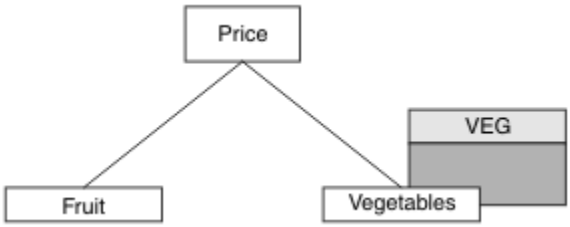


図 27. トピックへのパブリッシュ・アクセス権限の付与

表 92. 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。

  ALW **AIX, Linux, and Windows 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 ...
  ```

- ALW 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、500 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。

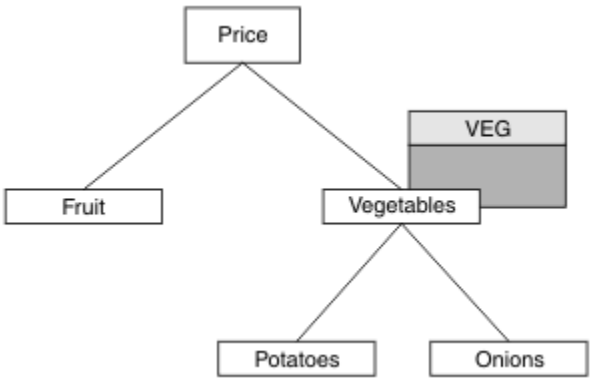


図 *28.* トピック・ツリー内のトピックへのパブリッシュ・アクセス権限の付与

表 *93.* 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	

表 93. パブリッシュ・アクセス権限の要件の例 (続き)		
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/Vegetables/Onions	USER1	

前のタスクでは、 USER1 に z/OS 上の hlq.PUBLISH.VEG プロファイルへのアクセス権限 Multiplatforms での VEG プロファイルへのパブリッシュ・アクセスを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ためのアクセス権限が付与されていました。 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 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 ...
```

- Multi Multiplatforms 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 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。
- Multi Multiplatforms で受け取るイベント・メッセージは、前のタスクで受け取ったものと似ていますが、実際のトピック・ストリングは異なります。

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、502 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。

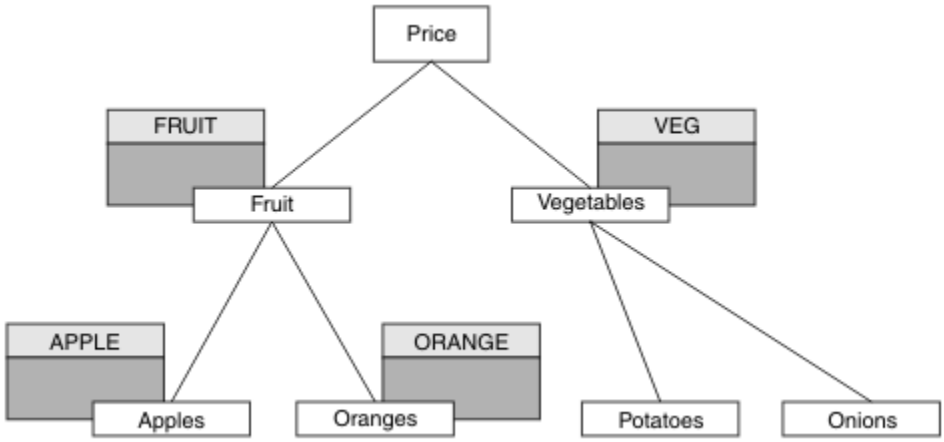


図 29. パブリッシュおよびサブスクライブのためのアクセス権限の付与

表 94. 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)
  ```

- Multi マルチプラットフォーム:

  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。

  ALW **AIX, Linux, and Windows 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 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 ...
```

- ALW AIX, Linux, and Windows 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 IBM i 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 95. 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

表 95. セキュリティーの例でのアクセス権限の完全なリスト (続き)			
トピック	必要なサブスクライブ・アクセス権限	必要なパブリッシュ・アクセス権限	トピック・オブジェクト
Price/ Vegetables/ Potatoes			
Price/ Vegetables/ Onions			

z/OS 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 96. 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)。 アクセス制御を確立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。 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、508 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。

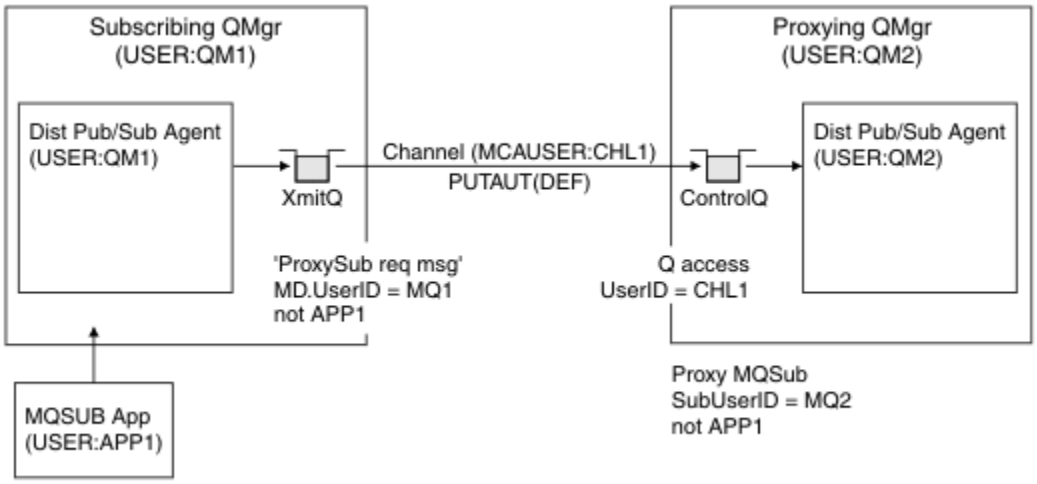


図 *30.* プロキシー・サブスクリプションのセキュリティー *(*サブスクリプションの作成*)*

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 (509 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。

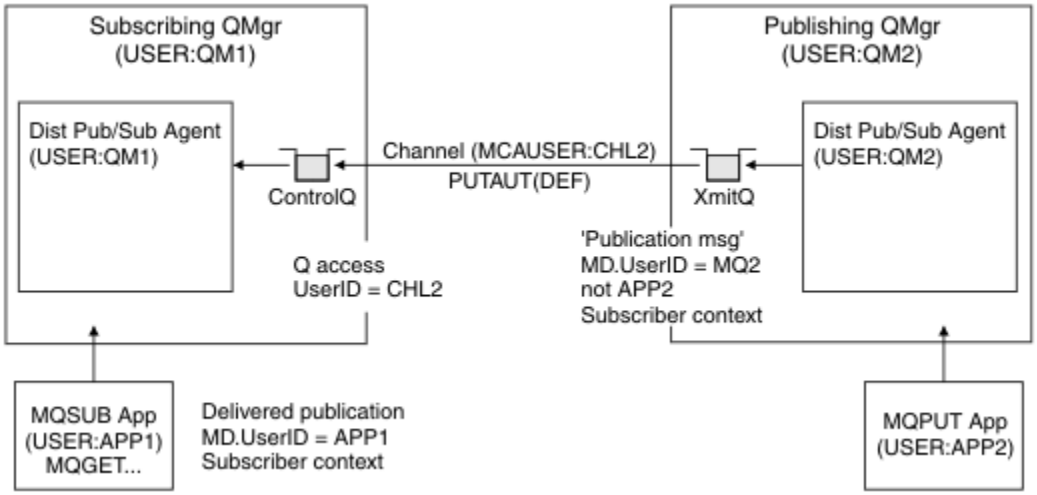


図 *31.* プロキシー・サブスクリプションのセキュリティー *(*パブリケーションの転送*)*

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 (509 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	mqm
AIX 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

IBM i 以外のプラットフォーム上のキュー・マネージャーが 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 を作成してアクセス権限を付与します。

IBM i または z/OS 上のキュー・マネージャーが AIX, Linux, and Windows 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、'mqm' ユーザー ID を作成してアクセス権限を付与します。

マルチプラットフォーム 上のキュー・マネージャーが 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・イニシエーターのアドレス・スペース・ユーザー ID を作成して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 (マルチプラットフォーム 上にある場合) では、ユーザー ID がすべて強制的に大文字に変換されます。 受信側のキュー・マネージャー (AIX, Linux, and Windows 上にある場合) では、ユーザー ID がすべて強制的に小文字になります。 そのため、AIX 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。

- AIX, 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 システムでは、ユーザー ID が大文字で指定されていることを確認してください。

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、523 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、512 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』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、530 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、531 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』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、534 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、535 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。

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、523 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、514 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、518 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。

- ALW

  `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、517 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、520 ページの『Configuring a SAF registry for the IBM MQ Console and REST API』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め、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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』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、530 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、531 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、

527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』を参照してください。

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、523 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。
  - Linux V 9.4.0 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

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。
   - IBM MQ インストール済み環境の場合:
     - ALW AIX, 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 for z/OS UNIX System Services Components のインストール・パスです。
   - Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

     ここで、 *MQWEB_INSTALLATION_PATH* は、 IBM MQ Web Server インストール・ファイルが解凍されたディレクトリーです。
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。
   - IBM MQ インストール済み環境の場合:
     - Linux AIX AIX または Linux の場合: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windows の場合: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
     - 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。

- Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

  ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・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、523 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』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、530 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、531 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』を参照してください。

ALW 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、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。
  - Linux V 9.4.0 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。
  - mqweb サーバーが IBM MQ インストールの一部である場合は、 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 の「QMQMADM」グループに属するユーザーには、 MQWebAdmin ロールと MFTWebAdmin ロールが付与されます。
- 他のすべてのユーザーには、 MQWebUser ロールが付与されます。

これらのロールについて詳しくは、523 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、AIX, Linux, and Windows 上だけです。 z/OS 同等の機能は、SAF レジストリーを構成することによって z/OS で提供されます。詳しくは、520 ページの『Configuring a SAF registry for the IBM MQ Console and REST API』を参照してください。

手順

1. 以下のいずれかのパスからサンプル XML ファイル local_os_registry.xml をコピーします。
   - Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

     ここで、 *MQWEB_INSTALLATION_PATH* は、 IBM MQ Web Server インストール・ファイルが解凍されたディレクトリーです。
   - IBM MQ インストール済み環境の場合: *MQ_INSTALLATION_PATH*/web/mq/samp/configuration
2. サンプル・ファイルを以下のいずれかのディレクトリーに配置します。
   - Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。
   - IBM MQ インストール済み環境の場合: *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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』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、530 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、531 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』を参照してください。

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、523 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。
  - Linux V 9.4.0 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

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。
   - IBM MQ インストール済み環境の場合:
     - ALW AIX, 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 for z/OS UNIX System Services Components のインストール・パスです。
   - Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

     ここで、 *MQWEB_INSTALLATION_PATH* は、 IBM MQ Web Server インストール・ファイルが解凍されたディレクトリーです。
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。
   - IBM MQ インストール済み環境の場合:
     - Linux AIX AIX または Linux の場合: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windows の場合: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
     - 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。
   - Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・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、523 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』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、530 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、531 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、527 ページの『REST API および IBM MQ Console を使用したクライアント証明書認証の構成』を参照してください。

z/OS Configuring a SAF registry for the IBM MQ Console and REST API

The System Authorization Facility (SAF) interface allows the mqweb server to call the external security manager for authentication and authorization checking. A user can then log in to the IBM MQ Console and REST API with a z/OS user ID and password.

Before you begin

- When you configure a SAF registry, you must assign users a role. Each role provides different levels of privilege to access the IBM MQ Console and REST API, and determines the security context that is used when an allowed operation is attempted. You need to understand these roles before you configure the registry. For more information about each of the roles, see “IBM MQ Console および REST API の役割” on page 523.

- You need the WebSphere Liberty Angel process running to use the authorized interface to SAF. See Enabling z/OS authorized services on Liberty for z/OS for more information.
- To complete this task, you must have write access to the `mqwebuser.xml` file, and authority to define security manager profiles.

Note: From IBM MQ 9.3.5 for Continuous Delivery and from IBM MQ 9.3.0 Fix Pack 20 for Long Term Support, the sample configuration file `zos_saf_registry.xml` is updated to remove a duplicate safAuthorization entry.

This update fixes an issue where an ICH408I error can occur when the IBM MQ Console on z/OS is upgraded to a level that ships WebSphere Liberty Profile 22.0.0.12 or later: that is, from IBM MQ 9.3.0 Fix Pack 2 for Long Term Support and from IBM MQ 9.3.1 CSU 1 and IBM MQ 9.3.2 for Continuous Delivery. Having more than one safAuthorization statement is not supported and might cause an ICH408I error when users who are not in either MQWebAdmin or MQWebAdminRO roles, in the EBJROLE class, try to access a z/OS queue manager through the IBM MQ Console.

The default for **racRouteLog**, which specifies the types of access attempts to log, is NONE. If you require an additional report or record for security auditing, see SAF Authorization (safAuthorization) for more information.

About this task

The SAF interface allows the mqweb server to call the external security manager for authentication and authorization checking for both the IBM MQ Console and REST API.

Procedure

1. Follow the steps in Enabling z/OS authorized services on Liberty for z/OS to give your mqweb server access to use z/OS authorized services.

   Sample JCL for starting the angel process is in USS_ROOT/web/templates/zos/procs/bbgzangl.jcl, where USS_ROOT is the path in z/OS UNIX System Services (z/OS UNIX) where z/OS UNIX components are installed.

   In bbgzangl.jcl, change the SET ROOT statement to point to USS_ROOT/web, for example, /usr/lpp/mqm/V9R2M0/web.

   See Administering Liberty on z/OS for further information on stopping and starting the angel process.
2. Follow the steps in Liberty: Setting up the System Authorization Facility (SAF) unauthenticated user to create the unauthenticated user needed by Liberty.
3. Copy the `zos_saf_registry.xml` file from the following path: `PathPrefix /web/mq/samp/configuration` where `PathPrefix` is the z/OS UNIX Components installation path.
4. Place the sample file in the *WLP_user_directory*`/servers/mqweb` directory, where *WLP_user_directory* is the directory that was specified when the **crtmqweb** script ran to create the mqweb server definition.
5. Optional: If you previously changed any configuration settings in `mqwebuser.xml`, copy them into the sample file.
6. Delete the existing `mqwebuser.xml` file and rename the sample file to `mqwebuser.xml`.
7. Customize the **safCredentials** element in `mqwebuser.xml`.
   a. Set **profilePrefix** to a name that is unique to your Liberty server. If you have more than one mqweb server running on a single system, you will need to choose a different name for each server; for example MQWEB920 and MQWEB915.
   b. Set **unauthenticatedUser** to the name of the unauthenticated user created in step "2" on page 521.
8. Define the mqweb server APPLID to RACF.

The APPLID resource name is the value you specified in the **profilePrefix** attribute in step “7” on page 521. The following example defines the mqweb server APPLID in RACF:

```
RDEFINE APPL profilePrefix UACC(NONE)
```

9. Grant all users, or groups, to be authenticated to the IBM MQ Console or REST API READ access to the mqweb server APPLID in the APPL class.

   You must also do this for the unauthenticated user defined in step “2” on page 521. The following example grants a user READ access to the mqweb server APPLID in RACF:

   ```
   PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
   ```

10. Use the **SETROPTS** RACF command to refresh the in-storage RACLISTed APPL class profiles:

    ```
    SETROPTS RACLIST(APPL) REFRESH
    ```

11. Define the profiles in the EJBROLE class needed to give users access to roles in the IBM MQ Console and REST API.

    The following example defines the profiles in RACF, where **profilePrefix** is the value specified for the **profilePrefix** attribute in step “7” on page 521.

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdminRO UACC(NONE)
```

12. Grant users access to roles in the IBM MQ Console and REST API.

    To do this, give users or groups READ access to one or more of the profiles in the EBJROLE class created in step “11” on page 522. For more information about the roles, see “IBM MQ Console および REST API の役割” on page 523.

    The following example gives a user access to the MQWebAdmin role for the REST API in RACF, where **profilePrefix** is the value specified for the **profilePrefix** attribute in step “7” on page 521.

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

Results

You have set up SAF authentication for the IBM MQ Console and REST API.

What to do next

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、“REST API および IBM MQ Console を使用したクライアント証明書認証の構成” on page 527 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、“REST API での HTTP 基本認証の使用” on page 530 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、“REST API でのトークン・ベースの認証の使用” on page 531 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、“REST API および IBM MQ Console を使用したクライアント証明書認証の構成” on page 527 を参照してください。

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、523 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。

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、512 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。

ALW IBM MQ Console によってブラウザーに表示される証明書の変更

認証の目的で CA 署名証明書を提示するように IBM MQ Console を構成できます。 CA 署名証明書を提示するように IBM MQ Console を構成すると、IBM MQ Console にアクセスしたときに、ブラウザーに自己署名証明書の警告が表示されなくなります。

このタスクについて

IBM MQ Console のセキュリティーは、 IBM MQ Console を実行する mqweb サーバーによって提供されます。 mqweb サーバーがブラウザーに提示する証明書を変更するには、まず新しい証明書を mqweb サーバ

ーの鍵ストアに追加します。 次に、 mqwebuser.xml ファイル内のセキュリティー構成を編集して、サーバーが提示する証明書を指定します。

この手順では、以下のことを前提としています。

- 特権ユーザーである。
- AIX、 Linux、 または Windows システムを使用している。
- mqwebuser.xml ファイルが、 basic_registry.xml、 local_os_registry.xml、 または ldap_registry.xml のいずれかのサンプル XML ファイルに基づいていること。

手順

1. オプション: **runmqktool** コマンドを使用して、mqweb サーバーの鍵ストア key.jks のデフォルト・パスワードを変更します。

   ```
   runmqktool -storepasswd -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass oldPassword
             -new newPassword
   ```

   oldPassword
   既存の key.jks パスワードを指定します。 デフォルトのパスワードは password です。

   newPassword
   新規 key.jks パスワードを指定します。
2. 認証局に送信する鍵ペアと認証要求を作成します。
   a) **runmqktool** コマンドを使用して、鍵ペアを作成します。

   ```
   runmqktool -genkeypair -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass password -storetype JKS
              -alias label -dname distinguished_name
              -sigalg signature_algorithm
   ```

   パスワード
   key.jks 鍵ストアのパスワードを指定します。

   ラベル
   証明書ラベルを指定します。 例えば、 MQWebConsole などです。

   distinguished_name
   証明書の X.500 識別名を指定します。 識別名は二重引用符で囲みます。

   例えば、"cn=MQWebConsole,o=myOrg,c=UK"

   signature_algorithm
   証明書の署名に使用するアルゴリズムを指定します。 詳しくは、 署名アルゴリズム を参照してください。

   b) **runmqktool** コマンドを使用して、証明書要求を作成します。

   ```
   runmqktool -certreq -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass password -alias label
              -file filename
   ```

   パスワード
   key.jks 鍵ストアのパスワードを指定します。

   ラベル
   サブステップ 525 ページの『2.a』の証明書ラベルを指定します。

   filename
   認証要求の完全修飾ファイル名を指定します。
3. この証明書要求ファイルを認証局 (CA) に送信します。
4. CA からの証明書がある場合は、 **runmqktool** コマンドを使用して、証明書チェーン内の証明書およびその他の証明書を、ルート CA 証明書から始めて keys.jks 鍵ストアにインポートします。

```
runmqktool -importcert -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password
           -alias label -file filename
```

パスワード
key.jks 鍵ストアのパスワードを指定します。

ラベル
サブステップ 525 ページの『2.a』 の証明書ラベルを指定します。

filename
インポートする証明書の完全修飾ファイル名を指定します。

5. CA 証明書を提示するように mqweb サーバーを構成します。
   a) mqwebuser.xml ファイルを開きます。

   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

   b) 以下の行をコメント化して、デフォルトのセキュリティー構成をオフにします。

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   クライアント証明書認証を使用するように mqweb サーバーを構成した場合、xml ファイルのこの行は既にコメント化されています。

   c) カスタム証明書構成を有効にする mqwebuser.xml ファイル内のセクションのコメントを外します。 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"/>
```

   クライアント証明書認証を使用するように mqweb サーバーを構成した場合、xml ファイルのこのセクションは既にアンコメントされています。

   d) オプション: ステップ 525 ページの『1』 で key.jks 鍵ストアのパスワードを変更した場合は、defaultKeyStore タグ内の **password** の値を、設定した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 ディレクトリーから、以下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。

   e) クライアント証明書認証を使用していない場合は、以下の行をコメント化します。

```
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
```

   f) **serverKeyAlias** の値を default から CA 証明書ラベルの値に変更します。

6. **endmqweb** コマンドを使用して mqweb サーバーを停止します。
7. **strmqweb** コマンドを使用して mqweb サーバーを始動します。

タスクの結果

Web サーバーが開始したら、IBM MQ Console を表示して、最新表示してください。 CA 証明書が使用され、ログイン・ページに直接移動します。

ALW REST API および IBM MQ Console を使用したクライアント証明書認証の構成

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、512 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` は、ユーザー名 `Fred` と突き合わされます。
  - 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。 フィルターおよびマッピングをセットアップして、マッチング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。

ALW AIX、 Linux、または Windows システムでは、この手順は以下の情報を前提としています。

- `mqwebuser.xml` ファイルが、 `basic_registry.xml`、 `local_os_registry.xml`、または `ldap_registry.xml` のいずれかのサンプル XML ファイルに基づいていること。
- 自分が特権ユーザーであること。

z/OS z/OS システムで RACF 鍵リングを使用してクライアント証明書認証を構成するには、 539 ページの『Configuring TLS for the REST API and IBM MQ Console on z/OS』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. **runmqktool** コマンドを使用して証明書を作成します。

```
runmqktool -genkeypair -keystore filename -storepass password -storetype PKCS12
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

filename
鍵ストア名 (例えば、 `user.p12`) を指定します。 鍵ストアが存在しない場合は、コマンドの実行時に作成されます。

パスワード
鍵ストア・パスワードを指定します。

ラベル
証明書ラベルを指定します。 例えば、 user1 などです。

distinguished_name
証明書の X.500 識別名を指定します。 識別名は二重引用符で囲みます。

基本ユーザー・レジストリーを使用している場合は、ユーザー・レジストリーのユーザーの名前を識別名の共通名 (CN) 部分に入力します。 例えば、ユーザー mqadmin の場合は、識別名 "CN=mqadmin"を使用します。

ローカル OS レジストリーを使用している場合は、識別名の共通名 (CN) 部分にローカル OS ユーザー ID の名前を入力します。 例えば、ユーザー mqadmin の場合は、識別名 "CN=mqadmin"を使用します。

LDAP ユーザー・レジストリーを使用している場合は、LDAP レジストリー内の識別名と一致する識別名を入力します。

signature_algorithm
証明書の署名に使用するアルゴリズムを指定します。 詳しくは、署名アルゴリズム を参照してください。

2. オプション: 認証局 (CA) から証明書を取得します。 あるいは、自己署名証明書を使用するには、ステップ 528 ページの『3』 に進みます。

a) 認証局から証明書を取得するには、 **runmqktool** コマンドを使用して認証要求を作成します。

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

filename
ステップ 527 ページの『1』 で指定した鍵ストア名を指定します。

パスワード
鍵ストア・パスワードを指定します。

ラベル
ステップ 527 ページの『1』 の証明書ラベルを指定します。

filename
認証要求の完全修飾ファイル名を指定します。

b) この証明書要求ファイルを認証局 (CA) に送信します。

c) CA から証明書を入手したら、 **runmqktool** コマンドを使用して、証明書を鍵ストアにインポートします。

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

filename
ステップ 527 ページの『1』 で指定した鍵ストア名を指定します。

パスワード
鍵ストア・パスワードを指定します。

ラベル
ステップ 527 ページの『1』 の証明書ラベルを指定します。

filename
CA 証明書の完全修飾ファイル名を指定します。

3. **runmqktool** コマンドを使用して、証明書の公開部分を抽出します。

```
runmqktool -exportcert -keystore filename -storepass password
           -alias label -file filename -rfc
```

filename
ステップ 527 ページの『1』 で指定した鍵ストア名を指定します。

パスワード
鍵ストア・パスワードを指定します。

ラベル
ステップ 527 ページの『1』 の証明書ラベルを指定します。

filename
抽出された証明書の完全修飾ファイル名を指定します。

4. 証明書の公開部分を mqweb サーバーのトラスト鍵ストアに署名者証明書としてインポートし、サーバーが **runmqktool** コマンドを使用してクライアント証明書を検証できるようにします。

```
runmqktool -importcert -keystore  MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/trust.jks -storepass password
           -alias label -file filename
```

パスワード
trust.jks 鍵ストアのパスワードを指定します。 既存の trust.jks 鍵ストアのパスワード、または新しい trust.jks 鍵ストアの新規パスワードのいずれかを指定できます。

ラベル
ステップ 527 ページの『1』 の証明書ラベルを指定します。

filename
抽出された証明書の完全修飾ファイル名を指定します。

5. クライアント証明書認証を使用するように mqweb サーバーを構成します。
   a) mqwebuser.xml ファイルを開きます。

   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

   b) 以下の行をコメント化して、デフォルトのセキュリティー構成をオフにします。

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   CA 証明書をブラウザーに提示するように mqweb サーバーを構成した場合、この行は既にコメント化されています。

   c) 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"/>
```

   CA 証明書をブラウザーに提示するように mqweb サーバーを構成した場合、このセクションのコメントは既に外されています。 ただし、**defaultTrustStore** 行のコメントを外す必要がある場合があります。

   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 ディレクトリーから、以下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。

6. **endmqweb** コマンドを使用して mqweb サーバーを停止します。
7. **strmqweb** コマンドを使用して mqweb サーバーを始動します。
8. 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。

- 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` ヘッダーの値をブランクを含む任意の値に設定してから、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ため、サポートしている cURL 実装を使用する必要があります。

以下の cURL の例は、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` 上に新規キュー `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 の作成に使用されたライブラリーによって異なります。 この例は、 OpenSSL に対して作成された cURL を持つ 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、次に証明書のパスワードを指定します。

```
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\"}"
```

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、512 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\"}"
```

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、512 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

削除 /login

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 を組み込むことできる 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
```

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。
- Linux V 9.4.0 mqweb サーバーがスタンドアロンの IBM MQ Web Server インストール済み環境の一部である場合は、 IBM MQ Web Server データ・ディレクトリー内の mqwebuser.xml ファイルに対する書き込み権限が必要です。

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。
   - IBM MQ インストール済み環境の場合:
     - Linux AIX AIX または Linux の場合: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windows の場合: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは C:\ProgramData\IBM\MQ です。
     - 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** コマンドを実行したときに指定したディレクトリーです。
   - Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     ここで、*MQ_OVERRIDE_DATA_PATH* は、**MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・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。

監査

IBM MQ Console および REST API で実行された操作の監査レコードは、キュー・マネージャーのコマンド・イベントおよび構成イベントを有効にすることによって作成でき、 AIX, 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

ALW

AIX, 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。

- IBM MQ インストール済み環境の場合:
  - Linux AIX AIX または Linux の場合: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`
  - Windows Windows の場合: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb\logs`。ここで、*MQ_DATA_PATH* は IBM MQ データ・パスです。 このパスは、IBM MQ のインストール時に選択されたデータ・パスです。 デフォルトでは、このパスは `C:\ProgramData\IBM\MQ` です。
- Linux V 9.4.0 スタンドアロンの IBM MQ Web Server インストール済み環境の場合: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb/logs`

  ここで、 *MQ_OVERRIDE_DATA_PATH* は、 **MQ_OVERRIDE_DATA_PATH** 環境変数が指す IBM MQ Web Server データ・ディレクトリーです。

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、およびログ・イベントの制御』を参照してください。

これらのコマンドおよび構成イベント・メッセージの場合、 **MQIACF_EVENT_ORIGIN** フィールドは MQEVO_REST に設定され、 **MQCACF_EVENT_APPL_IDENTITY** フィールドは認証されたプリンシパル名の最初の 32 文字を報告します。 ユーザーが MQWebAdmin 役割または MQWebAdminRO 役割を持っている場合、 **MQCACF_EVENT_USER_ID** フィールドは、コマンドを発行したプリンシパルのユーザー名ではなく、

mqweb サーバーのユーザー ID を報告します。 ただし、ユーザーが MQWebUser 役割を持っている場合、**MQCACF_EVENT_USER_ID** は、コマンドを発行したプリンシパルのユーザー名を報告します。

関連概念

476 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 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。 詳細については、537 ページの『Authority required by the mqweb server started task user ID』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、538 ページの『IBM MQ または IBM 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、539 ページの『Configuring TLS for the REST API and IBM MQ Console on z/OS』を参照してください。

z/OS Authority required by the mqweb server started task user ID

On z/OS, the mqweb server started task user ID requires certain authorities to issue PCF commands and access system resources.

The mqweb server started task user ID needs:

- A z/OS UNIX user identifier (UID) to be able to use z/OS UNIX System Services.
- Access to the `hlq.SCSQAUTH` and `hlq.SCSQANL*` data sets in the IBM MQ installation.
- Read access to the IBM MQ installation files in z/OS UNIX System Services.
- Read and write access to the Liberty user directory created by the **crtmqweb** script.
- Authority to connect to the queue manager. Grant the mqweb server started task user ID *READ* access to the `hlq.BATCH` profile in the MQCONN class.
- Authority to issue IBM MQ commands and access certain queues. These details are described in “IBM MQ Console - required command security profiles” on page 230, “System queue security” on page 209, and “Profiles for context security” on page 219.
- Authority to subscribe to the SYSTEM.FTE topic, in order to use the REST API for MFT. Grant the mqweb server started task user ID *ALTER* access to the `hlq.SUBSCRIBE.SYSTEM.FTE` profile in the MXTOPIC class.
- If you are are configuring a SAF registry, access to various security profiles. See “Configuring a SAF registry for the IBM MQ Console and REST API” on page 520 for more information.

Connection authentication

If your queue manager has been configured to require that all batch applications provide a valid user ID and password, by setting CHKLOCL(REQUIRED), you must give the mqweb server started task user ID *UPDATE* access to the `hlq.BATCH` profile in the MQCONN class.

This authority causes connection authentication to operate in CHKLOCL(OPTIONAL) mode for the mqweb server started task user ID.

If you have not configured the queue manager to require that all batch applications provide a valid user ID and password, it is sufficient to give the user ID that starts the mqweb server task *READ* access to the `hlq.BATCH` profile in the MQCONN class.

For more information about CHCKLOCL, see “Using CHCKLOCL on locally bound applications” on page 200.

z/OS IBM MQ または IBM 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 IBM 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

IBM MQ Console および REST API の役割について詳しくは、523 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、IBM 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 IBM 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、IBM 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、IBM 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。
   - IBM MQ Console でコマンドを実行するための権限。230 ページの『IBM MQ Console - required command security profiles』を参照してください。

z/OS Configuring TLS for the REST API and IBM MQ Console on z/OS

On z/OS, you can configure the mqweb server to use a RACF key ring to store certificates for secure connections with TLS, and client certificate authentication.

Before you begin

You must be a user that has write access to the `mqwebuser.xml` file, and authority to work with SAF key rings, to complete this procedure.

About this task

The default mqweb server configuration uses Java keystores for the server and trusted certificates. On z/OS, you can configure the mqweb server to use a RACF key ring, instead of the Java keystores. The server can also be configured to allow users to authenticate using a client certificate.

See Liberty: Keystores for information on using RACF key rings in Liberty.

Follow this procedure to configure the mqweb server to use a RACF key ring, and optionally configure client certificate authentication. This procedure describes the steps necessary to create and use certificates signed with your own certificate authority (CA) certificates. For production, you might prefer to use certificates obtained from an external certificate authority.

Procedure

1. Create a certificate authority (CA) certificate, which will be used to sign the server certificate. For example, enter the following RACF command:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb Certification Authority') -
    O('IBM') -
    OU('MQ')) -
  SIZE(2048) -
  WITHLABEL('mqwebCertauth')
```

2. Create a server certificate, signed with the CA certificate created in step 1, by entering the following command:

```
RACDCERT ID(mqwebUserId) GENCERT -
  SUBJECTSDN(CN('hostname') -
   O('IBM') -
   OU('MQ')) -
```

```
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebCertauth')) -
  WITHLABEL('mqwebServerCert')
```

where *mqwebUserId* is the mqweb server started task user ID, and *hostname* is the host name of the mqweb server.

3. Connect the CA certificate and server certificate to a SAF key ring by entering the following commands:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

where *mqwebUserId* is the mqweb server started task user ID, and *keyring* is the name of the key ring you want to use.

4. Export the CA certificate to a CER file by entering the following command:

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth')) -
        DSN('hlq.CERT.MQWEBCA') -
        FORMAT(CERTDER) -
        PASSWORD('password')
```

5. FTP the exported CA certificate in binary to your workstation, and import it into your browser as a certificate authority certificate.
6. Optional: If you want to configure client certificate authentication, create and export a client certificate.
   a) Create a certificate authority (CA) certificate, which will be used to sign the client certificate. For example, enter the following RACF command:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb User CA') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
  WITHLABEL('mqwebUserCertauth')
```

   b) Connect the CA certificate to a SAF key ring by entering the following command:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

   where *mqwebUserId* is the mqweb server started task user ID, and *keyring* is the name of the key ring you want to use.

   c) Create a client certificate, signed with the CA certificate. For example, enter the following command:

```
RACDCERT ID(clientUserId) GENCERT -
  SUBJECTSDN(CN('clientUserId') -
     O('IBM') -
     OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth')) -
  WITHLABEL('userCertLabel')
```

   where *clientUserId* is the user name.

   The method used to map a certificate to a principal depends on the type of user registry configured:

   - If you are using a basic registry, the Common Name field in the certificate is matched against the user in the registry.
   - If you are using a SAF registry, and the certificate is in the RACF database, the certificate owner, specified with the **ID** parameter when creating the certificate, is used.
   - If you are using an LDAP registry, the full distinguished name in the certificate is matched against the LDAP registry.

   d) Export the client certificate to a PKCS #12 file by entering the following command:

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) -
         PASSWORD('password') DSN('hlq.USER.CERT')
```

e) FTP the exported certificate in binary to your workstation. To use the client certificate with the IBM MQ Console, import it into the web browser used to access the IBM MQ Console as a personal certificate.

7. Edit the file *WLP_user_directory*/servers/mqweb/mqwebuser.xml, where *WLP_user_directory* is the directory that was specified when the **crtmqweb** script ran to create the mqweb server definition.

   Make the following changes to configure the mqweb server to use a RACF key ring:

   a) Remove, or comment out, the following line:

   ```
   <sslDefault sslRef="mqDefaultSSLConfig"/>
   ```

   b) Add the following statements:

   ```
   <keyStore id="defaultKeyStore" filebased="false"
             location="safkeyring://mqwebUserId/keyring"
             password="password" readOnly="true" type="JCERACFKS" />
   <ssl id="thisSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1.2"
             serverKeyAlias="mqwebServerCert" clientAuthenticationSupported="true" />
   <sslDefault sslRef="thisSSLConfig"/>
   ```

   where:

   - *mqwebUserId* is the mqweb server started task user ID.
   - *keyring* is the name of the RACF key ring.
   - *mqwebServerCert* is the label of the mqweb server certificate.

   Notes: The value of **keyStore password** is ignored.

8. Restart the mqweb server by stopping and restarting the mqweb server started task.
9. Optional: Use the client certificate to authenticate:

   - To use the client certificate with the IBM MQ Console, enter the URL for the IBM MQ Console in the web browser where you installed the client certificate.
   - To use the client certificate with the REST API, provide the client certificate with each REST request.

   Notes:

   a. If you are using only certificates to authenticate to the IBM MQ Console, the browser might display a list of certificates for you to select from.

   b. If you want to use a different certificate you might need to close and restart your browser.

   c. If you are using client certificates that are not in the RACF database, you can use RACF certificate name filtering, to map certificate attributes to a user ID. For example:

   ```
   RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
   ```

   maps certificates with a subject distinguished name containing OU=DEPT1 and C=US to user ID DEPT3USR.

Results

You have set up a TLS interface for the IBM MQ Console and REST API.

ALW 鍵と証明書の管理 (AIX, Linux, and Windows)

AIX, Linux, and Windows では、 **runmqakm** コマンドおよび V 9.4.0 V 9.4.0 **runmqktool** コマンドを使用して、鍵、証明書、および認証要求を管理します。

このタスクについて

runmqakm コマンドは、 **gskitcapicmd** と同様の機能を提供します。 V 9.4.0 V 9.4.0 **runmqktool** コマンドは、Java **keytool** 証明書管理ユーティリティーの機能と同様の機能を提供します。 **runmqakm** コマンドまたは **runmqktool** コマンドを使用する前に、 **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てください。

runmqktool コマンドを使用するには、 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 **runmqakm** コマンドを使用できます。

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は、**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。

手順

- **runmqakm** コマンドおよび **runmqktool** コマンドを使用して、以下のアクションを実行します。
  - IBM MQ がサポートする CMS および PKCS #12 鍵リポジトリーを作成します。
  - 認証要求を作成します。
  - 証明書をエクスポートします。
  - 個人証明書と CA 証明書をインポートします。
  - 自己署名証明書を管理します。
  - 秘密鍵を作成、抽出、および追加します。

関連情報
Keytool

ALW AIX, Linux, and Windows での runmqakm および runmqktool コマンド

AIX, Linux, and Windows システムでは、 **runmqakm** (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵および証明書を管理します。

注 : V 9.4.0 V 9.4.0

IBM MQ 9.4.0 以降、 **runmqckm** コマンドおよび **strmqikm** コマンドは削除されました。 PKCS #12 および JKS 鍵リポジトリーを管理するために、 **runmqckm** コマンドの代わりに **runmqktool** コマンドを使用できます。 **strmqikm** GUI に代わるものはありません。

runmqckm コマンドと **runmqktool** コマンドには、以下の重要な違いがあります。

- **runmqktool** コマンドは、鍵リポジトリー・パスワードを保管する stash ファイルをサポートしていません。 キー・リポジトリーにアクセスするためのパスワードは、 **runmqktool** コマンドの実行時に、コマンドへのパラメーターとして、またはコマンドによって発行されたプロンプトへの応答として、常に指定する必要があります。
- **runmqktool** コマンドは、 CMS キー・リポジトリーをサポートしません。 したがって、JKS から CMS 鍵リポジトリーに証明書をエクスポートするには、以下のステップを実行する必要があります。
  1. **runmqktool -importkeystore** コマンドを使用して、JKS 鍵リポジトリーから中間 PKCS #12 鍵リポジトリーに証明書をコピーします。 証明書のエクスポートについて詳しくは、 552 ページの『AIX, 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』を参照してください。
  2. **runmqakm -cert -import** コマンドを使用して、中間 PKCS #12 鍵リポジトリーから CMS 鍵リポジトリーに証明書をインポートします。 証明書のインポートについて詳しくは、 554 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 (AIX, Linux, and Windows)』を参照してください。

以下の IBM MQ コマンドを使用して、鍵および証明書を管理できます。

runmqakm

- **gskitcapicmd** の機能と同様の機能を提供します。
- CMS および PKCS #12 鍵リポジトリーをサポートします。
- 暗号化された鍵リポジトリー・パスワードを保管する stash ファイルの作成をサポートします。
- FIPS 140-2 準拠として認証され、 **-fips** パラメーターを使用して FIPS 準拠の方法で動作するように構成できます。

V 9.4.0 V 9.4.0 **runmqktool**

- Java **keytool** コマンドの機能と同様の機能を提供します。
- PKCS #12、JKS、および JCEKS 鍵リポジトリーをサポートします。
- IBM MQ Java runtime environment (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。

FIPS 準拠の方法で証明書を管理する必要がある場合は、 **runmqakm** コマンドを使用します。

runmqakm コマンドについて詳しくは、 runmqakm を参照してください。

V 9.4.0 V 9.4.0 **runmqktool** コマンドについて詳しくは、 runmqktool を参照してください。

このセクションのトピックには、これらのコマンドを使用して一般的な証明書管理タスクを実行する方法の例が含まれています。

ALW AIX, Linux, and Windows での自己署名個人証明書の作成

鍵リポジトリーに自己署名個人証明書を作成するには、以下の手順を実行します。

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。

Deprecated 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。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、自己署名証明書を作成できます。 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。

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 CipherSpecs 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。 詳しくは、48 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・パラメーターを指定する必要があります。 例えば、パラメーター **-sig_alg** EC_ecdsa_with_SHA384 を指定します。

runmqakm の使用

以下のコマンドを発行して、 **runmqakm** コマンドで自己署名個人証明書を作成します。

```
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*
鍵リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーは既に存在している必要があります。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
証明書ラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

IBM MQ によって使用される TLS 証明書のラベルは、 **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または IBM MQ MQI client ユーザー ID をすべて小文字で付加した値のいずれかです。 詳しくは、27 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。 識別名には少なくとも 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 : runmqakm コマンドは、郵便番号属性を PC ではなく POSTALCODE として参照します。 郵便番号を使用して証明書を要求するために **runmqakm** コマンドを使用する場合は、必ず **-dn** パラメーターに POSTALCODE を指定してください。

-size *key_size*
鍵のサイズを指定します。 値は 512、 1024、または 2048 です。

-x509version *バージョン*
作成する X.509 証明書のバージョン。 値は 1、2、または 3 にすることができます。 デフォルトは 3 です。

-expire *days*
証明書の有効期限 (日数)。 証明書の場合のデフォルトは 365 日です。

-fips
コマンドが FIPS モードで実行されるように指定します。 FIPS IBM Crypto for C (ICC) コンポーネントのみが使用され、このコンポーネントは FIPS モードで正常に初期化され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 検証済みの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
証明書の作成時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、証明書に関連付けられた署名を作成するために使用されます。 値は、md5、MD5_WITH_RSA、MD5WithRSA、 SHA_WITH_DSA、SHA_WITH_RSA、sha1、SHA1WithDSA、SHA1WithECDSA、SHA1WithRSA、sha224、SHA224_WITH_RSA、SHA224WithDSA、SHA224WithECDSA、SHA224WithRSA、sha256、SHA256_WITH_RSA、SHA256WithDSA、SHA256WithECDSA、SHA256WithRSA、SHA2WithRSA、sha384、SHA384_WITH_RSA、SHA384WithECDSA、SHA384WithRSA、sha512、SHA512_WITH_RSA、SHA512WithECDSA、SHA512WithRSA、SHAWithDSA、SHAWithRSA、EC_ecdsa_with_SHA1、EC_ecdsa_with_SHA224、EC_ecdsa_with_SHA256、EC_ecdsa_with_SHA384、または EC_ecdsa_with_SHA512 のいずれかです。

デフォルト値は SHA1WithRSA です。

これらのパラメータと指定できる値の詳細については、以下を参照してください。実行 mqakm -cert 。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行して、 **runmqktool** コマンドで自己署名個人証明書を作成します。

```
runmqktool -genkeypair -keystore filename -storepass password -storetype store_type
           -alias label -dname distinguished_name -validity days
           -keyalg key_algorithm -keysize key_size -sigalg signature_algorithm
```

ここで、

-keystore ファイル名
鍵リポジトリーの名前を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-storetype *store_type*
鍵リポジトリー・タイプを指定します。

-alias ラベル
証明書ラベルを指定します。 証明書ラベルは小文字に変換されます。

-dname *distinguished_name*
二重引用符で囲まれた証明書の X.500 識別名を指定します。

-validity 日
証明書が有効となる日数を指定します。

-keyalg *key_algorithm*
鍵ペアの作成に使用されるアルゴリズムを指定します。

-keysize キー・サイズ
鍵のサイズを指定します。

-sigalg *signature_algorithm*
証明書の署名に使用されるアルゴリズムを指定します。 指定できる署名アルゴリズムについて詳しくは、 署名アルゴリズムを参照してください。

これらのパラメーターおよび指定可能な値について詳しくは、 genkeypair を参照してください。

ALW AIX, Linux, and Windows での個人証明書の要求

個人証明書の要求を作成するには、以下の手順に従います。

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。

Deprecated 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。

個人証明書は、 **runmqakm** (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して要求できます。 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。

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 CipherSpecs 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。 詳しくは、48 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・パラメーターを指定する必要があります。 例えば、パラメーター **-sig_alg** EC_ecdsa_with_SHA384 を指定します。

暗号ハードウェアを使用している場合は、563 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。

runmqakm の使用

以下のコマンドを発行し、 **runmqakm** コマンドを使用して証明書要求を作成します。

```
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*
鍵リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーは既に存在している必要があります。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
証明書ラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

IBM MQ によって使用される TLS 証明書のラベルは、 **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または IBM MQ MQI client ユーザー ID をすべて小文字で付加した値のいずれかです。 詳しくは、27 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。 識別名には少なくとも 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 : runmqakm コマンドは、郵便番号属性を PC ではなく POSTALCODE として参照します。 郵便番号を使用して証明書を要求するために **runmqakm** コマンドを使用する場合は、必ず **-dn** パラメーターに POSTALCODE を指定してください。

-size *key_size*
鍵のサイズを指定します。 値は 512、 1024、または 2048 です。

-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 モードの場合、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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
認証要求の作成時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、認証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。

これらのパラメータと指定できる値の詳細については、 mqakm -certreq を実行します。

runmqktool の使用

V 9.4.0 V 9.4.0

runmqktool コマンドを使用して証明書要求を作成する前に、 **runmqktool -genkeypair** コマンドを使用して鍵ペアを生成する必要があります。 **runmqktool -genkeypair** コマンドについて詳しくは、543 ページの『AIX, 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。

以下のコマンドを発行し、 **runmqktool** コマンドを使用して証明書要求を作成します。

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

ここで、

-keystore ファイル名
鍵リポジトリーの名前を指定します。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
証明書ラベルを指定します。 これは、鍵ペアの生成時に指定された証明書ラベルです。 証明書ラベルは大/小文字を区別しません。

-file *filename*
認証要求のファイル名を指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 certreq を参照してください。

次のタスク

CA に証明書要求を送信します。 CA から署名済み証明書を受け取ったら、その署名済み証明書を鍵リポジトリーに追加します。 詳細については、547 ページの『個人証明書を鍵リポジトリーに受信する操作 (AIX, Linux, and Windows)』を参照してください。

ALW AIX, Linux, and Windows での既存の個人証明書の更新

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。 有効期限が切れる前に個人証明書を更新するには、以下の手順を実行します。

個人証明書は、 **runmqakm** (GSKCapiCmd) コマンドを使用して更新できます。

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。545 ページの『AIX, 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。

runmqakm の使用

以下のコマンドを発行して、**runmqakm** コマンドで個人証明書を更新するための証明書要求を作成します。

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

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
証明書ラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-target *filename*
認証要求のファイル名を指定します。

次のタスク

CA に証明書要求を送信します。 CA から署名済み証明書を受け取ったら、その署名済み証明書を鍵リポジトリーに追加します。 詳細については、547 ページの『個人証明書を鍵リポジトリーに受信する操作 (AIX, Linux, and Windows)』を参照してください。

ALW 個人証明書を鍵リポジトリーに受信する操作 (AIX, Linux, and Windows)

鍵リポジトリーに個人証明書を受信するには、この手順を使用します。

認証局 (CA) から新しい個人証明書が送信されたら、その証明書を、新しい証明書要求の生成元の鍵リポジトリーに追加します。 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。

CA 署名済み個人証明書を鍵リポジトリーに追加する前に、 551 ページの『AIX, Linux, and Windows での鍵リポジトリーへの CA 証明書またはトラステッド証明書の公開部分の追加』 のステップを実行して、CA 証明書を鍵リポジトリーに追加します。

runmqakm (GSKCapiCmd) コマンドまたは **runmqktool** (keytool) コマンドを使用して、鍵リポジトリーに個人証明書を受け取ることができます。 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。

暗号ハードウェアを使用している場合は、565 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。

runmqakm の使用

runmqakm コマンドを使用して、個人証明書を鍵リポジトリーに追加するには、以下のコマンドを発行します。

```
runmqakm -cert -receive -file filename -format format
         -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*
鍵リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーは既に存在している必要があり、認証要求を作成したのと同じリポジトリーでなければなりません。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-format 形式
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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

runmqktool コマンドを使用して、個人証明書を鍵リポジトリーに追加するには、以下のコマンドを発行します。

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

ここで、

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーは既に存在している必要があり、認証要求を作成したのと同じリポジトリーでなければなりません。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
認証要求の作成に使用された証明書のラベルを指定します。 証明書ラベルは小文字に変換されます。

-file *filename*
個人用証明書の完全修飾ファイル名を指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 importcert を参照してください。

次のタスク

証明書がキュー・マネージャーの TLS キー・リポジトリーに追加されている場合は、MQSC コマンド **REFRESH SECURITY TYPE(SSL)** を発行して、キュー・マネージャーの TLS キー・リポジトリー・キャッシュをリフレッシュします。

ALW 鍵リポジトリーから CA 証明書を抽出する操作 (AIX, Linux, and Windows)

鍵リポジトリーから認証局 (CA) 証明書を抽出するには、以下の手順に従います。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵リポジトリーから CA 証明書を抽出できます。 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。

runmqakm の使用

以下のコマンドを発行し、 **runmqakm** コマンドを使用して CA 証明書を抽出します。

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

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
CA 証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-target *filename*
宛先ファイルの完全修飾ファイル名を指定します。

-format 形式
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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行し、 **runmqktool** コマンドを使用して CA 証明書を抽出します。

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

ここで、

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
CA 証明書のラベルを指定します。 証明書ラベルは大/小文字を区別しません。

-file *filename*
宛先ファイルの完全修飾ファイル名を指定します。

-rfc (C)
インターネット RFC 1421 標準で定義されているように、出力ファイルが Base64-encoded ASCII フォーマットであることを指定します。 このオプションを指定しない場合、出力ファイルはバイナリー・フォーマットになります。

これらのパラメーターおよび指定可能な値について詳しくは、 exportcert を参照してください。

ALW 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 (AIX, Linux, and Windows)

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には、以下の手順を実行します。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵リポジトリーから証明書の公開部分を抽出できます。 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。

runmqakm の使用

以下のコマンドを発行し、 **runmqakm** コマンドを使用して自己署名証明書の公開部分を抽出します。

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

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
CA 証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-target *filename*
宛先ファイルの完全修飾ファイル名を指定します。

-format 形式
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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメータと指定できる値の詳細については、以下を参照してください。実行 mqakm -cert 。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行し、 **runmqktool** コマンドを使用して自己署名証明書の公開部分を抽出します。

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

ここで、

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
CA 証明書のラベルを指定します。 証明書ラベルは大/小文字を区別しません。

-file *filename*
宛先ファイルの完全修飾ファイル名を指定します。

-rfc (C)
インターネット RFC 1421 標準で定義されているように、出力ファイルが Base64-encoded ASCII フォーマットであることを指定します。 このオプションを指定しない場合、出力ファイルはバイナリー・フォーマットになります。

これらのパラメーターおよび指定可能な値について詳しくは、 exportcert を参照してください。

ALW AIX, Linux, and Windows での鍵リポジトリーへの CA 証明書またはトラステッド証明書の公開部分の追加

以下の手順に従って、CA 証明書またはトラステッド証明書の公開部分を鍵リポジトリーに追加します。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、CA 証明書、またはトラステッド証明書の公開部分を鍵リポジトリーに追加できます。 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。

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。

注 :

- 証明書が ASCII (UTF-8) またはバイナリー (DER) エンコードであることを確認します。
- IBM Java 8 **keytool** コマンドには制限があるため、ファイルにコメントが含まれている場合、**runmqktool** は、インターネット RFC 1421 で定義されている印刷可能エンコード形式 (Base64 エンコードとも呼ばれる) の証明書をインポートできません。 印刷可能なエンコード形式で証明書をインポートするには、ファイルからすべてのコメントを削除します。 このファイルは、「----- BEGIN」で始まり、「----- END」で始まるストリングで終わるストリングでなければなりません。

runmqakm の使用

runmqakm コマンドを使用して、トラステッド証明書を鍵リポジトリーに追加するには、以下のコマンドを発行します。

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

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーは既に存在している必要があります。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
証明書ラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメータと指定できる値の詳細については、実行 mqakm -cert 。

runmqktool の使用

V 9.4.0 V 9.4.0

runmqktool コマンドを使用して、トラステッド証明書を鍵リポジトリーに追加するには、以下のコマンドを発行します。

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

ここで、

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
証明書ラベルを指定します。 証明書ラベルは小文字に変換されます。

-file *filename*
個人用証明書の完全修飾ファイル名を指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 importcert を参照してください。

ALW AIX, 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

以下の手順に従って、鍵リポジトリーから個人証明書をエクスポートします。

証明書をエクスポートすると、その証明書とそれに関連する公開鍵と秘密鍵が別の鍵リポジトリーにコピーされます。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵リポジトリーから証明書をエクスポートできます。 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。

runmqakm の使用

以下のコマンドを発行して、 **runmqakm** コマンドで証明書をエクスポートします。

```
runmqakm -cert -export -db filename -pw password -label label
         -target filename -target_pw password -target_type type
         -encryption strength -fips
```

ここで、

-db *filename*
証明書を含む鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
証明書を含む鍵リポジトリーのパスワードを指定します。

-label *label*
エクスポートする証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-target *filename*
宛先キー・リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-target_pw パスワード
宛先キー・リポジトリーのパスワードを指定します。

-target_type 型
宛先キー・リポジトリーのタイプを指定します。 値は cms または pkcs12 です。 デフォルトは cms です。

-encryption 強度
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を指定します。 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。

-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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行して、 **runmqktool** コマンドで証明書をエクスポートします。

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

ここで、

-srckeystore ファイル名
証明書を含む鍵リポジトリーの完全修飾ファイル名を指定します。

-srcstorepass パスワード
証明書を含む鍵リポジトリーのパスワードを指定します。

-destkeystore ファイル名
宛先キー・リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-deststorepass パスワード
宛先キー・リポジトリーのパスワードを指定します。

-destkeypass パスワード
宛先キー・リポジトリー内のキーを保護するためのパスワードを指定します。 このパラメーターが指定されていない場合、鍵は、ソース鍵リポジトリー内の鍵を保護するために使用されるパスワードで保護されます。

-deststoretype タイプ
宛先キー・リポジトリーのタイプを指定します。

-srcalias ラベル
エクスポートする証明書のラベルを指定します。 証明書ラベルは大/小文字を区別しません。

-des タリアス *label*
宛先キー・リポジトリー内の証明書のラベルを指定します。 このパラメーターを指定しない場合、ソース・キー・リポジトリーと同じラベルが証明書に割り当てられます。

証明書ラベルは小文字に変換されます。

-file *filename*
宛先ファイルの完全修飾ファイル名を指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 importkeystore を参照してください。

ALW 個人証明書を鍵リポジトリーにインポートする操作 (AIX, Linux, and Windows)

個人証明書を鍵リポジトリーにインポートするには、以下の手順に従います。

証明書をインポートすると、証明書とそれに関連付けられた公開鍵と秘密鍵が、ある鍵リポジトリーから別の鍵リポジトリーにコピーされます。

個人証明書を鍵リポジトリーにインポートする前に、まず CA 証明書を発行する完全な有効なチェーンを鍵リポジトリーに追加する必要があります。 詳細については、551 ページの『AIX, Linux, and Windows での鍵リポジトリーへの CA 証明書またはトラステッド証明書の公開部分の追加』 を参照してください。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、証明書を鍵リポジトリーにインポートできます。 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。

runmqakm の使用

以下のコマンドを発行して、 **runmqakm** コマンドで証明書をインポートします。

```
runmqakm -cert -import -file filename -pw password -type type
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips
```

ここで、

-file *filename*
証明書を含む鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
証明書を含む鍵リポジトリーのパスワードを指定します。

-type 型
証明書を含む鍵リポジトリーのタイプを指定します。 値は cms または pkcs12 です。 デフォルトは cms です。

-target *filename*
宛先キー・リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-target_pw パスワード
宛先キー・リポジトリーのパスワードを指定します。

-target_type 型
宛先キー・リポジトリーのタイプを指定します。 値は cms または pkcs12 です。 デフォルトは cms です。

-label *label*
ソース鍵リポジトリーからインポートする証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-new_label ラベル
ターゲット鍵リポジトリー内の証明書に割り当てられるラベルを指定します。 このパラメーターを指定しない場合、ソース・キー・リポジトリーと同じラベルが証明書に割り当てられます。

-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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行して、 **runmqktool** コマンドで証明書をインポートします。

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

ここで、

-srckeystore ファイル名
証明書を含む鍵リポジトリーの完全修飾ファイル名を指定します。

-srcstorepass パスワード
証明書を含む鍵リポジトリーのパスワードを指定します。

-destkeystore ファイル名
宛先キー・リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-deststorepass パスワード
宛先キー・リポジトリーのパスワードを指定します。

-destkeypass パスワード
宛先キー・リポジトリー内のキーを保護するためのパスワードを指定します。 このパラメーターが指定されていない場合、鍵は、ソース鍵リポジトリー内の鍵を保護するために使用されるパスワードで保護されます。

注 : PKCS #12 鍵リポジトリーの場合、鍵は宛先鍵リポジトリーと同じパスワードで保護されている必要があります。

-deststoretype タイプ
宛先キー・リポジトリーのタイプを指定します。

-srcalias ラベル
ソース鍵リポジトリー内の証明書のラベルを指定します。 証明書ラベルは大/小文字を区別しません。

-des タリアス *label*
宛先キー・リポジトリー内の証明書のラベルを指定します。 このパラメーターを指定しない場合、ソース・キー・リポジトリーと同じラベルが証明書に割り当てられます。

証明書ラベルは小文字に変換されます。

-file *filename*
宛先ファイルの完全修飾ファイル名を指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 importkeystore を参照してください。

ALW Microsoft .pfx ファイルからの個人証明書のインポート

証明書をインポートするには、次の手順に従ってください。Microsoft .pfx ファイル AIX, Linux, and Windows。

.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、公開鍵と秘密鍵の両方を含む個人証明書またはサイト証明書です。 もう 1 つは、公開鍵のみを含む CA (署名者) 証明書です。 これらの証明書は同じ CMS 鍵リポジトリーに共存できないため、そのうちの 1 つのみをインポートできます。

証明書ラベルは、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。 以下の手順に従って、.pfx ファイルから個人証明書をインポートし、個人証明書ラベルを、.pfx ファイル内の CA 証明書に割り当てられているラベルに設定します。 発行元の CA 証明書は、ターゲット鍵データベースに既に追加されている必要があります。

runmqakm の使用

以下のコマンドを発行し、 **runmqakm** コマンドを使用して .pfx ファイルから証明書をインポートします。

```
runmqakm -cert -import -file filename -pw password -type pkcs12
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips -pfx
```

ここで、

-file *filename*
.pfx ファイルの完全修飾名を指定します。

-pw *password*
.pfx ファイルのパスワードを指定します。

-type pkcs12
鍵リポジトリーのタイプを指定します。

-target *filename*
宛先キー・リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-target_pw パスワード
宛先キー・リポジトリーのパスワードを指定します。

-target_type 型
宛先キー・リポジトリーのタイプを指定します。 値は cms または pkcs12 です。 デフォルトは cms です。

-label *label*
ソース鍵リポジトリーからインポートする証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-new_label ラベル
ターゲット鍵リポジトリー内の証明書に割り当てられるラベルを指定します。 このパラメーターを指定しない場合、ソース・キー・リポジトリーと同じラベルが証明書に割り当てられます。

-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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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 (P)
ソース・キー・リポジトリーが PFX 形式を使用することを示します。

これらのパラメータと指定できる値の詳細については、以下を参照してください。実行 mqakm -cert 。

ALW PKCS #7 ファイルからの個人証明書のインポート

AIX, Linux, and Windows 上の PKCS #7 ファイルから証明書をインポートするには、以下の手順を実行します。

runmqakm コマンドを使用して、 AIX, 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。

CA 証明書またはトラステッド証明書の公開部分の追加

以下のコマンドを発行して、CA 証明書、またはトラステッド証明書の公開部分を PKCS #7 ファイルから追加します。

```
runmqakm -cert -add -db filename -pw password -type type
         -label label -file filename
```

ここで、

-db *filename*
鍵リポジトリーの完全修飾名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-type 型
鍵リポジトリーのタイプを指定します。

-label *label*
追加する証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

ラベルは、追加された最初の証明書に割り当てられます。 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。

-file *filename*
PKCS #7 ファイルの完全修飾名を指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

個人証明書のインポート

以下のコマンドを発行して、PKCS #7 ファイルから個人証明書をインポートします。

```
runmqakm -cert -import -file filename -pw password -type pkcs7
         -target filename -target_pw password -target_type type
         -label label -new_label label
```

ここで、

-file *filename*
PKCS #7 ファイルの完全修飾名を指定します。

-pw *password*
PKCS #7 ファイルのパスワードを指定します。

-type pkcs7
PKCS #7 ファイルのタイプを指定します。

-target *filename*
宛先キー・リポジトリーの完全修飾ファイル名を指定します。 鍵リポジトリーが存在しない場合は作成されます。

-target_pw パスワード
宛先キー・リポジトリーのパスワードを指定します。

-target_type 型
宛先キー・リポジトリーのタイプを指定します。 値は cms または pkcs12 です。 デフォルトは cms です。

-label *label*
PKCS #7 ファイルからインポートする証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-new_label ラベル
ターゲット鍵リポジトリー内の証明書に割り当てられるラベルを指定します。 このパラメーターを指定しない場合、ソース・キー・リポジトリーと同じラベルが証明書に割り当てられます。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

ALW AIX, Linux, and Windows での鍵リポジトリー内の証明書のリスト表示

鍵リポジトリー内にある証明書をリストするには、この手順を使用します。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵リポジトリー内にある証明書に関する情報を表示できます。

runmqakm の使用

- **runmqakm** コマンドを使用して、鍵リポジトリー内の証明書のラベルをリストするには、以下のコマンドを発行します。

```
runmqakm -cert -list -db filename -pw password
```

- **runmqakm** コマンドを使用して鍵リポジトリー内の証明書の詳細をリストするには、以下のコマンドを発行します。

```
runmqakm -cert -details -showOID -db filename -pw password
         -label label
```

ここで、

-file *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
リストする証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

これらのパラメータと指定できる値の詳細については、実行 mqakm -cert 。

runmqktool の使用

V 9.4.0 V 9.4.0

- **runmqktool** コマンドを使用して、鍵リポジトリー内の証明書のラベルをリストするには、以下のコマンドを発行します。

```
runmqktool -list -keystore filename -storepass password
```

- **runmqktool** コマンドを使用して鍵リポジトリー内の証明書の詳細をリストするには、以下のコマンドを発行します。

```
runmqktool -list -keystore filename -storepass password -alias label -v
```

ここで、

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
リストする証明書のラベルを指定します。 証明書ラベルは大/小文字を区別しません。

-v
証明書の詳細を含む詳細出力を要求します。

これらのパラメーターおよび指定可能な値について詳しくは、 リストを参照してください。

ALW AIX, Linux, and Windows で鍵リポジトリーから証明書を削除する操作

鍵リポジトリーから個人証明書または CA 証明書を削除するには、この手順を使用します。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵リポジトリーから証明書を削除できます。 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。

runmqakm の使用

runmqakm コマンドを使用して証明書を削除するには、以下のコマンドを発行します。

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

ここで、

-file *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-label *label*
削除する証明書のラベルを指定します。 証明書ラベルでは大/小文字が区別されます。

-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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -cert を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

runmqktool コマンドを使用して証明書を削除するには、以下のコマンドを発行します。

```
runmqktool -delete -keystore filename -storepass password -alias label
```

ここで、

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。

-storepass パスワード
鍵リポジトリーのパスワードを指定します。

-alias ラベル
削除する証明書のラベルを指定します。 証明書ラベルは大/小文字を区別しません。

これらのパラメーターおよび指定可能な値について詳しくは、 delete を参照してください。

ALW AIX, Linux, and Windows での鍵リポジトリーの変換

この手順を使用して、鍵リポジトリーを別のタイプに変換します。

runmqakm (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して、鍵リポジトリー・パスワードを別のタイプに変換できます。

runmqakm の使用

以下のコマンドを発行して、 **runmqakm** コマンドで鍵リポジトリーを変換します。

```
runmqakm -keydb -convert -db filename -pw password
         -new_db filename -new_pw password
         -old_format type -new_format type
```

ここで、

-file *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-new_db ファイル名
新しい鍵リポジトリーの完全修飾ファイル名を指定します。

-new_pw パスワード
新しい鍵リポジトリーのパスワードを指定します。

-old_format 型
鍵リポジトリーの現在のタイプを指定します。 指定可能な値は、次のとおりです。

- pkcs12
- cms

-new_format タイプ
鍵リポジトリーの新しいタイプを指定します。 指定可能な値は、次のとおりです。

- pkcs12
- cms

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -keydb を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行して、 **runmqktool** コマンドで鍵リポジトリーを変換します。

```
runmqktool -importkeystore -srckeystore filename -destkeystore filename
           -srcstoretype type -deststoretype type
           -srcstorepass password -deststorepass password
```

ここで、

-all
鍵リポジトリーと同じパスワードで保護されているすべての項目のパスワードも変更することを指定します。

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。

-destkeystore ファイル名
新しい鍵リポジトリーの完全修飾ファイル名を指定します。

-srcstoretype 型
鍵リポジトリー・タイプを指定します。

-deststoretype タイプ
新しい鍵リポジトリー・タイプを指定します。

-srcstorepass パスワード
鍵リポジトリーのパスワードを指定します。

-deststorepass パスワード
新しい鍵リポジトリーのパスワードを指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 importkeystore を参照してください。

ALW AIX, Linux, and Windows での鍵リポジトリー・パスワードの変更

鍵リポジトリー・パスワードを変更するには、この手順を使用します。

鍵リポジトリーのパスワードは、**runmqakm** (GSKCapiCmd) または **runmqktool** (keytool) コマンドを使用して変更できます。

注:

- V 9.4.0 V 9.4.0 **runmqktool** コマンドを使用すると、個々の秘密鍵と秘密鍵を保護するパスワードとは無関係に、鍵リポジトリーのパスワードを変更することができます。 PKCS #12 鍵リポジトリーの場合、鍵リポジトリーのパスワードと、鍵リポジトリー内のすべての鍵を保護するパスワードは同じでなければなりません。 **runmqktool** コマンドを使用して鍵リポジトリーのパスワードを変更する場合は、鍵パスワードも変更されるように **-all** パラメーターが指定されていることを確認してください。
- 鍵リポジトリー・パスワードが stash ファイルに保管されていない場合は、キュー・マネージャー構成に保管されているパスワード、または鍵リポジトリーにアクセスするすべての IBM MQ client アプリケーションに保管されているパスワードも変更する必要があります。 詳しくは、301 ページの『キュー・マネージャーの鍵リポジトリー・パスワードの指定 (AIX, Linux, and Windows)』 および 302 ページの『AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定』 を参照してください。

runmqakm の使用

以下のコマンドを発行し、 **runmqakm** コマンドを使用して鍵リポジトリーのパスワードを変更します。

```
runmqakm -keydb -changepw -db filename -pw password -new_pw password -stash
```

ここで、

-file *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーの現在のパスワードを指定します。

-new_pw パスワード
鍵リポジトリーの新規パスワードを指定します。

-stash
オプション。 新しい鍵リポジトリー・パスワードを stash ファイルに保管するには、このオプションを指定します。 代わりに IBM MQ パスワード保護システムを使用してパスワードを暗号化する場合は、パスワードを stash ファイルに保管する必要はありません。

これらのパラメーターおよび指定可能な値について詳しくは、 runmqakm -keydb を参照してください。

runmqktool の使用

V 9.4.0 V 9.4.0

以下のコマンドを発行し、 **runmqktool** コマンドを使用して鍵リポジトリーのパスワードを変更します。

```
runmqktool -storepasswd -all -keystore filename -storepass password
           -new password
```

ここで、

-all
鍵リポジトリーと同じパスワードで保護されているすべての項目のパスワードも変更することを指定します。

-keystore ファイル名
鍵リポジトリーの完全修飾ファイル名を指定します。

-storepass パスワード
鍵リポジトリーの現在のパスワードを指定します。

-new パスワード
鍵リポジトリーの新規パスワードを指定します。

これらのパラメーターおよび指定可能な値について詳しくは、 storepasswd を参照してください。

ALW 秘密鍵の管理 AIX, Linux, and Windows

キー リポジトリ内の秘密キーを管理するには、次の手順に従います。

秘密鍵は、**runmqakm** （GSKCapiCmd） 指示。 秘密鍵は、**runmqktool** (keytool)コマンドは、IBM MQ 。

秘密鍵の作成

次のコマンドを発行してランダムな秘密鍵を作成します。**runmqakm** 指示：

```
runmqakm -secretkey -create -db filename -pw password
         -label label -size key_size
```

ここで、

-db *filename*
キー リポジトリの完全修飾ファイル名を指定します。 キー リポジトリがすでに存在している必要があります。

-pw *password*
キー リポジトリのパスワードを指定します。

-label *label*
キーに添付されるラベルを指定します。

-size *key_size*
キーのサイズをバイト単位で指定します。

これらのパラメータと指定できる値の詳細については、以下を参照してください。 runmqakm -秘密鍵。

秘密鍵の抽出

秘密鍵を抽出するには、次のコマンドを実行します。**runmqakm** 指示：

```
runmqakm -secretkey -extract -db filename -pw password
         -label label -target filename -format format
```

ここで、

-db *filename*
キー リポジトリの完全修飾ファイル名を指定します。 キー リポジトリがすでに存在している必要があります。

-pw *password*
キー リポジトリのパスワードを指定します。

-label *label*
抽出するキーのラベルを指定します。

-target *filename*
宛先ファイルの完全修飾ファイル名を指定します。

-フォーマットフォーマット
宛先ファイル内のキーの形式を指定します。 値は `ascii` のために Base64-encodedASCII または `binary` キーのバイナリコピー用。 デフォルトは `ascii` です。

これらのパラメータと指定できる値の詳細については、以下を参照してください。 runmqakm -秘密鍵。

秘密鍵の追加

秘密鍵を抽出するには、次のコマンドを実行します。**runmqakm** 指示：

```
runmqakm -secretkey -add -db filename -pw password
         -label label -file filename -format format
```

ここで、

-db *filename*
キー リポジトリの完全修飾ファイル名を指定します。 キー リポジトリがすでに存在している必要があります。

-pw *password*
キー リポジトリのパスワードを指定します。

-label *label*
キーに添付されるラベルを指定します。

-file *filename*
キーを含むファイルの名前を指定します。

-フォーマットフォーマット
キーの形式を指定します。 値は `ascii` のために Base64-encodedASCII または `binary` バイナリデータ用。 デフォルトは `ascii` です。

これらのパラメータと指定できる値の詳細については、以下を参照してください。 runmqakm -秘密鍵。

ALW 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。

IBM MQ 環境に証明書を保管しないが、すべての証明書を暗号ハードウェアに保管する場合でも、その環境を準備するために鍵リポジトリーを作成する必要があります。 鍵リポジトリーは、キュー・マネージャーがその **SSLKEYR** 属性で参照するため、またはクライアント・アプリケーションが MQSSLKEYR 環境変数で参照するために必要です。 この鍵リポジトリーは、認証要求を作成する場合にも必要です。

runmqakm (GSKCapiCmd) コマンドを使用して、鍵リポジトリーを作成します。

以下のコマンドを発行し、 **runmqakm** コマンドを使用して鍵リポジトリーを作成します。

```
runmqakm -keydb -create -db filename -pw password -type type -stash
```

ここで、

-db *filename*
鍵リポジトリーの完全修飾ファイル名を指定します。

-pw *password*
鍵リポジトリーのパスワードを指定します。

-type 型
データベースのタイプを指定します。 IBM MQ によって使用される鍵リポジトリーの場合、値は `cms` または `pkcs12` でなければなりません。

-stash
オプション。 指定すると、暗号化された鍵リポジトリー・パスワードがファイルに保存されます。

ALW *PKCS #11 ハードウェア用の個人用証明書の要求*

この手順を使用して、暗号ハードウェアを使用するキュー・マネージャーまたは IBM MQ MQI client の個人証明書を要求します。

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。

Deprecated 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。

暗号ハードウェアで証明書要求を作成する前に、 563 ページの『PKCS #11 ハードウェアでの証明書の管理』 で説明されている手順を実行して、鍵リポジトリーを作成します。

以下のコマンドを発行して、 **runmqakm** (GSKCapiCmd) コマンドを使用して認証要求を作成します。

```
runmqakm -certreq -create -crypto module_name -tokenlabel hardware_token
         -pw password -label label
```

```
          -dn distinguished_name -size key_size
          -file filename -fips -sig_alg algorithm
```

ここで、

-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 *password*
暗号ハードウェアにアクセスするためのパスワードを指定します。

-label *label*
証明書ラベルを指定します。

IBM MQ によって使用される TLS 証明書のラベルは、 **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または IBM MQ MQI client ユーザー ID をすべて小文字で付加した値のいずれかです。 詳しくは、27 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。 識別名には少なくとも 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 : **runmqakm** コマンドは、郵便番号属性を PC ではなく POSTALCODE として参照します。 郵便番号を使用して証明書を要求するために **runmqakm** コマンドを使用する場合は、必ず **-dn** パラメーターに POSTALCODE を指定してください。

-size *key_size*
鍵のサイズを指定します。 値は 512、 1024、または 2048 です。

-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 モードの場合、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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
認証要求の作成時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、認証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。

これらのパラメータと指定できる値の詳細については、 mqakm -certreq を実行します。

次のタスク

CA に証明書要求を送信します。 CA から署名済み証明書を受け取ったら、その署名済み証明書を鍵リポジトリーに追加します。 詳細については、565 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。

ALW *PKCS #11 ハードウェアでの個人証明書の受け取り*

この手順を使用して、キュー・マネージャーまたは IBM MQ MQI client の個人証明書を暗号ハードウェアに対して受信します。

個人証明書に署名した CA の CA 証明書を、暗号ハードウェアまたは 2 次キー・リポジトリーのいずれかに追加します。 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 証明書を鍵リポジトリー・ファイルに追加するには、 551 ページの『AIX, Linux, and Windows での鍵リポジトリーへの CA 証明書またはトラステッド証明書の公開部分の追加』の手順に従います。

runmqakm (GSKCapiCmd) コマンドを使用して、個人証明書を鍵リポジトリーに追加するには、以下のコマンドを発行します。

```
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*
暗号ハードウェアに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 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、 IBM Crypto for C (ICC) コンポーネントは FIPS 140-2 検証済みの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*
CA 証明書を保管するために使用される鍵リポジトリー・ファイルの完全修飾ファイル名を指定します。

-secondaryDBpw *password*
CA 証明書を保管するために使用される鍵リポジトリー・ファイルのパスワードを指定します。

IBM MQ コンポーネント構成ファイルでのパスワードの保護

IBM MQ の特定の機能を使用するには、その機能で使用されるパスワードの指定が必要になる場合があります。 IBM MQ に提供されるパスワードは、パスワード保護システムを使用して保護することができます。

以下のリストは、暗号化されたパスワードを処理する各コンポーネントで使用される用語を説明しています。

初期鍵
パスワードを保護するために使用される暗号鍵。

デフォルトの初期鍵
パスワードの暗号化時に初期鍵を指定しない場合に使用されるデフォルトの暗号鍵。

プレーン・テキスト・ストリング
暗号化されたストリング。通常はパスワードです。

暗号化されたパスワード・ストリング
IBM MQ が理解できる形式の暗号化されたパスワードを含むストリング。

初期キーの指定

コンポーネントごとに、パスワードの暗号化に使用される初期鍵を指定することを選択できます。

- 初期キーを指定しない場合は、コンポーネントのデフォルトの初期キーが使用されます。 デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 これは、デフォルトの初期鍵で暗号化されたパスワードは、別のインストール済み環境でパスワードを暗号化解除できる可能性があるため、安全に保護されないことを意味します。
- 独自の固有の初期鍵を指定した場合、指定した初期鍵へのアクセス権限を持つユーザーのみがパスワードを暗号化解除できます。

重要 : 保管されているパスワードに最高レベルのセキュリティーを提供するには、 IBM MQ コンポーネントごとに固有の初期キーを指定します。

独自の初期キーを使用することを選択した場合は、リストされているコンポーネントごとに固有の初期キーを指定します。 初期鍵は、そのコンポーネントの構成に保管されているパスワードを保護するために使用されます。 パスワードを暗号化解除するには、同じ初期鍵をコンポーネントに対しても使用可能にする必要があります。

ほとんどのコンポーネントでは、初期キーをファイルに指定する必要があります。 初期鍵ファイルに含まれている初期鍵は、以下の要件を満たしている必要があります。

- 少なくとも 1 文字の長さでなければなりません。
- これは単一行のテキストでなければなりません。

初期キーの最大長は無制限で、任意の文字を指定できます。 十分なセキュリティーを確保するために、少なくとも 16 文字の長さの初期キーを指定してください。 例えば、初期鍵ファイルに以下のストリングが含まれているとします。

```
Th1sIs@n3Ncypt|onK$y
```

初期鍵ファイルへのアクセスは、オペレーティング・システムのファイル許可を使用して初期鍵にアクセスする必要があるユーザーのみに制限する必要があります。

パスワード保護の利点と制限について詳しくは、 572 ページの『パスワード暗号化による保護の制限』を参照してください。

各 IBM MQ コンポーネントでのパスワードの保護

いくつかの IBM MQ コンポーネントは、保管されたパスワードを保護できます。 コンポーネントによっては、以下のいずれかのメカニズムを使用して、これらのパスワードを提供することができます。

- IBM MQ キュー・マネージャーまたは IBM MQ client に直接提供されます。
- 環境変数で指定します。
- 構成ファイルに保管されます。

各コンポーネントは、パスワードを暗号化する方法を提供します。 ほとんどのコンポーネントでは、パスワードを IBM MQ に提供したり構成に保管したりする前に、パスワードを暗号化する必要があります。

重要 : あるコンポーネントで使用するために生成された暗号化パスワードを、別のコンポーネントの構成ファイルにコピーすることはできません。 特定のコンポーネントで使用するために暗号化されたパスワードは、同じコンポーネントで提供されるユーティリティーを使用して保護する必要があります。

パスワード保護をサポートする各 IBM MQ コンポーネントのパスワードを保護する方法の詳細は、以下のセクションにリストされています。

- Advanced Message Security
- 568 ページの『Managed File Transfer』
- 569 ページの『IBM MQ Internet Pass-Thru』
- 569 ページの『暗号ハードウェアを使用する IBM MQ clients』
- 570 ページの『IBM MQ キュー・マネージャー』

- 570 ページの『IBM MQ C クライアント・アプリケーション』
- V 9.4.0 571 ページの『ネイティブ HA 構成』
- V 9.4.0 572 ページの『IBM MQ キュー・マネージャー (qm.ini ファイル内の AuthToken スタンザ)』

Advanced Message Security

Advanced Message Security (AMS) Java クライアントは、メッセージの保護に使用される秘密鍵を含む鍵ストアへのアクセスを必要とします。

Advanced Message Security (AMS) MCA インターセプトを実行するように構成された MQI クライアントまたはキュー・マネージャーでは、メッセージの保護に使用される秘密鍵を含む PKCS#11 暗号ハードウェアまたは PEM ファイルへのアクセスが必要になる場合があります。

これらの鍵リポジトリーにアクセスするには、keystore.conf という AMS 構成ファイルにパスワードを指定する必要があります。 **runamscred** コマンドを使用して、keystore.conf ファイルに含まれている機密情報を保護します。 例:

```
runamscred -f <keystore configuration file>
```

runamscred コマンドは、 **-f** パラメーターを使用して指定されたファイル内の機密パラメーターを保護します。

IBM MQ インストール済み環境では、以下の 2 つの **runamscred** コマンドを使用できます。

- <IBM MQ installation root>/bin に置かれている MQI **runamscred** コマンド
- Java **runamscred** コマンド (<IBM MQ installation root>/java/bin にあります)

重要 : 互換性を確保するために、

1. Java **runamscred** コマンドを使用して、 Java AMS クライアントで使用される構成ファイルを保護し、MQI **runamscred** コマンドを使用して、 AMS を使用する IBM MQ MQI clients の構成ファイルを保護します。
2. **runamscred** コマンドの実行後に、必要なすべての機密情報が保護されていることを確認します。
3. 保護されたパスワードを含むファイルを、通常どおり AMS 対応アプリケーションに提供します。

デフォルトでは、 **runamscred** コマンドは、構成ファイル内のパスワードをデフォルトの初期鍵で暗号化します。 特定の初期鍵を使用してパスワードを暗号化するには、以下のいずれかのメカニズムを使用して、初期鍵を含むファイルの名前を優先順に指定します。

1. **runamscred** コマンドに対する **-sf** パラメーター。
2. **MQS_AMSCRED_KEYFILE** 環境変数。
3. keystore.conf 構成ファイル内の **amscred.keyfile** パラメーター。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

runamscred コマンドを実行して AMS 構成内のパスワードを暗号化するときに初期鍵ファイルを指定する場合は、AMS アプリケーションの実行時にも同じ初期鍵ファイルを指定する必要があります。 以下のメカニズムを使用して、初期鍵ファイルの名前を優先順位の順に指定することができます。

1. **MQS_AMSCRED_KEYFILE** 環境変数。
2. keystore.conf 構成ファイル内の **amscred.keyfile** パラメーター。

デフォルトでは、**runamscred** コマンドは、IBM MQ 9.2 より前の AMS バージョンと互換性のない保護システムを使用して資格情報を保護します。 IBM MQ 9.2 より前のバージョンと互換性のある資格情報保護

システムを使用して構成ファイルを保護するには、 **runamscred** コマンドの実行時に **-sp 0** パラメーターを指定します。

Managed File Transfer

Managed File Transfer (MFT) は、キュー・マネージャーおよびその他のリソースへのアクセスに必要な資格情報を、以下の XML プロパティー・ファイルに保管します。

MQMFTCredentials.xml
このファイルには、以下の資格情報が含まれています。

- エージェント、調整、およびコマンド・キュー・マネージャーへの接続に使用される資格情報。
- セキュア通信に使用される鍵ストアへのアクセスに使用されるパスワード。

ProtocolBridgeCredentials.xml
このファイルには、FTP、SFTP、FTPS などのプロトコル・サーバーへの接続に使用される資格情報が含まれています。

ConnectDirectCredentials.xml
このファイルには、 Connect:Direct® エージェントが Connect:Direct ノードに接続するために使用する資格情報が含まれています。

これらのファイルに保管されている機密情報を保護するには、 fteObfuscate コマンドを使用します。 **-f** フラグを使用して、保護するファイルの名前を指定します。 以下に例を示します。

```
fteObfuscate -f <File to protect>
```

デフォルトでは、**fteObfuscate** コマンドは、デフォルトの初期鍵を使用して資格情報を保護します。 特定の初期鍵で資格情報を保護するには、 **-sf** パラメーターを使用して、初期鍵を含むファイルへのパスを指定します。 以下に例を示します。

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

重要 :

1. **fteObfuscate** の実行後に、すべての機密情報が保護されていることを確認します。
2. 通常どおりに保護ファイルを MFT に提供します。

MFT 構成で資格情報を保護するために **fteObfuscate** コマンドを実行するときに初期鍵ファイルを指定する場合は、MFT の始動時にも同じ初期鍵ファイルを指定する必要があります。 以下のメカニズムを使用して、初期鍵ファイルの名前を優先順位の順に指定することができます。

1. **com.ibm.wmqfte.cred.keyfile** Java システム・プロパティー。

   注 : IBM MQ 9.3.1 および IBM MQ 9.3.0 Fix Pack 10 より前では、この Java システム・プロパティーの名前のつづりが **com.ibm.wqmfte.cred.keyfile** と誤っていました。 IBM MQ 9.3.1 および IBM MQ 9.3.0 Fix Pack 10 以降、 Managed File Transfer は両方のバージョンの Java システム・プロパティーを使用して、以前のバージョンとの互換性を維持します。 両方の Java システム・プロパティーが設定されている場合、正しいスペルのプロパティー **com.ibm.wmqfte.cred.keyfile** の値が使用されます。
2. エージェント、ロガー、コマンド、および調整プロパティー・ファイル内のプロパティー。
3. `installation.properties` ファイル内の **commonCredentialsKeyFile** プロパティー。

詳細については、574 ページの『MFT での保管資格情報の暗号化』 を参照してください。

デフォルトでは、 **fteObfuscate** コマンドは、 IBM MQ 9.2 より前の MFT バージョンと互換性のない保護システムを使用して資格情報を保護します。 IBM MQ 9.2 より前のバージョンと互換性のある資格情報

保護システムを使用して構成ファイルを保護するには、 **fteObfuscate** コマンドの実行時に **-sp 0** パラメーターを指定します。

IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) 構成ファイルには、さまざまなリソースにアクセスするために使用されるパスワードを含めることができます。

mqiptPW コマンドを使用して、 MQIPT 構成ファイル内のパスワードを保護します。

mqiptPW コマンドは、暗号化するパスワードの入力を求めるプロンプトを出し、暗号化されたパスワードを返します。 暗号化されたパスワードを MQIPT 構成ファイルにコピーします。

デフォルトでは、 **mqiptPW** コマンドは、デフォルトの初期鍵を使用してパスワードを暗号化します。 特定の初期鍵を使用してパスワードを暗号化するには、 **-sf** パラメーターを使用して、初期鍵を含むファイルへのパスを指定します。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

詳しくは、 パスワード暗号鍵の指定を参照してください。

鍵リポジトリー・パスワードの暗号化時に初期鍵ファイルを指定する場合は、 MQIPT の始動時にも同じ初期鍵ファイルを指定する必要があります。 以下のメカニズムを使用して、初期鍵ファイルの名前を優先順位の順に指定することができます。

1. MQIPT を開始するために使用されるコマンドの **-sf** パラメーター。
2. **MQS_MQIPTCRED_KEYFILE** 環境変数。
3. **com.ibm.mq.ipt.cred.keyfile** Java プロパティー。
4. MQIPT ホーム・ディレクトリー内の `mqipt_cred.key` という名前のファイル。 MQIPT ホーム・ディレクトリーは、 MQIPT 構成ファイルを含むディレクトリーです。

デフォルトでは、 **mqiptPW** コマンドは、 IBM MQ 9.2 より前の MQIPT バージョンと互換性のない保護システムを使用して資格情報を保護します。 IBM MQ 9.2 より前のバージョンと互換性のある資格情報保護システムでパスワードを保護するには、 IBM MQ 9.2 より前のバージョンでサポートされている **mqiptPW** コマンド構文を使用します。

暗号ハードウェアを使用する IBM MQ clients

TLS 通信で使用される秘密鍵と証明書を保管するために PKCS #11 暗号ハードウェアを使用するように IBM MQ クライアントを構成できます。 PKCS #11 デバイスにアクセスするには、IBM MQ client に提供される構成ストリングの一部としてパスワードを指定する必要があります。

重要 : MQSCO 構造体の **CryptoHardware** フィールド、またはキュー・マネージャーの **SSLCRYP** 属性を使用して指定されたパスワードは、このメカニズムを使用して保護することはできません。

このパスワードは、 **runp11cred** コマンドを使用して保護できます。このコマンドは、 IBM MQ インストール・ディレクトリーの `bin` フォルダーにあります。

runp11cred コマンドは、暗号化するパスワードの入力を求めるプロンプトを出し、暗号化されたパスワードを返します。 暗号化されたパスワードは、暗号化ハードウェア構成ストリングにコピーする必要があります。

例えば、暗号ハードウェア構成ストリングが以下のようになっているとします。

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

runp11cred コマンドでパスワードの入力を求めるプロンプトが出されたら、 `Passw0rd` と入力します。このコマンドは、以下のようなストリングを返します。

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

暗号化されたパスワードを含む以下のストリングを指定するには、暗号化ハードウェア構成ストリング内のパスワードを、 **runp11cred** コマンドによって返されるストリングに置き換えます。

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

IBM MQ client アプリケーションの実行時に、以下のいずれかの方法で、暗号化されたパスワードを含む暗号ハードウェア構成ストリングを指定します。

- クライアント構成ファイルの SSL スタンザ内にある **SSLCryptoHardware** 属性。
- **MQSSLCRYP** 環境変数。

デフォルトでは、**runp11cred** コマンドはデフォルトの初期鍵を使用してパスワードを暗号化します。 独自の初期鍵を使用してパスワードを保護するには、以下のいずれかのメカニズムを優先順に使用して、初期鍵を含むファイルの名前を指定します。

1. **runp11cred** コマンドに対する **-sf** パラメーター。
2. **MQS_SSLCRYP_KEYFILE** 環境変数。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

鍵リポジトリー・パスワードの暗号化時に初期鍵ファイルを指定する場合は、IBM MQ client の実行時に初期鍵を含むファイルの名前も指定する必要があります。 以下のいずれかのメカニズムを優先度順に使用して、初期鍵ファイル名を指定します。

1. **MQS_SSLCRYP_KEYFILE** 環境変数。
2. クライアント構成ファイルの **SSL** スタンザ内にある **SSLCryptoHardwareKeyFile** 属性。

IBM MQ キュー・マネージャー

IBM MQ キュー・マネージャーは、パスワードをいくつかの属性に内部的に保管します。 例えば、キュー・マネージャーの **KEYRPWD** 属性などです。 キュー・マネージャーは、パスワードがディスク上のファイルに保管される前に、パスワードを自動的に暗号化します。

キュー・マネージャーの TLS キー・リポジトリーに対するパスワードは、 IBM MQ パスワード保護システムまたはキー・リポジトリー stash ファイルを使用して保護できます。 これら 2 つの方法について詳しくは、 297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』を参照してください。

キュー・マネージャーがパスワードを暗号化するときには、独自の初期鍵を指定しない限り、デフォルトの初期鍵が使用されます。 独自の初期鍵を使用するには、暗号化されたキュー・マネージャー属性を設定する前に、キュー・マネージャーの **INITKEY** 属性を固有の強い鍵に設定します。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

警告 : 暗号化された属性の値を設定した後に初期鍵が変更された場合、暗号化された属性は新しい初期鍵で再暗号化されません。 したがって、鍵リポジトリー・パスフレーズを再提供せずに初期鍵を変更すると、 IBM MQ は鍵リポジトリー・パスフレーズを暗号化解除できなくなり、鍵リポジトリーにアクセスできなくなります。

詳しくは、 INITKEY を参照してください。

IBM MQ C クライアント・アプリケーション

IBM MQ C クライアント・ライブラリーは、保護された特定のリソースにアクセスするためにパスワードを必要とします。 例えば、TLS を使用してキュー・マネージャーに接続するアプリケーション用の TLS キー・リポジトリーなどです。

鍵リポジトリー・パスワードは、 IBM MQ パスワード保護システムまたは鍵リポジトリー stash ファイルを使用して保護できます。 これら 2 つの方法について詳しくは、 297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』を参照してください。

IBM MQ パスワード保護システムを使用してパスワードを保護するには、 **runmqicred** コマンドを使用します。 このコマンドは、 *MQ_INSTALLATION_PATH*/bin ディレクトリーにあります。

runmqicred コマンドは、暗号化するパスワードの入力を求めるプロンプトを出し、暗号化されたパスワードを返します。 暗号化されたパスワードは、プレーン・テキスト・パスワードの代わりにクライアント・アプリケーションで使用できます。

例えば、*MQKEYRPWD* 環境変数を使用して TLS 鍵リポジトリー・パスワードを指定することを選択し、TLS 鍵ストア・パスワードが Passw0rd であるとします。 **runmqicred** を実行するときに、プロンプトが出されたら Passw0rd と入力します。 このコマンドは、以下のようなストリングを返します。

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

このストリングを *MQKEYRPWD* 環境変数の値として設定します。

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

デフォルトでは、 **runmqicred** コマンドは、デフォルトの初期鍵を使用してパスワードを暗号化します。 独自の初期鍵を使用してパスワードを保護するには、以下のいずれかのメカニズムを使用して、鍵を含むファイルの名前を優先順に指定します。

1. **runmqicred** コマンドに対する **-sf** パラメーター。
2. *MQS_MQI_KEYFILE* 環境変数。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

パスワードの暗号化時に初期鍵ファイルを指定する場合は、クライアント・アプリケーションの実行時に初期鍵を使用できるようにする必要もあります。

詳細については、302 ページの『AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの指定』を参照してください。

ネイティブ HA 構成

V 9.4.0

インスタンス間のネイティブ HA ログ複製トラフィックは、TLS を使用して暗号化できます。 ログ複製トラフィックを保護するために使用される証明書は、 qm.ini ファイルの **NativeHALocalInstance** スタンザで指定されている鍵リポジトリーに保管されます。

鍵リポジトリー・パスワードは、 IBM MQ パスワード保護システムまたは鍵リポジトリー stash ファイルを使用して保護できます。 これら 2 つの方法について詳しくは、 297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』を参照してください。

IBM MQ パスワード保護システムでネイティブ HA キー・リポジトリー・パスワードを保護するには、 **runmqicred** コマンドを使用します。

runmqicred コマンドは、暗号化するパスワードの入力を求めるプロンプトを出し、暗号化されたパスワードを返します。 プレーン・テキストのパスワードの代わりに、暗号化されたパスワードを使用する必要があります。 qm.ini ファイルの **NativeHALocalInstance** スタンザの **KeyRepositoryPassword** 属性の値を、コマンドが返す暗号化されたパスワードに設定します。

デフォルトでは、 **runmqicred** コマンドは、デフォルトの初期鍵を使用してパスワードを暗号化します。 独自の初期鍵を使用してパスワードを保護するには、以下のいずれかのメカニズムを使用して、鍵を含むファイルの名前を優先順に指定します。

1. **runmqicred** コマンドに対する **-sf** パラメーター。

2. *MQS_MQI_KEYFILE* 環境変数。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

鍵リポジトリー・パスワードの暗号化時に初期鍵ファイルを指定する場合は、 `qm.ini` ファイルの **NativeHALocalInstance** スタンザの **InitialKeyFile** 属性を使用して、同じ初期鍵ファイルも指定する必要があります。

詳しくは、 qm.ini ファイルの NativeHALocal インスタンス・スタンザを参照してください。

IBM MQ キュー・マネージャー (qm.ini ファイル内の AuthToken スタンザ)

Linux V 9.4.0 AIX

IBM MQ 9.3.4 以降、 AIX または Linux システム上で稼働する IBM MQ キュー・マネージャーに接続する IBM MQ MQI clients は、認証トークンを使用してキュー・マネージャーで認証を行うことができます。 キュー・マネージャーは、認証トークンを受け入れ、トークン発行者の公開鍵証明書またはトークンの署名に使用される秘密鍵にアクセスできるように構成されている必要があります。 信頼できる発行者の公開鍵証明書または秘密鍵を含む鍵リポジトリーは、パスワードで保護されます。

鍵リポジトリー・パスワードは、 IBM MQ パスワード保護システムまたは鍵リポジトリー stash ファイルを使用して保護できます。 これら 2 つの方法について詳しくは、 297 ページの『AIX, Linux, and Windows での鍵リポジトリー・パスワードの暗号化』 を参照してください。

IBM MQ パスワード保護システムで認証トークン・キー・リポジトリーのパスワードを保護するには、 **runqmcred** コマンドを使用してパスワードを暗号化します。

runqmcred コマンドは、暗号化するパスワードの入力を求めるプロンプトを出し、暗号化されたパスワードを返します。 プレーン・テキストのパスワードの代わりに、暗号化されたパスワードを使用する必要があります。 暗号化されたパスワードをファイルにコピーし、そのファイルへのパスを `qm.ini` ファイル内の **AuthToken** スタンザの **KeyStorePwdFile** 属性に含めます。

デフォルトでは、 **runqmcred** コマンドは、デフォルトの初期鍵を使用してパスワードを暗号化します。 特定の初期鍵を使用してパスワードを暗号化するには、 **-sf** パラメーターを使用して、初期鍵を含むファイルへのパスを指定します。

注意 : デフォルトの初期鍵は、すべての IBM MQ インストール済み環境で同じです。 パスワードを安全に保護するには、パスワードを暗号化するときに、インストール済み環境に固有の初期鍵を指定します。

重要 : パスワードの暗号化時に初期鍵を指定する場合は、キュー・マネージャーがパスワードを暗号化解除できるように、キュー・マネージャーの **INITKEY** 属性に同じ初期鍵を指定する必要があります。 キュー・マネージャーの **INITKEY** 属性が既に設定されている場合は、 **runqmcred** コマンドの実行時に同じ初期キーを使用します。 キュー・マネージャーの **INITKEY** 属性について詳しくは、 INITKEY を参照してください。

例えば、ファイル `/home/initial.key` 内の初期鍵を使用して認証トークン鍵ストアのパスワードを暗号化するには、次のコマンドを発行します。

```
runqmcred -sf /home/initial.key
```

詳細については、332 ページの『ローカル鍵ストアを使用して認証トークンを受け入れるためのキュー・マネージャーの構成』 を参照してください。

パスワード暗号化による保護の制限

IBM MQ は、さまざまな構成ファイルに保管されるパスワードの AES-128 暗号化をサポートします。 Advanced Encryption Standard (AES) 暗号化を使用して IBM MQ 構成内のパスワードを保護する場合は、提供される保護の制限について理解する必要があります。

IBM MQ 構成ファイル内のパスワードを暗号化しても、パスワードが保護または保護されることを意味するものではありません。 これは、暗号化されたパスワードにアクセスできるが、暗号鍵を知らないユーザーによってパスワードが容易にリカバリーされないようにするだけです。 IBM MQ プロセスでは、使用する平文パスワードを取得するために、暗号化されたパスワードと暗号化解除鍵の両方にアクセスする必要があります。 これらのデータ項目は両方とも、ファイル・システム上の、 IBM MQ がアクセスできる場所に保管する必要があります。 構成ファイルに配置されているパスワードを暗号化するユーザーは、暗号鍵にもアクセスする必要があります。 攻撃者が IBM MQ と同じファイル・セットにアクセスできる場合、パスワードに AES 暗号化を適用すると、最小レベルの保護のみが提供されます。

ただし、保存されているパスワードを暗号化することは、パスワードの偶発的な開示を防止し、暗号化解除鍵も共有されていない場合に構成ファイルの共有を可能にするため、考慮する必要があります。

暗号化解除鍵を含むファイルが共有されないようにすることに加えて、ファイルがシステム上の他のユーザーから保護されるように注意する必要があります。 IBM MQ 構成ファイルはすべてのユーザーがアクセスできますが、暗号化解除鍵が含まれているファイルに対する許可は、必要最小限に制限してください。 IBM MQ プロセスの実行に使用するユーザー ID には、暗号化解除鍵を含むファイルを読み取るためのアクセス権限が付与されている必要があります。 ただし、ファイルを読み取るアクセス権限をグループに付与したり、システム上のすべてのユーザーに付与したりする必要はありません。

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に、**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しなければなりません。
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

641 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 での保管資格情報の暗号化

Managed File Transfer (MFT) では、いくつかのユーザー ID と資格情報が必要です。これらは 2 つの XML ファイルに保管され、 **fteObfuscate** コマンドを使用して難読化することができます。

資格情報ファイル

MQMFTCredentials.xml
このファイルには、エージェント、調整、およびコマンドの各キュー・マネージャーに接続するためのユーザー ID と資格情報が含まれます。 キュー・マネージャーへのセキュア接続のための鍵ストアにアクセスするための資格情報も、同じファイルに保管されます。

578 ページの『MFT と IBM MQ の接続認証』 ファイルの場所を定義するプロパティー値の詳細については、MQMFTCredentials.xml を参照してください。

ProtocolBridgeCredentials.xml
このファイルには、プロトコル・サーバーに接続するためのユーザー ID と資格情報が含まれます。

fteObfuscate コマンドを使用した資格情報の暗号化

fteObfuscate コマンドは、以下のパラメーターを受け入れます。

- **-f** *credentials_file_name* (必須)

  注: Deprecated このパラメーターは、 IBM MQ 9.2.0 以降で非推奨になった **-credentialsFile** パラメーターに代わるものです。
- **-sp** 保護モード
- **-sf** 資格情報鍵ファイル *(credentials_key_file)*
- **-o** *output_file_name*

パラメーターの詳細については、**fteObfuscate** を参照してください。

保護モードも資格情報鍵ファイルも指定しない場合、このコマンドはデフォルトの保護モードを使用し、最新のアルゴリズムを使用しますが、資格情報を暗号化するための固定鍵が使用されます。

保護モード 0 を指定し、資格情報鍵ファイルを指定しない場合、このコマンドは、製品の以前のリリースと同じように機能します。 推奨されない保護の使用を示す警告メッセージがコンソールに表示されます。

保護モード 0 を指定し、資格情報鍵ファイルを指定すると、保護モード 0 の使用時の鍵ファイルの指定は無効であるということを示すエラー出力がコンソールに表示されます。

保護モード 1 を指定し、資格情報鍵ファイルを指定しない場合、このコマンドは最新のアルゴリズムを使用しますが、資格情報を暗号化するための固定鍵が使用されます。

保護モード 1 を指定し、資格情報鍵ファイルを指定すると、このコマンドは最新のアルゴリズムを使用して資格情報を暗号化します。

保護モード 1 を指定した場合、または保護モードを指定しない場合、存在しない資格情報鍵ファイルを指定すると、そのファイルが存在しないことを示すエラーがコンソールに出力されます。

保護モード 1 を指定した場合、または保護モードを指定しない場合、読み取り不可能な資格情報鍵ファイルを指定すると、そのファイルが読み取り不可能であることを示すエラーがコンソールに出力されます。

2 の保護モードを指定し、資格情報鍵ファイルを指定しない場合、コマンドは最新のアルゴリズムを使用して資格情報を暗号化するために保護モード 2 を使用し、暗号化するために固定鍵を使用します。

2 の保護モードを指定し、資格情報鍵ファイルを指定した場合、コマンドは、最新のアルゴリズムを使用して資格情報を暗号化するために保護モード 2 を使用し、暗号化するためにユーザー指定の鍵を使用します。

保護モード 2 を指定した場合、または保護モードを指定しない場合、存在しない資格情報鍵ファイルを指定すると、そのファイルが存在しないことを示すエラーがコンソールに出力されます。

保護モード 2 を指定した場合、または保護モードを指定しない場合、読み取り不可能な資格情報鍵ファイルを指定すると、そのファイルが読み取り不可能であることを示すエラーがコンソールに出力されます。

資格情報の復号

初期鍵ファイルへのパスは、さまざまな場所に指定できます。 デフォルト以外の初期鍵を使用して暗号化された資格情報を復号するには、優先順で示した以下のいずれかの方法で、初期鍵を含むファイルの名前を MFT に提供する必要があります。

1. Java システム・プロパティーを使用する。以下に例を示します。

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

   注:

   - IBM MQ 9.3.1 および IBM MQ 9.3.0 Fix Pack 10 より前では、この Java システム・プロパティーの名前のつづりが製品コード内で `com.ibm.wqmfte.cred.keyfile` と誤っていました。 IBM MQ 9.3.1 および IBM MQ 9.3.0 Fix Pack 10 以降、プロパティー名のスペルが `com.ibm.wmqfte.cred.keyfile` に修正されました。 Managed File Transfer は、資格情報の暗号化および暗号化解除に使用する初期鍵を含むファイルがユーザーによって指定されているかどうかを検査する際に、両方のバージョンの Java システム・プロパティーを使用します。 これにより、プロパティー名の正しいスペルを使用しながら、古いつづりの誤った名前との後方互換性を維持することができます。 両方の Java システム・プロパティーが設定されている場合、正しいスペルのプロパティー `com.ibm.wmqfte.cred.keyfile` の値が使用されることに注意してください。
   - IBM MQ 9.3.1 および IBM MQ 9.3.0 Fix Pack 10 より前では、プロパティー `com.ibm.wqmfte.cred.keyfile` を使用します。

   .
2. エージェント・プロパティー・ファイル、コマンド・プロパティー・ファイル、調整プロパティー・ファイル、またはロガー・プロパティー・ファイルにプロパティーを設定する。 プロパティー・ファイルの名前、およびその中で設定する必要があるプロパティーを以下の表に示します。

プロパティー・ファイル	**プロパティー名**
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. installation.properties ファイル内で。

   個々のプロパティー・ファイルにプロパティーを追加する代わりに、 **commonCredentialsKeyFile** プロパティーを既存の共通 `installation.properties` ファイルに追加して、エージェント、ロガー、およびコマンドが同じプロパティーを使用できるようにすることができます。

複数の場所でさまざまな **CredentialsKeyFile** プロパティーを定義した場合は、以下のようにします。

- エージェントおよびロガーに使用される資格情報鍵ファイルのパスは、そのエージェントまたはロガーの `output0.log` ファイルに記録されます。
- コマンドに使用されている資格情報鍵ファイルのパスがコンソールに表示されます。

Java システム・プロパティー **com.ibm.wmqfte.cred.keyfile** は、他のすべてをオーバーライドします。 システム・プロパティーが設定されていない場合、エージェントは `agent.properties` ファイルを調べ、その後に初期鍵ファイルの `installation.properties` ファイルを調べます。

それでも初期鍵ファイルが見つからず、**fteObfuscate** コマンドで保護モードを 1 に設定した場合、エージェントは `output0.log` ファイルにエラー・メッセージを記録します。

fteObfuscate コマンドで保護モードを `0` に設定した場合、非推奨を示す警告メッセージがログに記録されます。

ロガーとコマンドは、同じステップに従って初期鍵ファイルを見つけます。

プロトコル・ブリッジおよび Connect:Direct ブリッジ

プロトコル・ブリッジは、FTP、SFTP、および FTPS サーバーに接続するために、プロパティー・ファイル (ProtocolBridgeProperties.xml) を使用します。 このプロパティー・ファイルには、これらのサーバーへの接続に必要な接続属性が含まれています。

ProtocolBridgeProperties.xml ファイル内の **credentialsFile** 属性または **credentialsKeyFile** 属性の値を変更する場合は、ブリッジ・エージェントを再始動する必要があります。

これらの属性の 1 つは **credentialsFile** であり、値には、UID、PWD、またはこれらのサーバーに接続するために必要な鍵を含む XML ファイルへのパスが含まれます。 属性のデフォルト値は「*ProtocolBridgeCredentials.xml*」 で、ファイルは MQMFTCredentials.xml ファイルと同じように、ホーム・ディレクトリーにあります。

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

MQMFTCredentails.xml と同様、*ProtocolBridgeCredentials.xml* は **fteObfuscate** コマンドを使用して暗号化できます。 復号の目的で、次のテキストに示すように、追加の要素 **credentialsKeyFile** を使用して、資格情報鍵ファイルへの必要なパスを指定できます。 このパスには、環境変数を含めることができます。

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

注 : **agentCredentialsKeyFile** エージェント・プロパティーの値、 installation.properties の **commonCredentialsKeyFile** プロパティーの値、またはシステム・プロパティー **com.ibm.wqmfte.cred.keyfile** の値を指定しても、**credentialsKeyFile** 属性に指定された値には影響しません。

同様に、Connect:Direct ブリッジは、*ConnectDirectNodeProperties.xml* を使用して Connect:Direct サーバーに接続します。 この XML ファイルには、資格情報 XML ファイルへのパスを定義する属性とともに、必要な接続情報が含まれています。 この資格情報 XML ファイルには、UID または PWD、および Connect:Direct サーバーに接続するために必要な追加情報が含まれています。

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

ProtocolBridgeCredentials.xml ファイルと同様、 *ConnectDirectCredentials.xml* は **fteObfuscate** コマンドを使用して暗号化できます。 復号の目的で、次のテキストに示すように、追加の要素 **credentialsKeyFile** を使用して、資格情報鍵ファイルへの必要なパスを指定できます。 このパスには、環境変数を含めることができます。

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

注 : **agentCredentialsKeyFile** エージェント・プロパティー、installation.properties の **commonCredentialsKeyFile** プロパティー、またはシステム・プロパティー **com.ibm.wqmfte.cred.keyfile** を使用して値を指定しても、**credentialsKeyFile** 属性に指定された値に影響はありません。

credentialsKeyFile 要素は、*ProtocolBridgeProperties.xml* ファイル内で **credentialsFile** 要素を指定せずに指定できます。

credentialsFile 要素を指定しない場合は、プロトコル・ブリッジ・エージェントによってデフォルトの資格情報ファイル *ProtocolBridgeCredentials.xml* が使用され、**credentialsKeyFile** 属性で指定されている鍵ファイルの値が、資格情報ファイルの復号に使用されます。

同様に、**credentialsKeyFile** 要素は、*ConnectDirectNodeProperties.xml* ファイル内で **credentialsFile** 要素を指定せずに指定できます。

credentialsFile 要素を指定しない場合は、Connect:Direct ブリッジによってデフォルトの資格情報ファイル *ConnectDirectCredentials.xml* が使用され、**credentialsKeyFile** 属性で指定されている鍵ファイルの値が、資格情報ファイルの復号に使用されます。

z/OS 上のデータ・セットからの鍵の使用

z/OS

z/OS では、**MQMFTCredentials** を指定し、PDSE を使用して資格情報鍵ファイルを提供できます。 580 ページの『Configuring MQMFTCredentials.xml on z/OS』を参照してください。

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

fteObfuscate (機密データの暗号化)

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。

Multi 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、 579 ページの『Multiplatforms での MQMFTCredentials.xml の構成』を参照してください。

z/OS 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、580 ページの『Configuring MQMFTCredentials.xml on z/OS』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

Multiplatforms 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。

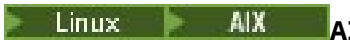
このタスクについて

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・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。



AIX 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 *97. :* 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 *98. :* 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

表 98. : 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。 (続き)		
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 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。

個々のプロパティー・ファイルにプロパティーを追加する代わりに、**commonCredentialsKeyFile** プロパティーを既存の共通 `installation.properties` ファイルに追加して、エージェント、ロガー、およびコマンドが同じプロパティーを使用できるようにすることができます。

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。

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **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

582 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。

IBM MQ File Transfer 構造の作成

関連資料

MFT の資格情報ファイルのフォーマット

MFT に保管されている資格情報の暗号化

fteObfuscate: 機密データの暗号化

z/OS Configuring MQMFTCredentials.xml on z/OS

If Managed File Transfer (MFT) is configured with security enabled, connection authentication requires all MFT agents, and commands that connect to a queue manager, to supply user ID and password credentials.

Similarly, MFT loggers might be required to specify a user ID and password when connecting to a database.

This credential information can be stored in the MFT credentials file. Note that the credentials files are optional, however, it is easier to define the file or files that you require before you customize the environment.

In addition to this, if you have credentials files, you receive fewer warning messages. The warning messages inform you that MFT considers that queue manager security is off, and therefore you are not supplying authentication details.

You can find a sample credentials file in the `MQ_INSTALLATION_PATH/mqft/samples/credentials` directory.

Here is an example of an `MQMFTCredentials.xml`file:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

When a job with userid ADMIN needs to connect to queue manager MQPH, it passes user ID *JOHNDOEH* and uses password *cXXXX*.

If the job is run by any other user ID, and connects MQPH, that job passes user ID *NONEH* and password *yXXXX*.

The default location for the `MQMFTCredentials.xml` file is the user's home directory on z/OS UNIX System Services (USS). It is also possible to store the file in either a different location on USS, or in a member within a partitioned data set.

If the credentials file is stored in a different location, then you can use the following properties to specify where the commands should look for it:

Table 99. : Properties that define the location of the MQMFTCredentials.xml file for various commands.

Type of command	**Property file**	**Property name**
Command which connects to the coordination queue manager	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Command which connects to the command queue manager	connection.properties	connectionQMgrAuthenticationCredentialsFile
Command that connects to an agent process	agent.properties	agentQMgrAuthenticationCredentialsFile
Command that connects to a logger process	logger.properties	loggerQMgrAuthenticationCredentialsFile

Table 100. : Properties that define the location of the MQMFTCredentials.xml file for agents and logger processes.

Type of command	**Property file**	**Property name**
MFT agents	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

For details about what commands and processes connect to which queue manager, see Which MFT commands and processes connect to which queue manager.

To create the credentials file within a partitioned data set, carry out the following steps:

- Create a PDSE with format VB and logical record length (Lrecl) 200.

- Create a member within the data set, make a note of the data set and member, and add the following code to the member:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!—credentials information goes here-->
</tns:mqmftCredentials>
```

You can protect the credentials file using a security product, for example, RACF, but the user IDs running the Managed File Transfer commands, and administering the agent and logger processes, need read access to this file.

You can obscure information in this file using the JCL in member BFGCROBS. This takes the file and encrypts the IBM MQ user ID and password. For example member BFGCROBS takes the line

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

and creates

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

If you want to keep the user ID to IBM MQ user ID mapping, you can add comments to the file. For example

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

These comments are unchanged by the obscuring process.

Note that the content is obscured, not strongly encrypted. You should limit which user IDs have access to the file.

Related tasks

“Multiplatforms での MQMFTCredentials.xml の構成” on page 579

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

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。

- デフォルトでは、「**ユーザー識別互換モード**」チェック・ボックスは選択されていません。 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。

- 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。

  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。

  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

31 ページの『MQCSP パスワード保護』
MQCSP 構造で指定された認証資格情報は、 IBM MQ MQCSP パスワード保護機能を使用して保護することも、TLS 暗号化を使用して暗号化することもできます。

関連資料

578 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

584 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。

585 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、585 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。

AIX, Linux, and Windows プラットフォームでのサンドボックスの機能

ALW AIX, Linux, and 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

588 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。

584 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

588 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

585 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。

584 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。 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、275 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、591 ページの『MFT 用の SSL/TLS 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

MFT 用の SSL/TLS プロパティー

一部の MFT プロパティー・ファイルには、SSL プロパティーと TLS プロパティーが含まれます。 SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化できます。

以下の MFT プロパティー・ファイルには、SSL プロパティーが含まれています。

- 「MFT agent.properties ファイルの SSL/TLS プロパティー」
- 「MFT coordination.properties ファイルの SSL/TLS プロパティー」
- 「MFT command.properties ファイルの SSL/TLS プロパティー」
- 「MFT logger.properties ファイルの SSL/TLS プロパティー」

Managed File Transfer での SSL または TLS の使用については、589 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。

IBM WebSphere MQ 7.5 以降、ファイルまたはディレクトリーの場所を表す Managed File Transfer プロパティーの一部で環境変数を使用できます。 これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更 (プロセスを実行しているユーザーなど) に合わせて変えることができます。 詳しくは、「MFT プロパティーでの環境変数の使用」を参照してください。

関連概念
MFT 構成オプション (Multiplatforms)
関連資料
MFT プロパティーでの環境変数の使用

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

IBM MQ は、チャネル認証レコードを使用して、チャネル・レベルでより正確にアクセスを制御します。 つまり、デフォルトでは、新しく作成されたキュー・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拒否します。

チャネル認証の詳細については、51 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 MQ 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 *agent_name* で終わるキュー名) について、クライアント接続を使用して IBM MQ 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、294 ページの『AIX, 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。 詳しくは、589 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、275 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。
   - AIX and Linux 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

ALW 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。 新しい IBM MQ キュー・マネージャーではデフォルト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。キュー・マネージャーの **CONNAUTH** 属性に指定されている IBM 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場合) 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。 キュー・マネージャーの **CONNAUTH** 属性に指定されている IBM 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する場合) の場合、ユーザー名とパスワードがキュー・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 101. 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。

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、294 ページの『AIX, Linux, and Windows での SSL/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。

V 9.4.0 V 9.4.0 IBM MQ 9.4.0 以降、AMQP チャネルはキュー・マネージャー上の CMS 鍵リポジトリーをサポートしなくなりました。 **runmqakm** コマンドを使用して、CMS 鍵リポジトリーを、サポートされている PKCS #12 形式に変換することができます。 例えば、以下のコマンドを使用して、

sslTest.kdb という名前の鍵リポジトリーを CMS 形式から PKCS #12 形式に変換できます。 新しい鍵リポジトリーには sslTest.p12 という名前が付けられ、パスワード passw0rd で保護されます。

```
runmqakm -keydb -convert -type cms -db sslTest.kdb -stashed -new_format pkcs12 -target
sslTest.p12 -new_pw passw0rd
```

Java 認証・承認サービス (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。 598 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

ALW 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 *102.* 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

ALW AMQP チャネルのための JAAS の構成

Java Authentication and Authorization Service (JAAS) カスタム・モジュールを使用して、接続時に AMQP クライアントによって 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 ベースのシステムでの認証に既に JAAS モジュールを使用しており、それらのモジュールを MQ への AMQP 接続の認証に再利用したい場合は、カスタム JAAS モジュールを使用することをお勧めします。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。 スタンザには、 JAAS javax.security.auth.spi.LoginModule インターフェースを実装する 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。
   - Linux AIX 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
   ```

   次に例を示します。

   ```
   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 サービスは、 JAAS 構成スタンザで構成された Java クラスにアクセスできる必要があります。

   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。

関連概念

651 ページの『メッセージ・チャネル・エージェント (MCA) のインターセプトと AMS』

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 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。
- ALW 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ます。 このチェックは、初期化の初期段階で実行され、環境変数 *AMQ_AMS_FIPS_OFF* の値を読み取ります。いずれかの値に設定されている場合、IBM Global Security Kit (GSKit) コードはそのアプリケーションで非 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。 対称鍵を後続のメッセージで再利用できるようにするには、アプリケーションは、メッセージをキューに書き込んだ後もキューを開いたままにしておく必要があります。 対称鍵を MQPUT1 操作で再利用することはできません。 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

644 ページの『AMS で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 鍵リングが使用されます。

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

11 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と *AMS*

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。

z/OS **サーバー間 MCA インターセプト**
IBM MQ for z/OS 9.1.3 以降、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 Advanced Message Security interception on message channels

On z/OS, Advanced Message Security (AMS) interception provides an additional option of security policy protection (SPLPROT) to sender, server, receiver, and requester channels, allowing you to support AMS and to communicate with business partners who do not support AMS.

Taking the example of a clearing house communicating with a bank, Figure 1 shows that, without AMS interception, both sides of the system need to support AMS.

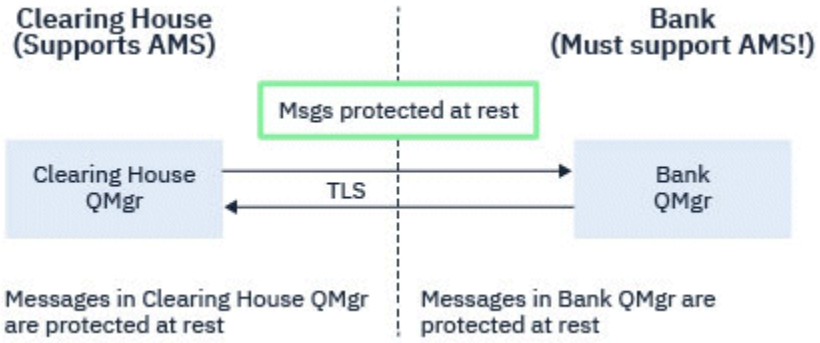


Figure 32. Usage of AMS without AMS interception

A key benefit of the AMS interception option is, that if your enterprise has AMS configured, and not all of your business partners support AMS, you can remove protection from outbound messages and protect inbound messages on channels to and from those business partners that do not support AMS.

Using the example of a clearing house and banks, this scenario is shown in Figure 2, where there is a message flow between the clearing house, banks, and business partners where some institutions have AMS, and others do not.

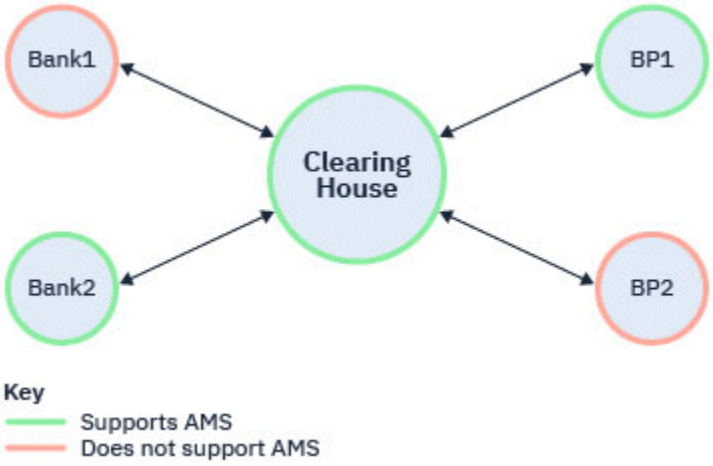


Figure 33. Some partners support AMS and some do not

Typically the channels are TLS enabled.

However, there might be a case where some banks and business partners do not support AMS, and there is a requirement to be able to exchange messages between all banks and business partners. This scenario is shown in Figure 3

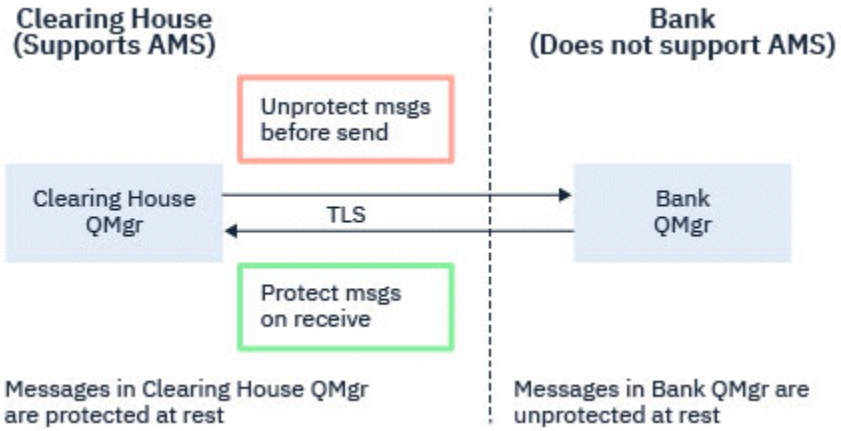


Figure 34. Message flow between business partners

Related tasks

Server-to-server message channel interception example configurations

z/OS *AMS interception on server-to-server message channels*

Server-to-server message channel interception provides a means to control if messages should have any applicable Advanced Message Security (AMS) policies applied to them, when sender type message channel agents get messages from transmission queues, and receiver type message channel agents put messages to target queues.

This allows AMS protection to be enabled on a queue manager when communicating, using server-to-server message channels of type sender, server, receiver, and requester, with a queue manager that does not have AMS enabled.

That is, AMS protected messages in AMS enabled queue managers can be unprotected prior to being sent to non-AMS enabled queue managers, and unprotected messages received from non-AMS enabled queue managers can be protected, by applicable AMS policies, on AMS enabled queue managers.

Configuring server-to-server message channel interception

Server-to-server message channel interception is configured with the SPLPROT attribute on channels with a channel type of sender, server, receiver, or requester. The available options to configure the behavior are dependent on the channel type specified:

PASSTHRU
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。

User ID for message channel interception

The requirement for user IDs used with server-to-server message channel interception are the same as those for existing AMS enabled applications. For a running channel, the sending message channel agent gets messages from a transmission queue and the receiving message channel agent puts messages to target queues. The message channel agent user ID (MCAUSER) field, set on server to server channels, defines the user ID under which message channel agents perform put and get requests.

With server-to-server message channel interception, AMS functions are performed during get and put requests, as with other AMS enabled applications. Therefore, message channel agent user Ids have the same requirements as those for AMS application user IDs.

The MCAUSER used to perform the put and get is configurable, and dependent on whether it is an outbound or inbound channel. See MCAUSER for details of how the chosen user ID performs actions on the message channel agent. As such, the user ID that the channel initiator is running under is the user ID that is to be used for AMS functions performed during server-to-server message channel interception. Therefore, these user IDs have the same requirements as those for AMS application user IDs.

Authentication is performed using the existing rules for the channel detailed for channels with PUTAUT configuration. See user IDs used by the channel initiator for more information.

Note: Server-to-server message channel interception does not take into account the value of the PUTAUT channel attribute.

Message size and MAXMSGL

Due to AMS protection, the message size of protected messages will be larger than the original message size.

Protected messages are larger than unprotected messages. Therefore, the value of the **MAXMSGL** attribute, on both queues and channels, might need to be altered to take into account the size of protected messages.

Related reference

Server-to-server message channel interception example configurations

AMS のエラー処理

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 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中の場合:

- 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、607 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 *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 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中の場合:

- 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 680 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

607 ページの『AMS のエラー処理』
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。

AMS のユーザー・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 (AMS) 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 (AMS)

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 613 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て Java アプリケーションを使用する予定の場合は、Java **keytool** コマンド V 9.4.0 V 9.4.0 または IBM MQ **runmqktool** コマ

ンドを使用して、JKS 鍵ストアと証明書を作成する必要があります。 詳しくは、631 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』を参照してください。 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 用の新しい鍵データベースを作成します。
   例えば、新しい鍵データベースを作成するには、次のコマンドを発行します。

   ```
   runmqakm -keydb -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -type cms -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します。
   - 暗号化された鍵データベース・パスワードをファイルに stash するには、 **-stash** パラメーターを含めます。
2. 暗号化で使用するユーザー alice を識別するための新しい自己署名証明書を作成します。
   例えば、以下のコマンドを発行して、新しい自己署名証明書を作成します。

   ```
   runmqakm -cert -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -stashed
   -label Alice_Cert -dn "CN=alice, O=IBM, C=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。 実動システムでは、認証局によって署名された証明書を使用することをお勧めします。
   - **-label** パラメーターは、必要な情報を受け取るためにインターセプターが検索する証明書の名前を指定します。
   - **-dn** パラメーターは、証明書の識別名 (DN) の詳細を指定します。 識別名はユーザーごとに固有でなければなりません。
3. ユーザー bob に対してステップ 611 ページの『1』 と 611 ページの『2』 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 = 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、645 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、644 ページの『AMS で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 Deprecated 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。

Linux AIX *AMS 上の AIX and Linux 用クイック・スタート・ガイド*

このガイドを使用して、 AIX and Linux 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。

- のランタイム
- サーバー
- サンプル・プログラム
- IBM Global Security Kit (GSKit)
- 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

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 620 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、631 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、645 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、644 ページの『AMS で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、メッセージはユーザー alice によって Deprecated SHA1 アルゴリズムを使用して署名され、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 *Example AMS configurations on z/OS*

This section provides example configurations of policies and certificates for Advanced Message Security queuing scenarios on z/OS.

See Configuring Advanced Message Security for z/OS for details on how you configure Advanced Message Security.

The examples cover the Advanced Message Security policies required, and the digital certificates that must exist relative to users and key rings. The examples assume that the users involved in the scenarios have been set up by following the instructions provided in Grant users resource permissions for Advanced Message Security.

See also server-to-server message channel interception examples.

z/OS *Local queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from a queue, local to the putting and getting applications.

The example queue manager and queue are:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

These users are used:

```
WMQBNK6  - AMS task user
```

```
TELLER5  - Sending user
FINADM2  - Recipient user
```

Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected messages. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK6.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue a user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security requires:

- The CA certificate (chain).
- The user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6.

When the certificates have been imported on the z/OS system running BNK6, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

In this example, no certificate is required for the recipient user.

Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6. To create the key rings use the RACDCERT ADDRING commands:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user, WMQBNK6, and a key ring for the sending user, 'TELLER5'. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

When the key rings have been created, the relevant certificates can be connected:

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

```
F BNK6AMSM,REFRESH KEYRING
```

Create the Advanced Message Security policy

In this example, integrity-protected messages are put to queue FIN.XFER.Q7 by an application running as user 'TELLER5', and retrieved from the same queue by an application running as user 'FINADM2', so only one Advanced Message Security policy is required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

After defining the policy, either restart the BNK6 queue manager, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configuration. For example:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS 上の AMS 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 (Manage RACF digital certificates) 」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、名前には大/小文字の区別があることに注意してください。

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 Deprecated SHA1 であり、送信側ユーザーの識別名 (DN) は 'CN=Teller5,O=BCO,C=US' であり、受信側ユーザーは 'CN=FinAdm2,O=BCO,C=US' です。 メッセージ・データの暗号化に使用されるアルゴリズムは Deprecated 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 *Remote queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from queues managed by two different queue managers. The two queue managers can be running on the same z/OS system, or on different z/OS systems, or one queue manager can be on a distributed system running Advanced Message Security.

The example queue managers and queues are:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Note: For this example, BNK6 and BNK7 are queue managers running on different z/OS systems.

These users are used:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

The steps to configure this scenario are as follows:

Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected message. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK7.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security require:

- The CA certificate (chain).
- The sending user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to RACDCERT (Manage RACF digital certificates) in the *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6 and BNK7.

In this example, the sending certificate must be imported on the z/OS system running BNK6, and the CA certificate must be imported on the z/OS system running BNK7. When the certificates have been

imported, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example, on BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6 and BNK7.

To create the key rings use the RACDCERT ADDRING command, on BNK6:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the sending user on BNK6. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

On BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user on BNK7. No user key ring is required for 'TELLER5' on BNK7.

When the key rings have been created, the relevant certificates can be connected.

On BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

On BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

On BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Create the Advanced Message Security policies

In this example, integrity-protected messages are put to remote queue FIN.XFER.Q7 on BNK6 by an application running as user 'TELLER5', and retrieved from local queue FIN.RCPT.Q7 on BNK7 by an application running as user 'FINADM2', so two Advanced Message Security policies are required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command to define an integrity policy for the remote queue on BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

Also, use the CSQ0UTIL utility to run the following command to define an integrity policy for the local queue on BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK7. The policy name and associated queue is FIN.RCPT.Q7. The algorithm expected for the sender's signature is MD5, and the distinguished name (DN) of the sending user is expected to be 'CN=Teller5,O=BCO,C=US'.

After defining the two policies, either restart the BNK6 and BNK7 queue managers, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configurations. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

On BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *z/OS 上の AMS 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 (Manage RACF digital certificates)」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 Deprecated 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・データの暗号化に使用されるアルゴリズムは Deprecated 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。 送信側の署名に予期されるアルゴリズムは Deprecated SHA1 であり、送信側ユーザーの識別名 (DN) は 'CN=Teller5,O=BCO,C=US' であり、受信側ユーザーは 'CN=FinAdm2,O=BCO,C=US' です。 メッセージ・データの暗号化解除に使用されるアルゴリズムは Deprecated 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

608 ページの『AMS プラットフォーム上の Windows のクイック・スタート・ガイド』 または 615 ページの『AMS 上の AIX and Linux 用クイック・スタート・ガイド』の説明に従って、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ます。

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 AIX and Linux) 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、636 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。 詳しくは、608 ページの『AMS プラットフォーム上の Windows のクイック・スタート・ガイド』 および 615 ページの『AMS 上の AIX and Linux 用クイック・スタート・ガイド』 を参照してください。

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。 これは、ご使用のプラットフォームの「クイック・スタート・ガイド」で使用されているのと同じディレクトリーに作成することもできます。 詳しくは、608 ページの『AMS プラットフォーム上の Windows のクイック・スタート・ガイド』 および 615 ページの『AMS 上の AIX and Linux 用クイック・スタート・ガイド』 を参照してください。

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. AIX and Linux 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 AIX and Linux) の指示に従っているため、既に `keystore.conf` ファイルを持っている場合は、既存のファイルを編集してこれらの行を追加することができます。
- 詳細については、645 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。

重要 : 抽出 および エクスポート という用語は、証明書管理コマンドごとに異なる方法で使用されます。

- IBM Global Security Kit (GSKit) **runmqakm** コマンドは、 抽出 という用語を使用して鍵ストアから証明書の公開部分のみをコピーするプロセスを指し、 エクスポート という用語は、証明書とそれに関連する公開鍵と秘密鍵をある鍵ストアから別の鍵ストアにコピーするプロセスを指します。
- Java **keytool** コマンド V 9.4.0 V 9.4.0 および IBM MQ **runmqktool** コマンドでは、エクスポート という用語を使用して、鍵ストアから証明書の公開部分のみをコピーするプロセスを指します。

export を誤って使用すると、秘密鍵を公開することによってアプリケーションが危険にさらされる可能性があるため、この区別は重要です。 この区別は非常に重要であるため、 IBM MQ の資料では、これらの用語を一貫して使用しています。 これらの理由から、以下の手順では、 **keytool** コマンドで `exportcert` オプションを使用して 証明書を抽出 します。

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 キューに定義され、ユーザー alice によって Deprecated SHA1 アルゴリズムを使用して署名され、ユーザー bob 用に 256 ビットの AES アルゴリズムを使用して暗号化されたポリシーの例です。

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

注 : DN は、鍵データベースからの各ユーザーの証明書に指定された DN と正確に一致します。

次のタスク

定義したポリシーを検証するには、以下のコマンドを実行します。

```
dspmqspl -m QM_VERIFY_AMS
```

一連の setmqspl コマンドとしてポリシーの詳細を出力するには、 -export フラグを指定します。 これにより、既に定義されているポリシーが格納されます。

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. セットアップのテスト

始める前に

使用している Java のバージョンに、無制限の 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てください。

注 : IBM MQ インストール済み環境で提供される Java のバージョンには、既にこれらのポリシー・ファイルがあります。 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。 異なるユーザーの下でのプログラムの実行について詳しくは、608 ページの『AMS プラットフォーム上の Windows のクイック・スタート・ガイド』 および 615 ページの『AMS 上の AIX and Linux 用クイック・スタート・ガイド』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。

AMS でのリモート・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)

IBM Integration Bus を使用した *AMS* での保護されたメッセージの経路指定

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 AIX) 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。 以下で説明されている方法を使用することが重要です。 **タスク 5. クイック・スタート・ガイド (Windows または AIX) 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)
   ```
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)
   ```

[1] cecil's

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。

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 AIX) 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。 以下で説明されている方法を使用することが重要です。 **タスク 5. クイック・スタート・ガイド (Windows または AIX) 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は、ローカル・インストール済み環境に IBM Global Security Kit (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、 Managed File Transfer シナリオで説明されているように、1 つのマシンに 2 つの Managed File Transfer キューと 2 つのエージェント AGENT1 および AGENT2 があり、単一のキュー・マネージャーを共有す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 AIX) 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、631 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。

手順

- Multi
  マルチプラットフォームに Advanced Message Security をインストールします。
- z/OS
  IBM MQ Advanced for z/OS をインストールします。
- z/OS

IBM MQ Advanced for z/OS Value Unit Edition をインストールします。

関連タスク

Advanced Message Security のアンインストール

z/OS Auditing for AMS on z/OS

Advanced Message Security (AMS) for z/OS provides a means for optional auditing of operations by applications on policy protected queues. When enabled, IBM System Management Facility (SMF) audit records are generated for the success and failure of these operations on policy-protected queues. Operations audited include MQPUT, MQPUT1, and MQGET.

Auditing is disabled by default, however, you can activate auditing by configuring _AMS_SMF_TYPE and _AMS_SMF_AUDIT in the configured Language Environment® _CEE_ENVFILE file for the AMS address space. For more information, see Create procedures for Advanced Message Security. The _AMS_SMF_TYPE variable is used to designate the SMF record type and is a number between 128 and 255. A SMF record type of 180 is usual, however is not mandatory. Auditing is disabled by specifying a value of 0. The _AMS_SMF_AUDIT variable configures whether audit records are created for operations that are successful, operations that fail, or both. The auditing options can also be dynamically changed while AMS is active using operator commands. For more information, see Operating Advanced Message Security.

The SMF record is defined using subtypes, with subtype 1 being a general auditing event. The SMF record contains all data relevant to the request being processed.

The SMF record is mapped by the CSQ0KSMF macro (note the zero in the macro name), which is provided in the target library SCSQMACS. If you are writing data-reduction programs for SMF data, you can include this mapping macro to aid in the development and customization of SMF post-processing routines.

In the SMF records produced by Advanced Message Security for z/OS, the data is organized into sections. The record consists of:

- a standard SMF header
- a header extension defined by Advanced Message Security for z/OS
- a product section
- a data section

The product section of the SMF record is always present in the records produced by Advanced Message Security for z/OS. The data section varies based on subtype. Currently, one subtype is defined and therefore a single data section is used.

SMF is described in the z/OS System Management Facilities manual (SA22-7630). Valid record types are described in the SMFPRMxx member of your system PARMLIB data set. See SMF documentation for more information.

Advanced Message Security audit report generator (CSQ0USMF)

Advanced Message Security for z/OS provides an audit report generator tool called CSQ0USMF which is provided in the installation SCSQAUTH library. Sample JCL to run the CSQ0USMF utility called CSQ40RSM is provided in the installation library SCSQPROC.

Before running the CSQ0USMF utility, the SMF type 180 records must be dumped from the system SMF data sets to a sequential data set. As an example, this JCL dumps SMF type 180 records from an SMF data set, and transfers them to a target data set:

```
//IFAUDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYSn.MANn.syst,DISP=SHR
//OUTDD1 DD DSN=your.target.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
```

```
OUTDD(OUTDD1,TYPE(180))
/*
```

You must verify the actual SMF data set names used by your installation. The target data set for the dumped records must have a record format of VBS, and a record length of 32760.

Note: If SMF logstreams are being used, you must use program IFASMFDL to dump a logstream out to a sequential dataset. See Processing type 116 SMF records for an example of the JCL used.

The target data set can then be used as input to the CSQ0USMF utility to produce an AMS audit report. For example:

```
//STEP1 EXEC PGM=CSQ0USMF,
// PARM=('/ -SMFTYPE 180 -M qmgr')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//    DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SMFIN DD DSN=your.target.dataset,DISP=SHR
//
```

The CSQ0USMF program accepts two optional parameters, which are listed in Table 103 on page 644:

Table 103. CSQ0USMF optional parameters

Parameter	Value	Description
SMFTYPE	nnn	The SMF record type applicable to the audit report. The CSQ0USMF program uses only SMF records that match the SMFTYPE value when generating the report. If you do not specify SMFTYPE, a default value of 180 is used.
M	qmgr	The IBM MQ queue manager name applicable to the audit report. If you do not specify the -M parameter, the audit report will include all audit records for all queue managers represented in the SMFIN data set.

AMS で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 鍵リングが使用されます。

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。

ALW AIX, Linux, and Windows では、鍵ストアのロケーションは鍵ストア構成ファイル (デフォルトでは `keystore.conf`) で提供されます。 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。

- Linux IBM i AIX IBM i では、以下の AIX and Linux $HOME/.mqs/keystore.conf なります。
- Windows Windows 上: *%HOMEDRIVE%%HOMEPATH%*\.mqs\keystore.conf

指定した鍵ストアのファイル名と場所を使用する場合は、以下のコマンド例に示すように、**MQS_KEYSTORE_CONF** 環境変数でこれを指定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:
  - Linux AIX AIX and Linux 上: `export MQS_KEYSTORE_CONF=path/filename`
  - Windows Windows 上: `set MQS_KEYSTORE_CONF=path\filename`

  注: 複数のドライブ名が使用可能な場合は、Windows 上のパスでドライブ名を指定できます。使用する場合は、指定する必要があります。

keystore.conf ファイル内の機密情報の保護

鍵ストア・ファイルの機密情報 (パスワードなど) にアクセスするには、IBM MQ Advanced Message Security (AMS) が鍵ストアにアクセスし、メッセージに署名して暗号化できるように、トークンを提供する必要があります。

AMS で提供されている **runamscred** コマンドを使用して、鍵ストア構成ファイルに含まれている機密情報を保護する必要があります。 構成ファイルを保護する方法について詳しくは、663 ページの『構成ファイルの AMS パスワード保護のセットアップ』を参照してください。

パスワードを保護する場合は、カスタムの強力な暗号鍵を使用する必要があります。 実行時にパスワードにアクセスするには、この暗号鍵が AMS に提供されている必要があります。

暗号鍵ファイルの場所を提供するには、以下の 2 つの方法があります。

- keystore.conf ファイル内の **amscred.keyfile** 構成プロパティー
- **MQS_AMSCRED_KEYFILE** 環境変数

優先順位は **MQS_AMSCRED_KEYFILE** の後に **amscred.keyfile** が続き、その後にデフォルト・キーが続きます。

関連概念

672 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。

673 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。

AMSCRED
パスワード保護システムに関連するパラメーター。

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。

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、650 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
pkcs11.encrypted = no
```

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
pem.encrypted = no
```

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
```

Java JCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
```

Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

Java PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.secondary_keystore_pass = password
pkcs11.encrypted = no
```

表 *104.* 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				
		Java (PKCS#11、JKS、JCEKS、および JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			
IBM i public	✓		IBM i ✓			
IBM i password	✓		IBM i ✓			
library	✓	✓		✓		
certificate	✓	✓		✓	✓	
token	✓	✓		✓		
token_pin	✓	✓		✓		
secondary_keystore	✓	✓		✓		
secondary_keystore_password	✓	✓				
encrypted		✓	IBM i ✓	✓		
keystore_pass	✓	✓				
key_pass		✓				
provider		✓				
keyfile						✓あなた

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。

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。
- 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。

AMS のネイティブ・パスワード保護ツールを使用してこのフィールドを保護する必要があります。649 ページの『パスワードの保護』を参照してください。

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。

Java 操作の場合のみ。Java AMS パスワード保護ツールを使用してこのフィールドを保護する必要があります。649 ページの『パスワードの保護』を参照してください。

ネイティブ操作の場合のみ。AMS のネイティブ・パスワード保護ツールを使用してこのフィールドを保護する必要があります。649 ページの『パスワードの保護』を参照してください。

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。

Java 環境の場合は、JKS 鍵ストアが必要であり、**secondary_keystore_password** を指定する必要があります。

secondary_keystore_password
Java PKCS#11 のみ。

secondary_keystore プロパティーで指定する JKS 鍵ストアのパスワード。 Java AMS パスワード保護ツールを使用してこのフィールドを保護する必要があります。649 ページの『パスワードの保護』を参照してください。

encrypted
Java 、および IBM MQ 9.3.0 からは PKCS#11 および 

PEM 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。

Java 操作の場合のみ。 Java AMS パスワード保護ツールを使用してこのフィールドを保護する必要があります。649 ページの『パスワードの保護』を参照してください。

key_pass
Java 構成のみ。

ユーザーの秘密鍵のパスワード。

Java 操作の場合のみ。Java AMS パスワード保護ツールを使用してこのフィールドを保護する必要があります。649 ページの『パスワードの保護』を参照してください。

keyfile
この構成ファイルに含まれているパスワードを保護または復号するときに使用する初期鍵の場所を提供します。649 ページの『パスワードの保護』を参照してください。

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。

パスワードの保護

パスワードおよび keystore.conf ファイルに含まれるその他の機密情報を保護する必要があります。 詳しくは、 **runamscred** を参照してください。

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 configuration
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

関連タスク
663 ページの『構成ファイルの AMS パスワード保護のセットアップ』

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 での 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す。 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。

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。

V 9.4.0 IBM MQ 9.4.0 以降、この JAR ファイルは `bcprov-jdk18on.jar` という名前になりました。

「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。

V 9.4.0 IBM MQ 9.4.0 以降、この JAR ファイルは `bcpkix-jdk18on.jar` という名前になりました。

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。

V 9.4.0 IBM MQ 9.4.0 以降、この JAR ファイルは `bcutil-jdk18on.jar` という名前になりました。

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。

V 9.4.0 IBM MQ 9.4.0 以降、これらのファイルは以下のとおりです。

mq_install_dir/java/lib/bcutil-jdk18on.jar
mq_install_dir/java/lib/bcpkix-jdk18on.jar
mq_install_dir/java/lib/bcprov-jdk18on.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) のインターセプトと AMS

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、654 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、644 ページの『AMS で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、651 ページの『AMS MCA インターセプトの例』を参照してください。

重要 : 許可されたクライアントのみが接続してこの機能を使用できるようにするために、SSL と SSLPEER または CHLAUTH TYPE (SSLPEERMAP) を使用するなどして、選択したチャネルのクライアント認証と暗号化を完了する必要があります。

IBM 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
```

AMS 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 MQ には AMS インターセプターが組み込まれており、 MQ クライアントおよびサーバーのランタイム環境でインターセプターを動的に使用可能にします。

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。
- 以下の手順は、クライアントで AMS インターセプトが非アクティブ化されていない限り、 IBM MQ では予期したとおりに機能しません。

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/mqm
KEYSTORELOC=/home/userID/var/mqm
mkdir -p $KEYSTORELOC
chmod -R 777 $KEYSTORELOC
chown -R mqm:mqm $KEYSTORELOC
export PATH=$PATH:$INSTLOC/gskit8/bin
echo "PATH = $PATH"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTLOC/gskit8/lib64

runmqakm -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
runmqakm -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
runmqakm -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd \
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
runmqakm -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd \
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. 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。

   企業で使用しているプラットフォームに応じて、「クイック・スタート・ガイド」に記載されている証明書の共有方法を使用することが重要です。

   Windows
   タスク 5 共有証明書

   AIX and Linux
   タスク 5 共有証明書

   Java クライアント
   タスク 5 共有証明書

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
```

重要：

a. 鍵ストアは、キュー・マネージャーが存在するシステム上になければなりません。

b. MCA の介入を有効にするには、 `cms.certificate` に特定のチャネルを指定する必要があります。その後、キュー・マネージャーは、そのチャネルを介してキューに接続するアプリケーションに対して、ポリシー・セットを使用して AMS 操作を実行します。

4. キュー・マネージャー AMSQMGR1 を作成して開始します。
5. QMGR 制御下で使用可能なポート番号を使用して TCP リスナーを定義します。
   以下に例を示します。

```
DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
```

6. リスナーを開始し、正しく開始されたことを確認します。
   以下に例を示します。

```
START  LISTENER(MY.LISTENER)
DISPLAY LSSTATUS(MY.LISTENER) PORT
```

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て、キュー・マネージャーが MQS_KEYSTORE_CONF 環境変数を使用できるようにします。
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. *MQSERVER* 環境変数を設定します。

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. 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
```

13. IBM MQ 9.4 のインストール環境からキューを参照します。

```
/opt/mq93/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。

14. 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
```

15. IBM MQ 9.4 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
```

関連概念

645 ページの『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。

関連資料

603 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

IBM MQ クライアントを使用して旧バージョンの製品からキュー・マネージャーに接続していて、 2085 (MQRC_UNKNOWN_OBJECT_NAME) エラーが報告される場合は、 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 MQ Advanced Message Security (AMS) は IBM MQ クライアントで自動的に使用可能になるため、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。

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** を設定する。
  - 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。
  - **DisableClientAMS** プロパティーを使用して、mqclient.ini ファイルの Security スタンザの下に指定します。
- C クライアントの場合は、環境変数 **MQS_DISABLE_ALL_INTERCEPT** を設定します。

注 : C クライアントに対して **AMQ_DISABLE_CLIENT_AMS** 環境変数を使用することはできません。 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 runtime environment (JRE) 以外の Java runtime environment (JRE) を使用している場合
  - IBM MQ 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 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。

  ```
  JM 3.0 java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/
  java/lib/com.ibm.mq.jakarta.client.jar  my.java.applicationClass
  ```

```
JMS 2.0 java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/
java/lib/com.ibm.mq.allclient.jar  my.java.applicationClass
```

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 を使用しており、クライアントで AMS を無効にする必要がある場合は、この環境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 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 クライアントにのみ使用できます。 Java クライアントの場合は、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

651 ページの『メッセージ・チャネル・エージェント (MCA) のインターセプトと AMS』
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

IBM MQ MQI client 構成ファイル、 `mqclient.ini`

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、47 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

675 ページの『AMS での保護品質』
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

656 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。

658 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。
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。 指定可能な値: • 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したい場合は、以下のプロパティーを使用して $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
   ```

   以下の行を使用して $JAVA_HOME/lib/security/java.security ファイルを編集することにより、OCSP チェックを有効にします。

   ```
   ocsp.enable=true
   ```

2. AIA がセットアップされている場合は、以下の行で $JAVA_HOME/lib/security/java.security ファイルを編集して OCSP チェックを有効にします。

   ```
   ocsp.enable=true
   ```

次のタスク

Java セキュリティー・マネージャーを使用している場合は、構成を完了させるために、以下の Java 許可を lib/security/java.policy に追加してください。

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

keystore.conf を使用する

手順

構成ファイルに以下の属性を追加します。

```
ocsp.enable=true
```

重要 : この属性を構成ファイルに設定すると、java.security 設定がオーバーライドされます。

次のタスク

構成を完了するには、以下の Java 許可を 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。

オプション	説明
crl.ldap.encrypted=*no/yes*	clr.ldap.pass を暗号化するかどうか。 詳しくは、AMS 構成ファイルでのパスワードの保護を参照してください。
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 を取り出すための HTTP プロキシー・サーバー。
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	IBM Global Security Kit (GSKit)によって受け入れられる HTTP サーバーから取得可能な CRL の最大サイズ (バイト単位)。
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。 指定可能な値: • 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。

ヘッダー	説明
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。 システム SSL 証明書の検証に関する詳細なドキュメントは、z/OS 暗号化サービスシステム Secure Sockets Layer プログラミングマニュアル。

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 *105. 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 *106. 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	送信側

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 *107. 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 MQ 証明書の検証では LDAP サーバーに接続可能であることが必要であるが、CRL を定義する必要がないセキュリティ設定を使用します。

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。

構成ファイルの AMS パスワード保護のセットアップ

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への読み取りおよび書き込みを許可されていることを確認する必要があります。 このトピック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。 さらに、この手順はセキュア・システム上で実行する必要があります。

構成ファイルを読み取る AMS クライアントのタイプに対して正しい **runamscred** バリアントを使用していることを確認してください。 AMS クライアントのタイプに応じて、以下のようにしてください。

- Java クライアントの場合は、Java の **runamscred** コマンドを使用してください。このコマンドは、<IBM MQ installation root>/java/bin にあります。
- MQI クライアントの場合は、<IBM MQ installation root>/bin にある MQI **runmqascred** コマンドを使用する必要があります。

手順

1. keystore.conf ファイルを編集して、保護を必要とするパスワードを含め、必要なすべての情報を組み込みます。

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

2. keystore.conf ファイルを保護しているユーザーがアクセスできるファイル内のパスワードを暗号化するには、暗号鍵を配置します。

   この鍵は、AMS クライアントが後で使用する鍵と同じでなければなりません。

```
ThisIsAnExampleEncryptionKey
```

3. **runamscred** コマンドを実行して、暗号鍵ファイルを提供する keystore.conf ファイルを保護します。

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. keystore.conf ファイルが保護され、暗号化されたパスワードが含まれていることを確認します。

例

以下の例は、保護された keystore.conf ファイルの外観を示しています。

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = yes
jceks.keystore_pass =
<AMS>1!62K/a4RinT+bks4RjFWx4A==!Vhi/RjIN2FH5qStUJ/0hsgKyn2IdMuhanemRRDrJq
HM=
jceks.key_pass =
<AMS>1!qmnxY++rsOUtZfDSgwcR1g==!VmWVREdVkNp1xYJstvuW64ph5vxxf7SPoqtsXxYh2
Tk=
jceks.provider = IBMJCE
```

関連情報

runamscred: AMS キーワードの保護

z/OS Using certificates with AMS on z/OS

About this task

Advanced Message Security implements three levels of protection: integrity, confidentiality, and privacy.

With an integrity policy, messages are signed using the private key of the originator (the application doing the MQPUT). Integrity provides detection of message modification, but the message text itself is not encrypted.

With a confidentiality policy, the message is encrypted when it is put to the queue. The message is encrypted using a symmetric key and an algorithm specified in the relevant Advanced Message Security policy. The symmetric key itself is encrypted with the public key of each recipient (the application doing the MQGET). Public keys are associated with certificates stored in key rings.

With a privacy policy, messages are both signed and encrypted.

When a message that is protected with privacy is dequeued by a recipient application doing an MQGET, the message must be decrypted. Because it was encrypted using the recipient's public key, it must be decrypted using the recipient's private key found in a key ring.

z/OS *Use of SAF key rings with AMS on z/OS*

Advanced Message Security (AMS) makes use of z/OS SAF key ring services to define and manage the certificates needed for signing and encryption. Security products that are functionally equivalent to RACF may be used instead of RACF if they provide the same level of support.

Efficient use of key rings can reduce the administration needed to manage the certificates.

After a certificate is generated (or imported), it must be connected to a key ring to become accessible. The same certificate can be connected to more than one key ring.

Advanced Message Security uses two sets of key rings. One set consists of key rings owned by the individual user IDs that originate or receive messages. Each key ring contains the private key associated with the certificate of the owning user ID. The private key of each certificate is used to sign messages for integrity protected or privacy protected queues. It is also used to decrypt messages from privacy protected or confidentiality protected queues when receiving messages.

The other set is a single key ring owned by the AMS address space user. It contains the chain of signing CA certificates necessary to validate the certificates of the message originator and recipients.

When privacy or confidentiality protection is used, the key ring owned by the AMS address space user also contains the certificates of the message recipients. The public keys in these certificates are used to

encrypt the symmetric key that was used to encrypt the message data when the message was put to the protected queue. When these messages are retrieved, the private key of relevant recipients is used to decrypt the symmetric key which is then used to decrypt the message data.

Advanced Message Security uses a key ring name of **drq.ams.keyring** when searching for certificates and private keys. This is the case for both the user and the AMS address space key rings.

For an illustration and further explanation of certificates and key ring, and their role in data protection, refer to Summary of the certificate-related operations.

The private key used for signing can have any label but must be connected as the default certificate. The private key or keys used for decryption can have any label, and must be connected to the key ring, but are no longer required to be connected as the default certificate.

Digital certificates and key rings are managed in RACF primarily by using the RACDCERT command.

For more information about certificates, labels, and the RACDCERT command, see the *z/OS: Security Server RACF Command Language Reference* and the *z/OS: Security Server RACF Security Administrator's Guide*.

z/OS *Replacing certificates*

When a certificate is renewed or replaced (for example, when the existing certificate is approaching its expiry date), it is not always possible to remove the protection from existing messages that are already on queues protected by confidentiality or privacy policies.

This can occur when the certificate was:

- Renewed with the same private key, and the reissued certificate has replaced the original certificate
- Re-keyed with a new private key and the RACDCERT ROLLOVER command has deleted the original private key

Messages will be decrypted, provided the necessary certificate is connected to the keyring of the user; it is no longer required to be connected as the default. This allows messages already on the queue, when the new certificate is connected, to be successfully decrypted.

The following example shows how a new certificate can be generated based on the existing certificate:

- A new certificate is created based on the existing certificate, with new public/private key pair.
- The new certificate is signed by the issuing authority.
- The public key of the old certificate is removed from the keyring of the AMS address space, and the public key of the new certificate is added.
- The new certificate and private key is added to the keyring of the user, in addition to the old certificate.

```
RACDCERT ID(user1) REKEY(LABEL('user1'))          -
         WITHLABEL('user1new')

RACDCERT GENREQ(LABEL('user1new')) ID(user1)      -
         DSN(output_data_set_name)

RACDCERT GENCERT(output_data_set_name) ID(user1)  -
         SIGNWITH(CERTAUTH LABEL('AMSCA'))

RACDCERT ID(user1) ALTER (LABEL('user1new'))      -
         TRUST

RACDCERT ID(WMQAMSD) REMOVE(ID(user1)             -
         LABEL('user1')                           -
         RING(drq.ams.keyring)  )

RACDCERT ID(WMQAMSD) CONNECT(ID(user1)            -
         LABEL('user1new') USAGE(SITE)            -
         RING(drq.ams.keyring)  )

RACDCERT ID(user1) CONNECT(ID(user1)              -
         LABEL('user1new') USAGE(PERSONAL)        -
         RING(drq.ams.keyring) DEFAULT )
```

For more information about certificates, labels, and the RACDCERT command, see the *z/OS: Security Server RACF Command Language Reference* and the *z/OS: Security Server RACF Security Administrator's Guide*.

z/OS *Authorizing access to the RACDCERT command for AMS on z/OS*

Authorization to use the RACDCERT command is a post-installation task that should have been completed by your z/OS system programmer. This task involves granting relevant permissions to the Advanced Message Security security administrator.

As a summary, these commands are needed to allow access to the RACF RACDCERT command:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

In this example, *admin* specifies the user ID of your security administrator, or any user you want to use the RACDCERT command.

z/OS *Creating the certificates and key rings for AMS users on z/OS*

This section documents the steps required to create the certificates and key rings necessary for z/OS users of Advanced Message Security (AMS), using a RACF Certificate Authority (CA).

Resolving problems with certificates when using Advanced Message Security on z/OS

If you are having problems with certificates and missing entries in key stores you can enable a IBM Global Security Kit (GSKit) trace.

In the file referenced by the ENVARS DD in the AMS started task procedure, add:

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

See Environment variables for more information.

For every access to the keystore, data is written to the trace file specified in `GSK_TRACE_FILE`.

To format the trace file use the command:

```
gsktrace inputtrace file > output_file
```

Scenario

A scenario of a sending application and a receiving application is used to explain the required steps.

In the examples that follow, `user1` is the originator of a message and `user2` is the recipient. The user ID of the Advanced Message Security address space is `WMQAMSD`.

All of the commands in the examples shown here are issued from ISPF option 6 by the administrative user ID `admin`.

z/OS *Defining a local Certificate Authority certificate for AMS on z/OS*

If you are using RACF as your CA, you must create a certificate authority certificate, if you have not already done so. The command shown here creates a certificate authority (or signer) certificate. This example creates a certificate called AMSCA to be used when creating subsequent certificates that reflect the identity of Advanced Message Security users and applications.

This command may be modified, specifically SUBJECTSDN, to reflect the naming structure and conventions used at your installation:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

Note: Certificates signed with this local certificate authority certificate show an issuer of CN=AMSCA,O=ibm,C=us when listed with the RACDCERT LIST command.

z/OS *Creating a digital certificate with a private key for AMS on z/OS*

A digital certificate with a private key must be generated for each Advanced Message Security user. In the example shown here, RACDCERT commands are used to generate certificates for user1 and user2, which are signed with the local CA certificate identified by the label AMSCA.

```
RACDCERT ID(user1) GENCERT SUBJECTSDN(CN('user1') O('ibm') C('us'))
WITHLABEL('user1') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user2) GENCERT SUBJECTSDN(CN('user2') O('ibm') C('us'))
WITHLABEL('user2') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user1) ALTER (LABEL('user1')) TRUST
RACDCERT ID(user2) ALTER (LABEL('user2')) TRUST
```

The RACDCERT ALTER command is required to add the TRUST attribute to the certificate. When a certificate is first created using this procedure, it has a different valid date range than the signing certificate. As a result, RACF marks it as NOTRUST, which means that the certificate is not to be used. Use the RACDCERT ALTER command to set the TRUST attribute.

The KEYUSAGE attributes HANDSHAKE, DATAENCRYPT and DOCSIGN must be specified for certificates used by Advanced Message Security.

Table 108. RACDCERT KEYUSAGE values and indicators

KEYUSAGE Value	**Indicators Set**
HANDSHAKE	digitalSignature and keyEncipherment
DATAENCRYPT	dataEncipherment
DOCSIGN	nonRepudiation
CERTSIGN	keyCertSign and cRLSign

z/OS *Creating the RACF key rings for AMS on z/OS*

The commands shown here create a key ring for RACF-defined user IDs user1, user2, and the Advanced Message Security address space task user WMQAMSD. The key ring name is fixed by Advanced Message Security and must be coded as shown, without quotes. The name is case-sensitive.

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS *Connecting the certificates to the key rings for AMS on z/OS*

Connect the user and CA certificates to the key rings:

```
RACDCERT ID(WMQAMSD) CONNECT(CERTAUTH LABEL('AMSCA')
RING(drq.ams.keyring))
RACDCERT ID(user1) CONNECT(ID(user1) LABEL('user1')
```

```
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(user2) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(WMQAMSD) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) USAGE(SITE))
```

Any certificates containing the private key or keys used for decryption must be connected to the key ring of the user, however they are no longer required to be connected as the default certificate.

The RACDCERT USAGE(SITE) attribute prevents the private key from being accessible in the key ring, while the RACDCERT USAGE(PERSONAL) attribute allows the private key to be used, if it exists. User2's certificate must be connected to the Advanced Message Security address space key ring because its public key is needed to encrypt messages as they are put to the queue. USAGE(SITE) limits exposure of user2's private key.

The CERTAUTH certificate with label AMSCA must be connected to the Advanced Message Security address space key ring because it was used to sign the certificate of user1, who is the message originator. It is used to validate user1's signing certificate.

z/OS *Key ring verification for AMS on z/OS*

The key ring should appear as shown here, after all commands have been entered:

```
RACDCERT ID(user1) LISTRING(drq.ams.keyring)
Digital ring information for user USER1:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user1                            ID(USER1)    PERSONAL YES

RACDCERT ID(user2) LISTRING(drq.ams.keyring)
Digital ring information for user USER2:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user2                            ID(USER2)    PERSONAL YES

RACDCERT ID(WMQAMSD) LISTRING(drq.ams.keyring)
Digital ring information for user WMQAMSD:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
AMSCA                            CERTAUTH     CERTAUTH NO
user2                            ID(USER2)    SITE     NO
```

Listing the individual certificates also shows the ring association.

```
RACDCERT ID(user2) LIST(label('user2'))
Digital certificate information for user USER2:

***
Label: user2
Certificate ID: 2QfH8Pny9/LzpKKFmfFA
Status: TRUST
Start Date: 2010/05/03 22:59:53
End Date:  2011/05/04 22:59:52
Serial Number:>15<:
Issuer's Name:>OU=AMSCA.O=ibm.C=us<:
Subject's Name:>CN=user2.O=ibm.C=us<:
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: USER2
Ring:>drq.ams.keyring<:
Ring Owner: WMQAMSD
Ring:>drq.ams.keyring<:
```

To improve performance, the contents of the drq.ams.keyring associated with the AMS address space is cached for the life of the address space. Changes to that key ring do not become effective automatically. The administrator can refresh the cache by either:

- Stopping and restarting the queue manager.
- Using the z/OS MODIFY command:

```
F qmgrAMSM,REFRESH KEYRING
```

Related tasks

Operating Advanced Message Security

z/OS *Summary of the certificate-related operations for AMS on z/OS*

Figure 35 on page 669 illustrates the relationships between sending and receiving applications and relevant certificates. The scenario illustrated involves remote queuing between two z/OS queue managers using a data-protection policy of privacy. In Figure 35 on page 669, "AMS" indicates " Advanced Message Security".

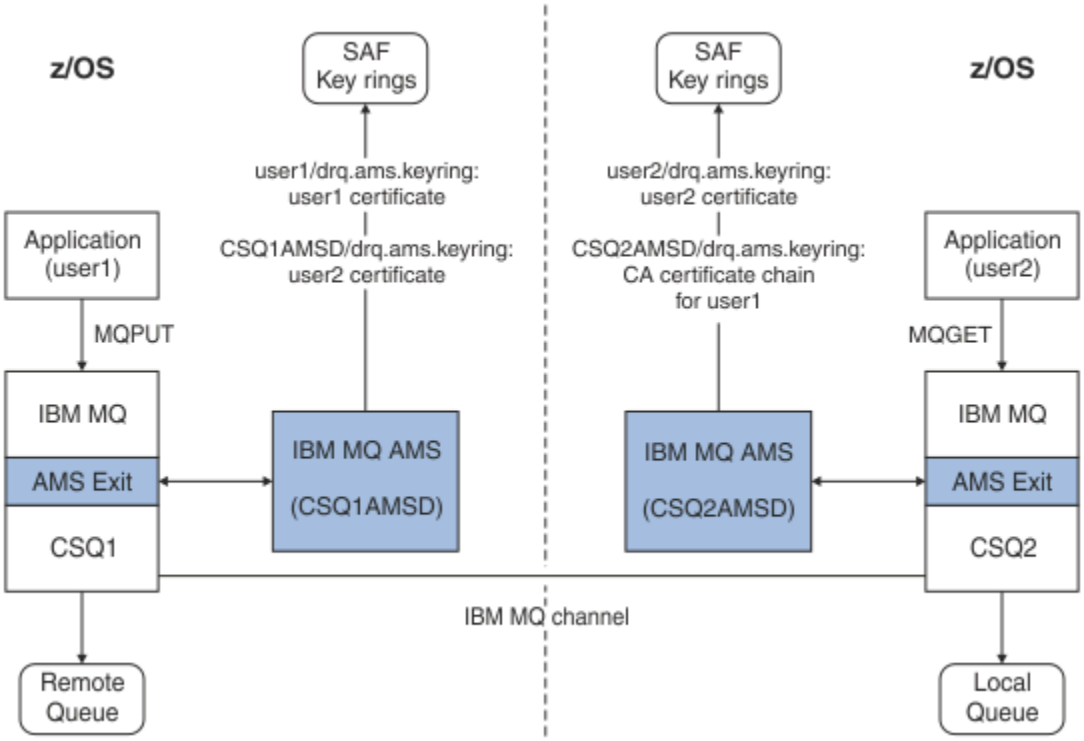


Figure 35. Application and certificate relationships

In this diagram, an application running as 'user1' puts a message to a remote queue managed by queue manager CSQ1, intended to be retrieved by an application running as 'user2' from a local queue managed by queue manager CSQ2. The diagram assumes an Advanced Message Security policy of privacy, which means the message is both signed and encrypted.

Advanced Message Security intercepts the message when a put occurs and uses user2's certificate (stored in the AMS address space user's key ring) to encrypt a symmetric key used to encrypt the message data.

Note that user2's certificate is connected to the AMS address space user key ring with option USAGE(SITE). This means the AMS address space user can access the certificate and public key, but not the private key.

On the receiving end, Advanced Message Security intercepts the get issued by user2, and uses user2's certificate to decrypt the symmetric key so that it can decrypt the message data. It then validates user1's signature using the CA certificate chain of user1's certificate stored in the AMS address space user's key ring.

Given this scenario, but with a data-protection policy of integrity, certificates for user2 would not be required.

To use Advanced Message Security to enqueue messages on IBM MQ-protected queues having a message protection policy of privacy or integrity, Advanced Message Security must have access to these data items:

- The X.509 V2 or V3 certificate and private key for the user enqueuing the message.
- The chain of certificates used to sign the digital certificates of all message signers.
- If the data protection policy is privacy, the X.509 V2 or V3 certificate of the intended recipients. The intended recipients are listed in the Advanced Message Security policy associated with the queue.

For processes and applications that run on z/OS, Advanced Message Security must have certificates in two places:

- In a SAF-managed key ring associated with the RACF identity of the sending application (the application that enqueues the protected message) or receiving application (if using privacy).

  The certificate that Advanced Message Security locates is the default certificate, and must include the private key. Advanced Message Security assumes the z/OS user identity of the sending application. That is, it acts as a surrogate, so it can access the user's private key.
- In a SAF-managed key ring associated with the AMS address space user.

  When sending messages protected with privacy, this key ring contains the public key certificates of the message recipients. When receiving messages, it contains the chain of Certificate Authority certificates needed to validate the message sender's signature.

The earlier examples shown have used RACF as the local CA. However, you may use another PKI provider (Certificate Authority) at your installation. If you intend to use another PKI product, remember that the private key and the certificate must be imported into a key ring associated with the z/OS RACF user IDs that originate IBM MQ messages protected by Advanced Message Security.

You can use the RACF RACDCERT command as the mechanism to generate certificate requests, which can be exported and sent to the PKI provider of your choice to be issued.

Here is a summary of the certificate-related steps:

1. Request the creation of a CA certificate, one in which RACF is the local CA. Omit this step if you are using another PKI provider.
2. Generate user certificates signed by the CA.
3. Create the key rings for the users and the Advanced Message Security AMS address space ID.
4. Connect the user certificate to the user key ring with the default attribute.
5. Connect the recipients certificates to the Advanced Message Security AMS address space user key ring using the usage(site) attribute (This step is necessary only for user certificates that will ultimately be the recipients of privacy-protected messages).
6. Connect the CA certificate chains for message senders to the Advanced Message Security AMS address space user key ring. (This step is necessary only for AMS tasks that will be verifying sender signatures.)

z/OS *Configuring a non-z/OS resident PKI for AMS*

Advanced Message Security for z/OS, uses X.509 V3 digital certificates in the protection-processing of messages placed on or received from IBM MQ queues. Advanced Message Security itself does not create or manage the life cycle of these certificates; that function is provided by a public key infrastructure (PKI). The examples in this publication that illustrate the use of certificates use z/OS Security Server RACF to fill certificate requests.

Whether or not a z/OS or non-z/OS resident PKI is used, AMS for z/OS uses only key rings that are managed by RACF or its equivalent. These key rings are based on Security Authorization Facility (SAF) and are the repository used by AMS for z/OS to retrieve certificates for originators and recipients of messages placed on or received from IBM MQ queues.

For messages that are originated from z/OS, which are protected by either integrity or encryption policy, the certificate and private key of the originating user ID must be stored in an SAF-managed key ring that is associated with the z/OS user ID of the message originator.

RACF includes the capability for importing certificates and private keys into RACF-managed key rings. See the z/OS Security Server RACF publications for the details and examples of how to load certificates to RACF managed key rings.

If your installation is using one of the supported PKI products, refer to the publications that accompany the product for information on how to deploy it.

Advanced Message Security 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

675 ページの『AMS での保護品質』
Advanced Message Security データ保護ポリシーは、保護品質 (QOP) を意味します。

674 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。

- Deprecated RC2
- Deprecated DES
- Deprecated 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いて、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 コンポーネント名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。 文字セットを使用して DN を作成するには、まず、 UTF-8 コーディングがオンになっている AIX and Linux 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 Linux または AIX プラットフォームからポリシーを作成するか、 IBM MQ に対して AMS プラグインを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

673 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 コンポーネント名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。 文字セットを使用して DN を作成するには、まず、 AIX を使用して UTF-8 コーディングで作成された DN、または UTF-8 コーディングがオンになっている Linux で作成された DN を使用して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 Linux または AIX プラットフォームからポリシーを作成するか、 IBM MQ に対して Advanced Message Security プラグインを使用する必要があります。

関連概念

672 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 *109. 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。

表 109. AMS でのセキュリティー・ポリシー属性 (続き)	
属性	**説明**
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。

- Deprecated RC2
- Deprecated DES
- Deprecated 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。

- Deprecated MD5
- Deprecated 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。

AMS での保護品質

Advanced Message Security データ保護ポリシーは、保護品質 (QOP) を意味します。

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、600 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。

AMS でのセキュリティー・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。

- ALW AIX, Linux, and 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。
  - Linux AIX AIX and Linux 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

677 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。

677 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。

678 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。

680 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 Multiplatforms では、 **setmqaut** コマンドを使用して、必要な +connect、 +inq 、および +chg 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、133 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 Multiplatforms では、 **setmqaut** コマンドを使用して、必要な +connect、 +inq 、および +chg 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、133 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 Multiplatforms では、 **setmqaut** コマンドを使用して、必要な +connect、 +inq 、および +chg 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、133 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 110. ***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 Multiplatforms では、出力をファイルにリダイレクトします。以下に例を示します。

```
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。

- Linux AIX On AIX and Linux:
  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。
- Windows Windows で、policies.bat 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 Multiplatforms では、 **setmqaut** コマンドを使用して、必要な +connect、 +inq 、および +chg 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、133 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。 677 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

Multi ストリーミング・キューと AMS

複製した Advanced Message Security (AMS) の保護メッセージをストリーミングすることができます。

キューに入れられるメッセージが署名および/または暗号化されるようにする AMS ポリシーがそのキューに定義されている場合、そのキューの **STREAMQ** 属性を構成して、保護されている各メッセージのコピーを 2 次キューに入れることもできます。 複製されストリーミングされたメッセージは、元のキューに構成されているポリシーと同じものを使用して、署名および/または暗号化されます。

以下の例では、 QUEUE1 と QUEUE2 の 2 つのキューを構成します。 QUEUE1 には、ストリーミングされたメッセージを QUEUE2 に入れる **STREAMQ** 属性が構成されています。

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

AMS の保護されたメッセージは、 証明書 CN=bob,O=IBM,C=GB を使用してユーザーによって QUEUE1 に入れられます。

証明書 CN=alice,O=IBM,C=GB を保有するアプリケーションは、QUEUE1 からメッセージを取り込みます。 証明書 CN=fred,O=IBM,C=GB を保有する別のアプリケーションは、QUEUE2 からメッセージを取り込みます。

QUEUE1 には、以下の AMS プライバシー・ポリシー が適用されています。

```
SET POLICY(QUEUE1) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=alice,O=IBM,C=GB') RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

この QUEUE1 のポリシーに暗号化アルゴリズムが構成されている場合、ポリシーにリストされている受信者には、QUEUE1 から元のメッセージを取り込む受信者と、QUEUE2 から複製されたメッセージを取り込む受信者の両方が含まれている必要があります。

アプリケーションが QUEUE2 からメッセージを取り込もうとすると、アプリケーションは整合性チェックを実行し、QUEEUE2 に設定されているポリシーに基づいて、メッセージの整合性チェックおよび/または暗号化解除を行います。 アプリケーションがストリーミングされたメッセージを QUEUE2 から取り込むことが望ましい場合、メッセージの整合性チェックと正確な暗号化解除を許可するための適切なポリシーを QUEUE2 に設定する必要があります。

具体的には、署名アルゴリズム、署名者、および暗号化アルゴリズムは、QUEUE1 に適用されているポリシーと同一である必要があります。 QUEUE2 のポリシー受信者には、QUEUE2 からメッセージを取り込む受信者の ID が含まれている必要があります。

注 : QUEUE2 に適用されているポリシーには、QUEUE1 に設定されているポリシーに指定されているすべての受信者を含める必要はありません。

例えば、証明書識別名 `CN=fred,O=IBM,C=GB` を保有するアプリケーションが AMS で保護されたメッセージを QUEUE2 から読み取ることができるようにするために、以下のポリシーを QUEUE2 に設定することが考えられます。

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

関連概念

ストリーミング・キュー

AMS での 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。

AMS での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。

- 684 ページの『AMSM アドレス・スペース』
- 684 ページの『CSQ0UTIL』
- 685 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。 z/OS これは、MXQUEUE、GMQQUEUE、または GMXQUEUE でも構いません。 詳しくは、203 ページの『Profiles for queue security』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 IBM i での AMS の証明書および鍵ストア構成ファイルのセットアップ

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 AMS での IBM i のポリシーの作成

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 *AMS での IBM i のポリシーのテスト*

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)

AMS のコマンドと構成イベント

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。

- 677 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
- 677 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
- 678 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
- 680 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

AMS のイベント・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

AMS のコマンド・イベント・メッセージ形式

コマンド・イベント・メッセージは、MQCFH 構造と、それに続く PCF パラメーターで構成されます。

選択された MQCFH 値は、以下のとおりです。

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

注 : ParameterCount 値は、MQCFGR タイプ (グループ) のパラメーターが常に 2 つあるため、2 になっています。 各グループは、適切なパラメーターで構成されます。 イベント・データは、CommandContext と CommandData の 2 つの グループから成ります。

CommandContext の内容は、以下のとおりです。

EventUserID

説明: コマンドを発行したユーザー ID、またはイベントを生成した呼び出し。 (これは、コマンドまたは呼び出しを発行する権限の検査に使用するものと同じユーザー ID です。キューから受け取ったコマンドの場合、これはコマンド・メッセージの MD からのユーザー ID (UserIdentifier) でもあります。)

ID: MQCACF_EVENT_USER_ID

データ型: MQCFST

最大長: MQ_USER_ID_LENGTH

戻り: 常時。

EventOrigin

説明: イベントを引き起こしたアクションの発信元。

ID: MQIACF_EVENT_ORIGIN

データ型: MQCFIN

値:

MQEVO_CONSOLE
コンソール・コマンド - コマンド・ライン

MQEVO_MSG
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。)

ID: MQCACF_EVENT_Q_MGR

データ型: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 常時。

EventAccountingToken

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアカウンティング・トークン (AccountingToken)。

ID: MQBACF_EVENT_ACCOUNTING_TOKEN

データ型: MQCFBS

最大長: MQ_ACCOUNTING_TOKEN_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

EventIdentityData

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーション識別データ (ApplIdentityData)。

ID: MQCACF_EVENT_APPL_IDENTITY

データ型: MQCFST

最大長: MQ_APPL_IDENTITY_DATA_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

EventApplType

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーションのタイプ (PutApplType)。

ID: MQIACF_EVENT_APPL_TYPE

データ型: MQCFIN

戻り: EventOrigin が MQEVO_MSG の場合のみ。

EventApplName

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーションの名前 (PutApplName)。

ID: MQCACF_EVENT_APPL_NAME

データ型: MQCFST

最大長: MQ_APPL_NAME_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

EventApplOrigin

説明: メッセージ (MQEVO_MSG) として受け取ったコマンドの場合、コマンド・メッセージの MD からのアプリケーションの発信元データ (ApplOriginData)。

ID: MQCACF_EVENT_APPL_ORIGIN

データ型: MQCFST

最大長: MQ_APPL_ORIGIN_DATA_LENGTH

戻り: EventOrigin が MQEVO_MSG の場合のみ。

コマンド

説明: コマンド・コード。

ID: MQIACF_COMMAND

データ型: MQCFIN

値: **MQCMD_INQUIRE_PROT_POLICY 数値 205**
MQCMD_CREATE_PROT_POLICY 数値 206
MQCMD_DELETE_PROT_POLICY 数値 207
MQCMD_CHANGE_PROT_POLICY 数値 208
これらは 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。

AMS の構成イベント・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。)

ID: **MQCACF_EVENT_USER_ID**

データ型: MQCFST

最大長: MQ_USER_ID_LENGTH

戻り: 常時。

SecurityId

説明: コマンド・サーバー・メッセージの場合は MQMD.AccountingToken の値、またはローカル・コマンドの場合は Windows SID。

ID: **MQBACF_EVENT_SECURITY_ID**

データ型: MQCBS.

最大長: MQ_SECURITY_ID_LENGTH

戻り: 常時。

EventOrigin

説明: イベントを引き起こしたアクションの発信元。

ID: **MQIACF_EVENT_ORIGIN**

データ型: MQCFIN

値:

MQEVO_CONSOLE
コンソール・コマンド - コマンド・ライン

MQEVO_MSG
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。)

ID: **MQCACF_EVENT_Q_MGR**

データ型: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 常時。

ObjectType

説明: オブジェクト・タイプ

ID: **MQIACF_OBJECT_TYPE**

データ型: MQCFIN

値:

MQOT_PROT_POLICY
Advanced Message Security 保護ポリシー。 **1019** - 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。

ID: **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。

ID: **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。

ID: **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。

ID: **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。

ID: **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。

ID: **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。

ID: **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。

IBM 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
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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。これらの名称はすべて架空のものであり、 名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。 お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。 このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。 従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性が あることをほのめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが IBM MQ のサービスを使用できるように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 氏の米国およびその他の国における登録商標です。

この製品には、 Eclipse Project (https://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: