9.4

Sécurisation d' IBM MQ

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 717.

Cette édition s'applique à la version 9 édition 4 d' IBM® MQ et à toutes les éditions et modifications ultérieures, sauf indication contraire dans les nouvelles éditions.

Table des matières

Sécurisation de IBM MQ

La sécurité est une considération importante pour les développeurs d'applications IBM MQ et pour les administrateurs système IBM MQ . En tant que minimum absolu, vous devez vous assurer que tous les matériels et logiciels à l'intérieur de la zone sécurisée et sur les postes de travail de l'opérateur sont dans leur cycle de vie de support, sont à jour avec les mises à jour logicielles obligatoires et que les mises à jour de sécurité sont appliquées rapidement.

Référence associée

Gestion des vulnérabilités de sécurité IBM

IBM Z et LinuxOne Security Portal

Présentation de la sécurité

Cette collection de rubriques présente les concepts de sécurité d' IBM MQ .

Les concepts et les mécanismes de sécurité, tels qu'ils s'appliquent à n'importe quel système informatique, sont présentés en premier, suivis d'une discussion de ces mécanismes de sécurité à mesure qu'ils sont implémentés dans IBM MQ.

Les aspects communément acceptés de la sécurité sont les suivants:

- «Identification et authentification», à la page 7
- «Autorisation», à la page 9
- «Audit», à la page 9
- «Confidentialité», à la page 10
- «Intégrité des données», à la page 10

Les *mécanismes de sécurité* sont des outils et des techniques techniques utilisés pour implémenter les services de sécurité. Un mécanisme peut fonctionner seul ou avec d'autres pour fournir un service particulier. Voici des exemples de mécanismes de sécurité communs:

- «Cryptographie», à la page 11
- «Historiques des messages et signatures numériques», à la page 13
- «Certificats numériques», à la page 13
- «Public key infrastructure (PKI)», à la page 18

Lorsque vous planifiez une implémentation IBM MQ , tenez compte des mécanismes de sécurité dont vous avez besoin pour implémenter les aspects de la sécurité qui sont importants pour vous. Pour plus d'informations sur les éléments à prendre en compte après avoir lu ces rubriques, voir «Planification de la sécurité», à la page 91.

Identification et authentification

L' *identification* est la possibilité d'identifier de manière unique un utilisateur d'un système ou d'une application qui s'exécute dans le système. L' *authentification* est la possibilité de prouver qu'un utilisateur ou une application est réellement qui est cette personne ou ce que cette application prétend être.

Par exemple, imaginez un utilisateur qui se connecte à un système en entrant un ID utilisateur et un mot de passe. Le système utilise l'ID utilisateur pour identifier l'utilisateur. Le système authentifie l'utilisateur au moment de la connexion en vérifiant que le mot de passe fourni est correct.

Identification et authentification dans IBM MQ

Lorsqu'une application se connecte à IBM MQ, une identité utilisateur est toujours associée à la connexion. L'identité de l'utilisateur est initialement l'ID utilisateur du système d'exploitation associé au

processus d'application. Cette identité est souvent suffisante pour les applications liées localement qui sont hébergées sur le même système que le gestionnaire de files d'attente. Toutefois, le gestionnaire de files d'attente peut également authentifier et modifier l'identité associée à la connexion de plusieurs manières. L'authentification de l'identité associée à une connexion est importante lorsque des applications client qui ne sont pas nécessairement dignes de confiance se connectent à un gestionnaire de files d'attente sur un réseau.

L'identité associée à une connexion d'application à un gestionnaire de files d'attente IBM MQ peut être établie à l'aide de l'un des mécanismes suivants:

- Lorsqu'une application se connecte à un gestionnaire de files d'attente, elle peut fournir un ID utilisateur et un mot de passe. Le gestionnaire de files d'attente valide les données d'identification en fonction de sa configuration. Par exemple, l'ID utilisateur et le mot de passe peuvent être transmis au système d'exploitation du gestionnaire de files d'attente ou au serveur LDAP pour être authentifiés.
- V 9.4.0 Depuis IBM MQ 9.3.4, une application peut également fournir un jeton d'authentification qu'elle obtient d'un serveur d'authentification externe. Pour plus d'informations sur les jetons d'authentification, voir «Utilisation des jetons d'authentification», à la page 335.
- Un canal client peut être configuré pour utiliser l'authentification mutuelle TLS, s'il est configuré avec un certificat numérique valide. L'authentification TLS peut être combinée à une règle d'authentification de canal (CHLAUTH) pour associer un ID utilisateur approprié à la connexion. Pour plus d'informations, voir «Comment TLS fournit l'identification, l'authentification, la confidentialité et l'intégrité», à la page 20,
- Les règles d'authentification de canal (CHLAUTH) peuvent remplacer l'identité en fonction des informations relatives à la connexion. Par exemple, une règle d'authentification de canal peut définir l'ID utilisateur associé à une connexion en fonction de l'adresse IP du client.
- Le code d'exit personnalisé peut définir une identité en fonction des critères que vous choisissez.

L'identité et l'authentification sont également applicables aux canaux entre deux gestionnaires de files d'attente. Ces canaux sont appelés canaux de message. Lorsqu'un canal de transmission de messages démarre, l'agent MCA à chaque extrémité du canal peut authentifier son partenaire. Cette technique est appelée *authentification mutuelle*. Pour l'agent MCA émetteur, il fournit l'assurance que le partenaire auquel il est sur le point d'envoyer des messages est authentique. De même, l'agent MCA récepteur est assuré qu'il est sur le point de recevoir des messages d'un véritable partenaire.

Lorsqu'une identité a été établie et authentifiée si nécessaire, elle est utilisée par IBM MQ de plusieurs manières:

- Il est important de noter que, par défaut, toutes les vérifications «Autorisation», à la page 9 ultérieures sont effectuées à l'aide de cette identité. Par exemple, si une application tente d'insérer un message dans une file d'attente, le gestionnaire de files d'attente confirme que l'identité associée à l'application dispose de l'autorisation'put'sur l'objet file d'attente.
- En outre, chaque message peut contenir des informations de *contexte de message* . Ces informations sont contenues dans le descripteur de message (MQMD). Le gestionnaire de files d'attente peut générer automatiquement le contexte de message lorsqu'une application place le message dans une file d'attente. L'application peut également fournir le contexte de message si l'ID utilisateur associé à l'application est autorisé à le faire. Ces informations de contexte dans un message fournissent à l'application qui reçoit les informations de message sur l'émetteur du message. Il contient, par exemple, le nom de l'application qui a inséré le message et l'ID utilisateur associé à l'application.

Non-répudiation

L'objectif global du service de non-répudiation est de pouvoir prouver qu'un message particulier est associé à un individu particulier.

Le service de *non-répudiation* peut être considéré comme une extension du service d'identification et d'authentification. En général, la non-répudiation s'applique lorsque les données sont transmises par voie électronique ; par exemple, un ordre donné à un courtier en actions pour acheter ou vendre des actions, ou un ordre donné à une banque pour transférer des fonds d'un compte à un autre.

Le service de non-répudiation peut contenir plusieurs composants, chaque composant fournissant une fonction différente. Si l'expéditeur d'un message refuse de l'envoyer, le service de non-répudiation avec une *preuve de l'origine* peut fournir au destinataire des preuves indéniables que le message a été envoyé par cette personne. Si le destinataire d'un message refuse de le recevoir, le service de non-répudiation avec *preuve de livraison* peut fournir à l'expéditeur des preuves indéniables que le message a été reçu par cette personne.

Dans la pratique, la preuve avec une quasi-certitude de 100%, ou des preuves indéniables, est un objectif difficile. Dans le monde réel, rien n'est totalement sécurisé. La gestion de la sécurité est plus axée sur la gestion des risques à un niveau acceptable pour l'entreprise. Dans un tel environnement, une attente plus réaliste du service de non-répudiation consiste à être en mesure de fournir des preuves qui sont admissibles, et à l'appui de votre cas, devant un tribunal.

La non-répudiation est un service de sécurité pertinent dans un environnement IBM MQ car IBM MQ est un moyen de transmettre des données par voie électronique. Par exemple, vous pouvez exiger des preuves simultanées qu'un message particulier a été envoyé ou reçu par une demande associée à un individu particulier.

IBM MQ avec Advanced Message Security ne fournit pas de service de non-répudiation dans le cadre de sa fonction de base. Toutefois, cette documentation du produit contient des suggestions sur la façon dont vous pouvez fournir votre propre service de non-répudiation dans un environnement IBM MQ en écrivant vos propres programmes d'exit.

Autorisation

L' *autorisation* protège les ressources critiques d'un système en limitant l'accès uniquement aux utilisateurs autorisés et à leurs applications. Il empêche l'utilisation non autorisée d'une ressource ou l'utilisation d'une ressource de manière non autorisée.

Autorisation dans IBM MQ

Vous pouvez utiliser l'autorisation pour limiter les actions que des individus ou des applications spécifiques peuvent effectuer dans votre environnement IBM MQ .

Voici quelques exemples d'autorisation dans un environnement IBM MQ :

- Autoriser uniquement un administrateur autorisé à émettre des commandes pour gérer les ressources IBM MQ .
- Autoriser une application à se connecter à un gestionnaire de files d'attente uniquement si l'ID utilisateur associé à l'application est autorisé à le faire.
- Autoriser une application à ouvrir uniquement les files d'attente nécessaires à sa fonction.
- Autoriser une application à s'abonner uniquement aux rubriques nécessaires à sa fonction.
- Autoriser une application à effectuer uniquement les opérations sur une file d'attente qui sont nécessaires à sa fonction. Par exemple, une application peut n'avoir besoin que de parcourir les messages d'une file d'attente particulière et non d'insérer ou d'extraire des messages.

Pour plus d'informations sur la configuration de l'autorisation, voir «Autorisation de planification», à la page 95 et les sous-rubriques associées.

Audit

L' *audit* est le processus d'enregistrement et de vérification des événements pour détecter si une activité inattendue ou non autorisée a eu lieu ou si une tentative a été effectuée pour effectuer cette activité.

Audit dans IBM MQ

IBM MQ peut émettre des messages d'événement pour enregistrer que l'activité inhabituelle a eu lieu.

Voici quelques exemples d'audit dans un environnement IBM MQ :

- Une application tente d'ouvrir une file d'attente qu'elle n'est pas autorisée à ouvrir. Un message d'événement d'instrumentation est émis. En inspectant le message d'événement, vous découvrez que cette tentative a eu lieu et pouvez décider de l'action nécessaire.
- Une application tente d'ouvrir un canal, mais la tentative échoue car la connexion TLS n'est pas autorisée. Un message d'événement d'instrumentation est émis. En inspectant le message d'événement, vous découvrez que cette tentative a eu lieu et pouvez décider de l'action nécessaire.

Confidentialité

Le service de *confidentialité* protège les informations sensibles contre toute divulgation non autorisée.

Lorsque des données sensibles sont stockées localement, les mécanismes de contrôle d'accès peuvent être suffisants pour les protéger en supposant que les données ne peuvent pas être lues si elles ne sont pas accessibles. Si un niveau de sécurité plus élevé est requis, les données peuvent être chiffrées.

Chiffrer des données sensibles lorsqu'elles sont transmises sur un réseau de communication, en particulier sur un réseau non sécurisé tel que l'Internet. Dans un environnement réseau, les mécanismes de contrôle d'accès ne sont pas efficaces contre les tentatives d'interception des données, telles que les écoutes téléphoniques.

Confidentialité dans IBM MQ

Vous pouvez implémenter la confidentialité dans IBM MQ en chiffrant les messages.

La confidentialité peut être assurée dans un environnement IBM MQ comme suit:

- Une fois qu'un agent MCA émetteur obtient un message d'une file d'attente de transmission, IBM MQ utilise TLS pour chiffrer le message avant qu'il ne soit envoyé sur le réseau à l'agent MCA récepteur. A l'autre extrémité du canal, le message est déchiffré avant que l'agent MCA récepteur ne le place dans sa file d'attente de destination.
- Lorsque les messages sont stockés dans une file d'attente locale, les mécanismes de contrôle d'accès fournis par IBM MQ peuvent être considérés comme suffisants pour protéger leur contenu contre toute divulgation non autorisée. Toutefois, pour un niveau de sécurité plus élevé, vous pouvez utiliser Advanced Message Security pour chiffrer les messages stockés dans les files d'attente.
- z/OS Les messages stockés dans les files d'attente locales peuvent être chiffrés au repos à l'aide du chiffrement de l'ensemble de données z/OS .

  Voir la section Confidentiality for data at rest on IBM MQ for z/OS with data set encryption pour plus d'informations.

Intégrité des données

Le service d' *intégrité des données* détecte s'il y a eu une modification non autorisée des données.

Les données peuvent être altérées de deux manières: accidentellement, par des erreurs de matériel et de transmission, ou en raison d'une attaque délibérée. De nombreux produits matériels et protocoles de transmission ont des mécanismes pour détecter et corriger les erreurs matérielles et de transmission. Le but du service d'intégrité des données est de détecter une attaque délibérée.

Le service d'intégrité des données vise uniquement à détecter si des données ont été modifiées. Il ne vise pas à restaurer les données dans leur état d'origine si elles ont été modifiées.

Les mécanismes de contrôle d'accès peuvent contribuer à l'intégrité des données dans la mesure où les données ne peuvent pas être modifiées si l'accès est refusé. Mais, comme pour la confidentialité, les mécanismes de contrôle de l'accès ne sont pas efficaces dans un environnement de réseautage.

Intégrité des données dans IBM MQ

L'intégrité des données peut être assurée dans un environnement IBM MQ comme suit:

- Vous pouvez utiliser TLS pour détecter si le contenu d'un message a été délibérément modifié alors qu'il était transmis sur un réseau. Dans TLS, l'algorithme de synthèse de message permet de détecter les messages modifiés en transit.

  Tous les CipherSpecs IBM MQ fournissent un algorithme de synthèse de message, à l'exception de TLS_RSA_WITH_NULL_NULL, qui ne fournit pas l'intégrité des données de message.

  IBM MQ détecte les messages modifiés lors de leur réception ; lors de la réception d'un message modifié, IBM MQ un message d'erreur AMQ9661 est consigné dans le journal des erreurs et le canal s'arrête.
- Lorsque des messages sont stockés dans une file d'attente locale, les mécanismes de contrôle d'accès fournis par IBM MQ peuvent être considérés comme suffisants pour empêcher une modification délibérée du contenu des messages.

  Toutefois, pour un niveau de sécurité plus élevé, vous pouvez utiliser Advanced Message Security pour détecter si le contenu d'un message a été délibérément modifié entre le moment où le message a été inséré dans la file d'attente et le moment où il a été extrait de la file d'attente.

  Si un message modifié est détecté, l'application qui tente de le recevoir reçoit un code retour MQRC_SECURITY_ERROR (2063). Si l'application utilise un appel MQGET , le message est également déplacé vers SYSTEM.PROTECTION.ERROR.QUEUE .

Concepts cryptographiques

Cette collection de rubriques décrit les concepts de cryptographie applicables à IBM MQ.

Le terme *entité* est utilisé pour désigner un gestionnaire de files d'attente, un IBM MQ MQI client, un utilisateur individuel ou tout autre système capable d'échanger des messages.

Cryptographie

La cryptographie est le processus de conversion entre du texte lisible, appelé *texte en clair*, et un format illisible, appelé *texte chiffré*.

Cela se produit comme suit:

1. L'expéditeur convertit le message en texte en clair en texte chiffré. Cette partie du processus est appelée *chiffrement* (parfois *chiffrement*).
2. Le texte chiffré est transmis au récepteur.
3. Le récepteur reconvertit le message chiffré en texte en clair. Cette partie du processus est appelée *déchiffrement* (parfois *déchiffrement*).

La conversion implique une séquence d'opérations mathématiques qui modifient l'apparence du message lors de la transmission mais n'affectent pas le contenu. Les techniques cryptographiques permettent de garantir la confidentialité et de protéger les messages contre l'affichage non autorisé (écoute clandestine), car un message chiffré n'est pas compréhensible. Les signatures numériques, qui garantissent l'intégrité des messages, utilisent des techniques de chiffrement. Pour plus d'informations, voir «Signatures numériques dans SSL/TLS», à la page 23.

Les techniques cryptographiques impliquent un algorithme général, rendu spécifique par l'utilisation de clés. Il existe deux classes d'algorithme:

- Ceux qui exigent que les deux parties utilisent la même clé secrète. Les algorithmes qui utilisent une clé partagée sont appelés algorithmes *symétriques* . La Figure 1, à la page 12 illustre la cryptographie à clé symétrique.
- Ceux qui utilisent une clé pour le chiffrement et une autre clé pour le déchiffrement. L'un d'eux doit être gardé secret, mais l'autre peut être public. Les algorithmes qui utilisent des paires de clés publiques et privées sont appelés algorithmes *asymétriques* . La Figure 2, à la page 12 illustre la cryptographie à clé asymétrique, également appelée *cryptographie à clé publique*.

Les algorithmes de chiffrement et de déchiffrement utilisés peuvent être publics, mais la clé secrète partagée et la clé privée doivent être gardées secrètes.

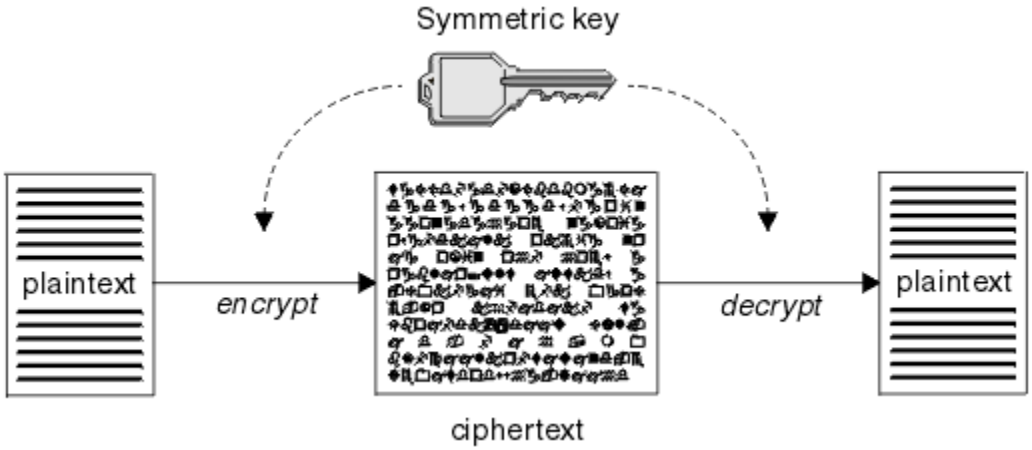


Figure 1. cryptographie à clé symétrique

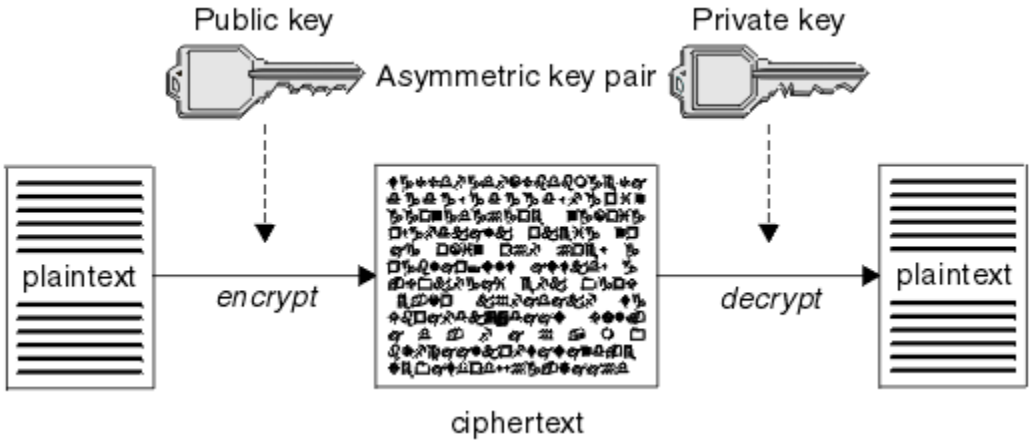


Figure 2. cryptographie à clé asymétrique

Figure 2, à la page 12 affiche le texte en clair chiffré avec la clé publique du récepteur et déchiffré avec la clé privée du récepteur. Seul le récepteur prévu détient la clé privée pour déchiffrer le texte chiffré. Notez que l'expéditeur peut également chiffrer les messages avec une clé privée, ce qui permet à quiconque détient la clé publique de l'expéditeur de déchiffrer le message, avec l'assurance que le message doit provenir de l'expéditeur.

Avec les algorithmes asymétriques, les messages sont chiffrés avec la clé publique ou la clé privée, mais ils ne peuvent être déchiffrés qu'avec l'autre clé. Seule la clé privée est secrète, la clé publique peut être connue de n'importe qui. Avec les algorithmes symétriques, la clé partagée ne doit être connue que des deux parties. Il s'agit du *problème de distribution de clé*. Les algorithmes asymétriques sont plus lents mais présentent l'avantage qu'il n'y a pas de problème de distribution de clé.

Une autre terminologie associée à la cryptographie est:

Force
La force du chiffrement est déterminée par la taille de la clé. Les algorithmes asymétriques requièrent des clés de grande taille, par exemple:

1 024 bits	Clé asymétrique de faible puissance
2048 bits	Clé asymétrique de niveau moyen
4096 bits	Clé asymétrique de haute résistance

Les clés symétriques sont plus petites: les clés 256 bits vous donnent un chiffrement renforcé.

Algorithme de chiffrement de bloc
Ces algorithmes chiffrent les données par blocs. Par exemple, l'algorithme RC2 de RSA Data Security Inc. utilise des blocs d'une longueur de 8 octets. Les algorithmes de bloc sont généralement plus lents que les algorithmes de flux.

Algorithme de chiffrement de flux
Ces algorithmes fonctionnent sur chaque octet de données. Les algorithmes de flux sont généralement plus rapides que les algorithmes de bloc.

Historiques des messages et signatures numériques

Un résumé de message est une représentation numérique de taille fixe du contenu d'un message. Le résumé du message est calculé par une fonction de hachage et peut être chiffré, formant une signature numérique.

La fonction de hachage utilisée pour calculer un résumé de message doit répondre à deux critères:

- Ça doit être un moyen. Il ne doit pas être possible d'inverser la fonction pour trouver le message correspondant à un résumé de message particulier, sauf en testant tous les messages possibles.
- Il doit être infaisable du point de vue du calcul de trouver deux messages qui se hachent dans le même condensé.

Le résumé du message est envoyé avec le message lui-même. Le destinataire peut générer un prétraitement pour le message et le comparer avec le prétraitement de l'expéditeur. L'intégrité du message est vérifiée lorsque les deux historiques de message sont identiques. Toute altération du message au cours de la transmission se traduit presque certainement par un résumé de message différent.

Un résumé de message créé à l'aide d'une clé symétrique secrète est appelé code d'authentification de message (MAC), car il peut fournir l'assurance que le message n'a pas été modifié.

L'expéditeur peut également générer un résumé de message, puis chiffrer le résumé à l'aide de la clé privée d'une paire de clés asymétriques, formant une signature numérique. La signature doit ensuite être déchiffrée par le récepteur, avant de la comparer à un prétraitement généré localement.

Concepts associés
«Signatures numériques dans SSL/TLS», à la page 23
Une signature numérique est formée par le chiffrement d'une représentation d'un message. Le chiffrement utilise la clé privée du signataire et, pour des raisons d'efficacité, opère généralement sur un résumé de message plutôt que sur le message lui-même.

Certificats numériques

Les certificats numériques constituent une protection contre l'usurpation d'identité en certifiant qu'une clé publique appartient à une entité spécifiée. Ils sont émis par une autorité de certification.

Les certificats numériques offrent une protection contre l'emprunt d'identité, car un certificat numérique lie une clé publique à son propriétaire, qu'il s'agisse d'un individu, d'un gestionnaire de files d'attente ou d'une autre entité. Les certificats numériques sont aussi appelés certificats de clé publique car ils donnent des garanties sur l'appartenance d'une clé publique lorsque vous utilisez un schéma de clé asymétrique. Un certificat numérique contient la clé publique d'une entité et établit que la clé publique appartient à cette entité :

- Lorsque le certificat est établi pour une entité individuelle, il est appelé *certificat personnel* ou *certificat d'utilisateur*.
- Lorsque le certificat est établi pour une autorité de certification, il est appelé *certificat d'autorité de certification* ou *certificat de signataire*.

Si les clés publiques sont envoyées directement par leur propriétaire à une autre entité, il existe un risque que le message soit intercepté et que la clé publique soit remplacée par une autre. C'est ce qu'on appelle une attaque de type *homme au milieu*. La solution à ce problème consiste à échanger les clés publiques par le biais d'un tiers sécurisé qui garantit fortement que la clé publique appartient réellement à l'entité

avec laquelle vous communiquez. Au lieu d'envoyer votre clé publique directement, vous demandez au tiers sécurisé de l'incorporer dans un certificat numérique. Le tiers sécurisé qui émet les certificats numériques est appelé autorité de certification, comme décrit dans «Autorités de certification», à la page 15.

Qu'est-ce qu'un certificat numérique ?

Les certificats numériques contiennent des éléments spécifiques d'informations, conformément à la norme X.509.

Les certificats numériques utilisés par IBM MQ sont conformes à la norme X.509, qui spécifie les informations requises et le format nécessaire pour leur envoi. X.509 constitue la partie de l'infrastructure d'authentification de la série X.500 de normes.

Les certificats numériques contiennent au moins les informations suivantes sur l'entité qui est certifiée :

- La clé publique du propriétaire
- Le nom distinctif du propriétaire
- Le nom distinctif de l'autorité de certification qui a émis le certificat
- La date à partir de laquelle le certificat est valide
- La date d'expiration du certificat
- Le numéro de version du format de données du certificat, comme défini dans la norme X.509. La version en cours de la norme X.509 est la version 3, et la plupart des certificats sont conformes à cette version.
- Un numéro de série. Il s'agit d'un identificateur unique affecté par l'autorité de certification qui a émis le certificat. Le numéro de série est unique au sein de l'autorité de certification qui a émis le certificat : deux certificats signés par la même autorité de certification ne peuvent pas avoir le même numéro de série.

Un certificat X.509 de version 2 contient aussi un identificateur d'émetteur et un identificateur d'objet, et un certificat X.509 de version 3 peut contenir un certain nombre d'extensions. Certaines extensions de certificat, comme l'extension Contrainte de base, sont *standard*, alors que d'autres sont propres à l'implémentation. Une extension peut être *critique*, auquel cas un système doit pouvoir reconnaître la zone ; s'il ne reconnaît pas la zone, il doit rejeter le certificat. Si une extension n'est pas critique, le système peut l'ignorer s'il ne la reconnaît pas.

La signature numérique dans un certificat personnel est générée avec la clé privée de l'autorité de certification qui a signé ce certificat. Toute personne devant vérifier le certificat personnel peut utiliser la clé publique de l'autorité de certification pour ce faire. Le certificat de l'autorité de certification contient sa clé publique.

Les certificats numériques ne contiennent pas votre clé privée. Vous devez garder votre clé privée secrète.

Exigences relatives aux certificats personnels

IBM MQ prend en charge les certificats numériques conformes à la norme X.509 . Elle requiert l'option d'authentification du client.

IBM MQ étant un système d'égal à égal, il est considéré comme une authentification client dans la terminologie SSL/TLS. Par conséquent, tout certificat personnel utilisé pour l'authentification SSL/TLS doit autoriser une utilisation de clé de l'authentification client. Cette option n'étant pas activée pour tous les certificats serveur, le fournisseur de certificat peut avoir besoin d'activer l'authentification client sur l'autorité de certification racine pour le certificat sécurisé.

En plus des normes qui spécifient le format de données d'un certificat numérique, il existe également des normes pour déterminer si un certificat est valide. Ces normes ont été mises à jour au fil du temps afin de prévenir certains types de violations de la sécurité. Par exemple, les anciens certificats X.509 versions 1 et 2 n'indiquent pas si le certificat peut être légitimement utilisé pour signer d'autres certificats. Il a donc été possible pour un utilisateur malveillant d'obtenir un certificat personnel d'une source légitime et de créer de nouveaux certificats destinés à usurper l'identité d'autres utilisateurs.

Lors de l'utilisation de certificats X.509 version 3, les extensions de certificat BasicConstraints et KeyUsage sont utilisées pour spécifier les certificats qui peuvent légitimement signer d'autres certificats.

La norme IETF RFC 5280 spécifie une série de règles de validation de certificat que les logiciels d'application conformes doivent implémenter afin d'éviter les attaques d'usurpation d'identité. Un ensemble de règles de certificat est appelé règle de validation de certificat.

Pour plus d'informations sur les règles de validation de certificat dans IBM MQ, voir «Règles de validation de certificat dans IBM MQ», à la page 47.

Autorités de certification

Une autorité de certification est un tiers sécurisé qui émet des certificats numériques pour garantir que la clé publique d'une entité appartient réellement à cette entité.

Les rôles d'une autorité de certification sont les suivants :

- A la réception d'une demande de certificat numérique, vérifier l'identité du demandeur avant de générer, signer et renvoyer le certificat personnel
- Fournir sa propre clé publique dans son certificat d'autorité de certification
- Publier des listes de certificats qui ne sont plus sécurisés dans une liste de révocation de certificat Pour plus d'informations, voir «Utilisation des certificats révoqués», à la page 350
- Fournir l'accès au statut de révocation de certificat via un serveur répondeur OSCP

Noms distinctifs

Le nom distinctif identifie de façon unique une entité dans un certificat X.509.

Avertissement : Seuls les attributs du tableau suivant peuvent être utilisés dans un filtre SSLPEER. Les noms distinctifs de certificat peuvent contenir d'autres attributs, mais le filtrage n'est pas autorisé sur ces attributs.

Tableau 1. Types d'attribut trouvés dans le nom distinctif pouvant être utilisés dans un filtre SSLPEER

Type d'attribut	**Description**
SERIALNUMBER	Numéro de série du certificat
MAIL	Adresse électronique
Deprecated E	Adresse électronique (dépréciée dans la préférence pour MAIL)
UID ou USERID	ID utilisateur
CN	Nom CN
T	Titre
OU	Nom d'unité organisationnelle
DC	Composant de domaine
O	Nom de l'organisation
STREET	Rue/Première ligne d'adresse
L	Nom du lieu
ST (ou SP ou S)	Nom du département
ordinateur personnel	Code postal
C	Pays
UNSTRUCTUREDNAME	Nom d'hôte
UNSTRUCTUREDADDRESS	Adresse IP
DNQ	Qualificateur de nom distinctif

La norme X.509 définit d'autres attributs qui ne font généralement pas partie du nom distinctif mais qui peuvent fournir des extensions en option au certificat numérique.

La norme X.509 permet de spécifier un nom distinctif au format chaîne. Exemple :

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

Le nom usuel (CN) peut décrire un utilisateur individuel ou toute autre entité, par exemple un serveur Web.

Le nom distinctif peut comporter plusieurs attributs OU et DC. Une instance seulement de chacun des autres attributs est admise. L'ordre des entrées OU est important : il spécifie une hiérarchie de noms d'unité organisationnelle, dans laquelle l'unité de niveau supérieur apparaît en premier. L'ordre des entrées DC est également important.

IBM MQ tolère certains noms distinctifs syntaxiquement inappropriés. Pour plus d'informations, voir Règles IBM MQ pour les valeurs SSLPEER.

Concepts associés

«Qu'est-ce qu'un certificat numérique ?», à la page 14
Les certificats numériques contiennent des éléments spécifiques d'informations, conformément à la norme X.509.

Obtention de certificats personnels d'une autorité de certification

Vous pouvez obtenir un certificat d'une autorité de certification externe sécurisée.

Vous obtenez un certificat numérique en envoyant des informations à une autorité de certification, sous la forme d'une demande de certificat. La norme X.509 définit un format pour ces informations, mais certaines autorités de certification proposent leur propre format. Les demandes de certificat sont généralement générées par l'outil de gestion des certificats utilisé par votre système ; par exemple:

- ALW Les commandes **runmqakm** et V 9.4.0 V 9.4.0 **runmqktool** sous AIX, Linux, and Windows.
- z/OS RACF sur z/OS.

Les informations contiennent votre nom distinctif et votre clé publique. Lorsque votre outil de gestion des certificats génère votre demande de certificat, il génère aussi votre clé privée, qui doit rester sécurisée. Ne la communiquez jamais.

Lorsque l'autorité de certification reçoit votre demande, elle vérifie votre identité avant de générer le certificat et de vous l'envoyer sous forme de certificat personnel.

La Figure 3, à la page 16 illustre le processus d'obtention d'un certificat numérique d'une autorité de certification.

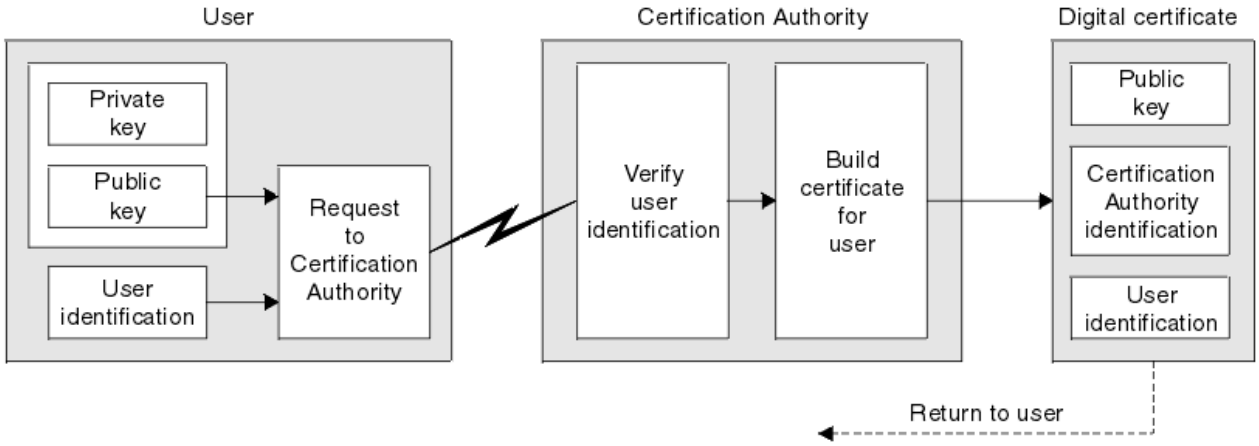


Figure 3. Obtention d'un certificat numérique

Dans le diagramme :

- L'identification de l'utilisateur inclut votre nom distinctif de sujet.
- L'identification de l'autorité de certification inclut le nom distinctif de l'autorité de certification qui émet le certificat.

Les certificats numériques contiennent des zones supplémentaires autres que celles affichées dans le diagramme. Pour plus d'informations sur les autres zones d'un certificat numérique, voir «Qu'est-ce qu'un certificat numérique ?», à la page 14.

Fonctionnement des chaînes de certificats

Lorsque vous recevez le certificat d'une autre entité, vous devrez peut-être utiliser une *chaîne de certificats* pour obtenir le certificat de l' *autorité de certification racine* .

La chaîne de certificats, également appelée *chemin de certification*, est une liste de certificats utilisés pour authentifier une entité. La chaîne, ou chemin, commence par le certificat de cette entité, et chaque certificat de la chaîne est signé par l'entité identifiée par le certificat suivant de la chaîne. La chaîne s'arrête avec un certificat d'autorité de certification racine. Le certificat de l'autorité de certification racine est toujours signé par l'autorité de certification elle-même. Les signatures de tous les certificats de la chaîne doivent être vérifiées jusqu'à ce que le certificat de l'autorité de certification racine soit atteint.

La Figure 4, à la page 17 illustre un chemin de certification entre le propriétaire du certificat et l'autorité de certification racine, où commence la chaîne de confiance.

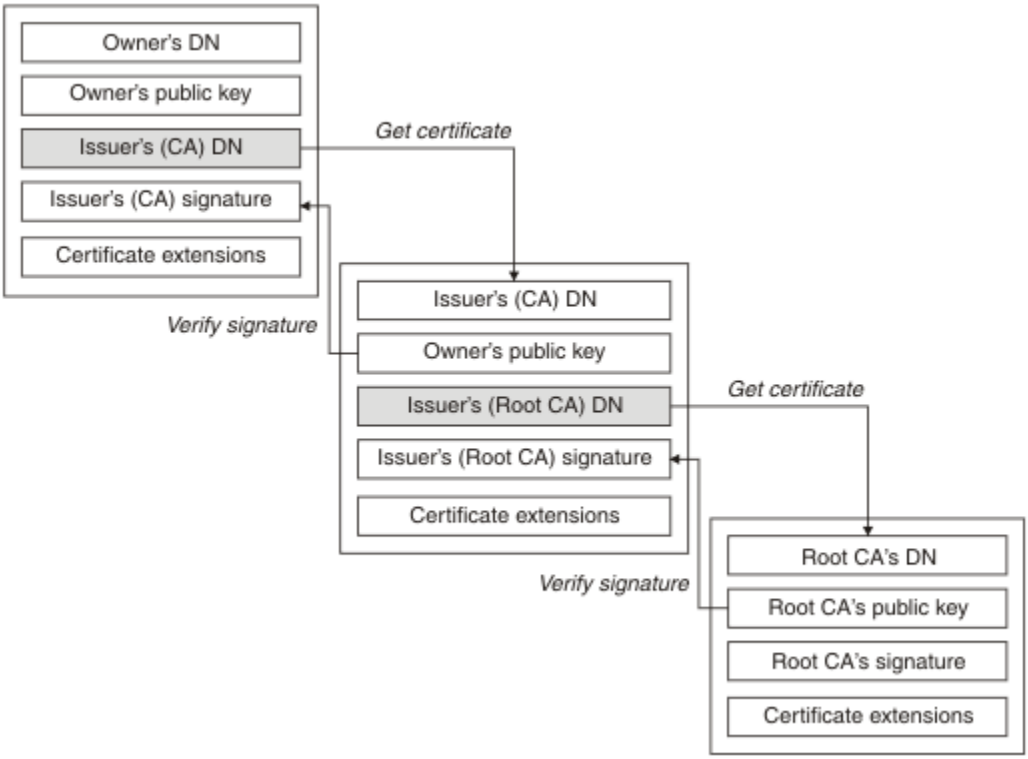


Figure 4. Chaîne de confiance

Chaque certificat peut contenir une ou plusieurs extensions. Un certificat appartenant à une autorité de certification contient généralement une extension BasicConstraints avec l'indicateur isCA défini pour indiquer qu'il est autorisé à signer d'autres certificats.

Lorsque les certificats ne sont plus valides

Les certificats numériques peuvent expirer ou être révoqués.

Les certificats numériques sont délivrés pour une période déterminée et ne sont pas valides après leur date d'expiration.

Les certificats peuvent être révoqués pour diverses raisons, notamment:

- Le propriétaire est parti dans une autre entreprise.
- La clé privée n'est plus secrète.

IBM MQ peut vérifier si un certificat est révoqué en envoyant une demande à un répondeur OCSP (Online Certificate Status Protocol) (sur AIX, Linux, and Windows uniquement). Ils peuvent également accéder à une liste de révocation de certificat (CRL) sur un serveur LDAP. Les informations de révocation OCSP et de LRC sont publiées par une autorité de certification. Pour plus d'informations, voir «Utilisation des certificats révoqués», à la page 350.

Public key infrastructure (PKI)

Une infrastructure à clé publique (ICP) est un système d'installations, de politiques et de services qui prend en charge l'utilisation de la cryptographie à clé publique pour authentifier les parties impliquées dans une transaction.

Il n'existe pas de norme unique qui définit les composants d'une infrastructure à clé publique, mais une ICP comprend généralement des autorités de certification (AC) et des autorités d'enregistrement (AR). Les autorités de certification fournissent les services suivants:

- Emission de certificats numériques
- Validation des certificats numériques
- Révocation de certificats numériques
- Distribution de clés publiques

Les normes X.509 constituent la base de l'infrastructure à clé publique conforme aux normes de l'industrie.

Pour plus d'informations sur les certificats numériques et les autorités de certification, voir «Certificats numériques», à la page 13 . Les autorités de certification vérifient les informations fournies lorsque des certificats numériques sont demandés. Si l'autorité de certification vérifie ces informations, l'autorité de certification peut émettre un certificat numérique au demandeur.

Une infrastructure PKI peut également fournir des outils pour la gestion des certificats numériques et des clés publiques. Une infrastructure PKI est parfois décrite comme une *hiérarchie de confiance* pour la gestion des certificats numériques, mais la plupart des définitions incluent des services supplémentaires. Certaines définitions comprennent des services de chiffrement et de signature numérique, mais ces services ne sont pas essentiels au fonctionnement d'une ICP.

Protocole de sécurité cryptographique TLS

Les protocoles cryptographiques fournissent des connexions sécurisées, permettant à deux parties de communiquer avec la confidentialité et l'intégrité des données. Le protocole TLS (Transport Layer Security) a évolué à partir de celui de la couche SSL (Secure Sockets Layer). IBM MQ prend en charge TLS.

Les objectifs principaux des deux protocoles sont de garantir la confidentialité (parfois appelée *confidentialité*), l'intégrité des données, l'identification et l'authentification à l'aide de certificats numériques.

Bien que les deux protocoles soient similaires, les différences sont suffisamment importantes pour que SSL 3.0 et les différentes versions de TLS n'interopèrent pas.

Concepts associés

«Protocoles de sécurité TLS dans IBM MQ», à la page 25
IBM MQ prend en charge le protocole TLS (Transport Layer Security) pour fournir une sécurité de niveau de liaison pour les canaux de message et les canaux MQI.

Concepts TLS (Transport Layer Security)

Le protocole TLS permet à deux parties de s'identifier et de s'authentifier et de communiquer avec la confidentialité et l'intégrité des données. Le protocole TLS a évolué à partir du protocole Netscape SSL 3.0 , mais TLS et SSL n'interopèrent pas.

Le protocole TLS assure la sécurité des communications sur Internet et permet aux applications client/serveur de communiquer de manière confidentielle et fiable. Les protocoles ont deux couches: un protocole d'enregistrement et un protocole d'établissement de liaison, et celles-ci sont superposées au-dessus d'un protocole de transport tel que TCP/IP. Ils utilisent tous deux des techniques de cryptographie asymétrique et symétrique.

Une connexion TLS est initiée par une application qui devient le client TLS. L'application qui reçoit la connexion devient le serveur TLS. Chaque nouvelle session commence par un établissement de liaison, tel que défini par les protocoles TLS.

La liste complète des CipherSpecs pris en charge par IBM MQ est disponible à l'adresse «Activation des CipherSpecs», à la page 435.

Pour plus d'informations sur le protocole SSL, voir les informations fournies à l'adresse https://developer.mozilla.org/docs/Mozilla/Projects/NSS. Pour plus d'informations sur le protocole TLS, voir les informations fournies par le groupe de travail TLS sur le site Web d'Internet Engineering Task Force à l'adresse https://www.ietf.org

Présentation de l'établissement de liaison SSL/TLS

L'établissement de liaison SSL/TLS permet au client et au serveur TLS d'établir les clés secrètes avec lesquelles ils communiquent.

Cette section fournit un récapitulatif des étapes permettant au client et au serveur TLS de communiquer entre eux.

- Convenir de la version du protocole à utiliser.
- Sélectionnez des algorithmes de cryptographie.
- Authentifiez-vous les uns les autres en échangeant et en validant des certificats numériques.
- Utilisez des techniques de chiffrement asymétrique pour générer une clé secrète partagée, ce qui évite le problème de distribution des clés. TLS utilise ensuite la clé partagée pour le chiffrement symétrique des messages, qui est plus rapide que le chiffrement asymétrique.

Pour plus d'informations sur les algorithmes de cryptographie et les certificats numériques, reportez-vous aux informations connexes.

Dans la présentation, les étapes impliquées dans l'établissement de liaison TLS sont les suivantes:

1. Le client TLS envoie un message "client hello" qui répertorie les informations cryptographiques telles que la version TLS et, dans l'ordre de préférence du client, les CipherSuites prises en charge par le client. Le message contient également une chaîne d'octets aléatoire qui est utilisée dans les calculs ultérieurs. Le protocole permet au "client hello" d'inclure les méthodes de compression de données prises en charge par le client.
2. Le serveur TLS répond avec un message "server hello" qui contient la suite de chiffrement CipherSuite choisie par le serveur dans la liste fournie par le client, l'ID de session et une autre chaîne d'octets aléatoires. Le serveur envoie également son certificat numérique. Si le serveur requiert un certificat numérique pour l'authentification du client, il envoie une "demande de certificat client" qui inclut une liste des types de certificat pris en charge et des noms distinctifs des autorités de certification (CA) acceptables.
3. Le client TLS vérifie le certificat numérique du serveur. Pour plus d'informations, voir «Comment TLS fournit l'identification, l'authentification, la confidentialité et l'intégrité», à la page 20.
4. Le client TLS envoie la chaîne d'octets aléatoires qui permet au client et au serveur de calculer la clé secrète à utiliser pour le chiffrement des données de message suivantes. La chaîne d'octets aléatoires elle-même est chiffrée avec la clé publique du serveur.

5. Si le serveur TLS a envoyé une "demande de certificat client", le client envoie une chaîne d'octets aléatoire chiffrée avec la clé privée du client, ainsi que le certificat numérique du client, ou une "alerte de non-certificat numérique". Cette alerte n'est qu'un avertissement, mais avec certaines implémentations, l'établissement de liaison échoue si l'authentification du client est obligatoire.
6. Le serveur TLS vérifie le certificat du client. Pour plus d'informations, voir «Comment TLS fournit l'identification, l'authentification, la confidentialité et l'intégrité», à la page 20.
7. Le client TLS envoie au serveur un message "finished" , qui est chiffré avec la clé secrète, indiquant que la partie client de l'établissement de liaison est terminée.
8. Le serveur TLS envoie au client un message "finished" , qui est chiffré avec la clé secrète, indiquant que la partie serveur de l'établissement de liaison est terminée.
9. Pendant la durée de la session TLS, le serveur et le client peuvent désormais échanger des messages qui sont chiffrés de manière symétrique avec la clé secrète partagée.

La Figure 5, à la page 20 illustre l'établissement de liaison TLS.

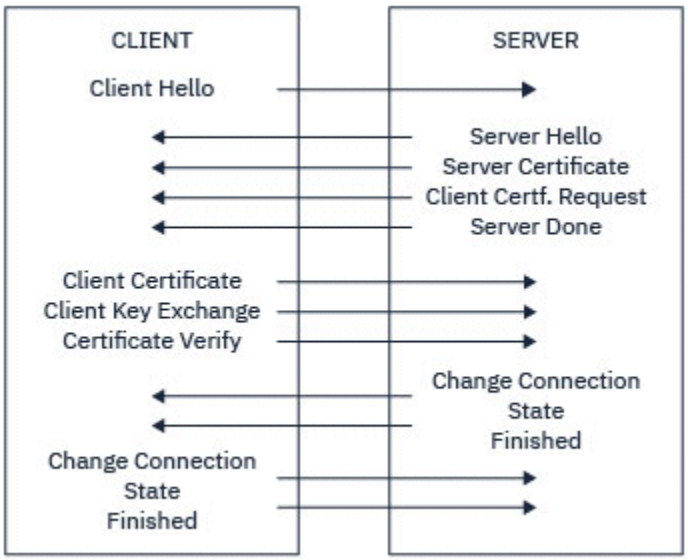


Figure 5. Présentation de l'établissement de liaison TLS

Comment TLS fournit l'identification, l'authentification, la confidentialité et l'intégrité

Lors de l'authentification du client et du serveur, une étape exige que les données soient chiffrées avec l'une des clés d'une paire de clés asymétriques et déchiffrées avec l'autre clé de la paire. Un résumé de message est utilisé pour assurer l'intégrité.

Pour une présentation des étapes impliquées dans l'établissement de liaison TLS, voir «Présentation de l'établissement de liaison SSL/TLS», à la page 19.

Comment TLS fournit l'authentification

Pour l'authentification du serveur, le client utilise la clé publique du serveur pour chiffrer les données utilisées pour calculer la clé secrète. Le serveur ne peut générer la clé secrète que s'il peut déchiffrer ces données avec la clé privée appropriée. La chaîne d'octets aléatoires elle-même est chiffrée à l'aide de la clé publique du serveur (étape «4», à la page 19 de la présentation).

Pour l'authentification client, le serveur utilise la clé publique dans le certificat client pour déchiffrer les données que le client envoie lors de l'étape «5», à la page 20 de l'établissement de liaison. L'échange des messages terminés qui sont chiffrés avec la clé secrète (étapes «7», à la page 20 et «8», à la page 20 de la présentation) confirme que l'authentification est terminée.

Si l'une des étapes d'authentification échoue, l'établissement de liaison échoue et la session se termine.

L'échange de certificats numériques lors de l'établissement de liaison TLS fait partie du processus d'authentification. Pour plus d'informations sur la façon dont les certificats fournissent une protection contre l'usurpation d'identité, reportez-vous aux informations connexes. Les certificats requis sont les suivants, où l'autorité de certification X émet le certificat sur le client TLS et l'autorité de certification Y émet le certificat sur le serveur TLS:

Pour l'authentification de serveur uniquement, le serveur TLS a besoin de:

- Certificat personnel émis sur le serveur par l'autorité de certification Y
- Clé privée du serveur

et le client TLS a besoin:

- Le certificat de l'autorité de certification pour l'autorité de certification Y

Si le serveur TLS requiert une authentification client, le serveur vérifie l'identité du client en vérifiant le certificat numérique du client avec la clé publique de l'autorité de certification qui a émis le certificat personnel au client, en l'occurrence l'autorité de certification X. Pour l'authentification du serveur et du client, le serveur a besoin:

- Certificat personnel émis sur le serveur par l'autorité de certification Y
- Clé privée du serveur
- Le certificat de l'autorité de certification pour l'autorité de certification X

et le client a besoin:

- Certificat personnel émis pour le client par l'autorité de certification X
- Clé privée du client
- Le certificat de l'autorité de certification pour l'autorité de certification Y

Le serveur et le client TLS peuvent avoir besoin d'autres certificats de l'autorité de certification pour former une chaîne de certificats pour le certificat de l'autorité de certification racine. Pour plus d'informations sur les chaînes de certificats, reportez-vous aux informations associées.

Ce qui se passe lors de la vérification de certificat

Comme indiqué dans les étapes «3», à la page 19 et «6», à la page 20 de la présentation, le client TLS vérifie le certificat du serveur et le serveur TLS vérifie le certificat du client. Cette vérification comporte quatre aspects:

1. La signature numérique est vérifiée (voir «Signatures numériques dans SSL/TLS», à la page 23).
2. La chaîne de certificats est vérifiée ; vous devez disposer de certificats d'autorité de certification intermédiaires (voir «Fonctionnement des chaînes de certificats», à la page 17).
3. Les dates d'expiration et d'activation ainsi que la période de validité sont vérifiées.
4. Le statut de révocation du certificat est vérifié (voir «Utilisation des certificats révoqués», à la page 350).

Réinitialisation de la clé secrète

Lors de l'établissement d'une liaison TLS, une *clé secrète* est générée pour chiffrer les données entre le client et le serveur TLS. La clé secrète est utilisée dans une formule mathématique qui est appliquée aux données pour transformer du texte en clair en texte chiffré illisible et du texte chiffré en texte en clair.

La clé secrète est générée à partir du texte aléatoire envoyé dans le cadre de l'établissement de liaison et est utilisée pour chiffrer du texte en clair en texte chiffré. La clé secrète est également utilisée dans l'algorithme MAC (Message Authentication Code), qui est utilisé pour déterminer si un message a été modifié. Pour plus d'informations, voir «Historiques des messages et signatures numériques», à la page 13.

Si la clé secrète est découverte, le texte en clair d'un message peut être déchiffré à partir du texte chiffré, ou le résumé du message peut être calculé, ce qui permet de modifier les messages sans détection.

Même pour un algorithme complexe, le texte en clair peut éventuellement être découvert en appliquant chaque transformation mathématique possible au texte chiffré. Pour minimiser la quantité de données pouvant être déchiffrées ou modifiées si la clé secrète est rompue, la clé secrète peut être renégociée périodiquement. Lorsque la clé secrète a été renégociée, la clé secrète précédente ne peut plus être utilisée pour déchiffrer les données chiffrées avec la nouvelle clé secrète.

Comment TLS assure la confidentialité

TLS utilise une combinaison de chiffrement symétrique et asymétrique pour garantir la confidentialité des messages. Lors de l'établissement de liaison TLS, le client et le serveur TLS conviennent d'un algorithme de chiffrement et d'une clé secrète partagée à utiliser pour une seule session. Tous les messages transmis entre le client TLS et le serveur sont chiffrés à l'aide de cet algorithme et de cette clé, ce qui garantit que le message reste privé même s'il est intercepté. Etant donné que TLS utilise le chiffrement asymétrique lors du transport de la clé secrète partagée, il n'y a pas de problème de distribution de clé. Pour plus d'informations sur les techniques de chiffrement, voir «Cryptographie», à la page 11.

Comment TLS assure l'intégrité

TLS assure l'intégrité des données en calculant un résumé de message. Pour plus d'informations, voir «Intégrité des données de messages», à la page 491.

L'utilisation de TLS garantit l'intégrité des données, à condition que le CipherSpec de votre définition de canal utilise un algorithme de hachage comme décrit dans le tableau dans «Activation des CipherSpecs», à la page 435.

En particulier, si l'intégrité des données pose problème, vous devez éviter de choisir un CipherSpec dont l'algorithme de hachage est répertorié comme "Aucun". L'utilisation de MD5 est également fortement déconseillée car elle est désormais très ancienne et n'est plus sécurisée pour des raisons pratiques.

CipherSpecs et CipherSuites

Les protocoles de sécurité cryptographique doivent convenir des algorithmes utilisés par une connexion sécurisée. CipherSpecs et CipherSuites définissent des combinaisons spécifiques d'algorithmes.

Un CipherSpec identifie une combinaison d'algorithme de chiffrement et d'algorithme MAC (Message Authentication Code). Les deux extrémités d'une connexion TLS doivent convenir du même CipherSpec pour pouvoir communiquer.

IBM MQ prend en charge les protocoles TLS1.3 et TLS1.2 et les CipherSpecs. Toutefois, vous pouvez activer les CipherSpecsobsolètes, si vous devez le faire.

Voir «Activation des CipherSpecs», à la page 435 pour plus d'informations sur:

- CipherSpecs pris en charge par IBM MQ
- Comment activer les CipherSpecs SSL 3.0 et TLS 1.0 CipherSpecs

Important : Lorsque vous utilisez des canaux IBM MQ , vous utilisez un CipherSpec. Lorsque vous utilisez des canaux Java , JMS ou MQTT, vous spécifiez une CipherSuite.

Pour plus d'informations sur les CipherSpecs, voir «Activation des CipherSpecs», à la page 435.

Une CipherSuite est une suite d'algorithmes de cryptographie utilisés par une connexion TLS. Une suite comprend trois algorithmes distincts:

- Algorithme d'échange de clés et d'authentification utilisé lors de l'établissement de liaison
- Algorithme de chiffrement utilisé pour chiffrer les données
- Algorithme MAC (Message Authentication Code) utilisé pour générer le résumé de message

Il existe plusieurs options pour chaque composant de la suite, mais seules certaines combinaisons sont valides lorsqu'elles sont spécifiées pour une connexion TLS. Le nom d'une CipherSuite valide définit la combinaison des algorithmes utilisés. Par exemple, CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA spécifie:

- Algorithme d'authentification et d'échange de clés RSA
- L'algorithme de chiffrement AES, utilisant une clé de 128 bits et le mode CBC (cipher block chaining)
- Code d'authentification de message (MAC) SHA-1

Signatures numériques dans SSL/TLS

Une signature numérique est formée par le chiffrement d'une représentation d'un message. Le chiffrement utilise la clé privée du signataire et, pour des raisons d'efficacité, opère généralement sur un résumé de message plutôt que sur le message lui-même.

Les signatures numériques varient avec les données en cours de signature, contrairement aux signatures manuscrites, qui ne dépendent pas du contenu du document en cours de signature. Si deux messages différents sont signés numériquement par la même entité, les deux signatures diffèrent, mais les deux signatures peuvent être vérifiées avec la même clé publique, c'est-à-dire la clé publique de l'entité qui a signé les messages.

Les étapes du processus de signature numérique sont les suivantes:

1. L'expéditeur calcule un résumé de message, puis il chiffre le résumé à l'aide de la clé privée de l'expéditeur, en formant la signature numérique.
2. L'émetteur transmet la signature numérique avec le message.
3. Le récepteur déchiffre la signature numérique à l'aide de la clé publique de l'expéditeur, en régénérant le résumé du message de l'expéditeur.
4. Le récepteur calcule un résumé de message à partir des données de message reçues et vérifie que les deux résumés sont identiques.

La Figure 6, à la page 23 illustre ce processus.

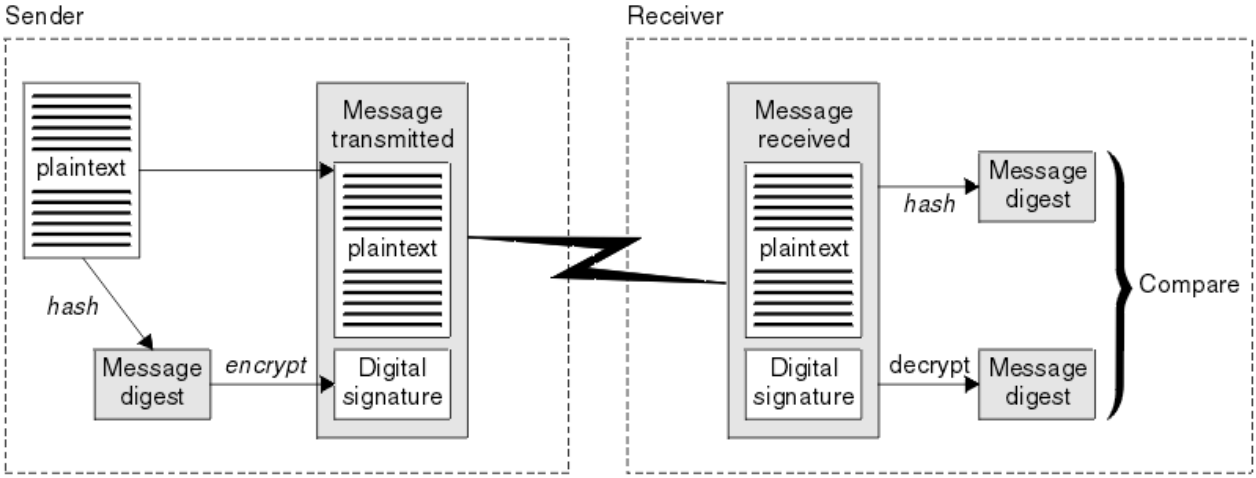


Figure 6. Processus de signature numérique

Si la signature numérique est vérifiée, le récepteur sait que:

- Le message n'a pas été modifié lors de la transmission.
- Le message a été envoyé par l'entité qui prétend l'avoir envoyé.

Les signatures numériques font partie des services d'intégrité et d'authentification. Les signatures numériques fournissent également une preuve de l'origine. Seul l'expéditeur connaît la clé privée, ce qui fournit des preuves solides que l'expéditeur est l'émetteur du message.

Remarque : Vous pouvez également chiffrer le message lui-même, ce qui protège la confidentialité des informations du message.

La norme FIPS (Federal Information Processing Standards)

Le gouvernement américain fournit des conseils techniques sur des systèmes informatiques et sur des systèmes de sécurité, y compris le chiffrement de données. Le National Institute for Standards and Technology (NIST) est un organisme important qui s'occupe des systèmes informatiques et de la sécurité. Il émet des recommandations et établit des normes, notamment la norme FIPS (Federal Information Processing Standards).

L'une de ces normes est la norme FIPS 140-2, qui nécessite l'utilisation d'algorithmes de cryptographie puissants. Elle spécifie également des exigences pour les algorithmes de hachage à utiliser afin de protéger les paquets contre toute modification en transit.

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

IBM MQ fournit une prise en charge FIPS 140-2 lorsqu'il a été configuré pour le faire.

Avec le temps, les analystes développent des attaques contre les algorithmes de chiffrement et de hachage existants. De nouveaux algorithmes sont adoptés pour résister à ces attaques. La norme FIPS 140-2 est mise à jour régulièrement afin de tenir compte de ces changements.

Concepts associés

«Agence de sécurité nationale (NSA) Suite B Cryptographie», à la page 24
Le gouvernement des États-Unis d'Amérique fournit des conseils techniques sur les systèmes informatiques et la sécurité, y compris le chiffrement des données. La National Security Agency (NSA) des États-Unis recommande un ensemble d'algorithmes cryptographiques interopérables dans sa norme Suite B.

Agence de sécurité nationale (NSA) Suite B Cryptographie

Le gouvernement des États-Unis d'Amérique fournit des conseils techniques sur les systèmes informatiques et la sécurité, y compris le chiffrement des données. La National Security Agency (NSA) des États-Unis recommande un ensemble d'algorithmes cryptographiques interopérables dans sa norme Suite B.

La norme Suite B spécifie un mode de fonctionnement dans lequel seul un ensemble spécifique d'algorithmes cryptographiques sécurisés est utilisé. La norme Suite B spécifie:

- L'algorithme de chiffrement (AES)
- L'algorithme d'échange de clés (Elliptic Curve Diffie-Hellman, également connu sous le nom d'ECDH)
- L'algorithme de signature numérique (Elliptic Curve Digital Signature Algorithm, également appelé ECDSA)
- Les algorithmes de hachage (SHA-256 ou SHA-384)

De plus, la norme IETF RFC 6460 spécifie des profils compatibles Suite B qui définissent la configuration détaillée de l'application et le comportement nécessaires pour se conformer à la norme Suite B. Il définit deux profils:

1. Profil compatible Suite B à utiliser avec TLS 1.2. Lorsqu'il est configuré pour une opération conforme à la suite B, seul l'ensemble restreint d'algorithmes de cryptographie répertorié est utilisé.
2. Profil de transition à utiliser avec TLS 1.0 ou TLS 1.1. Ce profil permet l'interopérabilité avec les serveurs non conformes à la norme Suite B. Lorsqu'il est configuré pour l'opération de transition Suite B, des algorithmes de chiffrement et de hachage supplémentaires peuvent être utilisés.

La norme Suite B est conceptuellement similaire à la norme FIPS 140-2, car elle restreint l'ensemble des algorithmes de cryptographie activés afin de fournir un niveau de sécurité assuré.

Sur les systèmes AIX, Linux, and Windows , IBM MQ, peut être configuré pour être conforme au profil TLS 1.2 compatible Suite B, mais ne prend pas en charge le profil de transition Suite B. Pour plus d'informations, reportez-vous à la section «NSA Suite B Cryptography dans IBM MQ», à la page 44.

Référence associée

«La norme FIPS (Federal Information Processing Standards)», à la page 24
Le gouvernement américain fournit des conseils techniques sur des systèmes informatiques et sur des systèmes de sécurité, y compris le chiffrement de données. Le National Institute for Standards and Technology (NIST) est un organisme important qui s'occupe des systèmes informatiques et de la sécurité. Il émet des recommandations et établit des normes, notamment la norme FIPS (Federal Information Processing Standards).

IBM MQ mécanismes de sécurité

Cette collection de rubriques décrit des mécanismes spécifiques dans IBM MQ qui implémentent les différents concepts de sécurité.

Protocoles de sécurité TLS dans IBM MQ

IBM MQ prend en charge le protocole TLS (Transport Layer Security) pour fournir une sécurité de niveau de liaison pour les canaux de message et les canaux MQI.

Les canaux de transmission de messages et les canaux MQI peuvent utiliser le protocole TLS pour assurer la sécurité au niveau de la liaison. Un agent MCA appelant est un client TLS et un agent MCA répondeur est un serveur TLS.

IBM MQ prend en charge les versions 1.2 et 1.3 du protocole TLS. Les versions antérieures de TLS, ainsi que SSL, ne sont pas activées par défaut, mais peuvent l'être si nécessaire. Vous pouvez spécifier les algorithmes de cryptographie utilisés par le protocole TLS en fournissant un CipherSpec dans la définition de canal.

Voir «Activation des CipherSpecs», à la page 435 pour la liste des CipherSpecs pris en charge par IBM MQ et «CipherSpecs obsolètes», à la page 450 pour ceux qui sont obsolètes.

Vous pouvez utiliser les paramètres SECPROT et SSLCIPH pour afficher le protocole de sécurité et CipherSpec utilisé sur un canal.

A chaque extrémité d'un canal de transmission de messages et à l'extrémité serveur d'un canal MQI, l'agent MCA agit pour le compte du gestionnaire de files d'attente auquel il est connecté. Lors de l'établissement de liaison TLS, l'agent MCA envoie le certificat numérique du gestionnaire de files d'attente à son agent MCA partenaire à l'autre extrémité du canal. Le code IBM MQ à l'extrémité client d'un canal MQI agit pour le compte de l'utilisateur de l'application client IBM MQ . Lors de l'établissement de liaison TLS, le code IBM MQ envoie le certificat numérique de l'utilisateur à l'agent MCA à l'extrémité serveur du canal MQI.

Les gestionnaires de files d'attente et les utilisateurs de client IBM MQ ne sont pas tenus d'avoir des certificats numériques personnels qui leur sont associés lorsqu'ils agissent en tant que clients TLS, sauf si SSLCAUTH (REQUIRED) est spécifié côté serveur du canal.

Les certificats numériques sont stockés dans un *référentiel de clés*. L'attribut de gestionnaire de files d'attente **SSLKeyRepository** indique l'emplacement du référentiel de clés qui contient le certificat numérique du gestionnaire de files d'attente. Sur un système client IBM MQ , la variable d'environnement MQSSLKEYR indique l'emplacement du référentiel de clés qui contient le certificat numérique de l'utilisateur. Une application client IBM MQ peut également spécifier son emplacement dans la zone **KeyRepository** de la structure d'options de configuration TLS, MQSCO, sur un appel MQCONNX. Consultez les rubriques connexes pour plus d'informations sur les référentiels de clés et pour savoir comment spécifier leur emplacement.

Prise en charge de TLS

IBM MQ prend en charge TLS 1.2 et TLS 1.3 sur toutes les plateformes. Pour plus d'informations sur le protocole TLS, reportez-vous aux informations des sous-rubriques.

Clients Java et JMS
Ces clients utilisent la machine virtuelle Java pour fournir la prise en charge TLS.

AIX, Linux, and Windows
La prise en charge de TLS est installée avec IBM MQ.

IBM i
La prise en charge de TLS fait partie intégrante du système d'exploitation IBM i .

z/OS
La prise en charge de TLS fait partie intégrante du système d'exploitation z/OS . La prise en charge de TLS sur z/OS est appelée *System SSL*.

Pour plus d'informations sur les prérequis pour la prise en charge de TLS dans IBM MQ , voir Configuration système requise pour IBM MQ.

Concepts associés
«Protocole de sécurité cryptographique TLS», à la page 18
Les protocoles cryptographiques fournissent des connexions sécurisées, permettant à deux parties de communiquer avec la confidentialité et l'intégrité des données. Le protocole TLS (Transport Layer Security) a évolué à partir de celui de la couche SSL (Secure Sockets Layer). IBM MQ prend en charge TLS.

Référentiel de clés SSL/TLS
Une connexion TLS mutuellement authentifiée requiert un référentiel de clés à chaque extrémité de la connexion. Le référentiel de clés inclut des certificats numériques et des clés privées.

Ces informations utilisent le terme général *référentiel de clés* pour décrire le magasin des certificats numériques et leurs clés privées associées. Le référentiel de clés est référencé par des noms différents sur des plateformes et des environnements différents qui prennent en charge TLS:

- 

Sous IBM i: *magasin de certificats*
- Sous Java et JMS: *magasin de clés* et *magasin de clés de confiance*
- 

Sous AIX, Linux, and Windows: *fichier de base de données de clés*
- 

Sous z/OS: *keyring*

Pour plus d'informations, reportez-vous aux sections «Certificats numériques», à la page 13 et «Concepts TLS (Transport Layer Security)», à la page 19.

Une connexion TLS mutuellement authentifiée requiert un référentiel de clés à chaque extrémité de la connexion. Le référentiel de clés peut contenir les certificats et demandes suivants:

- Un certain nombre de certificats de l'autorité de certification provenant de différentes autorités de certification qui permettent au gestionnaire de files d'attente ou au client de vérifier les certificats qu'il reçoit de son partenaire à l'extrémité éloignée de la connexion. Les certificats individuels peuvent se trouver dans une chaîne de certificats.
- Un ou plusieurs certificats personnels reçus d'une autorité de certification. Vous associez un certificat personnel distinct à chaque gestionnaire de files d'attente ou à IBM MQ MQI client. Les certificats personnels sont essentiels sur un client TLS si une authentification mutuelle est requise. Si l'authentification mutuelle n'est pas requise, les certificats personnels ne sont pas nécessaires sur le client. Le référentiel de clés peut également contenir la clé privée correspondant à chaque certificat personnel.
- Demandes de certificat qui attendent d'être signées par un certificat de l'autorité de certification digne de confiance.

Pour plus d'informations sur la protection de votre référentiel de clés, voir «Protection des référentiels de clés IBM MQ», à la page 27.

L'emplacement du référentiel de clés dépend de la plateforme que vous utilisez:

IBM i

Le référentiel de clés est un magasin de certificats. Le magasin de certificats de système par défaut se trouve à l'emplacement `/QIBM/UserData/ICSS/Cert/Server/Default` dans le système de fichiers intégré (IFS). IBM MQ stocke le mot de passe du magasin de certificats dans un *fichier de mot de passe secret*. Par exemple, le fichier de dissimulation du gestionnaire de files d'attente `QM1` est `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`.

Vous pouvez également indiquer que le magasin de certificats de système IBM i doit être utilisé à la place. Pour ce faire, remplacez la valeur de l'attribut **SSLKEYR** du gestionnaire de files d'attente par *SYSTEM. Cette valeur indique que le gestionnaire de files d'attente doit utiliser le magasin de certificats du système et que le gestionnaire de files d'attente est enregistré pour être utilisé en tant qu'application avec Digital Certificate Manager (DCM).

Le magasin de certificats contient également la clé privée du gestionnaire de files d'attente.

AIX, Linux, and Windows systèmes

Le référentiel de clés est un fichier de base de données de clés. Par exemple, sous AIX and Linux, le fichier de base de données de clés par défaut pour le gestionnaire de files d'attente `QM1` est `/var/mqm/qmgrs/QM1/ssl/key.kdb`. Si IBM MQ est installé dans l'emplacement par défaut, le chemin équivalent sous Windows est `C:\ProgramData\IBM\MQ\Qmgrs\QM1\ssl\key.kdb`.

Pour accéder à un fichier de base de données de clés, vous devez indiquer le mot de passe de la base de données de clés IBM MQ . Cette opération peut être effectuée directement ou via un fichier de mot de passe secret. Si un fichier de mot de passe secret est utilisé, il doit se trouver dans le même répertoire et avoir le même radical de fichier que la base de données de clés, et doit se terminer par le suffixe `.sth`, par exemple `/var/mqm/qmgrs/QM1/ssl/key.sth`.

Remarque : Les cartes matérielles de cryptographie PKCS #11 peuvent contenir les certificats et les clés qui sont conservés dans un fichier de base de données de clés. Lorsque des certificats et des clés sont détenus sur des cartes PKCS #11 , IBM MQ a toujours besoin d'accéder à la fois à un fichier de base de données de clés et à un fichier de mot de passe secret.

Sur les systèmes AIX, Linux, and Windows , la base de données de clés contient également la clé privée du certificat personnel associé au gestionnaire de files d'attente ou à IBM MQ MQI client.

z/OS

Les certificats sont stockés dans un fichier de clés dans z/OS.

D'autres gestionnaires de sécurité externes utilisent également des fichiers de clés pour stocker des certificats.

Les clés privées sont gérées par RACF.

Protection des référentiels de clés IBM MQ

Le référentiel de clés pour IBM MQ est un fichier. Assurez-vous que seul l'utilisateur prévu peut accéder au fichier de référentiel de clés. Cela empêche un intrus ou un autre utilisateur non autorisé de copier le fichier de référentiel de clés sur un autre système, puis de configurer un ID utilisateur identique sur ce système pour simuler les droits d'accès de l'utilisateur prévu.

Les droits sur les fichiers dépendent de l'umask de l'utilisateur et de l'outil utilisé. Sous Windows, les comptes IBM MQ requièrent des droits `BypassTraverseChecking` , ce qui signifie que les droits des dossiers dans le chemin d'accès au fichier n'ont aucun effet.

Vérifiez les droits d'accès aux fichiers du référentiel de clés et assurez-vous que les fichiers et le dossier qui les contient ne sont pas lisibles par tout le monde, de préférence pas même par groupe.

La mise en lecture seule du magasin de clés est recommandée, quel que soit le système que vous utilisez, seul l'administrateur étant autorisé à activer les opérations d'écriture afin d'effectuer la maintenance.

Dans la pratique, vous devez protéger tous les magasins de clés, quel que soit l'emplacement et s'ils sont protégés par mot de passe ou non ; protégez les référentiels de clés.

Labels de certificat numérique, compréhension des exigences
Lors de la configuration de TLS pour utiliser des certificats numériques, il peut y avoir des exigences de libellé spécifiques que vous devez respecter, en fonction de la plateforme utilisée et de la méthode que vous utilisez pour vous connecter.

Qu'est-ce que le label de certificat?

Un label de certificat est un identificateur unique représentant un certificat numérique stocké dans un référentiel de clés et fournit un nom lisible par l'utilisateur qui permet de faire référence à un certificat particulier lors de l'exécution de fonctions de gestion de clés. Vous affectez le libellé de certificat lors de l'ajout d'un certificat à un référentiel de clés pour la première fois.

Le libellé de certificat est distinct des zones **Subject Distinguished Name** ou **Subject Common Name** du certificat. Notez que **Subject Distinguished Name** et **Subject Common Name** sont des zones du certificat lui-même. Elles sont définies lors de la création du certificat et ne peuvent pas être modifiées. Toutefois, si nécessaire, vous pouvez modifier le libellé associé à un certificat numérique.

Syntaxe du libellé de certificat

Un label de certificat peut contenir des lettres, des chiffres et des signes de ponctuation avec les conditions suivantes:

- Multi Le libellé de certificat peut contenir jusqu'à 64 caractères.
- z/OS Le libellé de certificat peut contenir jusqu'à 32 caractères.
- Le libellé de certificat peut contenir des espaces.
- Les libellés sont sensibles à la casse.
- Sur les systèmes qui utilisent EBCDIC katakana, vous ne pouvez pas utiliser de caractères minuscules.

Des exigences supplémentaires pour les valeurs de label de certificat sont spécifiées dans les sections suivantes.

Comment le label de certificat est-il utilisé?

IBM MQ utilise des libellés de certificat pour localiser un certificat personnel envoyé lors de l'établissement de liaison TLS. Cela élimine l'ambiguïté lorsque plusieurs certificats personnels existent dans le référentiel de clés.

Vous pouvez définir le libellé de certificat sur une valeur de votre choix. Si vous ne définissez pas de valeur, un libellé par défaut est utilisé, qui suit une convention de dénomination en fonction de la plateforme que vous utilisez. Pour plus de détails, voir les sections suivantes sur des plateformes particulières.

Remarques :

1. Vous ne pouvez pas définir vous-même le libellé de certificat sur les systèmes Java ou JMS .
2. Les canaux définis automatiquement créés par un exit de définition automatique de canal (CHAD) ne peuvent pas définir le libellé de certificat car l'établissement de liaison TLS a eu lieu au moment de la création du canal. La définition du libellé de certificat dans un exit CHAD pour les canaux entrants n'a aucun effet.

Dans ce contexte, un client TLS fait référence au partenaire de connexion qui initie l'établissement de liaison, qui peut être un client IBM MQ ou un autre gestionnaire de files d'attente.

Lors de l'établissement de liaison TLS, le client TLS obtient toujours un certificat numérique du serveur et le valide. Avec l'implémentation IBM MQ , le serveur TLS demande toujours un certificat au client et le

client fournit toujours un certificat au serveur s'il en trouve un. Si le client ne parvient pas à localiser un certificat personnel, il envoie une réponse `no certificate` au serveur.

Le serveur TLS valide toujours le certificat client si celui-ci est envoyé. Si le client n'envoie pas de certificat, l'authentification échoue si l'extrémité du canal qui agit en tant que serveur TLS est définie avec le paramètre **SSLCAUTH** défini sur *REQUIRED* ou une valeur de paramètre **SSLPEER** définie.

Notez que les canaux entrants (y compris les canaux récepteur, demandeur, récepteur de cluster, serveur non qualifié et connexion serveur) n'envoient le certificat configuré que si la version IBM MQ de l'homologue distant prend entièrement en charge la configuration des libellés de certificat et que le canal utilise un CipherSpecTLS.

Un canal serveur non qualifié est un canal dont la zone CONNAME n'est pas définie.

Dans tous les autres cas, le paramètre **CERTLABL** du gestionnaire de files d'attente détermine le certificat envoyé. En particulier, les éléments suivants ne reçoivent que le certificat configuré par le paramètre **CERTLABL** du gestionnaire de files d'attente, quelle que soit la valeur de libellé spécifique au canal:

- Les clients Java et JMS prenant en charge l'indication de nom de serveur (SNI), c'est-à-dire les certificats canal par canal.
- Versions de IBM MQ antérieures à IBM MQ 8.0.
- Clients .NET gérés

En outre, le certificat utilisé par un canal doit être approprié pour le canal CipherSpec -voir «Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49 pour plus d'informations.

IBM MQ 8.0 et les versions ultérieures prennent en charge l'utilisation de plusieurs certificats sur le même gestionnaire de files d'attente, à l'aide d'un libellé de certificat par canal, spécifié à l'aide de l'attribut **CERTLABL** sur la définition de canal. Les canaux entrants dans le gestionnaire de files d'attente (par exemple, la connexion serveur ou le récepteur) dépendent de la détection du nom de canal à l'aide de TLS Server Name Indication (SNI), afin de présenter le certificat correct du gestionnaire de files d'attente. Pour plus d'informations sur l'utilisation de plusieurs certificats sur un gestionnaire de files d'attente, voir «Comment IBM MQ fournit plusieurs fonctions de certificats», à la page 31.

Si un canal se connecte au gestionnaire de files d'attente de destination via IBM MQ Internet Pass-Thru (MQIPT) et que la route MQIPT a à la fois **`SSLServer`** et **`SSLClient`** définis, il existe deux sessions TLS distinctes entre les noeuds finaux. MQIPT peut être configuré pour permettre l'utilisation de plusieurs certificats par le gestionnaire de files d'attente de destination en définissant le SNI sur le nom de canal ou en passant par le SNI reçu sur la connexion entrante à la route. Pour plus d'informations sur la prise en charge des certificats multiples et MQIPT, voir Prise en charge de plusieurs certificats IBM MQ avec MQIPT.

Pour plus d'informations sur la connexion d'un gestionnaire de files d'attente à l'aide de l'authentification unidirectionnelle, c'est-à-dire lorsque le client TLS n'envoie pas de certificat, voir Connexion de deux gestionnaires de files d'attente à l'aide de l'authentification unidirectionnelle.

Systèmes Multiplatforms

Multi

Sous Multiplateformes, le serveur TLS envoie un certificat au client.

Pour les gestionnaires de files d'attente et les clients respectivement, une valeur non vide est recherchée dans l'ordre dans les sources suivantes. La première valeur non vide détermine le libellé de certificat. Le libellé de certificat doit exister dans le référentiel de clés. Si aucun certificat correspondant dans la casse et le format corrects ne correspond à un libellé, une erreur se produit et l'établissement de liaison TLS échoue.

Gestionnaires de files d'attente

1. Attribut de libellé de certificat de canal **CERTLABL**.
2. Attribut de label de certificat de gestionnaire de files d'attente **CERTLABL**.

3. Une valeur par défaut, au format `ibmwebspheremq` avec le nom du gestionnaire de files d'attente ajouté, toutes en minuscules. Par exemple, pour un gestionnaire de files d'attente nommé `QM1`, le libellé de certificat par défaut est `ibmwebspheremqqm1`.

IBM MQ clients

1. Attribut de libellé de certificat **CERTLABL** dans la définition de canal CLNTCONN.
2. Attribut **`CertificateLabel`** de la structure MQSCO.
3. Variable d'environnement **MQCERTLABL**.
4. Attribut `.ini` file (dans sa section SSL) **`CertificateLabel`** du client
5. Valeur par défaut, au format: `ibmwebspheremq` avec l'ID utilisateur que l'application client exécute comme ajouté, le tout en minuscules. Par exemple, pour un ID utilisateur USER1, le libellé de certificat par défaut est `ibmwebspheremquser1`.

z/OS systèmes

z/OS

IBM MQ Les clients ne sont pas pris en charge sous z/OS. Toutefois, un gestionnaire de files d'attente z/OS peut jouer le rôle de client TLS lors du lancement d'une connexion ou de serveur TLS lors de l'acceptation d'une demande de connexion. Les exigences de label de certificat pour les gestionnaires de files d'attente z/OS s'appliquent dans ces deux rôles et diffèrent des exigences sur Multiplateformes.

Pour les gestionnaires de files d'attente et les clients respectivement, une valeur non vide est recherchée dans l'ordre dans les sources suivantes. La première valeur non vide détermine le libellé de certificat. Le libellé de certificat doit exister dans le référentiel de clés. Si aucun certificat correspondant dans la casse et le format corrects ne correspond à un libellé, une erreur se produit et l'établissement de liaison TLS échoue.

1. Attribut de libellé de certificat de canal, **CERTLABL**.
2. S'il est partagé, l'attribut de label de certificat de groupe de partage de files d'attente, **CERTQSGL**.

   S'il n'est pas partagé, l'attribut de label de certificat du gestionnaire de files d'attente, **CERTLABL**.
3. Valeur par défaut, au format: `ibmWebSphereMQ` avec le nom du gestionnaire de files d'attente ou du groupe de partage de files d'attente ajouté. Notez que cette chaîne est sensible à la casse et doit être écrite comme indiqué. Par exemple, pour un gestionnaire de files d'attente nommé `QM1`, le libellé de certificat par défaut est `ibmWebSphereMQQM1`.
4. Si aucun certificat n'est trouvé avec le format de l'option «3», à la page 30, IBM MQ tente d'utiliser le certificat marqué comme certificat par défaut dans le fichier de clés.

Pour plus d'informations sur l'affichage du référentiel de clés, voir «Locating the key repository for a queue manager on z/OS», à la page 319.

Clients IBM MQ Java et IBM MQ JMS

Les clients IBM MQ Java et IBM MQ JMS utilisent les fonctions de leur fournisseur JSSE (Java Secure Socket Extension) pour sélectionner un certificat personnel lors de l'établissement de liaison TLS et ne sont donc pas soumis aux exigences de label de certificat.

Le comportement par défaut est que le client JSSE itère via les certificats du référentiel de clés, en sélectionnant le premier certificat personnel acceptable trouvé. Cependant, ce comportement n'est qu'une valeur par défaut et dépend de l'implémentation du fournisseur JSSE.

En outre, l'interface JSSE est hautement personnalisable via la configuration et l'accès direct lors de l'exécution par l'application. Pour plus de détails, consultez la documentation fournie par votre fournisseur JSSE.

Pour le traitement des incidents ou pour mieux comprendre l'établissement de liaison effectué par l'application client IBM MQ Java en association avec votre fournisseur JSSE spécifique, vous pouvez activer le débogage en définissant `javax.net.debug=ssl` dans l'environnement JVM.

Vous pouvez définir la variable dans l'application, via la configuration ou en entrant `-Djavax.net.debug=ssl` sur la ligne de commande.

Linux *Comment IBM MQ fournit plusieurs fonctions de certificats*

L'indication de nom de serveur (SNI) est une extension du protocole TLS qui permet à un client d'indiquer le service dont il a besoin. Dans la terminologie IBM MQ , cela correspond à un canal.

L'extension SNI est utilisée par IBM MQ pour permettre la spécification de plusieurs certificats sur différents canaux à l'aide du paramètre CERTLABL dans la définition de canal.

L'adresse SNI utilisée par IBM MQ est basée sur le nom de canal demandé, suivi d'un suffixe `.chl.mq.ibm.com`.

Les noms de canal IBM MQ sont mappés pour être des noms SNI valides comme suit:

- Les lettres majuscules A à Z sont pliées en minuscules
- Les chiffres 0 à 9 restent inchangés
- Tous les autres caractères, y compris les lettres minuscules a à z, sont convertis en code de caractères ASCII hexadécimal à deux chiffres (en minuscules), suivi d'un trait d'union.
  - Les lettres minuscules a à z sont mappées au format hexadécimal `61-` à `7a-` respectivement
  - Le pourcentage (%) est mappé à la valeur hexadécimale `25-`
  - Le trait d'union (-) est mappé à la valeur hexadécimale `2d-`
  - Le point (.) est mappé à la valeur hexadécimale `2e-`
  - La barre oblique (/) est mappée à la valeur hexadécimale `2f-`
  - Le trait de soulignement (_) est mappé à la valeur hexadécimale `5f-`

Sur les plateformes EBCDIC, le nom de canal est converti en ASCII avant l'application de ce mappage.

Par exemple, le nom de canal `TO.QMGR1` est mappé à une adresse SNI de `to2e-qmgr1.chl.mq.ibm.com`.

En revanche, le nom de canal en minuscules `to.qmgr1` est mappé à l'adresse SNI de `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com`.

Remarque : Dans les environnements où l'URL SNI générée doit être conforme aux spécifications de formatage d'URL, par exemple lorsqu'un client se connecte à un gestionnaire de files d'attente s'exécutant dans Red Hat® OpenShift® via une route Red Hat OpenShift , le nom de canal ne doit pas se terminer par une lettre minuscule.

La propriété **OutboundSNI** de la strophe SSL vous permet de choisir si le SNI doit être défini sur le nom de canal IBM MQ cible sur le système distant lors du lancement d'une connexion TLS ou sur le nom d'hôte. Pour plus d'informations sur la propriété **OutboundSNI**, voir Strophe SSL du fichier qm.ini et strophe SSL du fichier de configuration du client.

Plusieurs certificats nécessitent que l'interface SNI soit définie sur le nom de canal IBM MQ . Si un nom d'hôte, personnalisé ou sans SNI est utilisé pour se connecter à un canal IBM MQ avec un libellé de certificat configuré, l'application de connexion est rejetée avec une erreur MQRC_SSL_INITIALIZATION_ERROR et un message AMQ9673 est imprimé dans les journaux d'erreurs du gestionnaire de files d'attente éloignées.

Si un canal se connecte au gestionnaire de files d'attente de destination via IBM MQ Internet Pass-Thru (MQIPT), MQIPT doit être configuré pour définir le SNI sur le nom de canal ou pour transmettre le SNI reçu sur la connexion entrante à la route, afin de permettre l'utilisation de plusieurs certificats par le gestionnaire de files d'attente de destination. Pour plus d'informations sur la prise en charge des certificats multiples et MQIPT, voir Prise en charge de plusieurs certificats IBM MQ avec MQIPT.

Pour plus d'informations sur l'utilisation de cette propriété, voir Connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

Régénération du référentiel de clés du gestionnaire de files d'attente
Lorsque vous modifiez le contenu d'un référentiel de clés, les processus de gestionnaire de files d'attente existants ne prennent pas en compte le nouveau contenu tant qu'une commande REFRESH SECURITY TYPE (SSL) n'est pas émise ou que le gestionnaire de files d'attente n'est pas redémarré.

Pour plus d'informations sur la commande REFRESH SECURITY TYPE (SSL), voir REFRESH SECURITY.

Si le gestionnaire de files d'attente crée un nouveau processus de canal (à l'aide de `amqrmppa` ou **`runmqchl`**) après avoir modifié le contenu du magasin de clés, le nouveau processus démarre immédiatement à l'aide des nouveaux certificats, tandis que les processus existants continuent d'utiliser leur copie en cache du magasin de clés. Pour plus d'informations, voir «Lorsque les modifications apportées aux certificats ou au référentiel de clés prennent effet sur AIX, Linux, and Windows», à la page 315.

Notez que plusieurs canaux en cours d'exécution peuvent utiliser différentes versions du référentiel de clés jusqu'à ce que vous exécutiez une commande REFRESH SECURITY TYPE (SSL).

Vous pouvez également actualiser un référentiel de clés à l'aide des commandes PCF ou du IBM MQ Explorer. Pour plus d'informations, voir la commande MQCMD_REFRESH_SECURITY et la rubrique *Actualisation de la sécurité TLS* dans la section IBM MQ Explorer de la documentation de ce produit.

Concepts associés

«Actualisation de la vue d'un client du contenu du référentiel de clés SSL/TLS et des paramètres SSl/TLS», à la page 32
Pour mettre à jour l'application client avec le contenu actualisé du référentiel de clés, vous devez arrêter et redémarrer l'application client.

Actualisation de la vue d'un client du contenu du référentiel de clés SSL/TLS et des paramètres SSl/TLS
Pour mettre à jour l'application client avec le contenu actualisé du référentiel de clés, vous devez arrêter et redémarrer l'application client.

Vous ne pouvez pas actualiser la sécurité sur un client IBM MQ ; il n'existe pas d'équivalent de la commande REFRESH SECURITY TYPE (SSL) pour les clients (voir REFRESH SECURITY) pour plus d'informations.

Vous devez arrêter et redémarrer l'application, chaque fois que vous modifiez le certificat de sécurité, pour mettre à jour l'application client avec le contenu actualisé du référentiel de clés.

Si le redémarrage du canal actualise les configurations et que votre application possède une logique de reconnexion, vous pouvez actualiser la sécurité sur le client en exécutant la commande STOP CHL STATUS (INACTIVE).

Concepts associés

«Régénération du référentiel de clés du gestionnaire de files d'attente», à la page 32
Lorsque vous modifiez le contenu d'un référentiel de clés, les processus de gestionnaire de files d'attente existants ne prennent pas en compte le nouveau contenu tant qu'une commande REFRESH SECURITY TYPE (SSL) n'est pas émise ou que le gestionnaire de files d'attente n'est pas redémarré.

Protection par mot de passe MQCSP

Les données d'authentification qui sont spécifiées dans la structure MQCSP peuvent être protégées à l'aide de la fonction de protection par mot de passe IBM MQ MQCSP ou chiffrées à l'aide du chiffrement TLS.

Les applications IBM MQ client peuvent fournir un ID utilisateur et un mot de passe lorsqu'elles se connectent à un gestionnaire de files d'attente. V 9.4.0 Depuis IBM MQ 9.4.0, les applications peuvent également fournir un jeton d'authentification comme méthode d'authentification alternative. Ces données d'identification sont envoyées au gestionnaire de files d'attente dans une structure MQCSP.

Si le canal utilise le chiffrement TLS, les données d'identification dans le MQCSP sont chiffrées conformément à la spécification de chiffrement TLS. Si le canal n'utilise pas le chiffrement TLS, IBM MQ peut protéger ces données d'identification avant qu'elles ne soient envoyées sur le réseau, afin d'éviter

l'envoi de données d'identification sur un réseau en texte clair. La fonction IBM MQ qui protège ces données d'identification est appelée protection par mot de passe MQCSP.

Si la protection par mot de passe MQCSP est utilisée, les données suivantes de la structure MQCSP sont protégées:

- Le mot de passe, si la zone `MQCSP.AuthenticationType` est définie sur `MQCSP_AUTH_USER_ID_AND_PW`.
- V 9.4.0 Le jeton d'authentification, si la zone `MQCSP.AuthenticationType` est définie sur `MQCSP_AUTH_ID_TOKEN`.

Important : La protection par mot de passe MQCSP est utile à des fins de test et de développement car l'utilisation de la protection par mot de passe MQCSP est plus simple que la configuration du chiffrement TLS, mais pas sécurisée. A des fins de production, utilisez le chiffrement TLS de préférence à la protection par mot de passe IBM MQ , en particulier lorsque le réseau entre le client et le gestionnaire de files d'attente n'est pas sécurisé, car le chiffrement TLS est plus sécurisé.

Si vous vous souciez du chiffrement utilisé et de la protection qu'il offre, vous devez utiliser le chiffrement TLS complet. Avec TLS, les algorithmes sont connus du public et vous pouvez sélectionner celui qui convient à votre entreprise à l'aide de l'attribut de canal **SSLCIPH** .

Pour plus d'informations sur la structure MQCSP, voir Structure MQCSP.

Les données d'identification de la structure MQCSP sont protégées par la protection par mot de passe IBM MQ si toutes les conditions suivantes sont remplies:

- Les deux extrémités de la connexion utilisent IBM MQ 8.0ou une version ultérieure.
- Le canal n'utilise pas le chiffrement TLS. Un canal n'utilise pas le chiffrement TLS si le canal possède un attribut **SSLCIPH** vide ou si l'attribut **SSLCIPH** est défini sur une spécification de chiffrement qui ne fournit pas de chiffrement. Les chiffrements null, par exemple, NULL_SHA, ne fournissent pas de chiffrement.
- La zone `MQCSP.AuthenticationType` est définie sur `MQCSP_AUTH_USER_ID_AND_PWD` ou sur `MQCSP_AUTH_ID_TOKEN`. Pour plus d'informations sur la zone `MQCSP.AuthenticationType` , voir **`AuthenticationType`**.
- Si le client est IBM MQ Explorer et que le mode de compatibilité d'identification d'utilisateur n'est pas activé. Ce mode n'est pas le mode par défaut utilisé par IBM MQ Explorer pour envoyer un ID utilisateur et un mot de passe. Cette condition s'applique uniquement à IBM MQ Explorer.

Si l'une de ces conditions n'est pas remplie, les données d'identification ne sont pas protégées par la protection par mot de passe MQCSP. Si la valeur de l'attribut **`PasswordProtection`** interdit l'envoi de données d'identification en texte en clair et que le canal n'utilise pas le chiffrement TLS, la connexion échoue et un code anomalie MQRC_PASSWORD_PROTECTION_ERROR (2594) est renvoyé.

Paramètre de configuration `PasswordProtection`

L'attribut **`PasswordProtection`** dans la section **`Channels`** des fichiers de configuration du client et du gestionnaire de files d'attente peut empêcher l'envoi des données d'identification en texte en clair.

Remarque : Cet attribut est pertinent uniquement pour les connexions qui n'utilisent pas le chiffrement TLS. Les données d'identification sont chiffrées à l'aide de TLS au lieu d'être protégées par la protection par mot de passe MQCSP si la connexion utilise le chiffrement TLS.

L'attribut peut être défini sur l'une des valeurs suivantes. La valeur par défaut est `compatible`.

compatible
: Les données d'identification sont envoyées en texte clair si le gestionnaire de files d'attente ou le client exécute une version de IBM MQ antérieure à IBM MQ 8.0. En d'autres termes, les données d'identification peuvent être envoyées sur un réseau en texte en clair à des fins de compatibilité avec les versions de IBM MQ qui ne prennent pas en charge la protection par mot de passe MQCSP.

Les données d'identification sont protégées par la protection par mot de passe MQCSP si le gestionnaire de files d'attente et le client exécutent une version de IBM MQ dans IBM MQ 8.0 ou une version ultérieure.

La connexion échoue avant l'envoi des données d'identification si le gestionnaire de files d'attente et le client exécutent une version de IBM MQ à la IBM MQ 8.0 ou une version ultérieure et que la zone `MQCSP.AuthenticationType` n'est pas définie sur `MQCSP_AUTH_USER_ID_AND_PW` ou `MQCSP_AUTH_ID_TOKEN`.

toujours
Les données d'identification ne doivent pas être envoyées sur un réseau non protégé.

Les données d'identification sont protégées par la protection par mot de passe MQCSP si le gestionnaire de files d'attente et le client exécutent une version de IBM MQ dans IBM MQ 8.0 ou une version ultérieure.

La connexion échoue avant l'envoi des données d'identification dans les cas suivants:

- La zone `MQCSP.AuthenticationType` n'est pas définie sur `MQCSP_AUTH_USER_ID_AND_PW` ou `MQCSP_AUTH_ID_TOKEN`.
- Le gestionnaire de files d'attente ou le client exécute une version de IBM MQ antérieure à IBM MQ 8.0.

facultatif
Les données d'identification sont protégées par la protection par mot de passe MQCSP si le gestionnaire de files d'attente et le client exécutent une version de IBM MQ à IBM MQ 8.0 ou une version ultérieure, et si la zone `MQCSP.AuthenticationType` est définie sur `MQCSP_AUTH_USER_ID_AND_PW` ou `MQCSP_AUTH_ID_TOKEN`. Sinon, les données d'identification sont envoyées en texte en clair.

avertissement
Tout client est autorisé à envoyer des données d'identification en texte clair. Si des données d'identification en texte normal sont reçues, le message d'avertissement AMQ9297W est consigné dans les journaux des erreurs du gestionnaire de files d'attente.

Cette option ne peut être spécifiée que dans le fichier de configuration du gestionnaire de files d'attente.

Pour les clients Java et JMS , le comportement de l'attribut **PasswordProtection** varie selon que le client utilise le mode compatibilité ou le mode MQCSP:

- Si les clients Java et JMS fonctionnent en mode compatibilité, une structure MQCSP n'est pas utilisée pour envoyer l'ID utilisateur et le mot de passe lorsque le client se connecte. Par conséquent, le comportement de l'attribut **PasswordProtection** est identique à celui décrit pour les clients qui exécutent une version de IBM MQ antérieure à IBM MQ 8.0.
- Si les clients Java et JMS fonctionnent en mode MQCSP, le comportement de l'attribut **PasswordProtection** est le comportement décrit.

Pour plus d'informations sur l'authentification de connexion avec les clients Java et JMS , voir «Authentification de connexion avec le client Java», à la page 88.

Protection par mot de passe MQCSP et MQIPT

V 9.4.0

Si un client se connecte à un gestionnaire de files d'attente via IBM MQ Internet Pass-Thru (MQIPT), la route MQIPT peut être configurée pour ajouter ou supprimer le chiffrement TLS. Autrement dit, la route MQIPT peut être configurée avec `SSLServer=true` et `SSLClient=false`, ou `SSLServer=true` et `SSLClient=false`. Dans cette situation, le client et le gestionnaire de files d'attente peuvent ne pas convenir d'un algorithme de protection par mot de passe car une extrémité du canal utilise le chiffrement TLS et l'autre ne l'est pas. La connexion échoue alors avec le code anomalie MQRC_PASSWORD_PROTECTION_ERROR (2594).

Depuis la IBM MQ 9.4.0, MQIPT peut ajouter ou supprimer une protection pour les données d'identification dans les structures MQCSP, afin de préserver la compatibilité entre le client et le gestionnaire de files d'attente pour les routes MQIPT qui ajoutent ou suppriment le chiffrement TLS. La protection par mot de passe MQCSP dans MQIPT est configurée à l'aide de la propriété de route **PasswordProtection** .

La valeur par défaut de la propriété **PasswordProtection** est `required`. Cette valeur signifie que MQIPT peut ajouter, mais pas supprimer, la protection par mot de passe MQCSP. Les connexions à une route MQIPT qui ajoute le chiffrement TLS peuvent échouer avec le code anomalie MQRC_PASSWORD_PROTECTION_ERROR (2594) avec cette valeur **PasswordProtection**. Pour résoudre ce problème, définissez la valeur de la propriété **PasswordProtection** sur `compatible` dans la configuration de route MQIPT .

Pour plus d'informations sur la propriété **PasswordProtection** dans MQIPT, voir PasswordProtection.

Gestionnaire de certificats numériques (DCM)

Utilisez DCM pour gérer les certificats numériques et les clés privées sur IBM i.

Digital Certificate Manager (DCM) vous permet de gérer les certificats numériques et de les utiliser dans des applications sécurisées sur le serveur IBM i . Avec Digital Certificate Manager, vous pouvez demander et traiter des certificats numériques auprès d'autorités de certification ou d'autres tiers. Vous pouvez également agir en tant qu'autorité de certification locale pour créer et gérer des certificats numériques pour vos utilisateurs.

DCM prend également en charge l'utilisation de listes de révocation de certificat (CRL) pour fournir un processus de validation de certificat et d'application plus fort. Vous pouvez utiliser DCM pour définir l'emplacement où réside une CRL d'autorité de certification spécifique sur un serveur LDAP afin que IBM MQ puisse vérifier qu'un certificat spécifique n'a pas été révoqué.

DCM prend en charge et peut détecter automatiquement les certificats dans divers formats. Lorsque DCM détecte un certificat codé PKCS #12 ou un certificat PKCS #7 contenant des données chiffrées, il invite automatiquement l'utilisateur à entrer le mot de passe utilisé pour chiffrer le certificat. DCM n'invite pas les certificats PKCS #7 qui ne contiennent pas de données chiffrées.

DCM fournit une interface utilisateur basée sur un navigateur que vous pouvez utiliser pour gérer les certificats numériques de vos applications et de vos utilisateurs. L'interface utilisateur est divisée en deux cadres principaux: un cadre de navigation et un cadre de tâche.

Vous utilisez le cadre de navigation pour sélectionner les tâches de gestion des certificats ou les applications qui les utilisent. Certaines tâches individuelles sont affichées directement dans le cadre de navigation principal, mais la plupart des tâches du cadre de navigation sont organisées en catégories. Par exemple, Gérer les certificats est une catégorie de tâche qui contient diverses tâches guidées individuelles, telles que Afficher le certificat, Renouveler le certificat et Importer le certificat. Si un élément du cadre de navigation est une catégorie qui contient plusieurs tâches, une flèche s'affiche à sa gauche. La flèche indique que lorsque vous sélectionnez le lien de catégorie, une liste développée de tâches s'affiche, vous permettant de choisir la tâche à effectuer.

Pour obtenir des informations importantes sur DCM, consultez les publications IBM Redbooks suivantes:

- *IBM i Wired Network Security: OS/400 V5R1 DCM and Cryptographic Enhancements*, SG24-6168. Plus précisément, consultez les annexes pour obtenir des informations essentielles sur la configuration de votre système IBM i en tant qu'autorité de certification locale.
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*, SG24-5659. Voir plus particulièrement le chapitre 5. *Digital Certificate Manager for AS/400* , qui décrit AS/400 DCM.

FIPS (Federal Information Processing Standards)

Cette rubrique présente le programme FIPS (Federal Information Processing Standards) Cryptomodule Validation Program du US National Institute of Standards and Technology et les fonctions cryptographiques qui peuvent être utilisées sur les canaux TLS.

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut

Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

Ces informations s'appliquent aux plateformes suivantes:

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

ALW Pour plus d'informations sur la conformité FIPS 140-2 d'une connexion TLS IBM MQ sur AIX, Linux, and Windows, voir «FIPS (Federal Information Processing Standards) pour AIX, Linux, and Windows», à la page 36.

z/OS Pour plus d'informations sur la conformité FIPS 140-2 d'une connexion TLS IBM MQ sur z/OS, voir «Federal Information Processing Standards (FIPS) for z/OS», à la page 39.

Si du matériel de cryptographie est présent, les modules de cryptographie utilisés par IBM MQ peuvent être configurés pour être ceux fournis par le fabricant du matériel. Dans ce cas, la configuration est uniquement conforme à la norme FIPS si ces modules cryptographiques sont certifiés FIPS.

Au fil du temps, les normes fédérales de traitement de l'information sont mises à jour pour refléter les nouvelles attaques contre les algorithmes et les protocoles de chiffrement. Par exemple, certains CipherSpecs peuvent ne plus être certifiés FIPS. Lorsque de telles modifications se produisent, IBM MQ est également mis à jour pour implémenter la dernière norme. Vous pouvez alors constater des changements de comportement une fois la maintenance appliquée.

Concepts associés

«Comment indiquer que seuls les CipherSpecs certifiés FIPS sont utilisés lors de l'exécution sur MQI Client», à la page 276
Créez vos référentiels de clés à l'aide d'un logiciel compatible FIPS, puis indiquez que le canal doit utiliser des CipherSpecscertifiés FIPS.

Tâches associées

Activation de TLS dans IBM MQ classes for Java

Utilisation du protocole TLS (Transport Layer Security) avec IBM MQ classes for JMS

Référence associée

Propriétés TLS des objets JMS

«Commandes runmqakm et runmqktool sous AIX, Linux, and Windows», à la page 558
Sur les systèmes AIX, Linux, and Windows , utilisez les commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool) pour gérer les clés et les certificats.

«La norme FIPS (Federal Information Processing Standards)», à la page 24
Le gouvernement américain fournit des conseils techniques sur des systèmes informatiques et sur des systèmes de sécurité, y compris le chiffrement de données. Le National Institute for Standards and Technology (NIST) est un organisme important qui s'occupe des systèmes informatiques et de la sécurité. Il émet des recommandations et établit des normes, notamment la norme FIPS (Federal Information Processing Standards).

ALW *FIPS (Federal Information Processing Standards) pour AIX, Linux, and Windows*

Lorsque la cryptographie est requise sur un canal SSL/TLS sur des systèmes AIX, Linux, and Windows , IBM MQ utilise un package de cryptographie appelé IBM Crypto for C (ICC). Sur les plateformes AIX, Linux, and Windows , le logiciel ICC a transmis le programme de validation Cryptomodule FIPS (Federal Information Processing Standards) de l'Institut national des normes et de la technologie des Etats-Unis, au niveau 140-2.

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

La conformité FIPS 140-2 d'une connexion TLS IBM MQ sur les systèmes AIX, Linux, and Windows est la suivante:

- Pour tous les canaux de transmission de messages IBM MQ (à l'exception des types de canal CLNTCONN), la connexion est conforme à la norme FIPS si les conditions suivantes sont remplies:
  - La version IBM Global Security Kit (GSKit) ICC installée a été certifiée conforme à la norme FIPS 140-2 sur la version du système d'exploitation et l'architecture matérielle installées.
  - L'attribut SSLFIPS du gestionnaire de files d'attente a été défini sur YES.
  - Tous les référentiels de clés ont été créés et manipulés uniquement à l'aide de logiciels conformes à la norme FIPS, tels que **runmqakm** avec l'option `-fips` .
  - L'accès à tous les référentiels de clés est fourni à l'aide d'un fichier de dissimulation et non de l'attribut **KEYRPWD** du gestionnaire de files d'attente.
- Pour toutes les applications IBM MQ MQI client , la connexion utilise GSKit et est conforme à la norme FIPS si les conditions suivantes sont remplies:
  - La version GSKit ICC installée a été certifiée conforme à la norme FIPS 140-2 sur la version du système d'exploitation et l'architecture matérielle installées.
  - Vous avez indiqué que seule la cryptographie certifiée FIPS doit être utilisée, comme décrit dans la rubrique connexe pour le client MQI.
  - Tous les référentiels de clés ont été créés et manipulés uniquement à l'aide de logiciels conformes à la norme FIPS, tels que **runmqakm** avec l'option `-fips` .
  - L'accès à tous les référentiels de clés est fourni à l'aide d'un fichier de dissimulation et non du mécanisme de mot de passe du référentiel de clés.
- Pour les applications IBM MQ classes for Java utilisant le mode client, la connexion utilise les implémentations TLS de l'environnement d'exécution Java et est conforme à la norme FIPS si les conditions suivantes sont remplies:
  - L'environnement d'exécution Java utilisé pour exécuter l'application est conforme à la norme FIPS sur la version de système d'exploitation installée et l'architecture matérielle.
  - Vous avez spécifié que seule la cryptographie certifiée FIPS doit être utilisée, comme décrit dans la rubrique connexe du client Java .
  - Tous les référentiels de clés ont été créés et manipulés uniquement à l'aide de logiciels conformes à la norme FIPS, tels que **runmqakm** avec l'option `-fips` .
- Pour les applications IBM MQ classes for JMS utilisant le mode client, la connexion utilise les implémentations TLS de l'environnement d'exécution Java et est conforme à la norme FIPS si les conditions suivantes sont remplies:
  - L'environnement d'exécution Java utilisé pour exécuter l'application est conforme à la norme FIPS sur la version de système d'exploitation installée et l'architecture matérielle.
  - Vous avez spécifié que seule la cryptographie certifiée FIPS doit être utilisée, comme décrit dans la rubrique connexe du client JMS .
  - Tous les référentiels de clés ont été créés et manipulés uniquement à l'aide de logiciels conformes à la norme FIPS, tels que **runmqakm** avec l'option `-fips` .

- Pour les applications client .NET non gérées, la connexion utilise GSKit et est conforme à la norme FIPS si les conditions suivantes sont remplies:
  - La version GSKit ICC installée a été certifiée conforme à la norme FIPS 140-2 sur la version du système d'exploitation et l'architecture matérielle installées.
  - Vous avez spécifié que seule la cryptographie certifiée FIPS doit être utilisée, comme décrit dans la rubrique connexe du client .NET .
  - Tous les référentiels de clés ont été créés et manipulés uniquement à l'aide de logiciels conformes à la norme FIPS, tels que **runmqakm** avec l'option `-fips` .
  - L'accès à tous les référentiels de clés est fourni à l'aide d'un fichier de dissimulation et non du mécanisme de mot de passe du référentiel de clés.
- Pour les applications client XMS .NET non gérées, la connexion utilise GSKit et est conforme à la norme FIPS si les conditions suivantes sont remplies:
  - La version GSKit ICC installée a été certifiée conforme à la norme FIPS 140-2 sur la version du système d'exploitation et l'architecture matérielle installées.
  - Vous avez indiqué que seule la cryptographie certifiée FIPS doit être utilisée, comme décrit dans la documentation XMS .NET .
  - Tous les référentiels de clés ont été créés et manipulés uniquement à l'aide de logiciels conformes à la norme FIPS, tels que **runmqakm** avec l'option `-fips` .
  - L'accès à tous les référentiels de clés est fourni à l'aide d'un fichier de dissimulation et non du mécanisme de mot de passe du référentiel de clés.

Toutes les plateformes prises en charge sont certifiées FIPS 140-2 sauf comme indiqué dans le fichier Readme inclus avec chaque groupe de correctifs ou groupe de mises à jour.

Pour les connexions TLS utilisant GSKit, le composant certifié FIPS 140-2 est nommé *ICC*. Il s'agit de la version de ce composant qui détermine la conformité à la norme GSKit FIPS sur une plateforme donnée. Pour déterminer la version de ICC actuellement installée, exécutez la commande **dspmqver -p 64 -v** .

Voici un exemple d'extrait de la sortie **dspmqver -p 64 -v** relative à ICC:

```
icc
============
@ (#)CompanyName: IBM Corporation
@(#)LegalTrademarks: IBM
@ (#)FileDescription: IBM Crypto for C-language
@(#)FileVersion:   8.0.0.0
@ (#)LegalCopyright: Eléments sous licence-Propriété d' IBM
@ (#) ICC
@ (#) (C) Copyright IBM Corp. 2002, 2024.
@ (#) Tous droits réservés. Utilisateurs du gouvernement américain
@ (#) Droits restreints-Utilisation, duplication ou divulgation
@ (#) restreint par GSA ADP Schedule Contract avec IBM Corp.
@ (#)ProductName: icc_8.0 (générationGoldCoast ) 100415
@(#)ProductVersion:  8.0.0.0
@(#)ProductInfo:   10/04/15.03:32:19.10/04/15.18:41:51
@ (#) CMVCInfo:
```

L'instruction de certification NIST pour GSKit ICC 8 (incluse dans GSKit 8) est disponible à l'adresse suivante: Cryptographic Module Validation Program.

Si du matériel de cryptographie est présent, les modules de cryptographie utilisés par IBM MQ peuvent être configurés pour être ceux fournis par le fabricant du matériel. Dans ce cas, la configuration est uniquement conforme à la norme FIPS si ces modules cryptographiques sont certifiés FIPS.

Restrictions DES triple imposées lors d'une opération conforme à la norme FIPS 140-2

Lorsque IBM MQ est configuré pour fonctionner conformément à la norme FIPS 140-2, des restrictions supplémentaires sont appliquées en ce qui concerne Triple DES (3DES) CipherSpecs. Ces restrictions permettent la conformité à la recommandation US NIST SP800-67 .

1. Toutes les parties de la clé Triple DES doivent être uniques.

2. Aucune partie de la clé Triple DES ne peut être une clé Weak, Semi-Weak ou Possiblement-Weak selon les définitions de la norme NIST SP800-67.
3. Vous ne pouvez pas transmettre plus de 32 Go de données via la connexion avant qu'une réinitialisation de clé secrète ne soit nécessaire. Par défaut, IBM MQ ne réinitialise pas la clé de session secrète. Cette réinitialisation doit donc être configurée. L'échec de l'activation de la réinitialisation de la clé secrète lors de l'utilisation d'un CipherSpec Triple DES et de la conformité à la norme FIPS 140-2 entraîne la fermeture de la connexion avec l'erreur AMQ9288 après le dépassement du nombre maximal d'octets. Pour plus d'informations sur la configuration de la réinitialisation des clés secrètes, voir «Réinitialisation des clés secrètes SSL et TLS», à la page 481.

IBM MQ génère des clés de session Triple DES qui sont déjà conformes aux règles 1 et 2. Toutefois, pour satisfaire à la troisième restriction, vous devez activer la réinitialisation de la clé secrète lors de l'utilisation de CipherSpecs Triple DES dans une configuration FIPS 140-2. Vous pouvez également éviter d'utiliser Triple DES.

Concepts associés

«Comment indiquer que seuls les CipherSpecs certifiés FIPS sont utilisés lors de l'exécution sur MQI Client», à la page 276
Créez vos référentiels de clés à l'aide d'un logiciel compatible FIPS, puis indiquez que le canal doit utiliser des CipherSpecscertifiés FIPS.

Tâches associées

Activation de TLS dans IBM MQ classes for Java

Utilisation du protocole TLS (Transport Layer Security) avec IBM MQ classes for JMS

Référence associée

Propriétés TLS des objets JMS

«Commandes runmqakm et runmqktool sous AIX, Linux, and Windows», à la page 558
Sur les systèmes AIX, Linux, and Windows , utilisez les commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool) pour gérer les clés et les certificats.

«La norme FIPS (Federal Information Processing Standards)», à la page 24
Le gouvernement américain fournit des conseils techniques sur des systèmes informatiques et sur des systèmes de sécurité, y compris le chiffrement de données. Le National Institute for Standards and Technology (NIST) est un organisme important qui s'occupe des systèmes informatiques et de la sécurité. Il émet des recommandations et établit des normes, notamment la norme FIPS (Federal Information Processing Standards).

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

"La norme FIPS (Federal Information Processing Standards)" on page 24
Le gouvernement américain fournit des conseils techniques sur des systèmes informatiques et sur des systèmes de sécurité, y compris le chiffrement de données. Le National Institute for Standards and Technology (NIST) est un organisme important qui s'occupe des systèmes informatiques et de la sécurité. Il émet des recommandations et établit des normes, notamment la norme FIPS (Federal Information Processing Standards).

Multi

Vérification de la configuration TLS de votre gestionnaire de files d'attente avec `mqcertck`

La commande **MQCERTCK** est un outil qui permet de rechercher les erreurs courantes dans la configuration TLS de votre gestionnaire de files d'attente et fournit des suggestions pour la résolution des problèmes.

Introduction

La commande **`mqcertck`** vérifie:

- Existence et droits du référentiel de clés du gestionnaire de files d'attente, référencé dans l'attribut **SSLKEYR** du gestionnaire de files d'attente.
- Existence et validité du certificat du gestionnaire de files d'attente, référencé dans l'attribut **CERTLABL** du gestionnaire de files d'attente.
- Existence et validité des certificats référencés dans les attributs **CERTLABL** du canal TLS activé.
- Le référentiel de clés et les certificats des applications client, y compris la vérification des certificats, sont autorisés avec le gestionnaire de files d'attente.

Remarque : La commande **mqcertck** n'est pas disponible sous z/OS ou IBM i.

Utilisation

Pour utiliser la commande **mqcertck** , exécutez la commande `mqcertck`, ainsi que ses paramètres obligatoires et tous les paramètres facultatifs dont vous avez besoin, à partir d'une ligne de commande.

Voir mqcertck pour une description de la commande et des paramètres utilisés par la commande.

Exemple

Vous venez de terminer la configuration de votre gestionnaire de files d'attente `QM1` pour autoriser les connexions TLS des clients se connectant au canal SVRCONN de votre gestionnaire de files d'attente.

Vous utilisez la fonction de certificats multiples et, par conséquent, votre gestionnaire de files d'attente et votre canal ont un libellé de certificat spécifié dans leurs attributs **CERTLABL** . Lors de la création du canal, vous avez fait une erreur dans l'attribut **CERTLABL** du canal. Par conséquent, lorsqu'un client tente de se connecter, le gestionnaire de files d'attente renvoie le code retour 2393 MQRC_SSL_INITIALIZATION_ERROR.

Avant d'activer le gestionnaire de files d'attente, utilisez la commande **mqcertck** pour vérifier la configuration TLS du gestionnaire de files d'attente.

Vous exécutez la commande `mqcertck QM1` et recevez la sortie suivante:

```
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.
|
+------------------------------------------------------------
```

Cette sortie vous invite à vérifier votre définition de canal pour le canal de connexion serveur MQCERTCK.CHANNEL. Ici, vous voyez l'erreur que vous avez faite et pouvez la corriger avant d'exécuter à nouveau la commande `mqcertck` pour vérifier que vous avez résolu le problème.

Vérification des connexions client

La commande **mqcertck** permet de vérifier les référentiels de clés client, ainsi que la configuration TLS du gestionnaire de files d'attente. Pour ce faire, **mqcertck** doit pouvoir accéder au référentiel de clés du client à partir de la machine exécutant le gestionnaire de files d'attente.

Lors de l'exécution de la commande **mqcertck** , si vous fournissez le paramètre **-clientkeyr** avec l'emplacement du référentiel de clés client (à l'exclusion de l'extension) **mqcertck** , ce référentiel de clés est vérifié par rapport au gestionnaire de files d'attente.

Si vous savez quel canal le client utilisera pour se connecter au gestionnaire de files d'attente, vous pouvez le spécifier avec l'indicateur **-clientchannel** .

Si le client utilise l'authentification mutuelle pour se connecter au gestionnaire de files d'attente, vous pouvez utiliser le paramètre **-clientusername** ou **-clientlabel** pour indiquer à la commande **mqcertck** le certificat à utiliser dans le référentiel de clés du client.

Si vous utilisez le certificat par défaut et que vous ne fournissez pas de libellé de certificat à l'application client, vous pouvez utiliser **-clientusername** et les paramètres **username** qui exécutent cette application.

Lors de l'exécution de la commande **mqcertck** , la commande génère le libellé de certificat `ibmwebspheremqXXXX` , où XXXX est la valeur transmise dans le paramètre **-clientusername** .

Afin de vérifier complètement le référentiel de clés du client, la commande **mqcertck** crée une connexion factice à l'aide de IBM Global Security Kit (GSKit). Pour ce faire, la commande doit disposer d'un port disponible auquel elle peut se connecter lors de ses tests client. Le port par défaut utilisé est 5857. Toutefois, s'il est déjà utilisé, vous pouvez spécifier un autre port à utiliser lors des tests client.

Remarque : Bien que la commande **mqcertck** se lie à un port, aucune communication externe n'est utilisée par **mqcertck**et tous les tests sont effectués localement.

SSL/TLS sur IBM MQ MQI client

IBM MQ prend en charge TLS sur les clients. Vous pouvez personnaliser l'utilisation de TLS de différentes manières.

IBM MQ fournit une prise en charge TLS pour IBM MQ MQI clients sur les systèmes AIX, Linux, and Windows . Si vous utilisez IBM MQ classes for Java, voir Utilisation de IBM MQ classes for Java et si vous utilisez IBM MQ classes for JMS, voir Utilisation de IBM MQ classes for JMS. Le reste de cette section ne s'applique pas aux environnements Java ou JMS .

Vous pouvez spécifier le référentiel de clés pour un IBM MQ MQI client avec la valeur MQSSLKEYR dans votre fichier de configuration client IBM MQ ou lorsque votre application effectue un appel MQCONNX. Vous disposez de trois options pour spécifier qu'un canal utilise TLS:

- Utilisation d'une table de définition de canal
- Utilisation de la structure des options de configuration SSL, MQSCO, sur un appel MQCONNX
- Utilisation d' Active Directory (sur les systèmes Windows)

Vous ne pouvez pas utiliser la variable d'environnement MQSERVER pour indiquer qu'un canal utilise TLS.

Vous pouvez continuer à exécuter vos applications IBM MQ MQI client existantes sans TLS, tant que TLS n'est pas spécifié à l'autre extrémité du canal.

Si des modifications sont apportées sur une machine client au contenu du référentiel de clés TLS, à l'emplacement du référentiel de clés TLS, aux informations d'authentification ou aux paramètres matériels de cryptographie, vous devez arrêter toutes les connexions TLS afin de refléter ces modifications dans les canaux de connexion client utilisés par l'application pour se connecter au gestionnaire de files d'attente. Une fois toutes les connexions terminées, redémarrez les canaux TLS. Tous les nouveaux paramètres TLS sont utilisés. Ces paramètres sont analogues à ceux actualisés par la commande REFRESH SECURITY TYPE (SSL) sur les systèmes de gestionnaire de files d'attente.

Lorsque votre IBM MQ MQI client s'exécute sur un système AIX, Linux, and Windows avec du matériel de cryptographie, vous configurez ce matériel avec la variable d'environnement MQSSLCRYP. Cette variable est équivalente au paramètre SSLCRYP de la commande ALTER QMGR MQSC. Pour obtenir une description du paramètre SSLCRYP dans la commande ALTER QMGR MQSC, voir ALTER QMGR . Si vous utilisez la version GSK_PCS11 du paramètre SSLCRYP, le libellé de jeton PKCS #11 doit être indiqué en minuscules.

La réinitialisation de la clé secrète TLS et la norme FIPS sont prises en charge sur IBM MQ MQI clients. Pour plus d'informations, voir «Réinitialisation des clés secrètes SSL et TLS», à la page 481 et «FIPS (Federal Information Processing Standards) pour AIX, Linux, and Windows», à la page 36.

Voir «Configuration de la sécurité IBM MQ MQI client», à la page 275 pour plus d'informations sur la prise en charge de TLS pour IBM MQ MQI clients.

Tâches associées

fichier de configuration IBM MQ MQI client , `mqclient.ini`

Spécification du fait qu'un canal MQI utilise SSL/TLS

Pour qu'un canal MQI utilise TLS, la valeur de l'attribut *SSLCipherSpec* du canal de connexion client doit être le nom d'un CipherSpec pris en charge par IBM MQ sur la plateforme client.

Vous pouvez définir un canal de connexion client avec une valeur pour cet attribut de l'une des manières suivantes. Ils sont répertoriés par ordre de priorité décroissante.

1. Lorsqu'un exit PreConnect fournit une structure de définition de canal à utiliser.

   Un exit PreConnect peut fournir le nom d'un CipherSpec dans la zone *SSLCipherSpec* d'une structure de définition de canal, MQCD. Cette structure est renvoyée dans la zone **ppMQCDArrayPtr** de la structure de paramètres d'exit MQNXP utilisée par l'exit PreConnect .
2. Lorsqu'une application IBM MQ MQI client émet un appel MQCONNX.

   L'application peut spécifier le nom d'un CipherSpec dans la zone *SSLCipherSpec* d'une structure de définition de canal, MQCD.Cette structure est référencée par la structure d'options de connexion, MQCNO, qui est un paramètre de l'appel MQCONNX.
3. Utilisation d'une table de définition de canal du client (CCDT).

   Une ou plusieurs entrées d'une table de définition de canal du client peuvent spécifier le nom d'un CipherSpec.Par exemple, si vous créez une entrée à l'aide de la commande MQSC DEFINE CHANNEL, vous pouvez utiliser le paramètre SSLCIPH dans la commande pour spécifier le nom d'un CipherSpec.
4. Utilisation de Active Directory sous Windows.

   Sur les systèmes Windows , vous pouvez utiliser la commande de contrôle **setmqscp** pour publier les définitions de canal de connexion client dans Active Directory. Une ou plusieurs de ces définitions peuvent spécifier le nom d'un CipherSpec.

Par exemple, si une application client fournit une définition de canal de connexion client dans une structure MQCD sur un appel MQCONNX, cette définition est utilisée de préférence aux entrées d'une table de définition de canal du client accessibles par le client IBM MQ .

Vous ne pouvez pas utiliser la variable d'environnement MQSERVER pour fournir la définition de canal à l'extrémité client d'un canal MQI qui utilise TLS.

Pour vérifier si un certificat client a transité, affichez le statut du canal à l'extrémité serveur d'un canal pour la présence d'une valeur de paramètre de nom d'homologue.

Concepts associés

«Spécification d'un CipherSpec pour un IBM MQ MQI client», à la page 458
Vous disposez de trois options pour spécifier un CipherSpec pour un IBM MQ MQI client.

CipherSpecs et CipherSuites dans IBM MQ

IBM MQ prend en charge TLS1.3 et TLS 1.2 CipherSpecs, ainsi que les algorithmes RSA et Diffie-Hellman. Toutefois, vous pouvez activer les CipherSpecsobsolètes, si vous devez le faire.

Voir «Activation des CipherSpecs», à la page 435 pour plus d'informations sur:

- CipherSpecs pris en charge par IBM MQ.
- Comment activer les CipherSpecs SSL 3.0 et TLS 1.0 CipherSpecs.

IBM MQ prend en charge les algorithmes d'authentification et d'échange de clés RSA et Diffie-Hellman. La taille de la clé utilisée lors de l'établissement de liaison TLS peut dépendre du certificat numérique que vous utilisez, mais certains CipherSpecs incluent une spécification de la taille de la clé d'établissement de liaison. Plus la taille de clé est élevée, plus l'authentification est solide. Avec des tailles de clé plus petites, l'établissement de la liaison est plus rapide.

Concepts associés

«CipherSpecs et CipherSuites», à la page 22
Les protocoles de sécurité cryptographique doivent convenir des algorithmes utilisés par une connexion sécurisée. CipherSpecs et CipherSuites définissent des combinaisons spécifiques d'algorithmes.

NSA Suite B Cryptography dans IBM MQ

Cette rubrique fournit des informations sur la manière de configurer IBM MQ for AIX, Linux, and Windows pour qu'il soit conforme au profil TLS 1.2 conforme à la suite B.

Au fil du temps, la norme NSA Cryptography Suite B est mise à jour pour refléter les nouvelles attaques contre les algorithmes et les protocoles de chiffrement. Par exemple, certains CipherSpecs peuvent ne plus être certifiés Suite B. Lorsque de telles modifications se produisent, IBM MQ est également mis à jour pour implémenter la dernière norme. Vous pouvez alors constater des changements de comportement une fois la maintenance appliquée. Le fichier Readme IBM MQ répertorie la version de Suite B appliquée par chaque niveau de maintenance du produit. Si vous configurez IBM MQ pour appliquer la conformité Suite B, consultez toujours le fichier Readme lors de la planification de l'application de la maintenance. Voir Fichiers Readme des produits IBM MQ, WebSphere MQet MQSeries.

Sur les systèmes AIX, Linux, and Windows , IBM MQ peut être configuré pour se conformer au profil TLS 1.2 compatible Suite B aux niveaux de sécurité indiqués dans le tableau 1.

Tableau 3. Niveaux de sécurité Suite B avec CipherSpecs et algorithmes de signature numérique autorisés

Niveau de sécurité	**CipherSpecs autorisés**	**Algorithmes de signature numérique autorisés**
128 bits	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA avec SHA-256 ECDSA avec SHA-384
192 bits	ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA avec SHA-384
Les deux [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA avec SHA-256 ECDSA avec SHA-384

1. Il est possible de configurer simultanément les niveaux de sécurité 128 bits et 192 bits. Etant donné que la configuration Suite B détermine les algorithmes de cryptographie minimaux acceptables, la configuration des deux niveaux de sécurité est équivalente à la configuration du niveau de sécurité 128 bits uniquement. Les algorithmes de cryptographie du niveau de sécurité 192 bits sont plus forts que le minimum requis pour le niveau de sécurité 128 bits, de sorte qu'ils sont autorisés pour le niveau de sécurité 128 bits même si le niveau de sécurité 192 bits n'est pas activé.

Remarque : Les conventions de dénomination utilisées pour le niveau de sécurité ne représentent pas nécessairement la taille de courbe elliptique ou la taille de clé de l'algorithme de chiffrement AES.

CipherSpec -conformation vers Suite B

Bien que le comportement par défaut de IBM MQ ne soit pas conforme à la norme Suite B, IBM MQ peut être configuré pour être conforme à l'un des niveaux de sécurité ou aux deux sur les systèmes AIX, Linux, and Windows . Suite à la configuration réussie de IBM MQ pour utiliser Suite B, toute tentative de démarrage d'un canal sortant à l'aide d'un CipherSpec non conforme à Suite B entraîne l'erreur AMQ9282. Cette activité a également pour conséquence que le client MQI renvoie le code anomalie MQRC_CIPHER_SPEC_NOT_SUITE_B. De même, la tentative de démarrage d'un canal entrant à l'aide d'un CipherSpec non conforme à la configuration Suite B entraîne l'erreur AMQ9616.

Pour plus d'informations sur les CipherSpecs IBM MQ CipherSpecs, voir «Activation des CipherSpecs», à la page 435

Suite B et certificats numériques

La suite B restreint les algorithmes de signature numérique qui peuvent être utilisés pour signer des certificats numériques. La suite B restreint également le type de clé publique que les certificats peuvent contenir. Par conséquent, IBM MQ doit être configuré pour utiliser des certificats dont l'algorithme de signature numérique et le type de clé publique sont autorisés par le niveau de sécurité Suite B configuré du partenaire distant. Les certificats numériques qui ne sont pas conformes aux exigences de niveau de sécurité sont rejetés et la connexion échoue avec l'erreur AMQ9633 ou AMQ9285.

Pour le niveau de sécurité Suite B 128 bits, la clé publique du sujet de certificat doit utiliser la courbe elliptique NIST P-256 ou la courbe elliptique NIST P-384 et être signée avec la courbe elliptique NIST P-256 ou la courbe elliptique NIST P-384 . Au niveau de la sécurité Suite B 192 bits, la clé publique du sujet de certificat est requise pour utiliser la courbe elliptique NIST P-384 et pour être signée avec la courbe elliptique NIST P-384 .

Pour obtenir un certificat adapté à une opération conforme à la suite B, utilisez la commande **runmqakm** et spécifiez le paramètre **-sig_alg** pour demander un algorithme de signature numérique approprié. Les valeurs des paramètres EC_ecdsa_with_SHA256 et EC_ecdsa_with_SHA384 **-sig_alg** correspondent à des clés de courbe elliptique signées par les algorithmes de signature numérique Suite B autorisés.

Pour plus d'informations sur la commande **runmqakm**, voir «Gestion des clés et des certificats sur AIX, Linux, and Windows», à la page 557.

Création et demande de certificats numériques

Pour créer un certificat numérique autosigné pour les tests Suite B, voir «Création d'un certificat personnel autosigné sur AIX, Linux, and Windows», à la page 559

Pour demander un certificat numérique signé par une autorité de certification pour une utilisation en production Suite B, voir «Demande d'un certificat personnel sur AIX, Linux, and Windows», à la page 561.

Remarque : L'autorité de certification utilisée doit générer des certificats numériques qui répondent aux exigences décrites dans le document IETF RFC 6460.

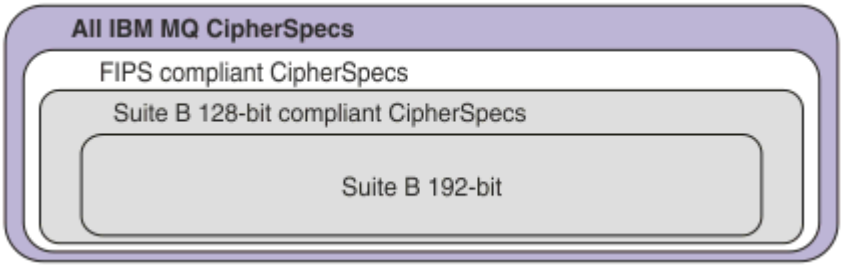
FIPS 140-2 et Suite B

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

La norme Suite B est conceptuellement similaire à la norme FIPS 140-2, car elle restreint l'ensemble des algorithmes de cryptographie activés afin de fournir un niveau de sécurité assuré. Les CipherSpecs Suite B actuellement pris en charge peuvent être utilisés lorsque IBM MQ est configuré pour une opération conforme à la norme FIPS 140-2. Il est donc possible de configurer IBM MQ pour la conformité FIPS et Suite B simultanément, auquel cas les deux ensembles de restrictions s'appliquent.

Le diagramme suivant illustre la relation entre ces sous-ensembles:



Configuration de IBM MQ pour une opération compatible Suite B

Pour plus d'informations sur la configuration de IBM MQ sur AIX, Linux, and Windows pour une opération compatible avec Suite B, voir «Configuration de IBM MQ pour Suite B», à la page 46.

IBM MQ ne prend pas en charge les opérations conformes à la suite B sur les plateformes et clients suivants:

- Plateforme IBM i
- Plateforme z/OS
- Java client
- JMS client

Concepts associés

«Comment indiquer que seuls les CipherSpecs certifiés FIPS sont utilisés lors de l'exécution sur MQI Client», à la page 276
Créez vos référentiels de clés à l'aide d'un logiciel compatible FIPS, puis indiquez que le canal doit utiliser des CipherSpecscertifiés FIPS.

ALW *Configuration de IBM MQ pour Suite B*

IBM MQ peut être configuré pour fonctionner conformément à la norme NSA Suite B sur les plateformes AIX, Linux, and Windows .

La suite B restreint l'ensemble des algorithmes de cryptographie activés afin de fournir un niveau de sécurité assuré. IBM MQ peut être configuré pour fonctionner conformément à la Suite B afin de fournir un niveau de sécurité amélioré. Pour plus d'informations sur la suite B, voir «Agence de sécurité nationale (NSA) Suite B Cryptographie», à la page 24. Pour plus d'informations sur la configuration de la suite B et son effet sur les canaux TLS, voir «NSA Suite B Cryptography dans IBM MQ», à la page 44.

Gestionnaire de files d'attente

Pour un gestionnaire de files d'attente, utilisez la commande **ALTER QMGR** avec le paramètre **SUITEB** pour définir les valeurs appropriées à votre niveau de sécurité requis. Pour plus d'informations, voir ALTER QMGR.

Vous pouvez également utiliser la commande PCF **MQCMD_CHANGE_Q_MGR** avec le paramètre **MQIA_SUITE_B_STRENGTH** pour configurer le gestionnaire de files d'attente pour une opération compatible avec Suite B.

Remarque : Si vous modifiez les paramètres Suite B d'un gestionnaire de files d'attente, vous devez redémarrer le service MQXR pour que ces paramètres prennent effet.

MQI Client

Par défaut, les clients MQI n'appliquent pas la conformité Suite B. Vous pouvez activer le client MQI pour la conformité Suite B en exécutant l'une des options suivantes:

1. En définissant la zone EncryptionPolicySuiteB dans la structure MQSCO d'un appel MQCONNX sur une ou plusieurs des valeurs suivantes:
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

   L'utilisation de MQ_SUITE_B_NONE avec une autre valeur n'est pas valide.

   Pour plus d'informations sur la structure MQSCO, voir MQSCO-Options de configuration SSL.
2. En définissant la variable d'environnement **MQSUITEB** sur une ou plusieurs des valeurs suivantes:
   - Aucun
   - 128_BIT
   - 192_BIT

   Vous pouvez spécifier plusieurs valeurs à l'aide d'une liste séparée par des virgules. L'utilisation de la valeur NONE avec une autre valeur n'est pas valide.
3. En définissant l'attribut **EncryptionPolicySuiteB** dans la strophe SSL du fichier de configuration du client sur une ou plusieurs des valeurs suivantes:
   - Aucun
   - 128_BIT
   - 192_BIT

   Vous pouvez spécifier plusieurs valeurs à l'aide d'une liste séparée par des virgules. L'utilisation de NONE avec une autre valeur n'est pas valide.

Remarque : Les paramètres du client MQI sont répertoriés par ordre de priorité. La structure MSCO de l'appel MQCONNX remplace le paramètre de la variable d'environnement **MQSUITEB** , qui remplace l'attribut dans la strophe SSL.

.NET

Pour les clients .NET non gérés, la propriété **MQC.ENCRYPTION_POLICY_SUITE_B** indique le type de sécurité Suite B requis.

Pour plus d'informations sur l'utilisation de la suite B dans IBM MQ classes for .NET, voir Classe MQEnvironment .NET.

AMQP

Les paramètres d'attribut Suite B d'un gestionnaire de files d'attente s'appliquent aux canaux AMQP de ce gestionnaire de files d'attente. Si vous modifiez les paramètres de la suite de gestionnaires de files d'attente B, vous devez redémarrer le service AMQP pour que les modifications soient prises en compte.

Règles de validation de certificat dans IBM MQ

La règle de validation de certificat détermine dans quelle mesure la validation de la chaîne de certificats est conforme aux normes de sécurité de l'industrie.

La règle de validation de certificat dépend de la plateforme et de l'environnement comme suit:

- Pour les applications Java et JMS sur toutes les plateformes, la règle de validation de certificat dépend du composant JSSE de l'environnement d'exécution Java . Pour plus d'informations sur les règles de validation de certificat, voir la documentation de votre environnement d'exécution Java.
- ALW Pour les systèmes AIX, Linux, and Windows , la règle de validation de certificat est fournie par IBM Global Security Kit (GSKit) et peut être configurée. V 9.4.0 V 9.4.0 Trois règles de validation de certificat différentes sont prises en charge:

- Une règle de validation de certificat existante, utilisée pour une compatibilité et une interopérabilité maximales en amont avec les anciens certificats numériques qui ne sont pas conformes aux normes de validation de certificat IETF actuelles. Cette règle est appelée règle de base.
- Une stratégie de validation de certificat stricte et conforme aux normes qui applique la norme RFC 5280. Cette règle est connue sous le nom de règle standard.
- V 9.4.0 V 9.4.0 Règle de validation de certificat qui n'authentifie pas le certificat de serveur TLS, disponible uniquement pour les applications client.

- IBM i Pour les systèmes IBM i , la règle de validation de certificat dépend de la bibliothèque de sockets sécurisés fournie par le système d'exploitation. Pour plus d'informations sur les règles de validation de certificat, voir la documentation du système d'exploitation.
- z/OS Pour les systèmes z/OS , la règle de validation de certificat dépend du composant System SSL fourni par le système d'exploitation. Pour plus d'informations sur les règles de validation de certificat, voir la documentation du système d'exploitation.

Pour plus d'informations sur la configuration de la règle de validation de certificat, voir «Configuration des règles de validation de certificat dans IBM MQ», à la page 48. Pour plus d'informations sur les différences entre les règles de validation de certificat de base et standard, voir Certificate validation and trust policy design on AIX, Linux, and Windows.

Configuration des règles de validation de certificat dans IBM MQ

Il existe plusieurs manières de spécifier la règle de validation de certificat TLS à utiliser pour valider les certificats numériques reçus des systèmes partenaires distants.

Pourquoi et quand exécuter cette tâche

La règle de validation de certificat détermine dans quelle mesure la validation de la chaîne de certificats est conforme aux normes de sécurité de l'industrie. La règle de validation de certificat dépend de la plateforme et de l'environnement. Pour plus d'informations sur les règles de validation de certificat, voir «Règles de validation de certificat dans IBM MQ», à la page 47.

Procédure

- Pour définir la règle de validation de certificat sur le gestionnaire de files d'attente, utilisez l'attribut de gestionnaire de files d'attente **CERTVPOL**.

  Pour plus d'informations sur la définition de cet attribut, voir ALTER QMGR (alter queue manager settings).
- Pour définir la règle de validation de certificat sur le client, utilisez les méthodes suivantes.

  Si plusieurs méthodes sont utilisées pour définir la règle, le client utilise les paramètres dans l'ordre de priorité suivant:

  1. Utilisez la zone `CertificateValPolicy` dans la structure MQSCO du client. Définissez la zone sur l'une des valeurs suivantes:

     MQ_CERT_VAL_POLICY_ANY
     Appliquez chacune des règles de validation de certificat prises en charge par la bibliothèque de sockets sécurisés. Acceptez la chaîne de certificats si l'une des politiques considère que la chaîne de certificats est valide.

     MQ_CERT_VAL_POLICY_RFC5280
     Appliquez uniquement la règle de validation de certificat RFC5280 . Ce paramètre fournit une validation plus stricte que le paramètre ANY, mais rejette certains certificats numériques plus anciens.

     V 9.4.0 V 9.4.0 **MQ_CERT_VAL_POLICY_NONE**
     N'appliquez aucune règle de validation de certificat. Ce paramètre est destiné aux applications client uniquement et accepte le certificat du serveur TLS sans valider la chaîne de confiance.

Pour plus d'informations sur l'utilisation de cette zone, voir MQSCO-Options de configuration SSL.

2. Utilisez la variable d'environnement client **MQCERTVPOL**. Pour définir cette variable d'environnement, utilisez l'une des commandes suivantes:
   - Linux AIX Pour les systèmes AIX and Linux :

     ```
     export MQCERTVPOL= value
     ```

   - Windows Pour les systèmes Windows :

     ```
     SET MQCERTVPOL= value
     ```

   - IBM i Pour les systèmes IBM i :

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

3. Utilisez l'attribut **CertificateValPolicy** de la strophe SSL dans le fichier de configuration du client. Définissez cet attribut sur l'une des valeurs suivantes:

   ANY
   Utilisez les règles de validation de certificat prises en charge par la bibliothèque de sockets sécurisés sous-jacente. Il s'agit du paramètre par défaut.

   RFC5280
   Utilisez uniquement la validation de certificat qui est conforme à la norme RFC 5280.

   V 9.4.0 V 9.4.0 **Aucun**
   N'appliquez aucune règle de validation de certificat. Ce paramètre accepte le certificat du serveur TLS sans valider la chaîne de confiance.

   Pour plus d'informations sur l'utilisation de cet attribut, voir Strophe SSL du fichier de configuration du client.

Certificats numériques et compatibilité CipherSpec dans IBM MQ

Cette rubrique fournit des informations sur la façon de choisir les CipherSpecs et les certificats numériques appropriés pour votre règle de sécurité, en soulignant la relation entre les CipherSpecs et les certificats numériques dans IBM MQ.

Seul un sous-ensemble des CipherSpecs pris en charge peut être utilisé avec tous les types de certificat numérique pris en charge. Il est donc nécessaire de choisir un CipherSpec approprié pour votre certificat numérique. De même, si la stratégie de sécurité de votre organisation requiert que vous utilisiez un CipherSpec particulier, vous devez obtenir un certificat numérique approprié pour ce CipherSpec.

L'algorithme de signature numérique MD5 et TLS 1.2

Les certificats numériques signés à l'aide de l'algorithme MD5 sont rejetés lorsque le protocole TLS 1.2 est utilisé. En effet, l'algorithme MD5 est désormais considéré comme faible par de nombreux analystes cryptographiques et son utilisation est généralement déconseillée. Pour utiliser des CipherSpecs plus récents basés sur le protocole TLS 1.2 , assurez-vous que les certificats numériques n'utilisent pas l'algorithme MD5 dans leurs signatures numériques. Les CipherSpecs plus anciens qui utilisent les protocoles TLS 1.0 ne sont pas soumis à cette restriction et peuvent continuer à utiliser des certificats avec des signatures numériques MD5 .

Pour afficher l'algorithme de signature numérique d'un certificat particulier, vous pouvez utiliser la commande **runmqakm** :

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

où *cert_label* est le libellé de certificat de l'algorithme de signature numérique à afficher. Pour plus de détails voir Labels de certificat numérique.

L'exécution de la commande **runmqakm** génère une sortie affichant l'utilisation de l'algorithme de signature spécifié:

```
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
```

La ligne `Signature Algorithm` indique que l'algorithme `MD5WithRSASignature` est utilisé. Cet algorithme étant basé sur MD5 , ce certificat numérique ne peut pas être utilisé avec les CipherSpecsTLS 1.2 .

Interopérabilité de Elliptic Curve et de RSA CipherSpecs

Les CipherSpecs ne peuvent pas tous être utilisés avec tous les certificats numériques. CipherSpecs sont indiqués par le préfixe de nom CipherSpec . Chaque type de CipherSpec impose des restrictions différentes sur le type de certificat numérique qui peut être utilisé. Ces restrictions s'appliquent à toutes les connexions TLS IBM MQ , mais sont particulièrement pertinentes pour les utilisateurs de la cryptographie Elliptic Curve.

Le tableau suivant récapitule les relations entre les CipherSpecs et les certificats numériques:

Tableau 4. Relations entre les CipherSpecs et les certificats numériques

Tapez	CipherSpec Préfixe de nom	Description	Type de clé publique requis	Algorithme de chiffrement de signature numérique	Méthode d'établissement de clé secrète
1	ECDHE_ECDSA_	CipherSpecs qui utilisent des clés publiques Elliptic Curve, des clés secrètes Elliptic Curve et des algorithmes de signature numérique Elliptic Curve.	Elliptic Curve	ECDSA	ECDHE
2	ECDHE_RSA_	CipherSpecs qui utilisent des clés publiques RSA, des clés secrètes Elliptic Curve et des algorithmes de signature numérique RSA.	RSA	RSA	ECDHE
3	(Tous les CipherSpecs TLS 1.3 CipherSpecs)	CipherSpecs qui utilisent des clés publiques Elliptic Curve ou RSA, des clés secrètes Elliptic Curve et des algorithmes de signature numérique Elliptic Curve ou RSA.	Courbe elliptique ou RSA	ECDSA ou RSA	ECDHE ou RSA
4	(Tous les autres)	CipherSpecs qui utilisent des clés publiques RSA et des algorithmes de signature numérique RSA.	RSA	RSA	RSA

Remarque : Les CipherSpecs de type 1 et 2 ne sont pas pris en charge par les gestionnaires de files d'attente IBM MQ et les clients MQI sur la plateforme IBM i .

La colonne de type de clé publique obligatoire indique le type de clé publique que le certificat personnel doit posséder lors de l'utilisation de chaque type de CipherSpec. Le certificat personnel est le certificat d'entité finale qui identifie le gestionnaire de files d'attente ou le client auprès de son partenaire distant.

Vous devez vous assurer que le certificat nommé dans le libellé de certificat est approprié pour le canal CipherSpec. En d'autres termes, si vous configurez un canal avec un CipherSpec qui requiert un certificat Elliptic Curve (EC), vous ne pouvez pas nommer un certificat RSA dans le libellé de certificat. Si vous configurez un canal avec un CipherSpec qui requiert un certificat RSA, vous ne pouvez pas nommer un certificat EC dans le libellé du certificat.

En supposant que vous avez correctement configuré IBM MQ, vous pouvez avoir:

- Un seul gestionnaire de files d'attente avec un mélange de certificats RSA et EC.
- Différents canaux sur le même gestionnaire de files d'attente à l'aide d'un certificat RSA ou EC.

L'algorithme de chiffrement de signature numérique fait référence à l'algorithme de chiffrement utilisé pour valider l'homologue. L'algorithme de chiffrement est utilisé avec un algorithme de hachage tel que MD5, SHA-1 ou SHA-256 pour calculer la signature numérique. Il existe différents algorithmes de signature numérique qui peuvent être utilisés, par exemple RSA avec MD5 ou ECDSA avec SHA-256. Dans le tableau, ECDSA fait référence à l'ensemble des algorithmes de signature numérique qui utilisent ECDSA ; RSA fait référence à l'ensemble des algorithmes de signature numérique qui utilisent RSA. Tout algorithme de signature numérique pris en charge dans l'ensemble peut être utilisé, à condition qu'il soit basé sur l'algorithme de chiffrement indiqué.

Les CipherSpecs de type 1 requièrent que le certificat personnel ait une clé publique Elliptic Curve. Lorsque ces CipherSpecs sont utilisés, l'accord de clé éphémère Elliptic Curve Diffie Hellman est utilisé pour établir la clé secrète pour la connexion.

Les CipherSpecs de type 2 requièrent que le certificat personnel ait une clé publique RSA. Lorsque ces CipherSpecs sont utilisés, l'accord de clé éphémère Elliptic Curve Diffie Hellman est utilisé pour établir la clé secrète pour la connexion.

Les CipherSpecs de type 3 requièrent que le certificat personnel ait une clé publique RSA. Lorsque ces CipherSpecs sont utilisés, l'échange de clés RSA est utilisé pour établir la clé secrète pour la connexion.

Cette liste de restrictions n'est pas exhaustive: selon la configuration, il peut y avoir des restrictions supplémentaires qui peuvent affecter davantage la possibilité d'interopérer. Par exemple, si IBM MQ est configuré pour être conforme aux normes FIPS 140-2 ou NSA Suite B, cela limitera également la plage des configurations autorisées. Pour plus d'informations, reportez-vous à la section suivante.

Si vous devez utiliser différents types de CipherSpec sur le même gestionnaire de files d'attente ou la même application client, configurez un libellé de certificat approprié et une combinaison de CipherSpec sur la définition du client.

Les trois types de CipherSpec n'interopèrent pas directement: il s'agit d'une limitation des normes TLS actuelles. Par exemple, supposons que vous ayez choisi d'utiliser le CipherSpec ECDHE_ECDSA_AES_128_CBC_SHA256 pour un canal récepteur nommé TO.QM1 sur un gestionnaire de files d'attente nommé QM1, le récepteur doit alors disposer d'un certificat personnel avec une clé Elliptic Curve et une signature numérique basée sur ECDSA. Si le canal récepteur ne répond pas à ces exigences, le démarrage du canal échoue.

Les autres canaux se connectant au gestionnaire de files d'attente QM1 peuvent utiliser d'autres CipherSpecs, à condition que chaque canal utilise un certificat du type approprié pour le CipherSpec de ce canal. Par exemple, supposons que QM1 utilise un canal émetteur nommé TO.QM2 pour envoyer des messages à un autre gestionnaire de files d'attente nommé QM2. Canal TO.QM2 peut utiliser le type 3 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 à condition que les deux extrémités du canal utilisent des certificats contenant des clés publiques RSA. L'attribut de canal de label de certificat peut être utilisé pour configurer un certificat différent pour chaque canal.

Lors de la planification de vos réseaux IBM MQ , réfléchissez soigneusement aux canaux qui requièrent TLS et assurez-vous que le type de certificat utilisé pour chaque canal est approprié pour une utilisation avec le CipherSpec sur ce canal.

Pour afficher l'algorithme de signature numérique et le type de clé publique d'un certificat numérique, vous pouvez utiliser la commande **runmqakm** :

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

où *cert_label* est le libellé du certificat dont vous devez afficher l'algorithme de signature numérique. Pour plus de détails voir Labels de certificat numérique.

L'exécution de la commande **runmqakm** génère une sortie affichant le type de clé publique:

```
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
```

Dans ce cas, la ligne Type de clé publique indique que le certificat possède une clé publique Elliptic Curve. Dans ce cas, la ligne Algorithme de signature indique que l'algorithme EC_ecdsa_with_SHA384 est en cours d'utilisation: il est basé sur l'algorithme ECDSA. Par conséquent, ce certificat ne peut être utilisé qu'avec des CipherSpecsde type 1.

TLS 1.3 CipherSpecs

TLS 1.3 CipherSpecs prend en charge les certificats ECDSA et RSA.

Elliptic Curve CipherSpecs et NSA Suite B

Lorsque IBM MQ est configuré pour se conformer au profil TLS 1.2 conforme à la suite B, les CipherSpecs et les algorithmes de signature numérique autorisés sont restreints, comme décrit dans «NSA Suite B Cryptography dans IBM MQ», à la page 44. De plus, la plage de clés Elliptic Curve acceptables est réduite en fonction des niveaux de sécurité configurés.

Au niveau de la sécurité Suite B 128 bits, la clé publique du sujet de certificat est requise pour utiliser la courbe elliptique NIST P-256 ou NIST P-384 et pour être signée avec la courbe elliptique NIST P-256 ou la courbe elliptique NIST P-384 . La commande **runmqakm** peut être utilisée pour demander des certificats numériques pour ce niveau de sécurité à l'aide d'un paramètre -sig_alg de EC_ecdsa_with_SHA256ou de EC_ecdsa_with_SHA384.

Au niveau de la sécurité de la suite B 192 bits, la clé publique du sujet de certificat est requise pour utiliser la courbe elliptique NIST P-384 et pour être signée avec la courbe elliptique NIST P-384 . La commande **runmqakm** peut être utilisée pour demander des certificats numériques pour ce niveau de sécurité à l'aide d'un paramètre -sig_alg de EC_ecdsa_with_SHA384.

Les courbes elliptiques NIST prises en charge sont les suivantes:

Tableau 5. Courbes elliptiques NIST prises en charge

Nom de courbe NIST FIPS 186-3	**Nom de courbe RFC 4492**	**Taille de clé de courbe elliptique (bits)**
P-256	secp256r1	256
P-384	secp384r1	384
P-521	secp521r1	521

Remarque : La courbe elliptique P-521 du NIST ne peut pas être utilisée pour une opération conforme à la suite B.

Concepts associés

«Activation des CipherSpecs», à la page 435
Activez un CipherSpec à l'aide du paramètre **SSLCIPH** dans la commande **DEFINE CHANNEL** ou **ALTER CHANNEL** MQSC.

«Comment indiquer que seuls les CipherSpecs certifiés FIPS sont utilisés lors de l'exécution sur MQI Client», à la page 276

Créez vos référentiels de clés à l'aide d'un logiciel compatible FIPS, puis indiquez que le canal doit utiliser des CipherSpecscertifiés FIPS.

«NSA Suite B Cryptography dans IBM MQ», à la page 44
Cette rubrique fournit des informations sur la manière de configurer IBM MQ for AIX, Linux, and Windows pour qu'il soit conforme au profil TLS 1.2 conforme à la suite B.

«Agence de sécurité nationale (NSA) Suite B Cryptographie», à la page 24
Le gouvernement des États-Unis d'Amérique fournit des conseils techniques sur les systèmes informatiques et la sécurité, y compris le chiffrement des données. La National Security Agency (NSA) des États-Unis recommande un ensemble d'algorithmes cryptographiques interopérables dans sa norme Suite B.

Enregistrements d'authentification de canal

Pour exercer un contrôle plus précis sur les accès accordés aux systèmes en cours de connexion au niveau d'un canal, vous pouvez utiliser les enregistrements d'authentification de canal.

Vous découvrirez peut-être que des clients essaient de se connecter à votre gestionnaire de files d'attente à l'aide d'un ID utilisateur vide ou d'un ID utilisateur de niveau supérieur, permettant à un client de procéder à des actions indésirables. Vous pouvez bloquer l'accès à ces clients à l'aide d'enregistrements d'authentification de canal. Il est également possible qu'un client accepte un ID utilisateur valide sur la plateforme client, mais inconnu ou sous un format non valide sur la plateforme serveur. Vous pouvez utiliser un enregistrement d'authentification de canal pour associer l'ID utilisateur accepté à un ID utilisateur valide.

Vous pouvez trouver une application client qui se connecte à votre gestionnaire de files d'attente et adopte un comportement indésirable. Pour protéger le serveur des problèmes que cette application pourrait provoquer, il convient de bloquer temporairement l'utilisation de l'adresse IP sur laquelle se trouve l'application client, le temps de mettre à jour les règles du pare-feu ou de corriger l'application. Vous pouvez utiliser un enregistrement d'authentification de canal pour bloquer l'adresse IP à partir de laquelle l'application client se connecte.

Si vous avez défini un outil d'administration tel qu'IBM MQ Explorer et un canal pour cette utilisation particulière, il est conseillé de limiter son utilisation à des ordinateurs client spécifiques. Vous pouvez utiliser un enregistrement d'authentification de canal pour permettre l'utilisation de ce canal uniquement à partir de certaines adresses IP.

Si vous venez de commencer avec des exemples d'applications s'exécutant en tant que clients, voir Préparation et exécution des exemples de programmes pour un exemple de configuration du gestionnaire de files d'attente en toute sécurité à l'aide d'enregistrements d'authentification de canal.

Pour obtenir des enregistrements d'authentification de canal afin de contrôler les canaux de communications entrantes, utilisez la commande MQSC **ALTER QMGR CHLAUTH(ENABLED)**.

Des règles **CHLAUTH** sont appliquées à un agent MCA de canal qui est créé en réponse à une nouvelle connexion entrante. Pour un agent MCA de canal créé en réponse au démarrage du canal en local, aucune règle **CHLAUTH** n'est appliquée.

Tableau 6. Emplacement de l'application des règles CHLAUTH dans différentes paires de canaux

Type de canal	**Agent MCA sur lequel les règles CHLAUTH sont appliquées**
Emetteur-récepteur	RCVR
Demandeur-serveur (démarré sur le serveur)	RQSTR
Demandeur-serveur (démarré sur le demandeur)	SVR
Demandeur-émetteur (démarré sur l'émetteur)	RQSTR
Demandeur-émetteur (démarré sur le demandeur)	Emetteur pour la connexion initiale. Demandeur pour la connexion de rappel.

Les enregistrements d'authentification de canal peuvent être créés pour l'exécution des fonctions suivantes :

- Bloquer les connexion en provenance d'adresses IP spécifiques.
- Bloquer les connexions associées à des ID utilisateur spécifiques.
- Définir une valeur MCAUSER à utiliser pour n'importe quel canal se connectant à partir d'une adresse IP spécifique.
- Définir une valeur MCAUSER à utiliser pour n'importe quel canal acceptant un ID utilisateur spécifique.
- Définir une valeur MCAUSER à utiliser pour n'importe quel canal associé à une adresse SSL ou un nom distinctif TLS spécifique.
- Définir une valeur MCAUSER à utiliser pour n'importe quel canal se connectant à partir d'un gestionnaire de files d'attente spécifique.
- Bloquer les connexions qui prétendent provenir d'un certain gestionnaire de files d'attente sauf si la connexion provient d'une adresse IP spécifique.
- Bloquer les connexions présentant un certain certificat SSL ou TLS, sauf si la connexion provient d'une adresse IP spécifique.

Ces utilisations sont expliquées plus en détails dans les sections suivantes.

Vous pouvez créer, modifier ou supprimer des enregistrements d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH** ou de la commande PCF **Set Channel Authentication Record**.

Remarque : Un grand nombre d'enregistrements d'authentification de canal peut avoir un impact négatif sur les performances d'un gestionnaire de files d'attente.

Blocage d'adresses IP

C'est normalement le rôle d'un pare-feu que de prévenir l'accès provenant de certaines adresses IP. Toutefois, il peut arriver que vous constatiez des tentatives de connexion provenant d'une adresse IP qui ne devrait pas avoir accès au système IBM MQ et que vous deviez temporairement bloquer l'adresse avant que le pare-feu ne puisse être mis à jour. Ces tentatives de connexion peuvent ne pas provenir de canaux IBM MQ, mais d'autres applications socket mal configurées pour cibler votre programme d'écoute IBM MQ. Bloquez les adresses IP en définissant un enregistrement d'authentification de canal de type BLOCKADDR. Vous pouvez spécifier une ou plusieurs adresses, ou des modèles avec des caractères génériques.

Lorsqu'une connexion entrante est refusée en raison d'un blocage de l'adresse IP de cette manière, un message d'événement MQRC_CHANNEL_BLOCKED avec un qualificateur de code anomalie MQRQ_CHANNEL_BLOCKED_ADDRESS généré, à condition que les événements du canal soient activés et que le gestionnaire de files d'attente soit en cours d'exécution. En outre, la connexion reste ouverte pendant 30 secondes avant de renvoyer l'erreur afin de garantir que le programme d'écoute n'est pas saturé par les tentatives de connexion répétées qui sont bloquées.

Pour bloquer des adresses IP uniquement sur des canaux spécifiques ou pour éviter le délai avant le signalement de l'erreur, définissez un enregistrement d'authentification de canal de type ADDRESSMAP avec le paramètre USERSRC(NOACCESS).

Lorsqu'une connexion entrante est refusée pour cette raison, un message d'événement MQRC_CHANNEL_BLOCKED avec le qualificateur de code anomalie MQRQ_CHANNEL_BLOCKED_NOACCESS est généré, à condition que les événements du canal soient activés et que le gestionnaire de files d'attente soit en cours d'exécution.

Pour voir un exemple, consultez «Blocage d'adresses IP spécifiques», à la page 397.

Blocage d'ID utilisateur

Pour empêcher certains ID utilisateur de se connecter sur un canal client, définissez un enregistrement d'authentification du canal de type BLOCKUSER. Ce type d'enregistrement s'applique uniquement aux canaux client disponibles et non aux canaux de message. Vous avez la possibilité d'indiquer un ou plusieurs ID utilisateur individuels, mais pas d'utiliser de caractères génériques.

Chaque fois qu'une connexion entrante est refusée pour cette raison, un message d'événement MQRC_CHANNEL_BLOCKED avec un qualificateur de code anomalie MQRQ_CHANNEL_BLOCKED_USERID est généré, à condition que les événements du canal soient activés.

Pour voir un exemple, consultez «Blocage d'ID utilisateur spécifiques», à la page 399.

Vous pouvez également bloquer l'accès d'un ID utilisateur quelconque sur certains canaux en définissant un enregistrement d'authentification du canal de type USERMAP avec le paramètre USERSRC(NOACCESS).

Lorsqu'une connexion entrante est refusée pour cette raison, un message d'événement MQRC_CHANNEL_BLOCKED avec le qualificateur de code anomalie MQRQ_CHANNEL_BLOCKED_NOACCESS est généré, à condition que les événements du canal soient activés et que le gestionnaire de files d'attente soit en cours d'exécution.

Pour voir un exemple, consultez «Blocage de l'accès pour un ID utilisateur client», à la page 402.

Blocage de gestionnaires de files d'attente

Pour bloquer l'accès à un canal se connectant à partir d'un gestionnaire de files d'attente spécifique, définissez un enregistrement d'authentification de canal de type QMGRMAP avec le paramètre USERSRC(NOACCESS). Vous pouvez indiquer un nom de gestionnaire de files d'attente ou un modèle comportant des caractères génériques. La fonction BLOCKUSER permettant de bloquer l'accès d'un gestionnaire de files d'attente ne comporte pas d'équivalent.

Lorsqu'une connexion entrante est refusée pour cette raison, un message d'événement MQRC_CHANNEL_BLOCKED avec le qualificateur de code anomalie MQRQ_CHANNEL_BLOCKED_NOACCESS est généré, à condition que les événements du canal soient activés et que le gestionnaire de files d'attente soit en cours d'exécution.

Pour voir un exemple, consultez «Blocage de l'accès à partir d'un gestionnaire de files d'attente éloignées», à la page 401.

Blocage des noms distinctifs SSL ou TLS

Pour bloquer l'accès à un utilisateur présentant un certificat personnel SSL ou TLS doté d'un nom distinctif spécifique, définissez un enregistrement d'authentification de canal de type SSLPEERMAP avec le paramètre USERSRC(NOACCESS). Vous pouvez indiquer un nom distinctif unique ou un modèle comportant des caractères génériques. La fonction BLOCKUSER permettant de bloquer l'accès aux noms distinctifs ne comporte pas d'équivalent.

Lorsqu'une connexion entrante est refusée pour cette raison, un message d'événement MQRC_CHANNEL_BLOCKED avec le qualificateur de code anomalie MQRQ_CHANNEL_BLOCKED_NOACCESS est généré, à condition que les événements du canal soient activés et que le gestionnaire de files d'attente soit en cours d'exécution.

Pour voir un exemple, consultez «Blocage de l'accès pour un nom distinctif SSL ou TLS», à la page 402.

Mappage d'adresses IP vers les ID utilisateur requis

Pour indiquer qu'un canal se connectant à partir d'une adresse IP spécifique doit utiliser une valeur MCAUSER spécifique, définissez un enregistrement d'authentification de canal de type ADDRESSMAP. Vous pouvez indiquer une adresse unique, une plage d'adresses ou un modèle comportant des caractères génériques.

Si vous utilisez un réexpéditeur de port, une rupture de session DMZ ou toute autre configuration modifiant l'adresse IP présentée au gestionnaire de files d'attente, le mappage des adresses IP ne convient pas forcément à votre situation.

Pour voir un exemple, consultez «Mappage d'une adresse IP à un ID utilisateur MCAUSER», à la page 403.

Mappage de gestionnaires de files d'attente vers les ID utilisateur requis

Pour indiquer qu'un canal se connectant à partir d'un gestionnaire de files d'attente spécifique doit utiliser une valeur MCAUSER spécifique, définissez un enregistrement d'authentification de canal de type

QMGRMAP. Vous pouvez indiquer un nom de gestionnaire de files d'attente ou un modèle comportant des caractères génériques.

Pour voir un exemple, consultez «Mappage d'un gestionnaire de files d'attente éloignées à un ID utilisateur MCAUSER», à la page 399.

Mappage des ID utilisateur vérifiés par un client vers les ID utilisateur requis

Pour indiquer qu'un ID utilisateur se connectant à partir d'un client IBM MQ MQI doit utiliser une valeur MCAUSER différente, définissez un enregistrement d'authentification de canal de type USERMAPP. Le mappage d'ID utilisateur ne se sert pas de caractères génériques.

Pour voir un exemple, consultez «Mappage d'un ID utilisateur client à un ID utilisateur MCAUSER», à la page 400.

Mappage des noms distinctifs SSL ou TLS vers les ID utilisateur requis

Pour indiquer qu'un utilisateur présentant un certificat personnel SSL/TLS doté d'un nom distinctif doit utiliser une valeur MCAUSER spécifique, définissez un enregistrement d'authentification de canal de type SSLPEERMAP. Vous pouvez indiquer un nom distinctif unique ou un modèle comportant des caractères génériques.

Pour voir un exemple, consultez «Mappage d'un nom distinctif SSL ou TLS à un ID utilisateur MCAUSER», à la page 401.

Mappage de gestionnaires de files d'attente, de clients ou de noms distinctifs SSL ou TLS en fonction d'une adresse IP

Dans certains, il est possible qu'une tierce partie falsifie le nom d'un gestionnaire de files d'attente. Un certificat SSL ou TLS ou un fichier de clés peut également être volé et réutilisé. Pour vous protéger contre ces menaces, vous pouvez indiquer qu'une connexion provenant d'un certain gestionnaire de files d'attente ou client ou utilisant un certain nom distinctif doit être établie à partir d'une adresse IP spécifique. Définissez un enregistrement d'authentification de canal de type USERMAP, QMGRMAP ou SSLPEERMAP et spécifiez l'adresse IP ou le modèle d'adresse autorisé à l'aide du paramètre ADDRESS.

Pour voir un exemple, consultez «Mappage d'un gestionnaire de files d'attente éloignées à un ID utilisateur MCAUSER», à la page 399.

Interaction entre les enregistrements d'authentification de canal

Il se peut qu'un canal tentant de se connecter corresponde à plusieurs enregistrements d'authentification de canaux et que leurs effets soient contradictoires. Par exemple, un canal peut vérifier un ID utilisateur qui soit bloqué par un enregistrement BLOCKUSER, mais qui comporte un certificat SSL ou TLS correspondant à un enregistrement SSLPEERMAP qui définisse un ID utilisateur différent. De plus, si les enregistrements utilisent des caractères génériques, il se peut qu'une adresse IP unique, un gestionnaire de files d'attente ou un nom distinctif SSL ou TLS corresponde à plusieurs modèles. Par exemple, l'adresse IP 192.0.2.6 correspond aux modèles 192.0.2.0-24, 192.0.2. *, et 192.0. * 6. L'action prise varie en fonction des éléments ci-dessous.

- L'enregistrement d'authentification de canal est sélectionné de la manière suivante :
  - Un enregistrement d'authentification de canal correspondant de manière explicite au nom du canal est prioritaire sur un enregistrement dont le canal comporte des caractères génériques.
  - Un enregistrement d'authentification de canal portant un nom distinctif SSL ou TLS est prioritaire sur un enregistrement associé à un ID utilisateur, un gestionnaire de files d'attente ou une adresse IP.
  - Un enregistrement d'authentification de canal associé à un ID utilisateur ou un gestionnaire de files d'attente est prioritaire sur un enregistrement faisant appel à une adresse IP.
- Si un enregistrement d'authentification de canal équivalent est détecté et qu'il indique une valeur MCAUSER, cette dernière est affectée au canal.

- Si un enregistrement d'authentification de canal équivalent est détecté et qu'il indique que le canal ne dispose d'aucun droit d'accès, une valeur MCAUSER de type *NOACCESS est affectée au canal. Cette valeur peut ensuite être modifiée par un programme exit de sécurité.
- Si aucun enregistrement d'authentification équivalent n'est détecté, ou si un enregistrement équivalent est détecté et qu'il indique que l'ID utilisateur du canal doit être utilisé, la zone MCAUSER est examinée.
  - Si la zone MCAUSER est vide, l'ID utilisateur client est affecté au canal.
  - Si la zone MCAUSER n'est pas vide, sa valeur est affectée au canal.
- Un programme exit de sécurité est exécuté. Ce programme peut définir l'ID utilisateur du canal ou déterminer si l'accès doit être bloqué.
- Si la connexion est bloquée ou si la valeur MCAUSER est définie sur *NOACCESS, le canal s'arrête.
- Si la connexion n'est pas bloquée, l'ID utilisateur du canal défini à l'étape précédente est vérifié par rapport à la liste des utilisateurs bloqués et ce, pour tous les canaux à l'exception d'un canal client.
  - Si l'ID utilisateur figure dans la liste des utilisateurs bloqués, le canal s'arrête.
  - Si l'ID utilisateur ne figure pas dans la liste des utilisateurs bloqués, le canal s'exécute.

Lorsque plusieurs enregistrements d'authentification de canal correspondent à un nom de canal, une adresse IP, un nom d'hôte, un nom de gestionnaire de files d'attente ou un nom distinctif SSL ou TLS, la correspondance la plus spécifique est utilisée. La correspondance considérée comme :

- La plus spécifique est un nom ne comportant pas de caractère générique, par exemple :
  - Un nom de canal A.B.C
  - Une adresse IP 192.0.2.6
  - Un nom d'hôte `hursley.ibm.com`
  - Un nom de gestionnaire de files d'attente 192.0.2.6
- La plus générique est l'astérisque unique (*), qui correspond, par exemple, à :
  - Tous les noms de canal
  - Toutes les adresses IP
  - Tous les noms d'hôte
  - Tous les noms de gestionnaire de files d'attente
- Un pattern de type chaîne commençant par un astérisque est plus générique qu'une chaîne dont le début est une valeur définie :
  - Pour les canaux, *.B.C est plus générique que A.*
  - Pour les adresses IP, *.0.2.6 est plus générique que 192.*
  - Pour les noms d'hôte, `*.ibm.com` est plus générique que `hursley.*`
  - Pour les noms de gestionnaire de files d'attente, *QUEUEMANAGER est plus générique que QUEUEMANAGER*
- Un pattern comportant un astérisque à une position spécifique dans une chaîne est plus générique qu'un pattern comportant une valeur définie à la même position dans une chaîne ; il en va de même pour chaque position suivante dans une chaîne :
  - Pour les canaux, A.*.C est plus générique que A.B.*
  - Pour les adresses IP, 192.*.2.6 est plus générique que 192.0.*.
  - Pour les noms d'hôte, `hursley.*.com` est plus générique que `hursley.ibm.*`
  - Pour les noms de gestionnaire de files d'attente, Q*MANAGER est plus générique que QUEUE*
- Lorsque plusieurs patterns comportent un astérisque à une position spécifique dans une chaîne, le pattern qui comporte le nombre de noeuds le moins élevé après l'astérisque est le plus générique :
  - Pour les canaux, A.* est plus générique que A.*.C
  - Pour les adresses IP, 192.* est plus générique que 192.*.2.*.

  - Pour les noms d'hôte, `hurlsey.*` est plus générique que `hursley.*.com`
  - Pour les noms de gestionnaire de files d'attente, Q* est plus générique que Q*MGR
- De plus, pour une adresse IP :
  - Une plage signalée par un trait d'union (-) est plus spécifique qu'un astérisque. Ainsi, 192.0.2.0-24 est plus spécifique que 192.0.2.*.
  - Une plage représentant un sous-ensemble d'une autre plage est plus spécifique que la plage la plus grande. Ainsi, 192.0.2.5-15 est plus spécifique que 192.0.2.0-24.
  - Les plages qui se chevauchent ne sont pas autorisées. Par exemple, vous ne pouvez pas avoir d'enregistrements d'authentification de canaux pour 192.0.2.0-15 et 192.0.2.10-20.
  - Un modèle ne peut pas contenir moins d'éléments que ce qui est obligatoire, sauf si le modèle se termine par une astérisque. Par exemple 192.0.2 n'est pas valide, mais 192.0.2.* est valide.
  - Un astérisque de fin doit être séparé du reste de l'adresse par le séparateur d'élément approprié (un point (.) pour IPv4, un deux-points (:) pour IPv6). Par exemple, 192.0* n'est pas valide parce que l'astérisque n'est pas un élément en soi.
  - Un pattern peut contenir des astérisques supplémentaires, à condition qu'aucun ne soit adjacent à l'astérisque de fin. Par exemple, 192.*.2.* est valide, mais 192.0.*.* est incorrect.
  - Un pattern d'adresse IPv6 ne peut pas contenir le signe deux-points et un astérisque de fin, car l'adresse résultante serait ambiguë. Par exemple, 2001::* pourrait devenir 2001:0000:*, 2001:0000:0000:* etc.
- Pour un nom distinctif SSL ou TLS, l'ordre de priorité des sous-chaînes est le suivant :

Tableau 7. Ordre de priorité des sous-chaînes

Commande	Sous-chaîne de nom distinctif	Nom
1	SERIALNUMBER=	Numéro de série du certificat
2	MAIL=	Adresse électronique
3	Deprecated E=	Adresse électronique (dépréciée dans la préférence pour MAIL)
4	UID=, USERID=	ID utilisateur
5	CN=	Nom usuel
6	T =	Titre
7	OU=	Unité organisationnelle
8	DC=	Composant de domaine
9	O=	Organisation
10	STREET=	Rue/Première ligne d'adresse
11	L=	Localité
12	ST=, SP=, S=	Nom de département
13	PC =	Code postal
14	C=	Pays
15	UNSTRUCTUREDNAME=	Nom d'hôte
16	UNSTRUCTUREDADDRESS=	Adresse IP
17	DNQ=	Qualificateur de nom distinctif

Par conséquent, si un certificat SSL ou TLS se présente avec un nom distinctif comportant les sous-chaînes O=IBM et C=UK, IBM MQ utilise un enregistrement d'authentification de canal pour O=IBM, mais pas pour C=UK si les deux sont présents.

Un nom distinctif peut contenir plusieurs OU, qui doit être spécifiée dans un ordre hiérarchique, les unités organisationnelles les plus grandes spécifiées en premier. Si deux noms distinctifs sont équivalents en tous points sauf en ce qui concerne leurs valeurs d'unités organisationnelles, le nom distinctif le plus spécifique est déterminé comme suit :

1. S'ils ont des nombres d'attributs d'unités organisationnelles différents, le nom distinctif possédant le plus de valeurs d'unités organisationnelles est le plus spécifique. Cela vient du fait que le nom distinctif possédant le plus d'unités organisationnelles qualifie le nom distinctif plus en détails et apporte plus de critères de correspondance. Même si l'unité organisationnelle de niveau supérieure est un caractère générique (OU=*), le nom distinctif possédant le plus d'unités organisationnelles est toujours considéré comme le plus spécifique globalement.
2. S'ils ont le même nombre d'attributs d'unités organisationnelles, les paires de valeurs d'unités organisationnelles correspondantes sont comparées dans l'ordre de gauche à droite, celles de gauche étant celles de plus haut niveau (les moins spécifiques), selon les règles suivantes.
   a. Une unité organisationnelle sans valeur indiquée par des caractères génériques est la plus spécifique car elle ne peut correspondre qu'à une seule chaîne exacte.
   b. Une unité organisationnelle avec un seul caractère générique, que ce soit au début ou à la fin (par exemple OU=ABC* ou OU=*ABC) est la plus spécifique suivante.
   c. Une unité organisationnelle avec deux caractères génériques par exemple OU=*ABC*) est la plus spécifique qui suit.
   d. Une unité organisationnelle constituée d'un seul astérisque (OU=*) est la moins spécifique.
3. Si la comparaison de chaîne est liée entre deux valeurs d'attributs de la même spécificité, alors la chaîne d'attribut la plus longue sera la plus spécifique.
4. Si la comparaison de chaîne est liée entre deux valeurs d'attributs de la même spécificité et de même longueur, le résultat est déterminé par une comparaison de chaîne ne respectant pas la casse de la portion de nom distinctif d'où sont exclus les caractères génériques.

Si deux DN sont égaux à tous égards, à l'exception de leurs valeurs de DC, les mêmes règles de correspondance s'appliquent que pour les OU, sauf que dans les valeurs de DC, la DC de gauche est le niveau le plus bas (le plus spécifique) et l'ordre de comparaison diffère en conséquence.

Affichage des enregistrements d'authentification de canaux

Pour afficher les enregistrements d'authentification de canaux, utilisez la commande MQSC **DISPLAY CHLAUTH** ou la commande PCF **Inquire Channel Authentication Records**. Vous pouvez choisir de renvoyer tous les enregistrements qui correspondent au nom de canal fourni ou seulement ceux qui correspondent à un élément particulier. La correspondance explicite vous indique quel enregistrement d'authentification de canal est utilisé lorsqu'un canal tente d'établir une connexion à partir d'une adresse IP ou d'un gestionnaire de files d'attente spécifique, qu'il utilise un ID utilisateur spécifique et qu'il présente un certificat personnel SSL/TLS doté d'un nom distinctif, le cas échéant.

Concepts associés

«Sécurité de la messagerie distante», à la page 107
Cette section traite des aspects de la sécurité liés à la messagerie distante.

Interaction entre CHLAUTH et CONNAUTH

Comment les enregistrements d'authentification de canal (CHLAUTH) et l'authentification de connexion (CONNAUTH) interagissent dans IBM MQ, dans le cas d'une conversation unique sur un canal.

Différents types de liaisons

IBM MQ prend en charge deux méthodes permettant à une application de se connecter:

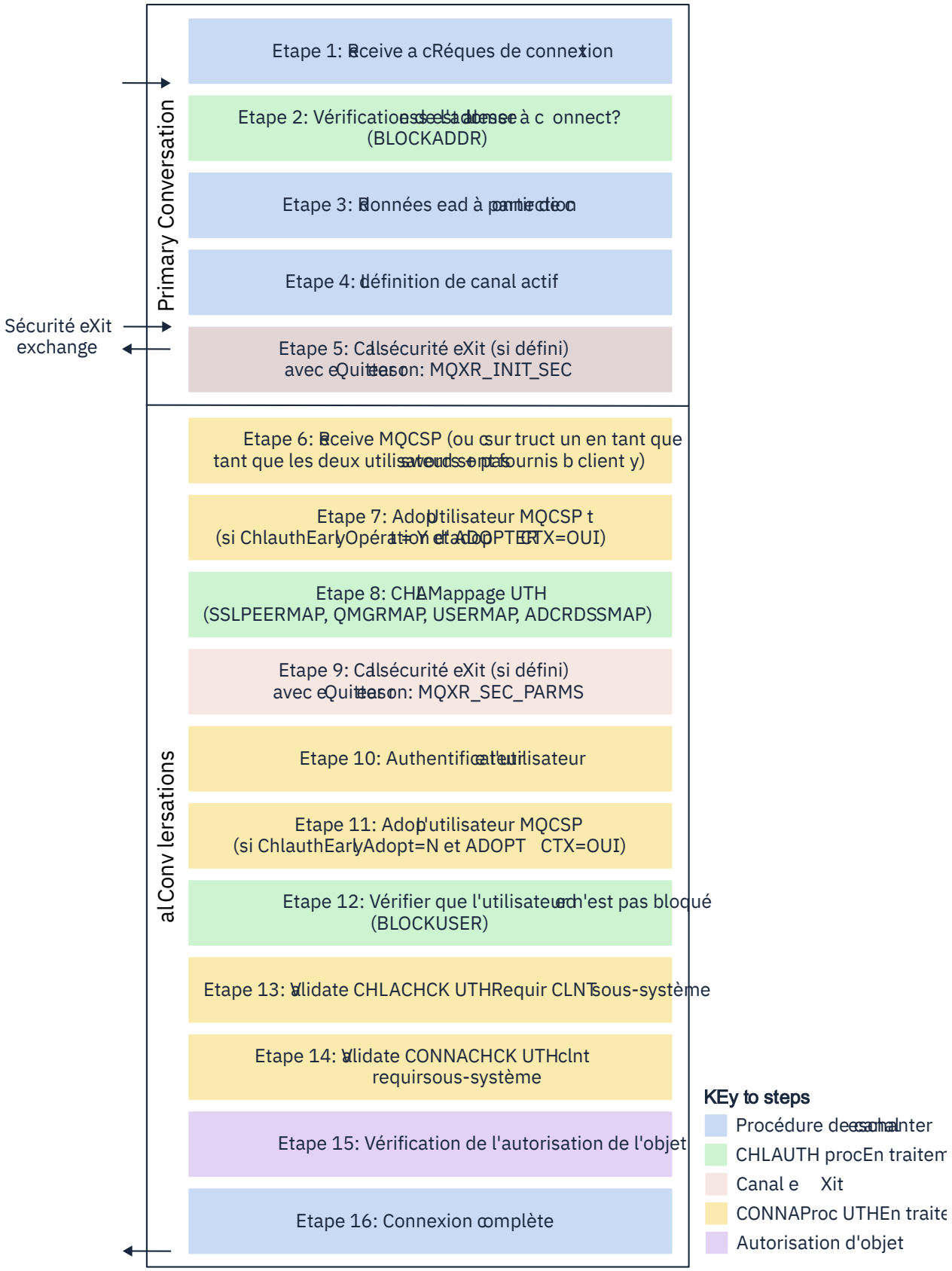
Liaisons locales
S'applique lorsque l'application et le gestionnaire de files d'attente se trouvent sur la même image d'exploitation. CHLAUTH n'est pas pertinent pour ce type de connexion d'application.

Liaisons client
S'applique lorsque l'application et le gestionnaire de files d'attente utilisent le réseau pour communiquer. L'application et le gestionnaire de files d'attente peuvent s'exécuter sur la même machine ou sur des machines différentes. Dans IBM MQ, une connexion client est gérée sous la forme d'un canal de connexion serveur (SVRCONN) et, dans ce cas, CONNAUTH et CHLAUTH sont applicables.

Etapes de liaison de l'extrémité réceptrice d'un canal

Lorsqu'une application se connecte à un gestionnaire de files d'attente, une vérification importante est effectuée pour s'assurer que les deux extrémités du canal comprennent ce qui est pris en charge par l'autre extrémité. L'extrémité réceptrice du canal effectue une vérification supplémentaire, impliquant CHLAUTH et CONNAUTH, pour s'assurer que le client est autorisé à se connecter, et ce processus peut également inclure un exit de sécurité car cela peut affecter le résultat. Cette phase de connexion de canal est également appelée *phase de liaison*.

Le diagramme suivant répertorie les étapes par lesquelles passe un canal SVRCONN lorsque l'extrémité du serveur (au niveau du gestionnaire de files d'attente) démarre:



Etape 1: Recevoir une demande de connexion
L'initiateur de canal ou le programme d'écoute reçoit une demande de connexion de quelque part sur le réseau.

Etape 2: L'adresse est-elle autorisée à se connecter?
Avant toute lecture de données, IBM MQ vérifie l'adresse IP du partenaire par rapport aux règles CHLAUTH, pour voir si l'adresse se trouve dans la règle *BLOCKADDR* . Si l'adresse est introuvable et donc non bloquée, le flux passe à l'étape suivante.

Etape 3: Lecture de données à partir du canal
IBM MQ lit maintenant les données dans une mémoire tampon et commence à traiter les informations envoyées.

Etape 4: Recherche de la définition de canal
Dans le premier flux de données, IBM MQ envoie, entre autres, le nom du canal que l'extrémité émettrice tente de démarrer. Le gestionnaire de files d'attente de réception peut ensuite rechercher la définition de canal, qui possède tous les paramètres spécifiés pour le canal.

Etape 5: Appel de l'exit de sécurité (s'il est défini)
Si un exit de sécurité (SCYEXIT) est défini pour le canal, il est appelé avec la raison de l'exit (MQCXP.**ExitReason**) défini sur MQXR_INIT_SEC.

Etape 6: Réception de MQCSP
Si nécessaire, construisez une valeur si le client a fourni des données d'authentification.

Si le client est une application Java ou JMS s'exécutant en mode compatibilité, il ne transmet pas de structure MQCSP au gestionnaire de files d'attente. A la place, si l'application a fourni un ID utilisateur et un mot de passe, une structure MQCSP est construite ici.

Etape 7: Adoptez l'utilisateur MQCSP (si `ChlauthEarlyAdopt` a pour valeur *Y* et ADOPTCTX=YES)
Les données d'identification fournies par le client sont authentifiées.

Si CONNAUTH utilise LDAP pour mapper un nom distinctif vérifié à un ID utilisateur court, le mappage se produit dans cette étape.

Si l'authentification aboutit, l'ID utilisateur est adopté par le canal et utilisé par l'étape de mappage CHLAUTH.

Remarque : A partir de IBM MQ 9.0.4 , le paramètre **`ChlauthEarlyAdopt`=** *Y* est automatiquement ajouté à la strophe channels du fichier qm.ini pour les nouveaux gestionnaires de files d'attente.

Etape 8: Mappage CHLAUTH
Le cache CHLAUTH est à nouveau inspecté pour rechercher les règles de mappage *SSLPEERMAP*, *USERMAP*, *QMGRMAP*et *ADDRESSMAP*.

La règle qui correspond le plus spécifiquement au canal entrant est utilisée. Si la règle comporte **USERSRC***(CHANNEL)* ou *(MAP)*, le canal se poursuit lors de la liaison.

Si les règles CHLAUTH ont pour résultat une règle avec **USERSRC***(NOACCESS)*, l'application ne peut pas se connecter au canal, sauf si les données d'identification sont ensuite remplacées par des données d'identification valides à l'étape 9.

Etape 9: Appel de l'exit de sécurité (s'il est défini)
Si un exit de sécurité (SCYEXIT) est défini pour le canal, il est appelé avec la raison de l'exit (MQCXP.**ExitReason**) défini sur MQXR_SEC_PARMS.

Un pointeur vers MQCSP sera présent dans la zone **SecurityParms** de la structure MQCXP.

La structure MQCSP comporte des pointeurs vers l'ID utilisateur (MQCSP.**CSPUserIdPtr**) et le mot de passe (MQCSP.**CSPPasswordPtr**). V 9.4.0 Depuis IBM MQ 9.3.4, la structure MQCSP contient également un pointeur vers le jeton d'authentification (MQCSP.**TokenPtr**).

Il est possible de modifier l'ID utilisateur et le mot de passe, ainsi que le jeton d'authentification dans l'exit. L'exemple suivant montre comment un exit de sécurité imprime les valeurs d'ID utilisateur et de mot de passe dans un journal d'audit:

```
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);
```

L'exit peut demander à IBM MQ de fermer le canal en renvoyant *MQXCC_CLOSE_CHANNEL* dans MQCXP.Zone **Exitresponse** . Sinon, le traitement du canal se poursuit jusqu'à la phase d'authentification de la connexion.

Remarque : Si l'utilisateur vérifié est modifié par l'exit de sécurité, les règles de mappage CHLAUTH ne sont pas réappliquées au nouvel utilisateur.

Etape 10: Authentification de l'utilisateur

La phase d'authentification se produit si CONNAUTH est activé sur le gestionnaire de files d'attente.

Pour vérifier cela, exécutez la commande MQSC'DISPLAY QMGR CONNAUTH'.

z/OS L'exemple suivant illustre la sortie de la commande **DISPLAY QMGR CONNAUTH** à partir d'un gestionnaire de files d'attente s'exécutant sous IBM MQ for z/OS.

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

Multi L'exemple suivant illustre la sortie de la commande'**DISPLAY QMGR CONNAUTH**'à partir d'un gestionnaire de files d'attente s'exécutant sous IBM MQ for Multiplatforms.

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

La valeur CONNAUTH est le nom d'un objet **AUTHINFO** IBM MQ .

Comme l'authentification du système d'exploitation (**AUTHTYPE**(*IDPWOS*)) est valide sur IBM MQ for Multiplatforms et IBM MQ for z/OS, les exemples utilisent l'authentification du système d'exploitation.

z/OS L'exemple suivant illustre l'objet AUTHINFO par défaut avec **AUTHTYPE**(*IDPWOS*) à partir d'un gestionnaire de files d'attente s'exécutant sous IBM MQ for z/OS.

```
 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 L'exemple suivant illustre l'objet AUTHINFO par défaut avec **AUTHTYPE**(*IDPWOS*) à partir d'un gestionnaire de files d'attente s'exécutant sous IBM MQ for Multiplatforms.

```
 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)
```

L'objet AUTHINFO TYPE (IDPWOS) possède un attribut appelé CHCKCLNT. Si la valeur est remplacée par *REQUIRED* , toutes les applications client doivent fournir des données d'identification valides.

Si l'utilisateur a été authentifié à l'étape 7, une autre vérification d'authentification n'est pas effectuée sauf si:

- L'ID utilisateur et le mot de passeou le jeton d'authentification dans la zone SecurityParms de la structure MQCXP ont été modifiés par un exit de sécurité à l'étape 9.
- L'application client est connectée avec des options demandant une fonctionnalité reconnectable.

Etape 11: Adopter le contexte de l'utilisateur MQCSP (si `ChlauthEarlyAdopt=`*N* et ADOPTCTX=YES)
Vous pouvez définir l'attribut ADOPTCTX , qui contrôle si le canal s'exécute sous MCAUSER ou l'ID utilisateur fourni par l'application.

Si l'ID utilisateur vérifié dans la zone MQCSP ou **`SecurityParms`** de la structure MQXCP a été authentifié avec succès et que **ADOPTCTX** est défini sur *YES*, le contexte de l'utilisateur résultant des étapes 7 et 8 est adopté comme contexte à utiliser pour cette application, sauf si l'ID utilisateur et le mot de passeou le jeton d'authentification dans la zone **`SecurityParms`** de la structure MQCXP ont été modifiés par un exit de sécurité à l'étape 9.

Cet ID utilisateur vérifié correspond à l'ID utilisateur vérifié pour l'autorisation d'utiliser les ressources IBM MQ .

Par exemple, vous n'avez pas de MCAUSER défini sur le canal SVRCONN et votre client s'exécute sous'`johndoe`'sur votre machine Linux . Votre application spécifie l'utilisateur'`fred`'dans le MQCSP, de sorte que le canal commence à s'exécuter avec'`johndoe`'en tant que MCAUSER actif. Après la vérification CONNAUTH, l'utilisateur'`fred`'est adopté et le canal s'exécute avec'`fred`'comme utilisateur MCAUSER actif.

Etape 12: Vérifier que l'utilisateur n'est pas bloqué (BLOCKUSER)
Si la vérification de CONNAUTH aboutit, le cache CHLAUTH est à nouveau inspecté pour vérifier si le MCAUSER actif est bloqué par une règle *BLOCKUSER* . Si l'utilisateur est bloqué, le canal se termine.

Etape 13: Validation des exigences CHLAUTH CHCKCLNT
Si la règle CHLAUTH sélectionnée à l'étape 8 spécifie en plus une valeur CHCKCLNT de REQUIRED ou REQDADM, la validation est effectuée pour s'assurer qu'un ID utilisateur CONNAUTH valide a été fourni pour répondre à l'exigence.

- Si CHCKCLNT (REQUIRED) est défini, un utilisateur doit avoir été authentifié à l'étape 7 ou 10. Sinon, la connexion est rejetée.
- Si CHCKCLNT (REQDADM) est défini, un utilisateur doit avoir été authentifié à l'étape 7 ou 10 si cette connexion est privilégiée. Sinon, la connexion est rejetée.
- Si CHCKCLNT (ASQMGR) est défini, cette étape est ignorée.

Remarques :

1. Si CHCKCLNT (REQUIRED) ou CHCKCLNT (REQDADM) est défini, mais que CONNAUTH n'est pas activé sur le gestionnaire de files d'attente, la connexion échoue avec un code retour MQRC_SECURITY_ERROR (2063) en raison du conflit dans la configuration.
2. L'utilisateur n'est pas réauthentifié dans cette étape.

Etape 14: Validez les exigences de CONNAUTH CHCKCLNT.
La phase d'authentification se produit si CONNAUTH est activé sur le gestionnaire de files d'attente.

La valeur de CONNAUTH CHCKCLNT est vérifiée pour déterminer les exigences définies pour les connexions entrantes:

- Si CHCKCLNT (NONE) est défini, cette étape est ignorée.
- Si CHCKCLNT (OPTIONAL) est défini, cette étape est ignorée.
- Si CHCKCLNT (REQUIRED) est défini, un utilisateur doit avoir été authentifié à l'étape 7 ou 10. Sinon, la connexion est rejetée.

- Si CHCKCLNT (REQDADM) est défini, un utilisateur doit avoir été authentifié à l'étape 7 ou 10 si cette connexion est privilégiée. Sinon, la connexion est rejetée.

Remarque : L'utilisateur n'est pas réauthentifié dans cette étape.

Multi **Etape 15: Vérification de l'autorisation de l'objet**

Une vérification est effectuée pour s'assurer que le MCAUSER actif dispose des droits appropriés pour se connecter au gestionnaire de files d'attente.

ALW Pour plus d'informations, voir Gestionnaire des droits d'accès aux objets.

IBM i Pour plus d'informations, voir «Gestionnaire des droits d'accès aux objets sous IBM i», à la page 167.

Etape 16: La connexion est terminée

Si les étapes précédentes aboutissent, la connexion se termine.

Concepts associés

CONNAUTH

Un gestionnaire de files d'attente peut être configuré pour authentifier les données d'identification fournies par une application lorsqu'elle se connecte.

Référence associée

SET CHLAUTH

ALTER AUTHINFO

Résolution des problèmes d'accès CHLAUTH

Etapes et exemples permettant de résoudre certains problèmes d'accès lors de l'utilisation d'enregistrements d'authentification de canal (CHLAUTH).

Avant de commencer

Remarque : Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir Administration d' IBM MQ à l'aide de commandes MQSC.

Pourquoi et quand exécuter cette tâche

Il existe trois règles par défaut pour le traitement CHLAUTH:

- AUCUN ACCES à tous les canaux par les utilisateurs `MQ-admin*`
- AUCUN ACCES à tous les SYSTEM.* canaux par tous les utilisateurs
- Accès ALLOW à SYSTEM.ADMIN.SVRCONN (non utilisateurs `MQ-admin`)

Les deux premières règles bloquent l'accès à tous les canaux. La troisième règle est plus spécifique et a donc priorité sur les deux autres, si le canal est SYSTEM.ADMIN.SVRCONN ADMIN.SVRCONN, permettant ainsi l'accès à ce canal.

Les règles HLAUTH permettent de déterminer si un canal peut être démarré et d'autoriser le mappage via MCAUSER vers un autre ID utilisateur. Si le canal ne peut pas être démarré, les erreurs suivantes se produisent généralement:

- `RC 2035 MQRC_NOT_AUTHORIZED`
- `RC 2059 MQRC_Q_MGR_NOT_AVAILABLE`
- `AMQ4036 Accès non autorisé`
- `AMQ9776: Le canal a été bloqué par l'ID utilisateur`
- `AMQ9777: Le canal a été bloqué`
- `MQJE001: Une exception MQException s'est produite: Code achèvement 2, Motif 2035`
- `MQJE036: Le gestionnaire de files d'attente a rejeté la tentative de connexion`

Vous devez bloquer l'accès strictement, puis ajouter d'autres règles CHLAUTH pour contrôler qui peut accéder aux canaux et les démarrer.

A titre de mesure temporaire, et pour identifier et résoudre les erreurs répertoriées, effectuez l'une des étapes suivantes.

Procédure

- **Désactiver les règles CHLAUTH**

  A titre de mesure temporaire, ainsi que pour identifier et résoudre les erreurs ci-dessus, vous pouvez désactiver les règles CHLAUTH. Les règles peuvent être réactivées à tout moment et si la désactivation des règles CHLAUTH résout le problème de connexion, vous savez que c'est la cause.

  Pour désactiver les règles CHLAUTH, exécutez la commande MQSC suivante:

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

  Notez que vous pouvez également définir CHLAUTH sur *WARN*, qui autorise l'accès et consigne le résultat de la règle.

- **Modifier ou supprimer des règles CHLAUTH**

  Vous pouvez également supprimer ou modifier la ou les règles CHLAUTH à l'origine de votre problème.

  Pour modifier une règle CHLAUTH, utilisez la commande SET CHLAUTH avec ACTION (REPLACE). Par exemple, pour modifier la règle par défaut qui empêche les utilisateurs `MQ-admin` d'accéder à WARN à tous les canaux, au lieu d'être bloqués, exécutez la commande MQSC suivante:

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

  Pour supprimer une règle CHLAUTH, utilisez la commande SET CHLAUTH avec ACTION (REMOVE). Par exemple, pour supprimer la règle par défaut qui empêche tous les utilisateurs `MQ-admin` d'accéder à tous les canaux, exécutez la commande MQSC suivante:

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

- **Tester l'accès à l'aide de MATCH (RUNCHECK)**

  Vous pouvez tester le résultat de vos règles CHLAUTH à l'aide de l'option MATCH (*RUNCHECK*) de la règle CHLAUTH. L'option **MATCH** (*RUNCHECK*) renvoie l'enregistrement correspondant à un canal entrant spécifique lors de l'exécution, si ce canal se connecte à ce gestionnaire de files d'attente. Vous devez fournir:

  – Nom du canal
  – attribut Adresse
  – Attribut SSLPEER, uniquement si le canal entrant utilise SSL ou TLS
  – QMNAME, si le canal entrant est un canal de gestionnaire de files d'attente, ou
  – CLNTUSER, si le canal entrant est un canal client

  L'exemple suivant exécute une commande MQSC pour vérifier quelle règle CHLAUTH, avec les règles par défaut en place, permet à un `MQ-admin` utilisateur `johndoe` d'accéder à un canal nommé `CHAN1`:

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

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

Pour l'utilisateur `johndoe`, le canal ne s'exécute pas, l'utilisateur sera bloqué en raison de la règle BLOCKUSER pour les utilisateurs *MQADMIN.

L'exemple suivant exécute une commande MQSC pour vérifier quelle règle CHLAUTH, avec les règles par défaut en place, permet à l'utilisateur `alice` qui n'est pas un utilisateur `MQ-admin` d'accéder à un canal nommé `CHAN1`:

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

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

Pour l'utilisateur `alice`, le canal s'exécute et le canal transmet `alice` en tant que MCAUSER. MCAUSER est l'ID utilisateur utilisé pour vérifier les droits sur les objets IBM MQ .

Référence associée
SET CHLAUTH
DISPLAYCHLAUTH

Création de nouvelles règles CHLAUTH pour les utilisateurs
Quelques scénarios courants pour les utilisateurs et des exemples de règles CHLAUTH pour les réaliser.

Avant de commencer

Remarque : Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir Administration d' IBM MQ à l'aide de commandes MQSC.

Pourquoi et quand exécuter cette tâche

Il existe trois règles par défaut pour le traitement CHLAUTH:

- AUCUN ACCES à tous les canaux par les utilisateurs `MQ-admin*`
- AUCUN ACCES à tous les SYSTEM.* canaux par tous les utilisateurs
- Accès ALLOW à SYSTEM.ADMIN.SVRCONN (non utilisateurs `MQ-admin`)

Les deux premières règles bloquent l'accès à tous les canaux. La troisième règle est plus spécifique et a donc priorité sur les deux autres, si le canal est SYSTEM.ADMIN.SVRCONN ADMIN.SVRCONN, permettant ainsi l'accès à ce canal.

Pour créer de nouvelles règles CHLAUTH pour les utilisateurs, configurez un ou plusieurs des scénarios suivants.

Procédure

- **Contrôle de l'accès pour des utilisateurs `MQ-admin` spécifiques**
  a) Configurez un canal de connexion serveur qui doit être utilisé exclusivement pour une perspective d'administration, c'est-à-dire pour la connexion à partir de IBM MQ Explorer.

  Vous disposez d'un canal spécifique pour cette utilisation, et d'une ou de plusieurs adresses IP définies, à partir desquelles vous souhaitez que les connexions soient acceptées, et d'un accès bloqué pour l'ID `'mqm'` , si la connexion ne provient pas de l'une des adresses IP spécifiées.

  b) Créez un canal SVRCONN pour les utilisateurs IBM MQ Explorer et `MQ-admin` appelés `ADMIN.CHAN`.

  Exécutez la commande MQSC suivante :

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

  c) A des fins de test, vérifiez que vous disposez d'un utilisateur défini dans le groupe `MQ-admin` et d'un autre utilisateur défini dans le groupe.

Pour ce scénario, mqadm se trouve dans le groupe MQ-admin et alice ne se trouve pas dans le groupe.

d) Vérifiez que les règles CHLAUTH par défaut sont en place.

e) Ajoutez trois règles pour permettre à un utilisateur spécifique d'accéder à ADMIN.CHAN en tant que MQ-admin à partir de certaines adresses IP:

- Définir NOACCESS à partir de n'importe quelle adresse
- Définissez BLOCKUSER pour ce canal afin de ne bloquer que l'utilisateur nobody, qui remplace *MQADMIN BLOCKUSER
- Accès ALLOW à l'utilisateur mqadm sur un sous-réseau spécifique d'adresses et MAP aux droits utilisateur mqadm

Pour ce faire, exécutez les commandes MQSC suivantes:

```
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)
```

A ce stade, l'utilisateur mqadm peut accéder à ADMIN.CHAN , à partir de la plage d'adresses IP spécifiée.

f) Facultatif : Vous pouvez exécuter la commande MQSC MATCH (RUNCHECK) à tout moment pour afficher les résultats de chacune de ces commandes:

```
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)
```

A ce stade, seuls les utilisateurs disposant d'un enregistrement CHLAUTH sont autorisés à accéder à l'aide de ADMIN.CHAN.

- **Contrôle de l'accès pour un utilisateur spécifique et une application client IBM MQ**

  Pour ce scénario, les règles CHLAUTH par défaut sont adéquates, en supposant que les droits IBM MQ doivent être définis pour un utilisateur spécifique, afin de fournir les droits IBM MQ appropriés (à l'aide de setmqaut).

  Dans ce scénario, les droits sont définis pour un utilisateur mqapp1, qui n'est pas un utilisateur MQ-admin .

  a) Utilisez la commande MQSC suivante pour créer un canal SVRCONN, APP1.CHAN, à utiliser par une application particulière et un utilisateur spécifique.

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

  b) Lorsque les règles CHLAUTH par défaut sont en place, l'utilisateur mqapp1 peut démarrer l'application APP1.CHAN .

  L'ID utilisateur provenant de l'application client IBM MQ est utilisé pour la vérification des droits d'accès aux objets IBM MQ . Dans ce cas, en supposant que l'utilisateur mqapp1 exécute

l'application client IBM MQ , celle-ci est utilisée pour la vérification des droits d'accès aux objets IBM MQ . Par conséquent, si mqapp1 a accès aux objets IBM MQ dont l'application a besoin, tout va bien ; si ce n'est pas le cas, vous obtiendrez des erreurs de droits d'accès.

Vous pouvez renforcer encore la sécurité en créant des règles CHLAUTH spécifiques pour l'ID utilisateur mqapp1 , mais sous les règles par défaut, aucun membre du groupe MQ-admin ne peut accéder à ce canal.

Exécutez les commandes MQSC suivantes:

```
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)
```

- **Contrôle de l'accès d'un utilisateur spécifique à l'aide du nom distinctif (DN) de certificat de cet utilisateur**

  Pour ce scénario, l'utilisateur doit disposer d'un certificat transmis au gestionnaire de files d'attente. Le nom distinctif est ensuite comparé au paramètre SSLPEER de la règle CHLAUTH et SSLPEER peut utiliser des caractères génériques.

  En cas de correspondance, l'utilisateur peut également être mappé à un autre utilisateur MCAUSER pour vérifier les droits sur les objets IBM MQ . Le mappage de MCAUSER peut réduire le nombre d'utilisateurs à gérer dans le gestionnaire des droits d'accès aux objets (OAM) IBM MQ .

  a) Vous disposez d'un canal TLS avec des certificats en cours d'utilisation et vous avez besoin de règles pour:

  – Bloquer tous les utilisateurs d'un canal particulier
  – N'autorisez que les utilisateurs ayant un SSLPEER particulier qui utilisent le client de cet utilisateur pour l'accès à IBM MQ OAM.

  Exécutez les commandes MQSC suivantes:

```
.
# 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)
```

  L'ID utilisateur client se connectant sur le canal est utilisé pour les droits IBM MQ OAM des objets IBM MQ ; par conséquent, l'ID utilisateur doit disposer des droits IBM MQ appropriés.

  b) Facultatif : Mapper à un autre ID utilisateur IBM MQ .

  Réexécutez la commande MQSC précédente, en remplaçant USERSRC(MAP) MCAUSER('mquser1') par USERSRC(CHANNEL).

- **Mapper un utilisateur particulier à l'utilisateur mqm**

  Il s'agit d'un ajout ou d'une modification à Control access for specific MQ-admin users.

  Utilisez les commandes MQSC pour ajouter la règle CHLAUTH suivante afin de mapper des utilisateurs particuliers à l'utilisateur mqm , ou à un ID utilisateur MQ-admin , pour lequel le droit d'accès aux objets IBM MQ est configuré dans le gestionnaire des droits d'accès aux objets IBM MQ .

```
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)
```

Cela permet et mappe l'utilisateur `johndoe` à l'utilisateur `mqm` pour le canal spécifique ADMIN.CHAN.

Concepts associés

«Création de nouvelles règles CHLAUTH pour les canaux», à la page 71
Pour vous aider à créer vos propres règles CHLAUTH, voici quelques scénarios courants pour les canaux, ainsi que des exemples de règles CHLAUTH pour les exécuter.

Tâches associées

«Résolution des problèmes d'accès CHLAUTH», à la page 66
Etapes et exemples permettant de résoudre certains problèmes d'accès lors de l'utilisation d'enregistrements d'authentification de canal (CHLAUTH).

Référence associée

SET CHLAUTH
DISPLAYCHLAUTH

Création de nouvelles règles CHLAUTH pour les canaux
Pour vous aider à créer vos propres règles CHLAUTH, voici quelques scénarios courants pour les canaux, ainsi que des exemples de règles CHLAUTH pour les exécuter.

Cette rubrique contient les scénarios suivants:

- «Autorisez uniquement l'accès à un canal particulier à partir d'une plage d'adresses IP spécifique.», à la page 71
- «Pour un canal spécifique, bloquez tous les utilisateurs, mais autorisez des utilisateurs spécifiques à se connecter.», à la page 71
- «Utilisation de CHLAUTH pour les canaux récepteur et émetteur», à la page 72

Autorisez uniquement l'accès à un canal particulier à partir d'une plage d'adresses IP spécifique.

Pour ce scénario, vous souhaitez:

- Aucun accès au canal depuis n'importe où
- Autoriser l'accès à partir d'une adresse IP ou d'une plage d'adresses IP spécifique

```
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)
```

Cela n'autorise que l'application APP2.CHAN à démarrer lorsque la connexion provient de la plage d'adresses IP spécifique spécifiée.

L'utilisateur se connectant en tant que MCAUSER est mappé à `mqapp2`et obtient par conséquent les droits OAM IBM MQ pour cet utilisateur.

Pour un canal spécifique, bloquez tous les utilisateurs, mais autorisez des utilisateurs spécifiques à se connecter.

Il existe trois règles par défaut pour le traitement CHLAUTH:

- AUCUN ACCES à tous les canaux par les utilisateurs `MQ-admin*`
- AUCUN ACCES à tous les SYSTEM.* canaux par tous les utilisateurs
- Accès ALLOW à SYSTEM.ADMIN.SVRCONN (non utilisateurs `MQ-admin`)

Les deux premières règles bloquent l'accès à tous les canaux. La troisième règle est plus spécifique et a donc priorité sur les deux autres, si le canal est SYSTEM.ADMIN.SVRCONN ADMIN.SVRCONN, permettant ainsi l'accès à ce canal.

Pour ce scénario, l'accès au canal MY.SVRCONN comporte les règles CHLAUTH par défaut.

Vous devez ajouter les éléments suivants:

```
# 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)
```

Cette première partie du code empêche toute personne de se connecter à MY.SVRCONN, puis le code autorise uniquement le canal MY.SVRCONN à être démarré lorsque la connexion provient de l'ID utilisateur spécifique `johndoe`.

L'utilisateur se connectant sur le canal `johndoe` est utilisé pour les droits IBM MQ OAM des objets IBM MQ . Par conséquent, l'ID utilisateur doit disposer des droits IBM MQ appropriés.

Vous pouvez effectuer un mappage vers un ID utilisateur IBM MQ différent si vous le souhaitez, à l'aide des éléments suivants:

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

au lieu de `USERSRC(CHANNEL)`.

Utilisation de CHLAUTH pour les canaux récepteur et émetteur

Vous pouvez utiliser les règles CHLAUTH pour ajouter une sécurité supplémentaire aux canaux récepteur et émetteur, afin de restreindre l'accès au canal récepteur. Notez que si vous ajoutez ou modifiez des règles CHLAUTH, les règles CHLAUTH mises à jour ne s'appliquent qu'au démarrage du canal. Par conséquent, si les canaux sont déjà en cours d'exécution, vous devez les arrêter et les redémarrer pour que les mises à jour CHLAUTH s'appliquent.

Les règles HLAUTH peuvent être utilisées sur n'importe quel canal, mais il existe certaines restrictions. Par exemple, les règles USERMAP s'appliquent uniquement aux canaux SVRCONN.

Cet exemple permet une connexion à partir d'une adresse IP particulière uniquement, pour démarrer TO.MYSVR1 :

```
# 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)
```

Cet exemple permet la connexion à partir d'un gestionnaire de files d'attente particulier uniquement:

```
# 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)
```

Tâches associées

«Résolution des problèmes d'accès CHLAUTH», à la page 66
Etapes et exemples permettant de résoudre certains problèmes d'accès lors de l'utilisation d'enregistrements d'authentification de canal (CHLAUTH).

«Création de nouvelles règles CHLAUTH pour les utilisateurs», à la page 68
Quelques scénarios courants pour les utilisateurs et des exemples de règles CHLAUTH pour les réaliser.

Référence associée

SET CHLAUTH
DISPLAYCHLAUTH

Création d'une règle de back-stop CHLAUTH
Lorsque vous pensez au contrôle des connexions entrantes dans votre gestionnaire de files d'attente, vous disposez de deux options. Vous pouvez soit essayer de répertorier toutes les connexions qui ne sont pas autorisées, soit vous pouvez commencer en disant que toutes les connexions ne sont pas autorisées, puis essayer de répertorier toutes les connexions qui sont autorisées. Cette deuxième option est décrite ici.

Pourquoi et quand exécuter cette tâche

La raison de l'utilisation de la deuxième option est que, si vous essayez de répertorier toutes les connexions qui ne sont pas autorisées et que tout ce qui n'est pas répertorié est autorisé dans la liste, le résultat de l'absence d'une connexion de la liste est qu'une connexion qui n'aurait pas dû être autorisée est en mesure de se connecter, ce qui entraîne une violation de sécurité potentielle.

A l'inverse, si au lieu de cela, vous commencez par dire que chaque connexion n'est pas autorisée, puis listez celles qui sont, le résultat de l'absence d'une de cette liste n'est pas une violation de la sécurité. Si votre entreprise requiert l'ajout de connexions supplémentaires, il s'agit d'une tâche relativement simple, mais il n'y a pas de violation de sécurité potentielle.

La première chose à faire est de créer une règle *back-stop* , qui est une règle qui intercepte les connexions qui ne correspondent pas à des règles plus spécifiques. Cette règle a pour effet d'empêcher les connexions distantes de pouvoir se connecter à votre gestionnaire de files d'attente.

Toutefois, si vous vous souciez de cette approche, vous pouvez configurer la règle *back-stop* en mode avertissement ; voir l'étape «2», à la page 73

Procédure

1. Pour créer une règle de back-stop qui arrête les connexions distantes à votre gestionnaire de files d'attente, exécutez la commande suivante:

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

   Maintenant que vous avez fermé la porte sur toutes les connexions à distance, vous pouvez commencer à mettre en place des règles plus spécifiques pour autoriser certaines connexions. Exemple :

   ```
   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. Si vous souhaitez créer la règle d'arrêt arrière en mode avertissement, exécutez la commande suivante:

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

Maintenant, vous pouvez continuer, et faire toutes vos règles positives. Lorsque vous pensez avoir créé toutes les règles dont vous avez besoin, activez les événements de canal en exécutant la commande suivante:

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

et surveillez SYSTEM.ADMIN.CHANNEL.EVENT EVENT pour les événements avec **Reason** défini sur MQRC_CHANNEL_BLOCKED_WARNING.

Ces événements détaillent les connexions qui correspondent à votre règle de back-stop, mais comme la commande s'exécute en mode avertissement, elles n'ont pas été bloquées pour le moment.

Examinez chacun de ces événements et déterminez si cette connexion doit avoir une règle positive en place pour l'autoriser ou si elle a été correctement mise en correspondance avec la règle *back-stop* . Vous pouvez exécuter dans ce mode, en examinant les événements au fur et à mesure qu'ils sont créés, jusqu'à ce que vous soyez satisfait d'avoir vu tous les canaux entrants et d'avoir des règles positives appropriées en place pour tous.

A ce stade, vous pouvez modifier la règle *back-stop* pour démarrer les connexions réellement bloquantes auxquelles elle correspond en exécutant la commande suivante:

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

Création d'un administrateur IBM MQ non privilégié
Comment créer un administrateur IBM MQ non privilégié à l'aide de CHLAUTH.

Pourquoi et quand exécuter cette tâche

Dans le cadre de cette tâche, les termes:

utilisateur privilégié
Utilisateur autorisé à effectuer une opération sans être explicitement autorisé à effectuer cette opération. Les utilisateurs du groupe mqm sont des exemples de ces utilisateurs privilégiés.

Administrateur IBM MQ
Désigne un utilisateur qui a besoin d'émettre des commandes d'administration sur IBM MQ, telles que **DEFINE QLOCAL** ou **START CHANNEL**.

Les étapes suivantes permettent de créer un administrateur IBM MQ non privilégié.

Procédure

1. Créez un ID utilisateur sur la machine du gestionnaire de files d'attente à l'aide des commandes appropriées pour la ou les plateformes utilisées par votre entreprise.

   Le nom d'utilisateur `alice` est utilisé dans cet exemple.
2. Accordez à ce nouvel utilisateur le droit d'émettre toutes les commandes d'administration IBM MQ en procédant comme suit:
   a) Démarrez IBM MQ Explorer à l'aide d'un utilisateur privilégié.
   b) Accédez à l' *assistant basé sur les rôles* en sélectionnant le gestionnaire de files d'attente approprié, puis `Object Authorities` et `Add Role Based Authorities`.
   c) Dans le panneau de l'assistant qui s'affiche, entrez l'ID utilisateur que vous avez créé à la première étape ou, si vous préférez utiliser des groupes, entrez le nom de groupe de l'utilisateur ou de l'ensemble d'utilisateurs que vous souhaitez transformer en administrateurs IBM MQ non privilégiés.
   d) Configurez l'assistant pour un accès administrateur complet.
   e) Si vous souhaitez permettre à votre administrateur IBM MQ non privilégié de parcourir les messages dans les files d'attente, cochez également cette case.

f) Passez en revue les commandes du panneau de prévisualisation situé dans la partie inférieure de l'assistant.

Vous pouvez couper et coller ces commandes pour générer vos propres scripts.

L'une des raisons pour lesquelles vous préférez le faire avec votre propre script est de réduire le nombre d'accès que vous accordez à cet utilisateur. Plutôt que d'accorder l'accès à tous les objets, vous préférerez peut-être n'accorder l'accès qu'à un certain groupe d'objets.

Si vous appuyez sur **OK** dans l'assistant, les commandes sont émises au fur et à mesure qu'elles s'affichent.

g) Vous devez configurer certaines règles CHLAUTH pour autoriser l'accès à distance pour cet ID utilisateur, si un administrateur IBM MQ non privilégié est également requis pour l'accès à distance.

En supposant que votre entreprise utilise les conseils de «Création d'une règle de back-stop CHLAUTH», à la page 73, il vous suffit d'ajouter une règle d'activation.

La règle que vous créez dépend plutôt de la manière dont vous choisissez d'authentifier vos administrateurs IBM MQ distants.

Si vous utilisez une authentification TCP/IP faible, vous pouvez configurer une règle CHLAUTH qui se présente comme suit:

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

9. Si vous utilisez l'authentification TLS, vous pouvez configurer une règle CHLAUTH qui se présente comme suit:

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

Désormais, lorsqu'un utilisateur se connecte à `admin-channel-name` (et correspond aux règles CHLAUTH), il peut émettre des commandes sous l'ID utilisateur `alice` sur le gestionnaire de files d'attente et un accès distant privilégié n'est donc pas requis.

Authentification de connexion

L'authentification de connexion permet aux applications de fournir des données d'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente. Le gestionnaire de files d'attente valide les données d'identification. L'ID utilisateur fourni dans les données d'identification peut également être utilisé dans les vérifications d'autorisation pour les ressources auxquelles l'application accède.

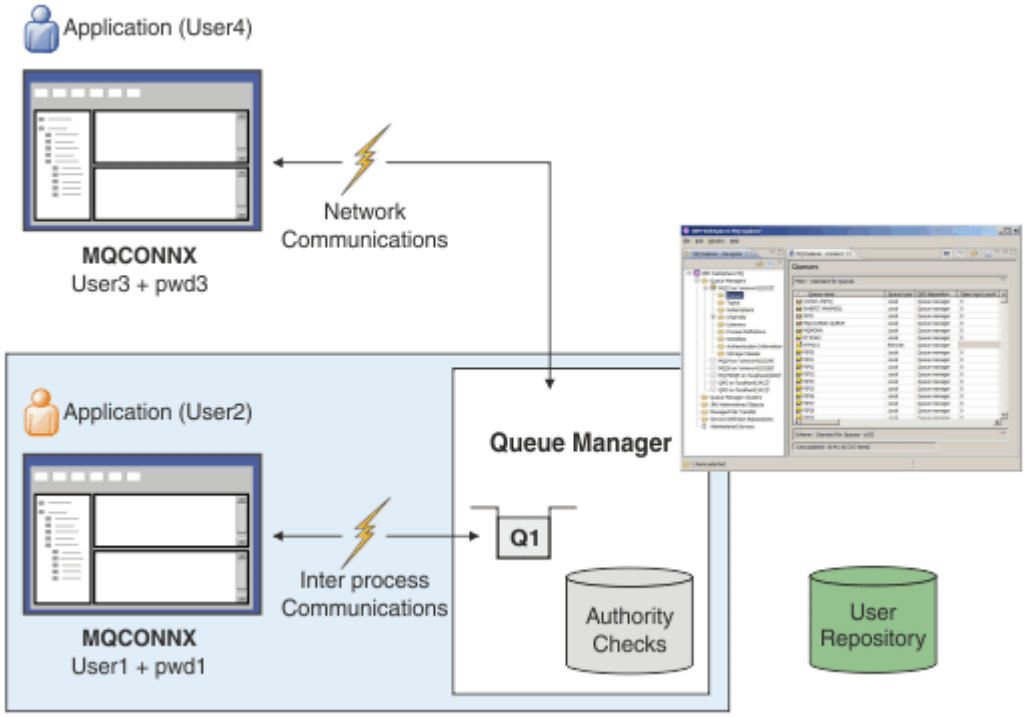
Les applications peuvent fournir un ID utilisateur et un mot de passe pour l'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente.

V 9.4.0 Depuis la IBM MQ 9.3.4, les applications IBM MQ client peuvent également fournir un jeton d'authentification comme méthode d'authentification alternative.

Le gestionnaire de files d'attente peut être configuré pour valider les données d'identification fournies par l'application.

Un ID utilisateur et un mot de passe fournis par une application sont vérifiés à l'aide du référentiel d'utilisateurs dans la configuration du gestionnaire de files d'attente. Pour plus d'informations sur le référentiel utilisé pour la vérification des ID utilisateur et des mots de passe, voir Référentiels d'utilisateurs.

V 9.4.0 Les jetons d'authentification sont validés à l'aide des certificats et des clés symétriques du magasin de clés d'authentification de jeton du gestionnaire de files d'attente pour valider la signature du jeton. Pour plus d'informations sur l'authentification des utilisateurs avec des jetons d'authentification, voir «Utilisation des jetons d'authentification», à la page 335.



Dans le diagramme, deux applications établissent des connexions avec un gestionnaire de files d'attente, une application en tant que client et une autre utilisant des liaisons locales. Les applications peuvent utiliser diverses API pour se connecter au gestionnaire de files d'attente, mais elles peuvent toutes fournir un ID utilisateur et un mot de passe. L'ID utilisateur sous lequel l'application s'exécute, `User2` et `User4` , dans le diagramme, qui est l'ID utilisateur de système d'exploitation habituel présenté à IBM MQ, peut être différent de l'ID utilisateur fourni par l'application, `User1` et `User3`.

Le gestionnaire de files d'attente reçoit les commandes de configuration (dans le diagramme, IBM MQ Explorer est utilisé) et gère l'ouverture des ressources et vérifie les droits d'accès à ces ressources. Il existe de nombreuses ressources différentes dans IBM MQ auxquelles une application peut avoir besoin de droits d'accès. Le diagramme illustre l'ouverture d'une file d'attente pour la sortie, mais les mêmes principes s'appliquent également aux autres ressources.

Concepts associés

«Authentification de connexion: Configuration», à la page 76
Un gestionnaire de files d'attente peut être configuré pour authentifier les données d'identification fournies par une application lorsqu'elle se connecte.

«Authentification de la connexion: modifications de l'application», à la page 81

«Authentification de connexion: référentiels d'utilisateurs», à la page 82
Pour chacun de vos gestionnaires de files d'attente, vous pouvez choisir différents types d'objet d'informations d'authentification pour l'authentification des ID utilisateur et des mots de passe.

Authentification de connexion: Configuration

Un gestionnaire de files d'attente peut être configuré pour authentifier les données d'identification fournies par une application lorsqu'elle se connecte.

Activation de l'authentification de connexion sur un gestionnaire de files d'attente

Sur un objet gestionnaire de files d'attente, l'attribut **CONNAUTH** peut être défini sur le nom d'un objet d'informations d'authentification (AUTHINFO). L'attribut **AUTHTYPE** d'un objet AUTHINFO spécifie le type de l'objet. Les objets AUTHINFO utilisés pour l'authentification de connexion peuvent être de l'un des deux types suivants:

IDPWOS
Le gestionnaire de files d'attente utilise le système d'exploitation local pour authentifier l'ID utilisateur et le mot de passe fournis par une application de connexion.

V 9.4.0 Linux AIX Depuis la IBM MQ 9.3.4, ce type d'objet AUTHINFO permet également à un gestionnaire de files d'attente qui s'exécute sur AIX ou Linux de valider des jetons d'authentification. Outre l'objet AUTHINFO utilisé pour configurer l'authentification de connexion, le gestionnaire de files d'attente doit être configuré pour accepter les jetons d'authentification avec la strophe **AuthInfo** du fichier `qm.ini` . Pour plus d'informations sur la configuration d'un gestionnaire de files d'attente pour qu'il accepte les jetons d'authentification, voir «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un magasin de clés local», à la page 343.

IDPWLDAP
Le gestionnaire de files d'attente utilise un serveur LDAP pour authentifier l'ID utilisateur et le mot de passe fournis par une application de connexion.

Remarque : Vous ne pouvez pas spécifier d'autre type d'objet d'informations d'authentification dans l'attribut **CONNAUTH** du gestionnaire de files d'attente.

Les objets AUTHINFO de type `IDPWOS` et `IDPWLDAP` sont similaires dans plusieurs de leurs attributs. Les attributs décrits ici sont communs aux deux types d'objet.

Les exemples de commandes MQSC suivants activent l'authentification de connexion avec les opérations suivantes:

1. Définissez un objet AUTHINFO nommé `USE.PW`.
2. Modifiez l'attribut **CONNAUTH** du gestionnaire de files d'attente pour qu'il fasse référence à cet objet AUTHINFO.
3. Exécutez la commande **REFRESH SECURITY** pour actualiser la configuration d'authentification de connexion du gestionnaire de files d'attente. La commande **REFRESH SECURITY** doit être émise avant que le gestionnaire de files d'attente ne reconnaisse les modifications apportées à la configuration d'authentification de connexion.

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

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

Pour contrôler si les données d'identification sont vérifiées pour les connexions établies par des applications liées localement, utilisez l'attribut AUTHINFO **CHCKLOCL** (vérifier les connexions locales). Pour contrôler si les données d'identification sont vérifiées pour les connexions établies par les applications client, utilisez l'attribut AUTHINFO **CHCKCLNT** (vérifier les connexions client).

CHCKLOCL accepte les valeurs de `NONE` et `OPTIONAL`, et **CHCKCLNT** autorise la valeur de `NONE` pour les exigences d'authentification à configurer:

Aucun
Les données d'authentification fournies par les applications ne sont pas vérifiées.

Facultatif
S'assure que toutes les données d'identification fournies par une application sont valides. Toutefois, il n'est pas obligatoire pour les applications de fournir des données d'authentification. Cette option peut être utile lors de la migration, par exemple.

Si vous:

- Indiquez le nom d'utilisateur et le mot de passe, ils sont authentifiés.
- N'indiquez pas le nom d'utilisateur et le mot de passe, la connexion est autorisée.
- Indiquez le nom d'utilisateur, mais pas le mot de passe pour lequel vous recevez une erreur.

Important : OPTIONAL est la valeur minimale que vous pouvez définir si vous souhaitez également définir une option plus restrictive dans les règles d'authentification de canal (CHLAUTH).

Si vous sélectionnez NONE et que la connexion client correspond à un enregistrement CHLAUTH avec **CHCKCLNT** défini sur REQUIRED (ou REQDADM sur les plateformes autres que z/OS), la connexion échoue. Vous recevez le message AMQ9793 sur Multiplatforms, et le message CSQX793E sur z/OS.

Pour plus d'informations sur l'utilisation des règles d'authentification de canal afin de définir des options **CHCKCLNT** plus restrictives pour certaines connexions client, voir «Granularité de la configuration», à la page 78.

required
Exige que toutes les applications fournissent des données d'identification valides. Voir aussi la remarque suivante.

REQDADM
Les utilisateurs privilégiés doivent fournir des données d'identification valides, mais les utilisateurs non privilégiés sont traités comme avec le paramètre OPTIONAL . Voir aussi la remarque suivante.

(Ce paramètre n'est pas autorisé sur les systèmes z/OS .)

Remarque :

Définir **CHCKLOCL** sur REQUIRED ou REQDADM signifie que vous ne pouvez pas administrer localement le gestionnaire de files d'attente à l'aide de **runmqsc** (erreur AMQ8135: Non autorisé) sauf si l'utilisateur spécifie le paramètre **-u** pour spécifier l'ID utilisateur dans la commande **runmqsc** . Lorsque ce paramètre est défini, **runmqsc** demande le mot de passe de l'utilisateur sur la console.

De même, un utilisateur qui exécute IBM MQ Explorer sur le système local verra l'erreur AMQ4036 lors de la tentative de connexion au gestionnaire de files d'attente. Pour spécifier un ID utilisateur et un mot de passe, cliquez avec le bouton droit de la souris sur l'objet du gestionnaire de files d'attente local et sélectionnez **Détails de connexion** > **Propriétés ...** dans le menu. Dans la section **ID utilisateur** , entrez l'ID utilisateur et le mot de passe à utiliser, puis cliquez sur **OK**.

Des considérations similaires s'appliquent aux connexions distantes avec **CHCKCLNT**.

L'attribut **CONNAUTH** du gestionnaire de files d'attente est vide pour les gestionnaires de files d'attente migrés à partir de versions antérieures à IBM MQ 8.0, mais il est défini sur *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* pour les gestionnaires de files d'attente nouvellement créés. Cette définition **AUTHINFO** par défaut a **CHCKCLNT** défini sur REQDADM par défaut.

Par conséquent, tous les clients existants qui utilisent un ID utilisateur privilégié pour se connecter doivent fournir des données d'identification valides.

Avertissement : Les données d'identification d'une structure MQCSP pour une application client sont parfois envoyées sur le réseau en texte en clair. Pour vous assurer que les données d'identification du client sont protégées, voir «Protection par mot de passe MQCSP», à la page 32.

Granularité de la configuration

Les attributs **CHCKLOCL** et **CHCKCLNT** de l'objet AUTHINFO définissent les exigences d'authentification pour toutes les connexions au gestionnaire de files d'attente. En plus de ces attributs, l'attribut **CHCKCLNT** des règles d'authentification de canal (CHLAUTH) permet de définir des exigences d'authentification plus strictes pour des connexions client spécifiques qui correspondent à la règle CHLAUTH.

Vous pouvez définir la valeur **CHCKCLNT** globale sur OPTIONAL, par exemple, sur l'objet AUTHINFO, puis la mettre à niveau pour qu'elle soit plus stricte pour certains canaux en définissant **CHCKCLNT** sur REQUIRED ou REQDADM sur la règle CHLAUTH. Par défaut, les règles CHLAUTH sont définies avec **CHCKCLNT(ASQMGR)**, de sorte que cette granularité n'a pas besoin d'être utilisée. Par exemple, ces commandes MQSC définissent une règle CHLAUTH qui remplace l'attribut **CHCKCLNT** de l'objet AUTHINFO et une règle CHLAUTH qui ne:

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

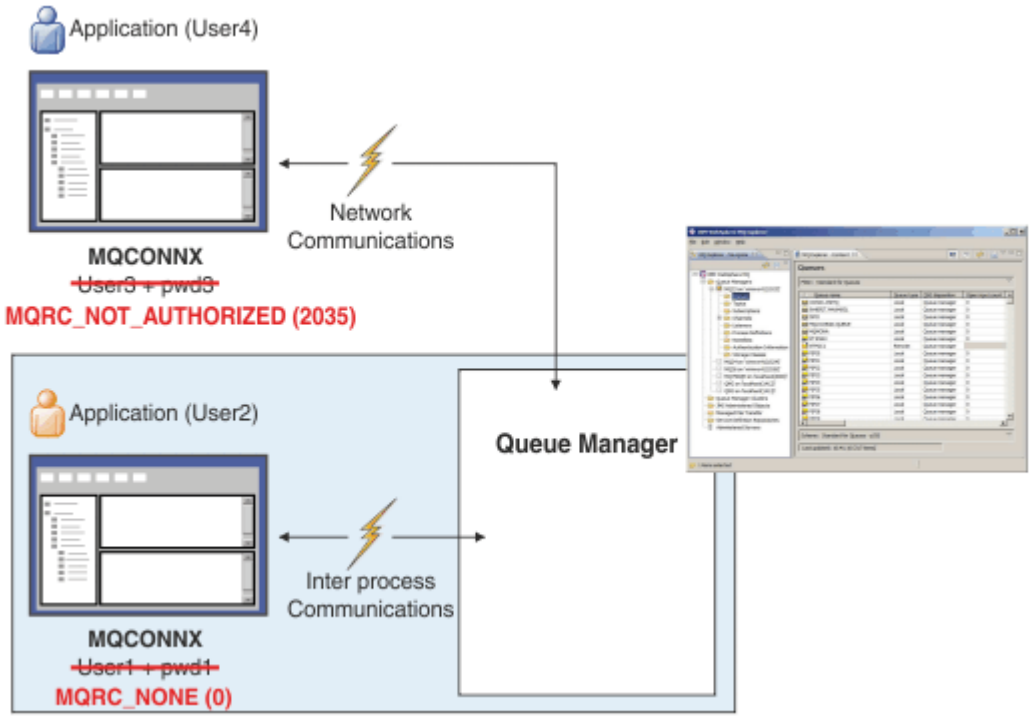
SET CHLAUTH('*') TYPE(ADDRESSMAP) +
```

```
ADDRESS('*') USERSRC(CHANNEL) +
CHCKCLNT(REQUIRED)

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

Pour plus d'informations sur les règles CHLAUTH, voir «Enregistrements d'authentification de canal», à la page 54.

Notification d'erreur



Une erreur est enregistrée dans les situations suivantes:

- Une application ne fournit pas de données d'authentification lorsqu'elles sont requises.
- Une application fournit des données d'authentification non valides. Cette situation est traitée comme une erreur même si la configuration indique qu'il est facultatif pour les applications de fournir des données d'identification.

Remarque : Lorsque **CHCKLOCL** ou **CHCKCLNT** est défini sur NONE, les données d'identification non valides fournies par les applications ne sont pas détectées.

Les échecs d'authentification sont conservés pendant le nombre de secondes spécifié par l'attribut **FAILDLAY** avant que l'erreur ne soit renvoyée à l'application. Ce délai offre une certaine protection contre les tentatives répétées de connexion d'une application.

L'erreur est enregistrée de plusieurs manières:

Application
Un code anomalie MQRC_NOT_AUTHORIZED (2035) est renvoyé à l'application.

Administrateur
Un administrateur IBM MQ voit l'événement signalé dans le journal des erreurs. Le message d'erreur indique que la connexion est rejetée car les données d'identification ne sont pas valides, plutôt que parce que, par exemple, l'utilisateur ne dispose pas des droits de connexion.

Outil de surveillance

Un outil de surveillance peut également être averti de l'échec, si vous activez des événements de droits d'accès, par un message d'événement dans la file d'attente SYSTEM.ADMIN.QMGR.EVENT . Pour activer les événements de droits d'accès, exécutez la commande MQSC suivante:

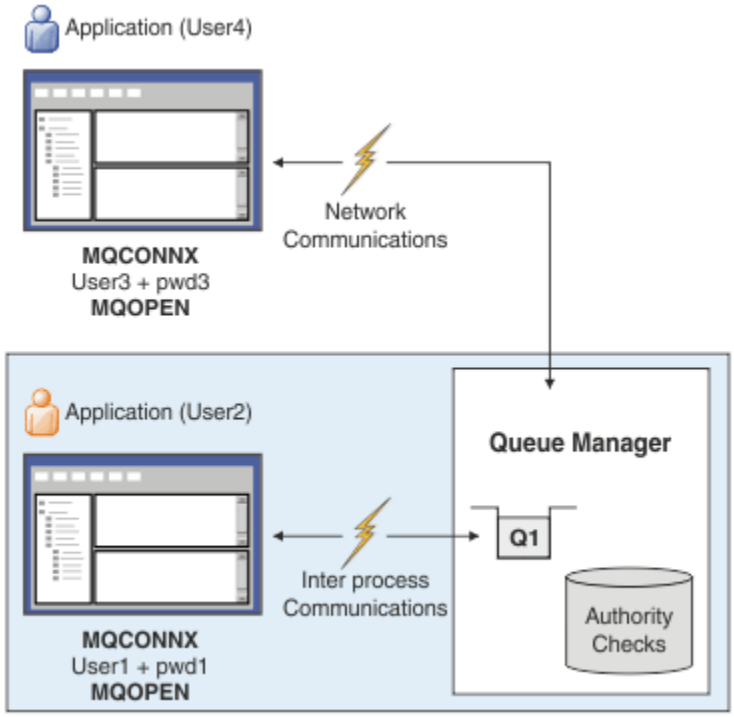
```
ALTER QMGR AUTHOREV(ENABLED)
```

Cet événement "Non autorisé" est un événement de connexion de type 1 et fournit les mêmes zones que les autres événements de type 1, avec une zone supplémentaire, l'ID utilisateur MQCSP fourni. Si l'application a fourni un mot de passe, il n'est pas inclus dans le message d'événement. Cela signifie que le message d'événement contient deux ID utilisateur:

- ID utilisateur sous lequel l'application s'exécute.
- ID utilisateur dans les données d'identification présentées par l'application.

Pour plus d'informations sur ce message d'événement, voir Non autorisé (type 1).

Adoption d'utilisateurs pour autorisation



Vous pouvez configurer le gestionnaire de files d'attente pour qu'il adopte les données d'identification présentées par l'application comme contexte de la connexion. L'adoption des données d'identification signifie que l'ID utilisateur fourni dans les données d'authentification est utilisé pour les vérifications d'autorisation, affiché sur les écrans d'administration et apparaît dans les messages. L'attribut **ADOPTCTX** de l'objet AUTHINFO contrôle si les données d'identification sont adoptées en tant que contexte pour l'application. Par exemple, les commandes MQSC suivantes définissent un objet AUTHINFO nommé USE.PWD qui est utilisé pour l'authentification de connexion et définissent l'attribut **ADOPTCTX** sur YES:

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

ALTER QMGR CONNAUTH(USE.PWD)
```

Les valeurs suivantes peuvent être spécifiées pour l'attribut **ADOPTCTX** :

ADOPTCTX (OUI)
Les données d'identification fournies par l'application sont adoptées en tant que contexte d'application pour la durée de la connexion. Toutes les vérifications d'autorisation pour une application sont effectuées avec l'ID utilisateur dans les données d'identification qui ont été authentifiées.

Avertissement : Lorsque vous utilisez des ID utilisateur **ADOPTCTX(YES)** et du système d'exploitation local, vous devez vous assurer que l'ID utilisateur adopté répond aux exigences relatives aux ID utilisateur dans IBM MQ. Pour plus d'informations, voir «ID utilisateur», à la page 94.

ADOPTCTX (NON)
Les données d'identification fournies par une application sont utilisées uniquement pour l'authentification lors de la connexion. L'ID utilisateur sous lequel l'application s'exécute continue d'être utilisé pour les vérifications d'autorisation ultérieures. Cette option peut s'avérer utile lors de la migration ou si vous prévoyez d'utiliser d'autres mécanismes, tels que des enregistrements d'authentification de canal, pour affecter l' ID utilisateur de l'agent de canal de message (MCAUSER).

Interaction avec l'authentification de canal

Les règles d'authentification de canal peuvent être utilisées pour modifier l'ID utilisateur utilisé comme contexte pour une connexion d'application, en fonction de l'ID utilisateur reçu du client. Pour un exemple d'utilisation d'une règle d'authentification de canal pour modifier l'ID utilisateur associé à une connexion, voir «Mappage d'un ID utilisateur client à un ID utilisateur MCAUSER», à la page 400.

L'ordre dans lequel les règles d'authentification de connexion et d'authentification de canal sont traitées est un facteur important dans la détermination du contexte de sécurité pour les connexions d'application client IBM MQ . Le paramètre **ChlauthEarlyAdopt** dans la section **channels** du fichier `qm.ini` contrôle l'ordre dans lequel le gestionnaire de files d'attente adopte le contexte à partir des données d'identification fournies par l'application et applique les règles d'authentification de canal. Pour plus d'informations sur **ChlauthEarlyAdopt**, voir Attributs de la strophe channels.

Avertissement : Lorsque vous utilisez le paramètre **ADOPTCTX(YES)** sur l'objet d'informations d'authentification, le contexte adopté à partir des données d'identification fournies par l'application ne peut être modifié par les règles d'authentification de canal que si le paramètre **ChlauthEarlyAdopt** est défini sur Y.

Pour plus d'informations sur l'interaction entre l'authentification de connexion et l'authentification de canal, ainsi que sur l'ordre dans lequel les vérifications sont effectuées lorsqu'une application client se connecte à un gestionnaire de files d'attente, voir «Interaction entre CHLAUTH et CONNAUTH», à la page 60.

Concepts associés

«Authentification de connexion», à la page 75
L'authentification de connexion permet aux applications de fournir des données d'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente. Le gestionnaire de files d'attente valide les données d'identification. L'ID utilisateur fourni dans les données d'identification peut également être utilisé dans les vérifications d'autorisation pour les ressources auxquelles l'application accède.

«Authentification de la connexion: modifications de l'application», à la page 81

«Authentification de connexion: référentiels d'utilisateurs», à la page 82
Pour chacun de vos gestionnaires de files d'attente, vous pouvez choisir différents types d'objet d'informations d'authentification pour l'authentification des ID utilisateur et des mots de passe.

Authentification de la connexion: modifications de l'application

Une application qui utilise l'interface de file d'attente de messages (MQI) peut fournir un ID utilisateur et un mot de passe dans la structure des paramètres de sécurité de connexion (MQCSP) lorsque MQCONNX est appelé. Dans les autres interfaces de programme d'application, la structure MQCSP est généralement construite pour le compte de l'application par les bibliothèques IBM MQ .

V 9.4.0 Depuis la IBM MQ 9.3.4, les applications client qui se connectent à un gestionnaire de files d'attente exécuté sur des systèmes AIX ou Linux peuvent également envoyer un jeton d'authentification dans la structure MQCSP comme moyen d'identification alternatif.

L'ID utilisateur et le mot de passe, ou jeton d'authentification, sont transmis pour vérification au gestionnaire des droits d'accès aux objets (OAM) fourni avec le gestionnaire de files d'attente ou au composant de service d'autorisation fourni avec le gestionnaire de files d'attente sur les systèmes z/OS . Vous n'avez pas besoin d'écrire votre propre interface personnalisée.

Si l'application s'exécute en tant que client, l'ID utilisateur et le mot de passe, ou le jeton d'authentification, est également transmis aux exits de sécurité côté client et côté serveur pour traitement. Ils peuvent également être utilisés pour définir l' attribut d'ID utilisateur d'agent de canal de message (MCAUSER) d'une instance de canal.

Avertissement : Les données d'identification d'une structure MQCSP pour une application client sont parfois envoyées sur le réseau en texte en clair. Pour vous assurer que les données d'identification de l'application client sont protégées, voir «Protection par mot de passe MQCSP», à la page 32.

En utilisant la chaîne XAOPEN pour fournir un ID utilisateur et un mot de passe, vous pouvez éviter d'avoir à modifier le code de l'application.

Remarque :

Depuis la IBM WebSphere MQ 6.0, l'exit de sécurité permet de définir le MQCSP. Par conséquent, les clients de ce niveau ou d'un niveau ultérieur n'ont pas besoin d'être mis à niveau.

Toutefois, dans les versions de IBM MQ antérieures à IBM MQ 8.0, MQCSP ne plaçait aucune restriction sur l'ID utilisateur et le mot de passe fournis par l'application. Lorsque vous utilisez ces valeurs avec des fonctions fournies par IBM MQ , il existe des limites qui s'appliquent à l'utilisation de ces fonctions, mais si vous les transmettez uniquement à vos propres exits, ces limites ne s'appliquent pas.

Concepts associés

«Authentification de connexion», à la page 75
L'authentification de connexion permet aux applications de fournir des données d'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente. Le gestionnaire de files d'attente valide les données d'identification. L'ID utilisateur fourni dans les données d'identification peut également être utilisé dans les vérifications d'autorisation pour les ressources auxquelles l'application accède.

«Authentification de connexion: Configuration», à la page 76
Un gestionnaire de files d'attente peut être configuré pour authentifier les données d'identification fournies par une application lorsqu'elle se connecte.

«Authentification de connexion: référentiels d'utilisateurs», à la page 82
Pour chacun de vos gestionnaires de files d'attente, vous pouvez choisir différents types d'objet d'informations d'authentification pour l'authentification des ID utilisateur et des mots de passe.

Authentification de connexion: référentiels d'utilisateurs

Pour chacun de vos gestionnaires de files d'attente, vous pouvez choisir différents types d'objet d'informations d'authentification pour l'authentification des ID utilisateur et des mots de passe.

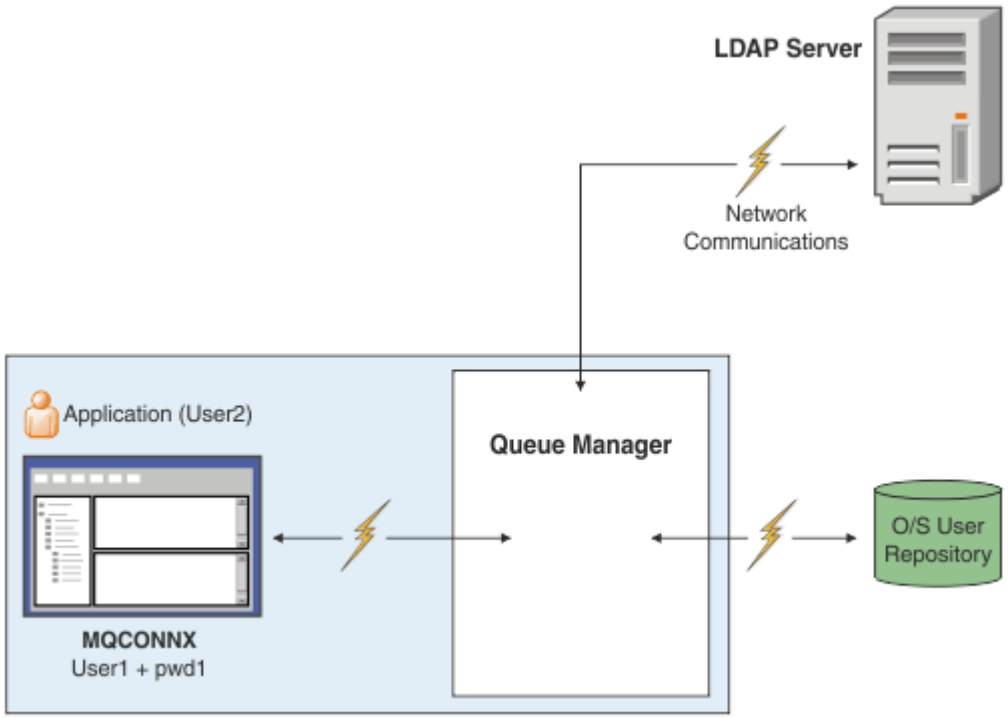


Figure 7. Types d'objets d'informations d'authentification

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

Il existe deux types d'objet d'informations d'authentification, comme représenté dans le diagramme:

- `IDPWOS` est utilisé pour indiquer que le gestionnaire de files d'attente utilise le système d'exploitation local pour authentifier l'ID utilisateur et le mot de passe. Si vous choisissez d'utiliser le système d'exploitation local, vous devez définir les attributs communs, comme décrit dans les rubriques précédentes.
- `IDPWLDAP` est utilisé pour indiquer que le gestionnaire de files d'attente utilise un serveur LDAP pour authentifier l'ID utilisateur et le mot de passe. Si vous choisissez d'utiliser un serveur LDAP, des informations supplémentaires sont fournies dans cette rubrique.

Un seul type d'objet d'informations d'authentification peut être choisi pour chaque gestionnaire de files d'attente à utiliser, en nommant l'objet approprié dans l'attribut **CONNAUTH** du gestionnaire de files d'attente.

Utilisation d'un serveur LDAP pour l'authentification.

Définissez la zone **CONNAME** sur l'adresse du serveur LDAP pour le gestionnaire de files d'attente. Vous pouvez fournir des adresses supplémentaires pour le serveur LDAP dans une liste séparée par des virgules, ce qui peut faciliter la redondance si le serveur LDAP ne fournit pas cette fonction lui-même.

Définissez l'ID et le mot de passe du serveur LDAP requis dans les zones **LDAPUSER** et **LDAPPWD** afin que le gestionnaire de files d'attente puisse accéder au serveur LDAP et rechercher des informations sur les enregistrements utilisateur.

Connexion sécurisée à un serveur LDAP

Contrairement aux canaux, il n'existe pas de paramètre **SSLCIPH** pour activer l'utilisation de TLS pour la communication avec le serveur LDAP. Dans ce cas, IBM MQ agit en tant que client sur le serveur LDAP et une grande partie de la configuration est effectuée sur le serveur LDAP. Certains paramètres existants dans IBM MQ sont utilisés pour configurer le fonctionnement de cette connexion.

Définissez la zone **SECCOMM** pour contrôler si la connectivité au serveur LDAP utilise TLS.

En plus de cet attribut, les attributs de gestionnaire de files d'attente **SSLFIPS** et **SUITEB** restreignent l'ensemble des spécifications de chiffrement qui sont choisies. Le certificat utilisé pour identifier le gestionnaire de files d'attente sur le serveur LDAP est le certificat du gestionnaire de files d'attente, `ibmwebspheremq` *`qmgr-name`* ou la valeur de l'attribut **CERTLABL** . Pour plus de détails voir Labels de certificat numérique.

Référentiel d'utilisateurs LDAP

Lors de l'utilisation d'un référentiel d'utilisateurs LDAP, d'autres opérations de configuration doivent être effectuées sur le gestionnaire de files d'attente, mais pas uniquement pour indiquer au gestionnaire de files d'attente où trouver le serveur LDAP.

Les ID utilisateur définis dans un serveur LDAP possèdent une structure hiérarchique qui les identifie de manière unique. Par conséquent, une application peut se connecter au gestionnaire de files d'attente et présenter son ID utilisateur en tant qu'ID utilisateur hiérarchique qualifié complet.

Toutefois, pour simplifier les informations qu'une application doit fournir, il est possible de configurer le gestionnaire de files d'attente pour qu'il considère que la première partie de la hiérarchie est commune à tous les ID et de l'ajouter automatiquement avant l'ID abrégé fourni par l'application. Le gestionnaire de files d'attente peut alors présenter un ID complet au serveur LDAP.

Définissez BASEDNU sur le point initial où la recherche LDAP recherche l'ID dans la hiérarchie LDAP. Lorsque vous définissez BASEDNU, vous devez vous assurer qu'un seul résultat est renvoyé lorsque vous recherchez l'ID dans la hiérarchie LDAP.

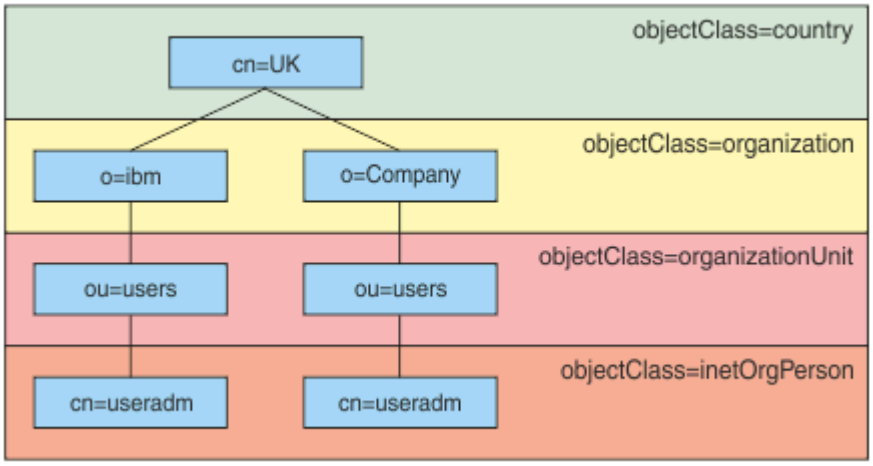


Figure 8. Exemple de hiérarchie LDAP

Par exemple, dans Figure 8, à la page 84 BASEDNU, la valeur peut être "ou=users, o=ibm, c = UK" ou ", o=ibm, c = UK". Cependant, comme un nom distinctif contenant "cn = useradm" existe à la fois dans la branche "o = ibm" et dans la branche "o=Company", BASEDNU ne peut pas être défini sur "c = UK". Pour des raisons de performances et de sécurité, utilisez le point le plus élevé de votre hiérarchie LDAP à partir duquel vous pouvez référencer tous les ID utilisateur dont vous avez besoin. Dans cet exemple, il s'agit de "ou=users, o=ibm, c = UK".

Votre application peut soumettre au gestionnaire de files d'attente l'ID utilisateur sans fournir le nom d'attribut LDAP, CN= par exemple. Si vous définissez USRFIELD sur le nom d'attribut LDAP, cette valeur

est ajoutée en tant que préfixe à l'ID utilisateur provenant de l'application. Il peut s'agir d'une aide à la migration utile lorsque vous passez des ID utilisateur du système d'exploitation aux ID utilisateur LDAP, car l'application peut alors présenter la même chaîne dans les deux cas et vous pouvez éviter de modifier l'application.

Par conséquent, l'ID utilisateur complet présenté au serveur LDAP se présente comme suit:

```
USRFIELD = ID_from_application BASEDNU
```

Concepts associés

«Authentification de connexion», à la page 75
L'authentification de connexion permet aux applications de fournir des données d'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente. Le gestionnaire de files d'attente valide les données d'identification. L'ID utilisateur fourni dans les données d'identification peut également être utilisé dans les vérifications d'autorisation pour les ressources auxquelles l'application accède.

«Authentification de connexion: Configuration», à la page 76
Un gestionnaire de files d'attente peut être configuré pour authentifier les données d'identification fournies par une application lorsqu'elle se connecte.

«Authentification de la connexion: modifications de l'application», à la page 81

Exit de sécurité côté client pour l'insertion d'un ID utilisateur et d'un mot de passe (`mqccred`)

Si vous disposez d'applications client qui sont requises pour envoyer un ID utilisateur ou un mot de passe mais que vous ne pouvez pas encore modifier la source, un exit de sécurité fourni avec IBM MQ 8.0 appelé **`mqccred`** est disponible. **`mqccred`** fournit un ID utilisateur et un mot de passe pour le compte de l'application client, à partir d'un fichier `.ini` . Cet ID utilisateur et ce mot de passe sont envoyés au gestionnaire de files d'attente qui, s'il est configuré pour le faire, les authentifiera.

Présentation

`mqccred` est un exit de sécurité qui s'exécute sur la même machine que votre application client. Il permet de fournir des informations d'ID utilisateur et de mot de passe pour le compte de l'application client, lorsque ces informations ne sont pas fournies par l'application elle-même. Les informations d'ID utilisateur et de mot de passe sont fournies dans une structure appelée Connection Security Parameters (MQCSP) et sont authentifiées par le gestionnaire de files d'attente si l' authentification de connexion est configurée.

Les informations d'ID utilisateur et de mot de passe sont extraites d'un fichier `.ini` sur la machine client. Les mots de passe du fichier sont protégés par le brouillage à l'aide de la commande **`runmqccred`** et en s'assurant que les droits d'accès au fichier `.ini` sont définis de sorte que seul l'ID utilisateur exécutant l'application client (et donc l'exit) puisse le lire.

Emplacement

`mqccred` est installé:

Windows plateformes
Dans le répertoire *`installation_directory`*`\Tools\c\Samples\mqccred\`

AIX and Linux plateformes
Dans le répertoire *`installation_directory`*`/samp/mqccred`

Remarques : L'exit:

1. Agit uniquement comme un exit de canal de sécurité et doit être le seul exit de ce type défini sur un canal.
2. Est généralement nommé via la table de définition de canal du client (CCDT), mais un client Java peut avoir directement l'exit mentionné dans les objets JNDI, ou l'exit peut être configuré pour les applications qui construisent manuellement la structure MQCD .

3. Vous devez copier les programmes **mqccred** et **mqccred_r** dans le répertoire `var/mqm/exits` .

   Par exemple, sur un système AIX ou Linux 64 bits, exécutez la commande suivante:

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

   Pour plus d'informations, voir Exemple étape par étape de test de mqccred .
4. Est capable de s'exécuter sur des versions précédentes d' IBM MQ, aussi anciennes que IBM WebSphere MQ 7.0.1.

Configuration des ID utilisateur et des mots de passe

Le fichier `.ini` contient des sections pour chaque gestionnaire de files d'attente, avec un paramètre global pour les gestionnaires de files d'attente non spécifiés. Chaque section contient le nom du gestionnaire de files d'attente, un ID utilisateur et un texte en clair ou un mot de passe brouillé.

Vous devez éditer le fichier `.ini` à la main, à l'aide de l'éditeur de votre choix, et ajouter l'attribut de mot de passe en texte en clair aux sections. Exécutez le programme **runmqccred** fourni, qui utilise le fichier `.ini` et remplace l'attribut **Password** par l'attribut **OPW** , une forme brouillée du mot de passe.

Voir runmqccred pour une description de la commande et de ses paramètres.

Le fichier `mqccred.ini` contient vos informations d'ID utilisateur et de mot de passe.

Un modèle de fichier `.ini` est fourni dans le même répertoire que l'exit pour fournir un point de départ à votre entreprise.

Par défaut, ce fichier est recherché dans $HOME/`.mqs/mqccred.ini`. Si vous souhaitez le localiser ailleurs, vous pouvez utiliser la variable d'environnement *MQCCRED* pour le localiser:

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

Si vous utilisez *MQCCRED*, la variable doit inclure le nom complet du fichier de configuration, y compris tout type de fichier `.ini` . Etant donné que ce fichier contient des mots de passe (même s'ils sont brouillés), vous devez protéger le fichier à l'aide des privilèges du système d'exploitation pour vous assurer que les personnes non autorisées ne peuvent pas le lire. Si vous ne disposez pas des droits d'accès au fichier appropriés, l'exit ne s'exécutera pas correctement.

Si l'application a déjà fourni une structure MQCSP , l'exit le respecte normalement et n'insère aucune information du fichier `.ini` . Toutefois, vous pouvez le remplacer à l'aide de l'attribut **Force** de la section.

La définition de **Force** sur la valeur *TRUE* supprime l'ID utilisateur et le mot de passe fournis par l'application et les remplace par la version du fichier ini.

Vous pouvez également définir l'attribut **Force** dans la section globale du fichier pour définir la valeur par défaut de ce fichier.

La valeur par défaut de **Force** est *FALSE*.

Vous pouvez fournir un ID utilisateur et un mot de passe pour tous les gestionnaires de files d'attente ou pour chaque gestionnaire de files d'attente individuel. Voici un exemple de fichier `mqccred.ini` :

```
# 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
```

Remarques :

1. Les définitions de gestionnaire de files d'attente individuelles sont prioritaires sur le paramètre global.
2. Les attributs sont insensibles à la casse.

Contraintes

Lorsque cet exit est utilisé, l'ID utilisateur local de la personne exécutant l'application n'est pas transmis du client au serveur. Les seules informations d'identité disponibles proviennent du contenu du fichier ini.

Par conséquent, vous devez configurer le gestionnaire de files d'attente pour qu'il utilise **ADOPTCTX(YES)** ou mapper la demande de connexion entrante à un ID utilisateur approprié via l'un des mécanismes disponibles, par exemple, «Enregistrements d'authentification de canal», à la page 54.

Important : Si vous ajoutez de nouveaux mots de passe ou mettez à jour les anciens, la commande **runmqccred** ne traite que les mots de passe en texte en clair, sans toucher à ceux qui sont brouillés.

Débogage

L'exit écrit dans la trace IBM MQ standard lorsqu'elle est activée.

Pour faciliter le débogage des problèmes de configuration, l'exit peut également écrire directement dans stdout.

Aucune donnée d'exit de sécurité de canal (**SCYDATA**) la configuration est normalement requise pour le canal. Toutefois, vous pouvez spécifier:

ERREUR
Seules les informations d'impression sont associées à des conditions d'erreur, telles que l'impossibilité de trouver le fichier de configuration.

DEBOGAGE
Affiche ces conditions d'erreur et des instructions de trace supplémentaires.

NOCHECKS
Ignore les contraintes sur les droits d'accès aux fichiers et la contrainte supplémentaire selon laquelle le fichier `.ini` ne doit pas contenir de mots de passe non protégés.

Vous pouvez placer un ou plusieurs de ces éléments dans la zone **SCYDATA** , séparés par des virgules, dans n'importe quel ordre. Par exemple, `SCYDATA=(NOCHECKS,DEBUG)`.

Notez que les éléments sont sensibles à la casse et doivent être entrés en majuscules.

Utilisation `mqccred`

Une fois votre fichier configuré, vous pouvez appeler l'exit de canal en mettant à jour votre définition de canal de connexion client pour inclure l'attribut `SCYEXIT('mqccred(ChlExit)')` :

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

Référence associée
SCYDATA
SCYEXIT
runmqccred

Authentification de connexion avec le client Java

L'authentification de connexion est une fonction d' IBM MQ qui permet de configurer les gestionnaires de files d'attente de sorte qu'ils puissent authentifier les applications à l'aide d'un ID utilisateur et d'un mot de passe fournis. Lorsque l'application est une application Java qui utilise le transport client, l'authentification de connexion peut être exécutée en mode compatibilité ou en mode d'authentification MQCSP.

L'ID utilisateur et le mot de passe à authentifier sont spécifiés par l'application à l'aide de l'une des méthodes suivantes:

- Dans une application IBM MQ classes for Java , dans la classe `MQEnvironment` ou dans la table de hachage des propriétés qui est transmise au constructeur `com.ibm.mq.MQQueueManager` .
- Dans une application IBM MQ classes for JMS , en tant qu'arguments de la méthode `createConnection(String username, String Password)` ou `createContext(String username, String password)` .

Mode d'authentification MQCSP

Dans ce mode, l'ID utilisateur côté client sous lequel l'application s'exécute est envoyé au gestionnaire de files d'attente, ainsi que l'ID utilisateur et le mot de passe à authentifier. IBM MQ classes for Java et IBM MQ classes for JMS envoient l'ID utilisateur et le mot de passe à authentifier auprès du gestionnaire de files d'attente dans une structure MQCSP .

L'ID utilisateur et le mot de passe sont disponibles pour un exit de sécurité de connexion serveur dans la structure MQCSP. L'adresse de la structure MQCSP se trouve dans la zone **`SecurityParms`** de la structure MQCXP du canal.

Le mode d'authentification MQCSP offre les avantages suivants:

- La longueur maximale de l'ID utilisateur à authentifier est de 1024 caractères.
- La longueur maximale du mot de passe pour l'authentification est de 256 caractères.
- Les vérifications d'autorisation d'accès pour l'utilisation des ressources IBM MQ peuvent être effectuées à l'aide de l'ID utilisateur côté client sous lequel l'application s'exécute, lorsque l'objet d'informations d'authentification utilisé pour contrôler l'authentification de connexion sur le gestionnaire de files d'attente est configuré avec ADOPTCTX (NO).

Mode compatibilité

Avant IBM MQ 8.0, le client Java pouvait envoyer un ID utilisateur et un mot de passe via le canal de connexion client au canal de connexion serveur, et les fournir à un exit de sécurité dans les zones **`RemoteUserIdentifier`** et **`RemotePassword`** de la structure MQCD. En mode compatibilité, ce comportement est conservé.

Vous pouvez utiliser ce mode en combinaison avec l'authentification de connexion et effectuer une migration à partir de tous les exits de sécurité précédemment utilisés pour effectuer le même travail.

Ce mode comporte les restrictions suivantes:

- La longueur de l'ID utilisateur et du mot de passe doit être inférieure ou égale à 12 caractères. Les ID utilisateur de plus de 12 caractères sont tronqués à 12 caractères. Cela peut entraîner l'échec de la connexion avec le code anomalie MQRC_NOT_AUTHORIZED.
- L'ID utilisateur côté client sous lequel l'application s'exécute n'est pas envoyé au gestionnaire de files d'attente. Vous devez définir ADOPTCTX (YES) sur l'objet d'informations d'authentification qui est utilisé pour contrôler l'authentification de connexion sur le gestionnaire de files d'attente, ou utiliser une autre méthode, telle qu'une règle d'authentification de canal basée sur un certificat TLS, pour définir l'ID utilisateur MCA de canal dont l'autorisation d'utilisation des ressources IBM MQ est vérifiée.

Mode d'authentification par défaut

Le mode d'authentification par défaut utilisé par une application client IBM MQ classes for Java ou IBM MQ classes for JMS varie selon que l'application spécifie un ID utilisateur et un mot de passe.

- Si un ID utilisateur et un mot de passe sont spécifiés, l'authentification MQCSP est utilisée par défaut.
- Si un ID utilisateur, mais qu'aucun mot de passe n'est spécifié, le mode de compatibilité est utilisé par défaut.
- Si aucun ID utilisateur n'est spécifié, le mode de compatibilité est toujours utilisé.

Dans les cas où un ID utilisateur est spécifié, un mode d'authentification spécifique peut être choisi par l'application pour chaque connexion individuelle, ou défini globalement avant le démarrage de l'application, comme décrit dans «Choix du mode d'authentification», à la page 89.

Remarque : Les applications qui utilisent IBM MQ classes for JMS peuvent être affectées par la modification du mode d'authentification par défaut dans IBM MQ 9.3.0. Après la mise à niveau de IBM MQ classes for JMS vers IBM MQ 9.3.0, les applications qui utilisaient auparavant le mode de compatibilité par défaut utilisent l'authentification MQCSP à la place. Cela peut entraîner l'échec de la connexion des applications qui se sont précédemment connectées à un gestionnaire de files d'attente avec un `JMSException` contenant le code anomalie 2035 (MQRC_NOT_AUTHORIZED). Dans ce cas, utilisez l'une des méthodes décrites dans «Choix du mode d'authentification», à la page 89 pour indiquer que l'application utilise le mode compatibilité.

Les applications Java qui se connectent au gestionnaire de files d'attente à l'aide de liaisons locales utilisent toujours le mode d'authentification MQCSP.

Choix du mode d'authentification

Le mode d'authentification utilisé par les applications client Java qui spécifient un ID utilisateur lors de la connexion au gestionnaire de files d'attente peut être spécifié à l'aide de l'une des méthodes suivantes. Ces méthodes sont répertoriées par ordre de priorité décroissant. Si le mode d'authentification n'est pas spécifié à l'aide de l'une de ces méthodes, le mode d'authentification par défaut est utilisé.

Remarque : L'utilisation de ces méthodes pour sélectionner le mode d'authentification a été clarifiée dans IBM MQ 9.3.0. Dans certains cas, le mode d'authentification utilisé par une application client Java peut changer lorsque IBM MQ classes for Java ou IBM MQ classes for JMS sont mis à niveau vers IBM MQ 9.3.0. Cela peut entraîner l'échec de la connexion des applications qui se sont précédemment connectées à un gestionnaire de files d'attente avec un `JMSException` contenant le code anomalie 2035 (MQRC_NOT_AUTHORIZED). Dans ce cas, utilisez l'une des méthodes suivantes pour sélectionner le mode d'authentification requis.

- Spécifiez le mode d'authentification pour chaque connexion individuelle en définissant la propriété appropriée dans l'application avant de vous connecter au gestionnaire de files d'attente.
  - Lors de l'utilisation de IBM MQ classes for Java, définissez la propriété *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* dans la table de hachage des propriétés qui est transmise au constructeur `com.ibm.mq.MQQueueManager`.
  - Lorsque vous utilisez IBM MQ classes for JMS, définissez la propriété *JmsConstants.USER_AUTHENTICATION_MQCSP* sur la fabrique de connexions appropriée avant de créer la connexion.

  Définissez la valeur de ces propriétés sur l'une des valeurs suivantes:

  Oui
  Utilisez le mode d'authentification MQCSP lors de l'authentification avec un gestionnaire de files d'attente.

  false
  Utilisez le mode compatibilité lors de l'authentification avec un gestionnaire de files d'attente.
- Indiquez le mode d'authentification pour toutes les connexions client établies par une application en définissant la propriété système *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java lors du démarrage de l'application. Définissez la valeur de la propriété sur l'une des valeurs suivantes:

Y
Utilisez le mode d'authentification MQCSP lors de l'authentification avec un gestionnaire de files d'attente.

N
Utilisez le mode compatibilité lors de l'authentification avec un gestionnaire de files d'attente.

Par exemple, la commande suivante définit la propriété permettant de sélectionner le mode de compatibilité et démarre une application Java :

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

- Spécifiez le mode d'authentification pour toutes les connexions client établies par les applications démarrées dans le même environnement en définissant la variable d'environnement *com.ibm.mq.jmqi.useMQCSPauthentication* dans l'environnement dans lequel l'application est démarrée. Définissez la valeur de la variable d'environnement sur l'une des valeurs suivantes:

  Y
  Utilisez le mode d'authentification MQCSP lors de l'authentification avec un gestionnaire de files d'attente.

  N
  Utilisez le mode compatibilité lors de l'authentification avec un gestionnaire de files d'attente.
- Indiquez le mode d'authentification pour toutes les applications qui utilisent un fichier de configuration client IBM MQ MQI client spécifique en spécifiant l'attribut **useMQCSPauthentication** dans la section JMQI du fichier de configuration client. Définissez la valeur de l'attribut sur l'une des valeurs suivantes:

  YES
  Utilisez le mode d'authentification MQCSP lors de l'authentification avec un gestionnaire de files d'attente.

  NO
  Utilisez le mode compatibilité lors de l'authentification avec un gestionnaire de files d'attente.

  Pour plus d'informations sur l'attribut **useMQCSPauthentication** , voir Strophe JMQI du fichier de configuration du client.

Choix du mode d'authentification dans IBM MQ Explorer

IBM MQ Explorer étant une application Java , ces deux modes, le mode compatibilité et le mode d'authentification MQCSP, lui sont également applicables.

Le mode d'authentification MQCSP est le mode par défaut.

Dans les panneaux où l'identification de l'utilisateur est fournie, une case à cocher permet d'activer ou de désactiver le mode de compatibilité:

- Par défaut, cette case n'est pas cochée. Pour utiliser le mode compatibilité, cochez cette case.

Concepts associés

«Authentification de connexion», à la page 75
L'authentification de connexion permet aux applications de fournir des données d'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente. Le gestionnaire de files d'attente valide les données d'identification. L'ID utilisateur fourni dans les données d'identification peut également être utilisé dans les vérifications d'autorisation pour les ressources auxquelles l'application accède.

«Authentification de la connexion: modifications de l'application», à la page 81

«Authentification de connexion: référentiels d'utilisateurs», à la page 82
Pour chacun de vos gestionnaires de files d'attente, vous pouvez choisir différents types d'objet d'informations d'authentification pour l'authentification des ID utilisateur et des mots de passe.

Sécurité des messages dans IBM MQ

La sécurité des messages dans l'infrastructure IBM MQ est fournie par Advanced Message Security.

Advanced Message Security (AMS) développe les services de sécurité IBM MQ pour fournir la signature et le chiffrement des données au niveau des messages. Les services étendus garantissent que les données de message n'ont pas été modifiées entre le moment où elles sont placées à l'origine dans une file d'attente et le moment où elles sont extraites. En outre, AMS vérifie qu'un expéditeur de données de message est autorisé à placer des messages signés dans une file d'attente cible.

Concepts associés

«Advanced Message Security», à la page 617
Advanced Message Security (AMS) est un composant de IBM MQ qui offre un niveau de protection élevé pour les données sensibles transitant par le réseau IBM MQ , sans affecter les applications finales.

Planification de la sécurité

Cette collection de rubriques explique ce que vous devez prendre en compte lors de la planification de la sécurité dans un environnement IBM MQ .

Vous pouvez utiliser IBM MQ pour une grande variété d'applications sur une large gamme de plateformes. Les exigences de sécurité sont susceptibles d'être différentes pour chaque application. Pour certains, la sécurité sera une considération cruciale.

IBM MQ fournit une gamme de services de sécurité au niveau des liens, y compris la prise en charge du protocole TLS (Transport Layer Security).

Vous devez prendre en compte certains aspects de la sécurité lors de la planification de l'installation de IBM MQ:

- Multi Sous Multiplateformes, si vous ignorez ces aspects et ne faites rien, vous ne pouvez pas utiliser IBM MQ.
- z/OS Sous z/OS, le fait d'ignorer ces aspects a pour effet que vos ressources IBM MQ ne sont pas protégées. C'est-à-dire que tous les utilisateurs peuvent accéder à toutes les ressources IBM MQ et les modifier.

Droit d'administration de IBM MQ

Les administrateurs IBM MQ doivent disposer des droits suivants:

- Emettez des commandes pour administrer IBM MQ
- Utilisez laIBM MQ Explorer
- IBM i Utilisez les panneaux et les commandes d'administration IBM i .
- z/OS Utilisation des panneaux d'opérations et de contrôle sous z/OS
- z/OS Utilisez le programme utilitaire IBM MQ , CSQUTIL, sous z/OS
- z/OS Accès aux fichiers du gestionnaire de files d'attente sous z/OS

Pour plus d'informations, voir :

- ALW «Droit d'administration de IBM MQ sur AIX, Linux, and Windows», à la page 414
- IBM i «Droit d'administration de IBM MQ sur IBM i», à la page 96
- z/OS «Authority to administer IBM MQ on z/OS», à la page 97

Droits d'utilisation des objets IBM MQ

Les applications peuvent accéder aux objets IBM MQ suivants en émettant des appels MQI:

- Gestionnaires de files d'attente
- Files d'attente
- Processus
- Listes de noms
- Rubriques

Les applications peuvent également utiliser les commandes PCF (Programmable Command Format) pour accéder à ces objets IBM MQ , ainsi que pour accéder aux canaux et aux objets d'informations d'authentification. Ces objets peuvent être protégés par IBM MQ de sorte que les ID utilisateur associés aux applications aient besoin de droits d'accès.

Pour plus d'informations, voir «Autorisation pour les applications d'utiliser IBM MQ», à la page 99.

Sécurité des canaux

Les ID utilisateur associés aux agents MCA (Message Channel Agent) doivent être autorisés à accéder à diverses ressources IBM MQ . Par exemple, un agent MCA doit pouvoir se connecter à un gestionnaire de files d'attente. S'il s'agit d'un agent MCA émetteur, il doit pouvoir ouvrir la file d'attente de transmission du canal. S'il s'agit d'un agent MCA récepteur, il doit pouvoir ouvrir des files d'attente de destination. Les ID utilisateur associés aux applications qui doivent administrer les canaux, les initiateurs de canal et les programmes d'écoute doivent disposer des droits d'utilisation des commandes PCF appropriées. Cependant, la plupart des applications n'ont pas besoin d'un tel accès.

Pour plus d'informations, voir «Autorisation de canal», à la page 121.

Autres considérations

Vous devez prendre en compte les aspects suivants de la sécurité uniquement si vous utilisez certaines fonctions IBM MQ ou certaines extensions de produit de base:

- «Sécurité des clusters de gestionnaires de files d'attente», à la page 134
- «Sécurité pour la publication / abonnement IBM MQ», à la page 135

Planification de l'identification et de l'authentification

Choisissez les ID utilisateur à utiliser, ainsi que la manière et les niveaux auxquels vous souhaitez appliquer les contrôles d'authentification.

Vous devez décider de la manière dont vous allez identifier les utilisateurs de vos applications IBM MQ , en gardant à l'esprit que différents systèmes d'exploitation prennent en charge des ID utilisateur de longueurs différentes. Vous pouvez utiliser des enregistrements d'authentification de canal pour effectuer un mappage d'un ID utilisateur à un autre ou pour spécifier un ID utilisateur en fonction d'un attribut de la connexion. Les canaux IBM MQ utilisant TLS utilisent des certificats numériques comme mécanisme d'identification et d'authentification. Chaque certificat numérique possède un nom distinctif de sujet qui peut être mappé à des identités spécifiques à l'aide d'enregistrements d'authentification de canal. En outre, les certificats de l'autorité de certification dans le référentiel de clés déterminent quels certificats numériques peuvent être utilisés pour l'authentification auprès de IBM MQ. Pour plus d'informations, voir :

- «Mappage d'un gestionnaire de files d'attente éloignées à un ID utilisateur MCAUSER», à la page 399
- «Mappage d'un ID utilisateur client à un ID utilisateur MCAUSER», à la page 400
- «Mappage d'un nom distinctif SSL ou TLS à un ID utilisateur MCAUSER», à la page 401
- «Mappage d'une adresse IP à un ID utilisateur MCAUSER», à la page 403

Planification de l'authentification pour une application client

Vous pouvez appliquer des contrôles d'authentification à quatre niveaux: au niveau des communications, dans les exits de sécurité, avec des enregistrements d'authentification de canal et en termes d'identification transmise à un exit de sécurité.

Il y a quatre niveaux de sécurité à prendre en compte. Le diagramme montre un IBM MQ MQI client connecté à un serveur. La sécurité est appliquée à quatre niveaux, comme décrit dans le texte suivant. MCA est un agent MCA.

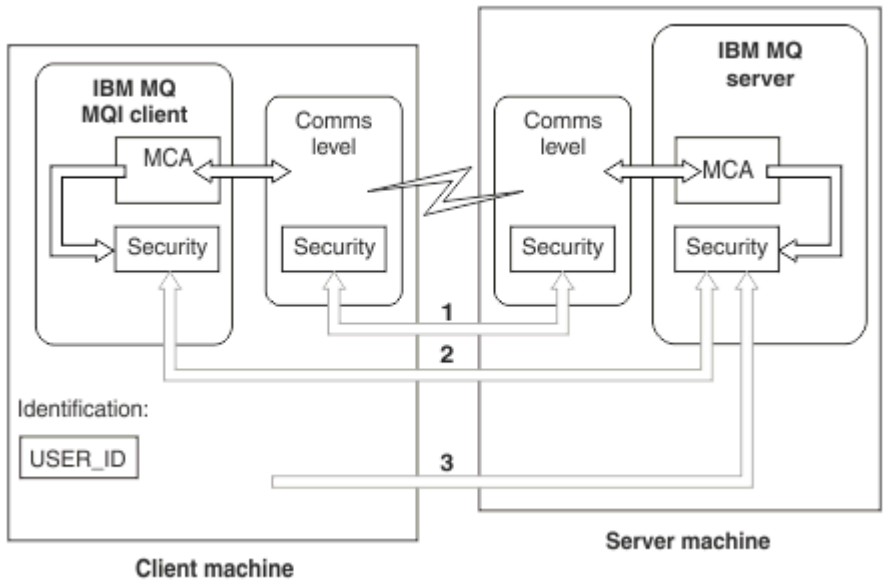


Figure 9. Sécurité dans une connexion client/serveur

1. Niveau de communication

   Voir la flèche 1. Pour implémenter la sécurité au niveau des communications, utilisez TLS. Pour plus d'informations, voir «Protocole de sécurité cryptographique TLS», à la page 18

2. Enregistrements d'authentification de canal

   Voir les flèches 2 et 3. L'authentification peut être contrôlée à l'aide de l'adresse IP ou des noms distinctifs TLS au niveau de la sécurité. Un ID utilisateur peut également être bloqué ou un ID utilisateur vérifié peut être mappé à un ID utilisateur valide. Une description complète est fournie dans «Enregistrements d'authentification de canal», à la page 54.

3. Authentification de connexion

   Voir la flèche 3. Le client envoie un ID utilisateur et un mot de passe, ou un jeton d'authentification. Pour plus d'informations, voir «Authentification de connexion: Configuration», à la page 76.

4. Exits de sécurité de canal

   Voir la flèche 2. Les exits de sécurité de canal pour la communication de client à serveur peuvent fonctionner de la même manière que pour la communication de serveur à serveur. Une paire d'exits indépendants du protocole peut être écrite pour permettre l'authentification mutuelle du client et du serveur. Une description complète est fournie dans Programmes d'exit de sécurité de canal.

5. Identification transmise à un exit de sécurité de canal

   Voir la flèche 3. Dans les communications client-serveur, les exits de sécurité de canal n'ont pas besoin de fonctionner en tant que paire. L'exit côté client IBM MQ peut être omis. Dans ce cas, l'ID utilisateur est placé dans le descripteur de canal (MQCD) et l'exit de sécurité côté serveur peut le modifier, si nécessaire.

   IBM MQ MQI clients envoie également des informations supplémentaires pour faciliter l'identification.

   - L'ID utilisateur transmis au serveur est l'ID utilisateur actuellement connecté sur le client.

- ID de sécurité de l'utilisateur actuellement connecté.

  Les valeurs de l'ID utilisateur et, si elles sont disponibles, de l'ID de sécurité, peuvent être utilisées par l'exit de sécurité du serveur pour établir l'identité du IBM MQ MQI client.

Depuis IBM MQ 8.0, vous pouvez envoyer des mots de passe inclus dans la structure MQCSP.

V 9.4.0 Linux AIX Depuis la IBM MQ 9.3.4, la IBM MQ MQI clients connexion à des gestionnaires de files d'attente IBM MQ s'exécutant sur des systèmes AIX ou Linux peut également envoyer des jetons d'authentification dans la structure MQCSP.

Avertissement : Dans certains cas, le mot de passe ou le jeton d'authentification dans une structure MQCSP pour une application client est envoyé sur le réseau en texte clair. Pour vous assurer que les mots de passe d'application client et les jetons d'authentification sont protégés de manière appropriée, voir «Protection par mot de passe MQCSP», à la page 32.

ID utilisateur

Lorsque vous créez des ID utilisateur pour des applications client, les ID utilisateur ne doivent pas dépasser la longueur maximale autorisée. Vous ne devez pas utiliser les ID utilisateur réservés UNKNOWN et NOBODY. Si le serveur auquel le client se connecte est un serveur IBM MQ for Windows , vous devez mettre fin à l'utilisation du signe @. La longueur autorisée des ID utilisateur dépend de la plateforme utilisée pour le serveur:

- Linux z/OS AIX Sous z/OS, AIX and Linux, la longueur maximale d'un ID utilisateur est de 12 caractères.
- IBM i Sous IBM i, la longueur maximale d'un ID utilisateur est de 10 caractères.
- Windows Sous Windows, si le serveur IBM MQ MQI client et le serveur IBM MQ sont sous Windowset que le serveur a accès au domaine dans lequel l'ID utilisateur client est défini, la longueur maximale d'un ID utilisateur est de 20 caractères. Toutefois, si le serveur IBM MQ n'est pas un serveur Windows , l'ID utilisateur est tronqué à 12 caractères.
- Si vous utilisez la structure MQCSP pour transmettre des données d'identification, la longueur maximale d'un ID utilisateur est de 1024 caractères. L'ID utilisateur de la structure MQCSP ne peut pas être utilisé pour contourner la longueur maximale de l'ID utilisateur utilisée par IBM MQ pour l'autorisation. Pour plus d'informations sur la structure MQCSP, voir «Identification et authentification des utilisateurs à l'aide de la structure MQCSP», à la page 331.

Sur les systèmes AIX and Linux , les ID utilisateur sont utilisés par défaut pour l'authentification et les groupes sont utilisés pour l'autorisation. Toutefois, vous pouvez configurer ces systèmes pour une autorisation sur les ID utilisateur. Pour plus d'informations, voir «Droits utilisateur OAM sur AIX and Linux», à la page 365. Les systèmes Windows peuvent utiliser les deux ID utilisateur pour l'authentification et l'autorisation et les groupes pour l'autorisation.

Si vous créez des comptes de service, sans tenir compte des groupes, et que vous autorisez tous les ID utilisateur différemment, chaque utilisateur peut accéder aux informations de chaque autre utilisateur.

ID utilisateur restreints

Les ID utilisateur `UNKNOWN` et le groupe `NOBODY` ont une signification particulière pour IBM MQ. La création d'un ID utilisateur dans le système d'exploitation appelé `UNKNOWN` ou d'un groupe appelé `NOBODY` peut avoir des résultats inattendus.

ID utilisateur lors de la connexion à un serveur IBM MQ for Windows

Windows

Un serveur IBM MQ for Windows ne prend pas en charge la connexion d'un IBM MQ MQI client si le client s'exécute sous un ID utilisateur contenant le caractère @, par exemple, `abc@d`. Le code retour de l'appel `MQCONN` au niveau du client est `MQRC_NOT_AUTHORIZED`.

Toutefois, vous pouvez spécifier l'ID utilisateur à l'aide de deux caractères @, par exemple, `abc@@d`. Il est recommandé d'utiliser le format `id@domain` pour s'assurer que l'ID utilisateur est résolu de manière cohérente dans le domaine approprié ; par conséquent, `abc@@d@domain`.

Autorisation de planification

Planifiez les utilisateurs qui auront des droits d'administration et planifiez la manière d'autoriser les utilisateurs des applications à utiliser les objets IBM MQ de manière appropriée, y compris ceux qui se connectent à partir d'un IBM MQ MQI client.

Les personnes ou les applications doivent disposer d'un accès leur permettant d'utiliser IBM MQ. L'accès dont ils ont besoin dépend des rôles qu'ils assument et des tâches qu'ils doivent effectuer. L'autorisation dans IBM MQ peut être divisée en deux catégories principales:

- Autorisation d'effectuer des opérations d'administration
- Autorisation pour les applications d'utiliser IBM MQ

Les deux classes d'opération sont contrôlées par le même composant et un individu peut être autorisé à effectuer les deux catégories d'opération.

Les rubriques suivantes fournissent des informations supplémentaires sur des domaines d'autorisation spécifiques que vous devez prendre en compte:

Droit d'administration de IBM MQ

Les administrateurs IBM MQ doivent disposer des droits nécessaires pour exécuter diverses fonctions. Ces droits sont obtenus de différentes manières sur différentes plateformes.

Les administrateurs IBM MQ doivent disposer des droits suivants:

- Exécutez des commandes pour administrer IBM MQ.
- Windows Linux Utilisez la console IBM MQ Explorer.
- z/OS Utilisez les panneaux d'opérations et de contrôle sous z/OS.
- z/OS Utilisez le programme utilitaire IBM MQ , CSQUTIL, sous z/OS.
- z/OS Accédez aux fichiers du gestionnaire de files d'attente sur z/OS.

Pour plus d'informations, voir la rubrique correspondant à votre système d'exploitation.

ALW *Droits d'administration de IBM MQ sur les systèmes AIX, Linux, and Windows*

Un administrateur IBM MQ est membre du groupe mqm. Ce groupe a accès à toutes les ressources IBM MQ et peut émettre des commandes de contrôle IBM MQ . Un administrateur peut accorder des droits spécifiques à un groupe d'utilisateurs.

Pour être un administrateur IBM MQ sur les systèmes AIX, Linux, and Windows , un utilisateur doit être membre du *groupe mqm*. Ce groupe est créé automatiquement lorsque vous installez IBM MQ. Pour permettre aux utilisateurs d'émettre des commandes de contrôle, vous devez les ajouter au groupe mqm. Cela inclut l'utilisateur root sous AIX and Linux.

Les utilisateurs qui ne sont pas membres du groupe mqm peuvent se voir accorder des privilèges d'administration, mais ils ne peuvent pas émettre de commandes de contrôle IBM MQ et ils sont autorisés à exécuter uniquement les commandes auxquelles ils ont accès.

De plus, sur les systèmes Windows , les comptes SYSTEM et Administrator disposent d'un accès complet aux ressources IBM MQ .

Tous les membres du groupe mqm ont accès à toutes les ressources IBM MQ sur le système, y compris la possibilité d'administrer tout gestionnaire de files d'attente en cours d'exécution sur le système. Cet accès peut être révoqué uniquement en supprimant un utilisateur du groupe mqm. Sur les systèmes Windows , les membres du groupe Administrateurs ont également accès à toutes les ressources IBM MQ .

Les administrateurs peuvent utiliser la commande de contrôle **runmqsc** pour émettre des commandes IBM MQ Script (MQSC). Lorsque **runmqsc** est utilisé en mode indirect pour envoyer des commandes MQSC à un gestionnaire de files d'attente éloignées, chaque commande MQSC est encapsulée dans une commande Escape PCF. Les administrateurs doivent disposer des droits requis pour que les commandes MQSC soient traitées par le gestionnaire de files d'attente éloignées.

IBM MQ Explorer émet des commandes PCF pour effectuer des tâches d'administration. Les administrateurs n'ont pas besoin de droits supplémentaires pour utiliser IBM MQ Explorer afin d'administrer un gestionnaire de files d'attente sur le système local. Lorsque IBM MQ Explorer est utilisé pour administrer un gestionnaire de files d'attente sur un autre système, les administrateurs doivent disposer des droits requis pour que les commandes PCF soient traitées par le gestionnaire de files d'attente éloignées.

Pour plus d'informations sur les vérifications des droits d'accès effectuées lors du traitement des commandes PCF et MQSC, voir les rubriques suivantes:

- Pour les commandes qui fonctionnent sur les gestionnaires de files d'attente, les files d'attente, les canaux, les processus, les listes de noms et les objets d'informations d'authentification, voir «Autorisation pour les applications d'utiliser IBM MQ», à la page 99.
- Pour les commandes qui fonctionnent sur les canaux, les initiateurs de canal, les programmes d'écoute et les clusters, voir Sécurité des canaux.
- z/OS Pour les commandes MQSC traitées par le serveur de commandes sous IBM MQ for z/OS, voir «Command security and command resource security on z/OS», à la page 97.

Pour plus d'informations sur les droits dont vous avez besoin pour administrer les systèmes IBM MQ for AIX, Linux, and Windows , voir les informations connexes.

IBM i *Droit d'administration de IBM MQ sur IBM i*

Pour être un administrateur IBM MQ sous IBM i, vous devez être membre du *groupe QMQMADM*. Ce groupe possède des propriétés similaires à celles du groupe mqm sur les systèmes AIX, Linux, and Windows . En particulier, le groupe QMQMADM est créé lorsque vous installez IBM MQ for IBM iet les membres du groupe QMQMADM ont accès à toutes les ressources IBM MQ sur le système. Vous avez également accès à toutes les ressources IBM MQ si vous disposez des droits *ALLOBJ.

Les administrateurs peuvent utiliser des commandes CL pour administrer IBM MQ.L'une de ces commandes est GRTMQMAUT, qui permet d'accorder des droits à d'autres utilisateurs. Une autre commande, STRMQMMQSC, permet à un administrateur d'émettre des commandes MQSC vers un gestionnaire de files d'attente local.

Deux groupes de commandes CL sont fournis par IBM MQ for IBM i:

Groupe 1
Pour émettre une commande dans cette catégorie, un utilisateur doit être membre du groupe QMQMADM ou disposer des droits *ALLOBJ. GRTMQMAUT et STRMQMMQSC appartiennent à cette catégorie, par exemple.

Groupe 2
Pour émettre une commande dans cette catégorie, un utilisateur n'a pas besoin d'être membre du groupe QMQMADM ou de disposer des droits *ALLOBJ. Au lieu de cela, deux niveaux d'autorité sont requis:

- L'utilisateur doit disposer des droits IBM i pour utiliser la commande. Ces droits sont accordés à l'aide de la commande GRTOBJAUT.
- L'utilisateur doit disposer des droits IBM MQ pour accéder à tout objet IBM MQ associé à la commande. Ces droits sont accordés à l'aide de la commande GRTMQMAUT.

Les exemples suivants présentent les commandes de ce groupe:

- CRTMQMQ, Création d'une file d'attente MQM
- CHGMQMPRC, modification du processus MQM

- DLTMQMNL, Suppression de la liste de noms MQM
- DSPMQMAUTI, Affichage des informations d'authentification MQM
- CRTMQMCHL, création d'un canal MQM

Pour plus d'informations sur ce groupe de commandes, voir «Autorisation pour les applications d'utiliser IBM MQ», à la page 99.

Pour la liste complète des commandes du groupe 1 et du groupe 2, voir «Droits d'accès pour les objets IBM MQ sous IBM i», à la page 168

Pour plus d'informations sur les droits d'accès dont vous avez besoin pour administrer IBM MQ sur IBM i, voir Administration d' IBM i .

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.

Autorisation pour les applications d'utiliser IBM MQ

Lorsque les applications accèdent à des objets, les ID utilisateur associés aux applications doivent disposer des droits appropriés.

Les applications peuvent accéder aux objets IBM MQ suivants en émettant des appels MQI:

- Gestionnaires de files d'attente
- Files d'attente
- Processus
- Listes de noms
- Rubriques

Les applications peuvent également utiliser des commandes PCF pour administrer des objets IBM MQ . Lorsque la commande PCF est traitée, elle utilise le contexte de droits de l'ID utilisateur qui a inséré le message PCF.

Dans ce contexte, les applications incluent celles écrites par les utilisateurs et les fournisseurs, ainsi que celles fournies avec IBM MQ for z/OS.

z/OS Les applications fournies avec IBM MQ for z/OS sont les suivantes:

- Les panneaux d'opérations et de contrôle
- Le programme utilitaire IBM MQ , CSQUTIL
- L'utilitaire de gestionnaire de files d'attente de rebut, CSQUDLQH

Les applications qui utilisent IBM MQ classes for Java, IBM MQ classes for JMS, IBM MQ classes for .NETou les clients de service de messagerie pour C/C++ et .NET utilisent l'interface MQI indirectement.

Les agents MCA émettent également des appels MQI et les ID utilisateur associés aux agents MCA ont besoin de droits d'accès à ces objets IBM MQ . Pour plus d'informations sur ces ID utilisateur et sur les droits dont ils ont besoin, voir «Autorisation de canal», à la page 121.

z/OS Sous z/OS, les applications peuvent également utiliser des commandes MQSC pour accéder à ces objets IBM MQ , mais la sécurité des commandes et la sécurité des ressources de commandes permettent de vérifier les droits d'accès dans ces circonstances. z/OS Pour plus d'informations, voir «Command security and command resource security on z/OS», à la page 97 et «MQSC commands and the system command input queue on z/OS», à la page 98.

IBM i Sous IBM i, un utilisateur qui émet une commande CL dans le groupe 2 peut avoir besoin de droits d'accès à un objet IBM MQ associé à la commande. Pour plus d'informations, voir «Lorsque des vérifications des droits d'accès sont effectuées», à la page 100.

Lorsque des vérifications des droits d'accès sont effectuées

Les vérifications des droits d'accès sont effectuées lorsqu'une application tente d'accéder à un gestionnaire de files d'attente, à une file d'attente, à un processus ou à une liste de noms.

Sous IBM i, des vérifications des droits d'accès peuvent également être effectuées lorsqu'un utilisateur émet une commande CL dans le groupe 2 qui accède à l'un de ces objets IBM MQ . Les vérifications sont effectuées dans les cas suivants:

Lorsqu'une application se connecte à un gestionnaire de files d'attente à l'aide d'un appel MQCONN ou MQCONNX

Le gestionnaire de files d'attente demande au système d'exploitation l'ID utilisateur associé à l'application. Le gestionnaire de files d'attente vérifie ensuite que l'ID utilisateur est autorisé à s'y connecter et conserve l'ID utilisateur pour les vérifications ultérieures.

Les utilisateurs n'ont pas besoin de se connecter à IBM MQ. IBM MQ suppose que les utilisateurs sont connectés au système d'exploitation sous-jacent et qu'ils ont été authentifiés par celui-ci.

Lorsqu'une application ouvre un objet IBM MQ à l'aide d'un appel MQOPEN ou MQPUT1

Toutes les vérifications de droits sont effectuées lorsqu'un objet est ouvert, et non lors d'un accès ultérieur. Par exemple, des vérifications des droits d'accès sont effectuées lorsqu'une application ouvre une file d'attente. Elles ne sont pas effectuées lorsque l'application insère des messages dans la file d'attente ou extrait des messages de la file d'attente.

Lorsqu'une application ouvre un objet, elle indique les types d'opération qu'elle doit effectuer sur l'objet. Par exemple, une application peut ouvrir une file d'attente pour parcourir les messages qu'elle contient, en extraire des messages, mais pas pour y placer des messages. Pour chaque type d'opération, le gestionnaire de files d'attente vérifie que l'ID utilisateur associé à l'application dispose des droits permettant d'effectuer cette opération.

Lorsqu'une application ouvre une file d'attente, les vérifications des droits d'accès sont effectuées sur l'objet nommé dans la zone `ObjectName` du descripteur d'objet. La zone `ObjectName` est utilisée sur les appels `MQOPEN` ou `MQPUT1` . Si l'objet est une file d'attente alias ou une définition de file d'attente éloignée, les vérifications des droits sont effectuées sur l'objet lui-même. Elles ne sont pas effectuées sur la file d'attente dans laquelle la file d'attente alias ou la définition de file d'attente éloignée est résolue. Cela signifie que l'utilisateur n'a pas besoin de droits pour y accéder. Limitez les droits de création de files d'attente aux utilisateurs privilégiés. Si vous ne le faites pas, les utilisateurs peuvent ignorer le contrôle d'accès normal simplement en créant un alias.

Une application peut faire explicitement référence à une file d'attente éloignée. Il définit les zones `ObjectName` et `ObjectQMgrName` du descripteur d'objet sur les noms de la file d'attente éloignée et du gestionnaire de files d'attente éloignées. Les vérifications des droits d'accès sont effectuées sur la file d'attente de transmission portant le même nom que le gestionnaire de files d'attente éloignées:

- z/OS Sous z/OS, une vérification est effectuée sur le profil de file d'attente RACF qui correspond au nom du gestionnaire de files d'attente éloignées et elle est effectuée, que cette file d'attente de transmission soit définie localement ou non.
- Multi Sous Multiplateformes, une vérification est effectuée par rapport au profil RQMNAME qui correspond au nom du gestionnaire de files d'attente éloignées, si la mise en cluster est utilisée.

Une application peut référencer une file d'attente de cluster de manière explicite en définissant la zone `ObjectName` dans le descripteur d'objet sur le nom de la file d'attente de cluster. Les vérifications des droits d'accès sont effectuées sur la file d'attente de transmission du cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Les droits d'accès à une file d'attente dynamique sont basés sur la file d'attente modèle dont elle est dérivée, mais ne sont pas nécessairement les mêmes ; voir la remarque 1.

L'ID utilisateur utilisé par le gestionnaire de files d'attente pour les vérifications des droits d'accès est obtenu à partir du système d'exploitation. L'ID utilisateur est obtenu lorsque l'application se connecte au gestionnaire de files d'attente. Une application dûment autorisée peut émettre un appel `MQOPEN` en spécifiant un autre ID utilisateur ; des vérifications de contrôle d'accès sont ensuite effectuées

sur l'autre ID utilisateur. L'utilisation d'un autre ID utilisateur ne modifie pas l'ID utilisateur associé à l'application, mais uniquement celui utilisé pour les vérifications de contrôle d'accès.

Lorsqu'une application s'abonne à une rubrique à l'aide d'un appel MQSUB

Lorsqu'une application s'abonne à une rubrique, elle spécifie le type d'opération qu'elle doit effectuer. Il s'agit de créer un abonnement, de modifier un abonnement existant ou de reprendre un abonnement existant sans le modifier. Pour chaque type d'opération, le gestionnaire de files d'attente vérifie que l'ID utilisateur associé à l'application est autorisé à effectuer l'opération.

Lorsqu'une application s'abonne à une rubrique, les vérifications des droits d'accès sont effectuées sur les objets de rubrique qui se trouvent dans l'arborescence de rubriques. Les objets de rubrique se trouvent au niveau ou au-dessus du point de l'arborescence de rubriques auquel l'application s'est abonnée. Les vérifications des droits d'accès peuvent impliquer des vérifications sur plusieurs objets de rubrique. L'ID utilisateur utilisé par le gestionnaire de files d'attente pour les vérifications des droits d'accès est obtenu à partir du système d'exploitation. L'ID utilisateur est obtenu lorsque l'application se connecte au gestionnaire de files d'attente.

Le gestionnaire de files d'attente effectue des vérifications des droits d'accès sur les files d'attente d'abonné, mais pas sur les files d'attente gérées.

Lorsqu'une application supprime une file d'attente dynamique permanente à l'aide d'un appel MQCLOSE

Le descripteur d'objet spécifié dans l'appel MQCLOSE n'est pas nécessairement le même que celui renvoyé par l'appel MQOPEN qui a créé la file d'attente dynamique permanente. S'il est différent, le gestionnaire de files d'attente vérifie l'ID utilisateur associé à l'application qui a émis l'appel MQCLOSE . Il vérifie que l'ID utilisateur est autorisé à supprimer la file d'attente.

Lorsqu'une application qui ferme un abonnement pour le supprimer ne l'a pas créé, les droits appropriés sont requis pour le supprimer.

Lorsqu'une commande PCF qui agit sur un objet IBM MQ est traitée par le serveur de commandes

Cette règle inclut le cas où une commande PCF agit sur un objet d'informations d'authentification.

L'ID utilisateur utilisé pour les vérifications des droits d'accès est celui qui se trouve dans la zone UserIdentifier du descripteur de message de la commande PCF. Cet ID utilisateur doit disposer des droits requis sur le gestionnaire de files d'attente dans lequel la commande est traitée. La commande MQSC équivalente encapsulée dans une commande Escape PCF est traitée de la même manière. Pour plus d'informations sur la zone UserIdentifier et sur la manière dont elle est définie, voir «Contexte de message», à la page 102.

IBM i **Sous IBM i, lorsqu'un utilisateur émet une commande CL dans le groupe 2 qui fonctionne sur un objet IBM MQ**

Cette règle inclut le cas où une commande CL du groupe 2 s'applique à un objet d'informations d'authentification.

Des vérifications sont effectuées pour déterminer si l'utilisateur a le droit d'opérer sur un objet IBM MQ associé à la commande. Les vérifications sont effectuées sauf si l'utilisateur est membre du groupe QMQMADM ou dispose des droits *ALLOBJ . Les droits requis dépendent du type d'opération que la commande exécute sur l'objet. Par exemple, la commande **CHGMQMQ**, Modifier la file d'attente MQM, requiert le droit de modifier les attributs de la file d'attente spécifiée par la commande. En revanche, la commande **DSPMQMQ**, Afficher la file d'attente MQM, requiert le droit d'afficher les attributs de la file d'attente spécifiée par la commande.

De nombreuses commandes s'exécutent sur plusieurs objets. Par exemple, pour exécuter la commande **DLTMQMQ**, Supprimer une file d'attente MQM, les droits suivants sont requis:

- Droit de connexion au gestionnaire de files d'attente spécifié par la commande
- Droit de suppression de la file d'attente indiqué par la commande

Certaines commandes ne fonctionnent sur aucun objet du tout. Dans ce cas, l'utilisateur n'a besoin que des droits IBM i pour exécuter l'une de ces commandes. **STRMQMLSR**, Démarrer le programme d'écoute MQM est un exemple de commande de ce type.

Droits de l'utilisateur de remplacement

Lorsqu'une application ouvre un objet ou s'abonne à une rubrique, elle peut fournir un ID utilisateur sur l'appel MQOPEN, MQPUT1ou MQSUB. Il peut demander au gestionnaire de files d'attente d'utiliser cet ID utilisateur pour les vérifications des droits d'accès au lieu de celui associé à l'application.

L'application réussit à ouvrir l'objet uniquement si les deux conditions suivantes sont remplies:

- L'ID utilisateur associé à l'application a le droit de fournir un ID utilisateur différent pour les vérifications des droits. L'application est dite disposer de *droits d'utilisateur de remplacement*.
- L'ID utilisateur fourni par l'application a le droit d'ouvrir l'objet pour les types d'opération demandés ou de s'abonner à la rubrique.

Contexte de message

Les informations de *contexte de message* permettent à l'application qui extrait un message de découvrir l'origine du message. Les informations sont conservées dans les zones du descripteur de message et les zones sont divisées en trois parties logiques

Ces parties sont les suivantes:

contexte d'identité
: Ces zones contiennent des informations sur l'utilisateur de l'application qui a inséré le message dans la file d'attente.

contexte d'origine
: Ces zones contiennent des informations sur l'application elle-même et sur le moment où le message a été inséré dans la file d'attente.

contexte d'utilisateur
: Ces zones contiennent les propriétés de message que les applications peuvent utiliser pour sélectionner les messages que le gestionnaire de files d'attente doit distribuer.

Lorsqu'une application insère un message dans une file d'attente, elle peut demander au gestionnaire de files d'attente de générer les informations de contexte dans le message. Il s'agit de l'action par défaut. Il peut également indiquer que les zones de contexte ne doivent pas contenir d'informations. L'ID utilisateur associé à une application ne nécessite aucun droit spécial pour effectuer l'une ou l'autre de ces opérations.

Une application peut définir les zones de contexte d'identité dans un message, ce qui permet au gestionnaire de files d'attente de générer le contexte d'origine ou de définir toutes les zones de contexte. Une application peut également transmettre les zones de contexte d'identité d'un message qu'elle a extrait à un message qu'elle place dans une file d'attente, ou transmettre toutes les zones de contexte. Toutefois, l'ID utilisateur associé à une application requiert des droits d'accès pour définir ou transmettre des informations de contexte. Une application indique qu'elle a l'intention de définir ou de transmettre des informations de contexte lorsqu'elle ouvre la file d'attente dans laquelle elle est sur le point d'insérer des messages et que ses droits sont vérifiés à ce stade.

Voici une brève description de chacune des zones de contexte:

contexte d'identité

UserIdentifier
: ID utilisateur associé à l'application qui a inséré le message. Si le gestionnaire de files d'attente définit cette zone, elle est définie sur l'ID utilisateur obtenu à partir du système d'exploitation lorsque l'application se connecte au gestionnaire de files d'attente.

AccountingToken
: Informations pouvant être utilisées pour facturer le travail effectué à la suite du message.

ApplIdentityData
: Si l'ID utilisateur associé à une application est autorisé à définir les zones de contexte d'identité ou à définir toutes les zones de contexte, l'application peut définir cette zone sur n'importe quelle valeur liée à l'identité. Si le gestionnaire de files d'attente définit cette zone, elle est mise à blanc.

Contexte d'origine

PutApplType
Type de l'application qui a inséré le message ; une transaction CICS , par exemple.

PutApplName
Nom de l'application qui a inséré le message.

PutDate
Date à laquelle le message a été inséré.

PutTime
Heure à laquelle le message a été inséré.

ApplOriginData
Si l'ID utilisateur associé à une application a le droit de définir toutes les zones de contexte, l'application peut définir cette zone sur n'importe quelle valeur liée à l'origine. Si le gestionnaire de files d'attente définit cette zone, elle est mise à blanc.

Contexte utilisateur

Les valeurs suivantes sont prises en charge pour **MQINQMP** ou **MQSETMP**:

MQPD_USER _CONTEXT

La propriété est associée au contexte utilisateur.

Aucune autorisation spéciale n'est requise pour pouvoir définir une propriété associée au contexte utilisateur à l'aide de l'appel MQSETMP.

Sur un gestionnaire de files d'attente V7.0 ou ultérieure, une propriété associée au contexte utilisateur est sauvegardée comme décrit pour MQOO_SAVE_ALL_CONTEXT. Une instruction MQPUT avec MQOO_PASS_ALL_CONTEXT spécifiée entraîne la copie de la propriété du contexte sauvegardé dans le nouveau message.

MQPD_NO_CONTEXT

La propriété n'est pas associée à un contexte de message.

Une valeur non reconnue est rejetée avec MQRC_PD_ERROR. La valeur initiale de cette zone est **MQPD_NO_CONTEXT**.

Pour une description détaillée de chacune des zones de contexte, voir MQMD-Descripteur de message. Pour plus d'informations sur l'utilisation du contexte de message, voir Contexte de message.

IBM i ALW *Droits d'utilisation des objets IBM MQ sur les systèmes IBM i , AIX, Linux, and Windows*

Le composant de service d'autorisation fourni avec IBM MQ est appelé *gestionnaire des droits d'accès aux objets* (OAM). Il fournit un contrôle d'accès via des vérifications d'authentification et d'autorisation.

Authentification.
La vérification de l'authentification effectuée par la méthode d'accès aux objets (OAM) fournie avec IBM MQ est de base et n'est effectuée que dans des circonstances spécifiques. Il n'est pas destiné à répondre aux exigences strictes attendues dans un environnement hautement sécurisé.

La méthode d'accès aux objets (OAM) effectue sa vérification d'authentification lorsqu'une application se connecte à un gestionnaire de files d'attente et les conditions suivantes sont remplies:

- Si une structure MQCSP a été fournie par l'application de connexion, et
- L'attribut *AuthenticationType* de la structure MQCSP reçoit la valeur MQCSP_AUTH_USER_ID_AND_PWD, et
- La valeur CHCKLOCL ou CHKCCLNT de l'objet AUTHINFO configuré n'est pas'NONE'

Les étapes d'authentification de la méthode d'accès aux objets (OAM) valident le mot de passe à l'aide des services du système d'exploitation, qui ont peut-être été configurés pour effectuer des

vérifications supplémentaires, par exemple pour s'assurer que le nom d'utilisateur n'a pas fait trop de tentatives de test de mot de passe incorrectes.

Il est possible d'utiliser d'autres mécanismes d'authentification si vous écrivez un nouveau composant de service d'autorisation ou si vous en obtenez un auprès d'un fournisseur.

Autorisation.
Les vérifications d'autorisation sont exhaustives et visent à répondre à la plupart des exigences normales.

Les vérifications d'autorisation sont effectuées lorsqu'une application émet un appel MQI pour accéder à un gestionnaire de files d'attente, une file d'attente, un processus, une rubrique ou une liste de noms. Ils sont également exécutés à d'autres moments, par exemple lorsqu'une commande est exécutée par le serveur de commandes.

Sur les systèmes IBM i IBM i , AIX, Linux, and Windows , le *service d'autorisation* fournit le contrôle d'accès lorsqu'une application émet un appel MQI pour accéder à un objet IBM MQ qui est un gestionnaire de files d'attente, une file d'attente, un processus, une rubrique ou une liste de noms. Cela inclut la vérification des droits d'utilisateur de remplacement et des droits de définition ou de transmission des informations de contexte.

Windows Sous Windows , la méthode d'accès aux objets (OAM) accorde aux membres du groupe Administrateurs le droit d'accéder à tous les objets IBM MQ , même lorsque le contrôle UAC est activé. En outre, sur les systèmes Windows , le compte SYSTEM dispose d'un accès complet aux ressources IBM MQ .

Le service d'autorisation permet également de vérifier les droits d'accès lorsqu'une commande PCF s'exécute sur l'un de ces objets IBM MQ ou sur un objet d'informations d'authentification. La commande MQSC équivalente encapsulée dans une commande Escape PCF est traitée de la même manière.

IBM i Sous IBM i , à moins que l'utilisateur ne soit membre du groupe QMQMADM ou qu'il ne dispose des droits *ALLOBJ, le service d'autorisation fournit également des vérifications de droits lorsqu'un utilisateur émet une commande CL dans le groupe 2 qui s'applique à l'un de ces objets IBM MQ ou à un objet d'informations d'authentification.

Le service d'autorisation est un *service installable*, ce qui signifie qu'il est implémenté par un ou plusieurs *composants de service installables*. Chaque composant est appelé à l'aide d'une interface documentée. Cela permet aux utilisateurs et aux fournisseurs de fournir des composants pour augmenter ou remplacer ceux fournis par les produits IBM MQ .

Le composant de service d'autorisation fourni avec IBM MQ est appelé gestionnaire des droits d'accès aux objets (OAM). La méthode d'accès aux objets (OAM) est automatiquement activée pour chaque gestionnaire de files d'attente que vous créez.

La méthode d'accès aux objets (OAM) gère une liste de contrôle d'accès (ACL) pour chaque objet IBM MQ auquel elle contrôle l'accès. Sur les systèmes AIX and Linux , seuls les ID groupe peuvent apparaître dans une liste de contrôle d'accès. Cela signifie que tous les membres d'un groupe ont les mêmes droits. Sur les systèmes IBM i IBM i et Windows , les ID utilisateur et les ID de groupe peuvent apparaître dans une liste de contrôle d'accès. Cela signifie que des droits peuvent être accordés à des utilisateurs et à des groupes individuels.

Une limitation de 12 caractères s'applique au groupe et à l'ID utilisateur. Les plateformes UNIX limitent généralement la longueur d'un ID utilisateur à 12 caractères. AIX et Linux ont augmenté cette limite, mais IBM MQ continue d'observer une restriction de 12 caractères sur toutes les plateformes UNIX . Si vous utilisez un ID utilisateur de plus de 12 caractères, IBM MQ le remplace par la valeur "UNKNOWN". Ne définissez pas d'ID utilisateur avec la valeur "UNKNOWN".

La méthode d'accès aux objets (OAM) peut authentifier un utilisateur et modifier les zones de contexte d'identité appropriées. Vous pouvez l'activer en spécifiant une structure de paramètres de sécurité de connexion (MQCSP) sur un appel MQCONNX. La structure est transmise à la fonction OAM Authenticate User (MQZ_AUTHENTICATE_USER), qui définit les zones de contexte d'identité appropriées. Si une connexion MQCONNX est établie à partir d'un client IBM MQ , les informations du MQCSP sont transmises

au gestionnaire de files d'attente auquel le client se connecte via la connexion client et le canal de connexion serveur. Si des exits de sécurité sont définis sur ce canal, le MQCSP est transmis à chaque exit de sécurité et peut être modifié par l'exit. Les exits de sécurité peuvent également créer le MQCSP. Pour plus de détails sur l'utilisation des exits de sécurité dans ce contexte, voir Programmes d'exit de sécurité de canal.

Avertissement : Dans certains cas, le mot de passe dans une structure MQCSP pour une application client est envoyé sur un réseau en texte clair. Pour vous assurer que les mots de passe d'application client sont correctement protégés, voir Protection par mot de passe CSPIBM MQ.

Sur les systèmes AIX, Linux, and Windows , la commande de contrôle **setmqaut** accorde et révoque les droits et est utilisée pour gérer les listes de contrôle d'accès. Par exemple, la commande

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

permet aux membres du groupe VOYAGER de parcourir les messages dans la file d'attente MOON.EUROPA appartenant au gestionnaire de files d'attente JUPITER. Il permet également aux membres d'extraire des messages de la file d'attente. Pour révoquer ces droits ultérieurement, entrez la commande suivante:

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

La commande :

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

permet aux membres du groupe VOYAGER d'insérer des messages dans n'importe quelle file d'attente dont le nom commence par les caractères `MOON.`. MOON.* est le nom d'un profil générique.Un *profil générique* vous permet d'accorder des droits pour un ensemble d'objets à l'aide d'une seule commande **setmqaut** .

La commande de contrôle **dspmqaut** permet d'afficher les droits en cours d'un utilisateur ou d'un groupe sur un objet spécifié.La commande de contrôle **dmpmqaut** est également disponible pour afficher les droits en cours associés aux profils génériques.

IBM i Sous IBM i, un administrateur utilise la commande CL GRTMQMAUT pour accorder des droits et la commande CL RVKMQMAUT pour révoquer des droits. Les profils génériques peuvent également être utilisés.Par exemple, la commande CL:

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

fournit la même fonction que l'exemple précédent d'une commande **setmqaut** ; elle permet aux membres du groupe VOYAGER de placer des messages dans n'importe quelle file d'attente dont le nom commence par les caractères `MOON.`

IBM i La commande CL DSPMQMAUT permet d'afficher les droits en cours dont dispose l'utilisateur ou le groupe pour un objet spécifié. Les commandes CL WRKMQMAUT et WRKMQMAUTD sont également disponibles pour gérer les droits en cours associés aux objets et aux profils génériques.

Si vous ne souhaitez pas de vérification des droits d'accès, par exemple, dans un environnement de test, vous pouvez désactiver la méthode d'accès aux objets (OAM).

Multi *Utilisation de PCF pour accéder aux commandes OAM*
Sur les systèmes IBM i, AIX, Linux, and Windows , vous pouvez utiliser les commandes PCF pour accéder aux commandes d'administration OAM.

Les commandes PCF et les commandes OAM équivalentes sont les suivantes:

Tableau 8. Commandes PCF et commandes OAM équivalentes	
Commande PCF	**Commande OAM**
Consulter des enregistrements de droits	dmpmqaut
Consulter les droits de l'entité	dspmqaut
Définir l'enregistrement de droits d'accès	setmqaut
Supprimer l'enregistrement de droits d'accès	setmqaut avec l'option -remove

Les commandes **setmqaut** et **dmpmqaut** sont limitées aux membres du groupe mqm. Les commandes PCF équivalentes peuvent être exécutées par des utilisateurs de n'importe quel groupe disposant des droits dsp et chg sur le gestionnaire de files d'attente.

Pour plus d'informations sur l'utilisation de ces commandes, voir Introduction to Programmable Command Formats.

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 97.

Sécurité de la messagerie distante

Cette section traite des aspects de la sécurité liés à la messagerie distante.

Vous devez autoriser les utilisateurs à utiliser les fonctions IBM MQ . Il est organisé en fonction des actions à entreprendre en ce qui concerne les objets et les définitions. Exemple :

- Les gestionnaires de files d'attente peuvent être démarrés et arrêtés par des utilisateurs autorisés
- Les applications doivent se connecter au gestionnaire de files d'attente et disposer des droits d'utilisation des files d'attente
- Les canaux de transmission de messages doivent être créés et contrôlés par des utilisateurs autorisés
- Les objets sont conservés dans des bibliothèques et l'accès à ces bibliothèques peut être restreint

L'agent MCA sur un site distant doit vérifier que le message en cours de distribution provient d'un utilisateur autorisé à le faire sur ce site distant. En outre, comme les agents MCA peuvent être démarrés à distance, il peut être nécessaire de vérifier que les processus distants qui tentent de démarrer vos agents MCA sont autorisés à le faire. Il y a quatre façons possibles pour vous de faire face à cette situation:

1. Utilisez de manière appropriée l'attribut PutAuthority de votre définition de canal RCVR, RQSTR ou CLUSRCVR pour contrôler l'utilisateur utilisé pour les vérifications d'autorisation au moment où les messages entrants sont placés dans vos files d'attente. Voir la description de la commande DEFINE CHANNEL dans le guide des commandes MQSC.
2. Implémentez des enregistrements d'authentification de canal pour rejeter les tentatives de connexion non souhaitées ou pour définir une valeur MCAUSER basée sur les éléments suivants: l'adresse IP distante, l'ID utilisateur distant, le nom distinctif du sujet TLS fourni ou le nom du gestionnaire de files d'attente distant.
3. Implémentez la vérification de la sécurité de l' *exit utilisateur* pour vous assurer que le canal de transmission de messages correspondant est autorisé. La sécurité de l'installation hébergeant le canal correspondant garantit que tous les utilisateurs sont correctement autorisés, de sorte que vous n'avez pas besoin de vérifier les messages individuels.
4. Implémentez le traitement des messages de l' *exit utilisateur* pour vous assurer que les messages individuels sont vérifiés pour l'autorisation.

IBM i *Sécurité des objets IBM MQ for IBM i*

Cette section traite des aspects de la sécurité liés à la messagerie distante.

Vous devez autoriser les utilisateurs à utiliser les fonctions IBM MQ for IBM i . Ce droit est organisé en fonction des actions à entreprendre en ce qui concerne les objets et les définitions. Exemple :

- Les gestionnaires de files d'attente peuvent être démarrés et arrêtés par des utilisateurs autorisés
- Les applications doivent se connecter au gestionnaire de files d'attente et disposer des droits permettant d'utiliser les files d'attente

- Les canaux de message doivent être créés et contrôlés par des utilisateurs autorisés

L'agent MCA sur un site distant doit vérifier que le message en cours de distribution est dérivé d'un utilisateur disposant des droits permettant d'afficher le message sur ce site distant. En outre, comme les agents MCA peuvent être démarrés à distance, il peut être nécessaire de vérifier que les processus distants qui tentent de démarrer vos agents MCA sont autorisés à le faire. Il y a quatre façons possibles pour vous de faire face à cette situation:

- Dans la définition de canal, décréter que les messages doivent contenir des droits *context* acceptables, sinon ils sont supprimés.
- Implémentez des enregistrements d'authentification de canal pour rejeter les tentatives de connexion non souhaitées ou pour définir une valeur MCAUSER basée sur l'un des éléments suivants: l'adresse IP distante, l'ID utilisateur distant, le nom distinctif TLS fourni ou le nom du gestionnaire de files d'attente distant.
- Implémentez la vérification de la sécurité de l'exit utilisateur pour vous assurer que le canal de message correspondant est autorisé. La sécurité de l'installation hébergeant le canal correspondant garantit que tous les utilisateurs sont correctement autorisés, de sorte que vous n'avez pas besoin de vérifier les messages individuels.
- Implémentez le traitement des messages d'exit utilisateur pour vous assurer que les messages individuels sont vérifiés pour l'autorisation.

Voici quelques faits sur la façon dont IBM MQ for IBM i opère la sécurité:

- Les utilisateurs sont identifiés et authentifiés par IBM i.
- Les services de gestionnaire de files d'attente appelés par les applications sont exécutés avec les droits du profil utilisateur de gestionnaire de files d'attente, mais dans le processus de l'utilisateur.
- Les services de gestionnaire de files d'attente appelés par les commandes utilisateur sont exécutés avec les droits du profil utilisateur de gestionnaire de files d'attente.

Linux AIX *Sécurité des objets sur AIX and Linux*

Les utilisateurs d'administration doivent faire partie du groupe mqm sur votre système (y compris root) si cet ID doit utiliser les commandes d'administration IBM MQ .

Vous devez toujours exécuter amqcrsta en tant qu'ID utilisateur "mqm".

ID utilisateur sous AIX and Linux

Le gestionnaire de files d'attente convertit tous les identificateurs utilisateur en majuscules ou en casse mixte en minuscules. Le gestionnaire de files d'attente insère ensuite les identificateurs d'utilisateur dans la partie contextuelle d'un message ou vérifie leur autorisation. Les autorisations sont donc basées uniquement sur des identificateurs en minuscules.

Windows *Sécurité des objets sur les systèmes Windows*

Les utilisateurs d'administration doivent faire partie du groupe mqm et du groupe d'administrateurs sur les systèmes Windows si cet ID doit utiliser les commandes d'administration IBM MQ .

ID utilisateur sur les systèmes Windows

Sur les systèmes Windows , *si aucun exit de message n'est installé*, le gestionnaire de files d'attente convertit tous les identificateurs d'utilisateur en majuscules ou en minuscules. Le gestionnaire de files d'attente insère ensuite les identificateurs d'utilisateur dans la partie contextuelle d'un message ou vérifie leur autorisation. Les autorisations sont donc basées uniquement sur des identificateurs en minuscules.

ID utilisateur sur tous les systèmes

Les plateformes autres que les systèmes AIX, Linux, and Windows utilisent des majuscules pour les ID utilisateur dans les messages. Pour permettre aux systèmes AIX, Linux, and Windows d'utiliser des ID utilisateur en minuscules dans les messages, l'agent MCA doit effectuer les conversions appropriées de caractères alphabétiques.

Pour permettre aux systèmes AIX, Linux, and Windows d'utiliser des ID utilisateur en minuscules dans les messages, les conversions suivantes sont effectuées par l'agent MCA sur ces plateformes:

A la fin de l'envoi
Les caractères alphabétiques de tous les ID utilisateur sont convertis en caractères majuscules si aucun exit de message n'est installé.

A l'extrémité réceptrice
Les caractères alphabétiques de tous les ID utilisateur sont convertis en minuscules si aucun exit de message n'est installé.

Les conversions automatiques ne sont pas effectuées si vous fournissez un exit de message sur AIX, Linux, and Windows pour une autre raison.

Utilisation d'un service d'autorisation personnalisé

IBM MQ fournit un service d'autorisation installable. Vous pouvez choisir d'installer un autre service.

Le composant de service d'autorisation fourni avec IBM MQ est appelé Object Authority Manager (OAM). Si la méthode d'accès aux objets (OAM) ne fournit pas les fonctions d'autorisation dont vous avez besoin, vous pouvez écrire votre propre composant de service d'autorisation. Les fonctions de service installables qui doivent être implémentées par un composant de service d'autorisation sont décrites dans Informations de référence de l'interface des services installables.

Contrôle d'accès pour les clients

Le contrôle d'accès est basé sur les ID utilisateur. Il peut y avoir de nombreux ID utilisateur à administrer, et les ID utilisateur peuvent être dans des formats différents. Vous pouvez définir la propriété de canal de connexion serveur MCAUSER sur une valeur d'ID utilisateur spéciale à utiliser par les clients.

Le contrôle d'accès dans IBM MQ est basé sur les ID utilisateur. L'ID utilisateur du processus qui effectue des appels MQI est normalement utilisé. Pour les clients MQ MQI, l'agent MCA de connexion serveur effectue des appels MQI pour le compte des clients MQ MQI. Vous pouvez sélectionner un autre ID utilisateur pour l'agent MCA de connexion serveur à utiliser pour effectuer des appels MQI. L'ID utilisateur alternatif peut être associé au poste de travail client ou à tout élément que vous choisissez pour organiser et contrôler l'accès des clients. L'ID utilisateur doit disposer des droits nécessaires sur le serveur pour émettre des appels MQI. Il est préférable de choisir un autre ID utilisateur que d'autoriser les clients à effectuer des appels MQI avec les droits de l'agent MCA de connexion serveur.

Tableau 9. ID utilisateur utilisé par un canal de connexion serveur

ID utilisateur	En cas d'utilisation
ID utilisateur défini par un exit de sécurité	Utilisé sauf s'il est bloqué par une règle **CHLAUTH TYPE(BLOCKUSER)** . Pour plus d'informations, voir la section suivante, «Définition de l'ID utilisateur dans un exit de sécurité», à la page 110 .
ID utilisateur défini par une règle CHLAUTH	Utilisé sauf si remplacé par un exit de sécurité. Pour plus d'informations, voir Enregistrements d'authentification de canal .
ID utilisateur défini dans l'attribut **MCAUSER** de la définition de canal SVRCONN	Utilisé sauf s'il est remplacé par un exit de sécurité ou une règle CHLAUTH.
ID utilisateur transmis à partir de la machine client	Utilisé lorsqu'aucun ID utilisateur n'est défini par d'autres moyens.
ID utilisateur ayant démarré le canal de connexion serveur	Utilisé lorsqu'aucun ID utilisateur n'est défini par d'autres moyens et qu'aucun ID utilisateur client n'est transmis. Pour plus d'informations, voir la section suivante, «ID utilisateur qui exécute le programme de canal», à la page 110 .

Etant donné que l'agent MCA de connexion serveur effectue des appels MQI pour le compte d'utilisateurs distants, il est important de prendre en compte les implications de sécurité de l'agent MCA de connexion serveur qui émet des appels MQI pour le compte de clients distants et de savoir comment administrer l'accès d'un nombre potentiellement élevé d'utilisateurs.

- L'une des approches consiste pour l'agent MCA de connexion serveur à émettre des appels MQI avec ses propres droits d'accès. Mais attention, il est normalement indésirable pour l'agent MCA de connexion serveur, avec ses puissantes fonctions d'accès, d'émettre des appels MQI pour le compte des utilisateurs client.
- Une autre approche consiste à utiliser l'ID utilisateur qui provient du client. L'agent MCA de connexion serveur peut émettre des appels MQI à l'aide des fonctions d'accès de l'ID utilisateur client. Cette approche pose un certain nombre de questions à prendre en considération:
  1. Il existe différents formats pour l'ID utilisateur sur différentes plateformes. Cela provoque parfois des problèmes si le format de l'ID utilisateur sur le client diffère des formats acceptables sur le serveur.
  2. Il existe potentiellement de nombreux clients, avec des ID utilisateur différents et qui changent. Les ID doivent être définis et gérés sur le serveur.
  3. L'ID utilisateur est-il digne de confiance? Tout ID utilisateur peut être transmis à partir d'un client, pas nécessairement l'ID de l'utilisateur connecté. Par exemple, le client peut transmettre un ID avec des droits `mqm` complets qui ont été définis intentionnellement uniquement sur le serveur pour des raisons de sécurité.
- L'approche préférée consiste à définir des jetons d'identification client sur le serveur, et donc à limiter les capacités des applications connectées au client. Cette opération est généralement effectuée en définissant la propriété de canal de connexion serveur MCAUSER sur une valeur d'ID utilisateur spéciale à utiliser par les clients et en définissant quelques ID à utiliser par les clients ayant un niveau d'autorisation différent sur le serveur.

Définition de l'ID utilisateur dans un exit de sécurité

Pour IBM MQ MQI clients, le processus qui émet les appels MQI est l'agent MCA de connexion serveur. L'ID utilisateur utilisé par l'agent MCA de connexion serveur est contenu dans les zones `MCAUserIdentifier` ou `LongMCAUserIdentifier` du MQCD. Le contenu de ces zones est défini par:

- Toutes les valeurs définies par les exits de sécurité
- ID utilisateur du client
- MCAUSER (dans la définition de canal de connexion serveur)

L'exit de sécurité peut remplacer les valeurs qui lui sont visibles lorsqu'il est appelé.

- Si l'attribut MCAUSER du canal de connexion serveur est défini sur une valeur non vide, la valeur MCAUSER est utilisée.
- Si l'attribut MCAUSER du canal de connexion serveur est vide, l'ID utilisateur reçu du client est utilisé.
- Si l'attribut MCAUSER du canal de connexion serveur est vide et qu'aucun ID utilisateur n'est reçu du client, l'ID utilisateur qui a démarré le canal de connexion serveur est utilisé.

Le client IBM MQ ne transite pas l'ID utilisateur vérifié vers le serveur lorsqu'un exit de sécurité côté client est en cours d'utilisation.

ID utilisateur qui exécute le programme de canal

Lorsque les zones d'ID utilisateur sont dérivées de l'ID utilisateur qui a démarré le canal de connexion serveur, la valeur suivante est utilisée:

- z/OS Pour z/OS, ID utilisateur affecté à la tâche démarrée de l'initiateur de canal par la table des procédures démarrées z/OS .
- Pour TCP/IP (non z/OS), l'ID utilisateur de l'entrée `inetd.conf` ou l'ID utilisateur qui a démarré le programme d'écoute.

- Pour SNA (non z/OS), l'ID utilisateur de l'entrée SNA Server ou (s'il n'y en a aucune) la demande de connexion entrante, ou l'ID utilisateur qui a démarré le programme d'écoute.
- Pour NetBIOS ou SPX, l'ID utilisateur qui a démarré le programme d'écoute.

S'il existe des définitions de canal de connexion serveur dont l'attribut MCAUSER est à blanc, les clients peuvent utiliser cette définition de canal pour se connecter au gestionnaire de files d'attente avec des droits d'accès déterminés par l'ID utilisateur fourni par le client. Il peut s'agir d'un risque de sécurité si le système sur lequel s'exécute le gestionnaire de files d'attente autorise des connexions réseau non autorisées. Le canal de connexion serveur par défaut IBM MQ (SYSTEM.DEF.SVRCONN) a l'attribut MCAUSER à blanc. Pour éviter les accès non autorisés, mettez à jour l'attribut MCAUSER de la définition par défaut avec un ID utilisateur qui n'a pas accès aux objets IBM MQ MQ .

Casse des ID utilisateur

Lorsque vous définissez un canal avec `runmqsc`, l'attribut MCAUSER est mis en majuscules sauf si l'ID utilisateur est placé entre apostrophes.

ALW Pour les serveurs sous AIX, Linux, and Windows, le contenu de la zone `MCAUserIdentifier` reçue du client est remplacé par des minuscules.

IBM i Pour les serveurs sous IBM i, le contenu de la zone `LongMCAUserIdentifier` reçue du client est mis en majuscules.

Linux AIX Pour les serveurs sur les systèmes AIX and Linux , le contenu de la zone `LongMCAUserIdentifier` reçue du client est remplacé par des minuscules.

Par défaut, l'ID utilisateur transmis lorsqu'une application de liaison IBM MQ JMS est utilisée correspond à l'ID utilisateur de la machine virtuelle Java sur laquelle l'application est exécutée.

Il est également possible de transmettre un ID utilisateur via la méthode `createQueueConnection` .

Planification de la confidentialité

Planifiez le maintien de la confidentialité de vos données.

Vous pouvez implémenter la confidentialité au niveau de l'application ou au niveau du lien. Vous pouvez choisir d'utiliser TLS, auquel cas vous devez planifier votre utilisation des certificats numériques. Vous pouvez également utiliser des programmes d'exit de canal si les fonctions standard ne répondent pas à vos besoins.

Concepts associés

«Comparatif de la sécurité au niveau des liaisons et de la sécurité au niveau de l'application», à la page 111
Cette rubrique contient des informations sur divers aspects de la sécurité au niveau des liens et de la sécurité au niveau des applications, et compare les deux niveaux de sécurité.

«Programmes d'exit de canal», à la page 117
Les *programmes d'exit de canal* sont des programmes appelés à des endroits définis dans la séquence de traitement d'un agent MCA. Les utilisateurs et les fournisseurs peuvent écrire leurs propres programmes d'exit de canal. Certains sont fournis par IBM.

«Protection des canaux avec SSL/TLS», à la page 123
La prise en charge de TLS dans IBM MQ utilise l'objet d'informations d'authentification du gestionnaire de files d'attente et diverses commandes MQSC. Vous devez également tenir compte de votre utilisation des certificats numériques.

Comparatif de la sécurité au niveau des liaisons et de la sécurité au niveau de l'application

Cette rubrique contient des informations sur divers aspects de la sécurité au niveau des liens et de la sécurité au niveau des applications, et compare les deux niveaux de sécurité.

La sécurité au niveau des liens et des applications est illustrée dans Figure 10, à la page 112.

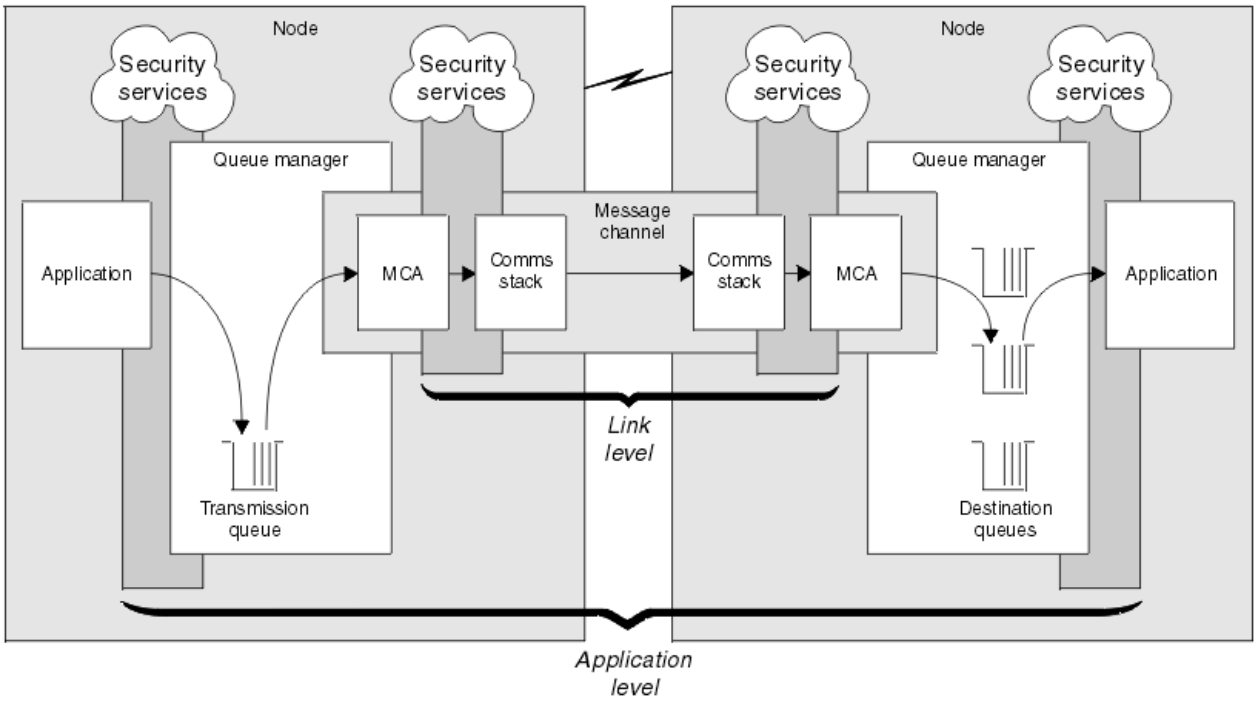


Figure 10. Sécurité au niveau des liens et sécurité au niveau des applications

Protection des messages dans les files d'attente

La sécurité au niveau des liaisons peut protéger les messages lorsqu'ils sont transférés d'un gestionnaire de files d'attente à un autre. Il est particulièrement important lorsque les messages sont transmis sur un réseau non sécurisé. Toutefois, il ne peut pas protéger les messages lorsqu'ils sont stockés dans des files d'attente d'un gestionnaire de files d'attente source, d'un gestionnaire de files d'attente de destination ou d'un gestionnaire de files d'attente intermédiaire.

z/OS Le chiffrement des fichiers z/OS peut fournir une certaine protection des messages stockés dans les files d'attente, mais uniquement pour les données au repos sur un gestionnaire de files d'attente local. Voir la section Confidentiality for data at rest on IBM MQ for z/OS with data set encryption pour plus d'informations.

La sécurité au niveau de l'application, par comparaison, peut protéger les messages lorsqu'ils sont stockés dans des files d'attente et s'applique même lorsque la mise en file d'attente répartie n'est pas utilisée. Il s'agit de la différence majeure entre la sécurité au niveau de la liaison et la sécurité au niveau de l'application. Elle est illustrée dans la Figure 10, à la page 112.

Les gestionnaires de files d'attente ne s'exécutent pas dans des environnements contrôlés et sécurisés

Si un gestionnaire de files d'attente s'exécute dans un environnement contrôlé et sécurisé, les mécanismes de contrôle d'accès fournis par IBM MQ peuvent être considérés comme suffisants pour protéger les messages stockés dans ses files d'attente. Cela est particulièrement vrai si seule la mise en file d'attente locale est impliquée et que les messages ne quittent jamais le gestionnaire de files d'attente. Dans ce cas, la sécurité au niveau de l'application peut être considérée comme inutile.

La sécurité au niveau de l'application peut également être considérée comme inutile si des messages sont transférés vers un autre gestionnaire de files d'attente qui s'exécute également dans un environnement contrôlé et sécurisé ou s'ils sont reçus d'un tel gestionnaire de files d'attente. La sécurité au niveau de l'application est d'autant plus nécessaire lorsque des messages sont transférés vers ou reçus d'un gestionnaire de files d'attente qui ne s'exécute pas dans un environnement contrôlé et sécurisé.

Différences de coût

La sécurité au niveau de l'application peut coûter plus cher que la sécurité au niveau de la liaison en termes d'administration et de performances.

Le coût de l'administration est probablement plus élevé car il y a potentiellement plus de contraintes à configurer et à gérer. Par exemple, vous pouvez être amené à vous assurer qu'un utilisateur particulier n'envoie que certains types de message et qu'il n'envoie des messages qu'à certaines destinations. A l'inverse, il peut être nécessaire de s'assurer qu'un utilisateur particulier ne reçoit que certains types de message et qu'il ne reçoit que des messages provenant de certaines sources. Au lieu de gérer les services de sécurité au niveau des liens sur un canal de messages unique, vous devrez peut-être configurer et gérer des règles pour chaque paire d'utilisateurs qui échangent des messages sur ce canal.

Il peut y avoir un impact sur les performances si les services de sécurité sont appelés à chaque fois qu'une application insère ou reçoit un message.

Les organisations ont tendance à prendre en compte d'abord la sécurité au niveau des liens car elle peut être plus facile à mettre en oeuvre. Ils prennent en compte la sécurité au niveau de l'application s'ils découvrent que la sécurité au niveau des liens ne répond pas à toutes leurs exigences.

Disponibilité des composants

Généralement, dans un environnement distribué, un service de sécurité requiert un composant sur au moins deux systèmes. Par exemple, un message peut être chiffré sur un système et déchiffré sur un autre. Cela s'applique à la sécurité au niveau de la liaison et à la sécurité au niveau de l'application.

Dans un environnement hétérogène, avec des plateformes différentes en cours d'utilisation, chacune avec des niveaux de fonction de sécurité différents, les composants requis d'un service de sécurité peuvent ne pas être disponibles pour chaque plateforme sur laquelle ils sont nécessaires et sous une forme facile à utiliser. Il s'agit probablement d'un problème plus important pour la sécurité au niveau de l'application que pour la sécurité au niveau de la liaison, en particulier si vous prévoyez de fournir votre propre sécurité au niveau de l'application en achetant des composants à partir de diverses sources.

Messages dans une file d'attente de rebut

Si un message est protégé par la sécurité au niveau de l'application, il peut y avoir un problème si, pour une raison quelconque, le message n'atteint pas sa destination et est placé dans une file d'attente de messages non livrés. Si vous ne savez pas comment traiter le message à partir des informations du descripteur de message et de l'en-tête de la lettre morte, vous devrez peut-être inspecter le contenu des données d'application. Vous ne pouvez pas effectuer cette opération si les données de l'application sont chiffrées et que seul le destinataire prévu peut les déchiffrer.

Ce que la sécurité au niveau de l'application ne peut pas faire

La sécurité au niveau de l'application n'est pas une solution complète. Même si vous implémentez la sécurité au niveau de l'application, vous pouvez tout de même avoir besoin de certains services de sécurité au niveau de la liaison. Exemple :

- Lorsqu'un canal démarre, l'authentification mutuelle des deux agents MCA peut toujours être requise. Cette opération ne peut être effectuée que par un service de sécurité au niveau de la liaison.
- La sécurité au niveau de l'application ne peut pas protéger l'en-tête de la file d'attente de transmission, MQXQH, qui inclut le descripteur de message imbriqué. Il ne peut pas non plus protéger les données dans les flux de protocole de canal IBM MQ autres que les données de message. Seule la sécurité au niveau de la liaison peut fournir cette protection.
- Si les services de sécurité au niveau de l'application sont appelés à l'extrémité serveur d'un canal MQI, les services ne peuvent pas protéger les paramètres des appels MQI envoyés via le canal. En particulier, les données d'application d'un appel MQPUT, MQPUT1ou MQGET ne sont pas protégées. Seule la sécurité au niveau des liens peut fournir la protection dans ce cas.

sécurité au niveau des liaisons

La *sécurité de niveau liaison* fait référence aux services de sécurité qui sont appelés, directement ou indirectement, par un agent MCA, le sous-système de communication ou une combinaison des deux.

La sécurité au niveau des liens est illustrée dans la Figure 10, à la page 112.

Voici quelques exemples de services de sécurité au niveau des liens:

- L'agent MCA à chaque extrémité d'un canal de transmission de messages peut authentifier son partenaire. Cette opération est effectuée lorsque le canal démarre et qu'une connexion de communication a été établie, mais avant que les messages ne commencent à circuler. Si l'authentification échoue à l'une des extrémités, le canal est fermé et aucun message n'est transféré. Il s'agit d'un exemple de service d'identification et d'authentification.
- Un message peut être chiffré à l'extrémité émettrice d'un canal et déchiffré à l'extrémité réceptrice. Il s'agit d'un exemple de service de confidentialité.
- Un message peut être vérifié à l'extrémité réceptrice d'un canal pour déterminer si son contenu a été volontairement modifié lors de sa transmission sur le réseau. Voici un exemple de service d'intégrité des données.

Sécurité au niveau de la liaison fournie par IBM MQ

Le principal moyen de mise à disposition de la confidentialité et de l'intégrité des données dans IBM MQ consiste à utiliser TLS. Pour plus d'informations sur l'utilisation de TLS dans IBM MQ, voir «Protocoles de sécurité TLS dans IBM MQ», à la page 25. Pour l'authentification, IBM MQ fournit la fonction permettant d'utiliser les enregistrements d'authentification de canal. Les enregistrements d'authentification de canal offrent un contrôle précis de l'accès accordé aux systèmes de connexion, au niveau des canaux individuels ou des groupes de canaux. Pour plus d'informations, voir «Enregistrements d'authentification de canal», à la page 54.

Mise à disposition de votre propre sécurité de niveau de liaison

Vous pouvez fournir vos propres services de sécurité au niveau des liens. L'écriture de vos propres programmes d'exit de canal est le principal moyen de fournir vos propres services de sécurité de niveau de liaison.

Les programmes d'exit de canal sont introduits dans «Programmes d'exit de canal», à la page 117. La même rubrique décrit également le programme d'exit de canal fourni avec IBM MQ for Windows (programme d'exit de canal SSPI). Ce programme d'exit de canal est fourni au format source afin que vous puissiez modifier le code source en fonction de vos besoins. Si ce programme d'exit de canal ou les programmes d'exit de canal disponibles auprès d'autres fournisseurs ne répondent pas à vos besoins, vous pouvez concevoir et écrire vos propres programmes. Cette rubrique explique comment les programmes d'exit de canal peuvent fournir des services de sécurité. Pour plus d'informations sur l'écriture d'un programme d'exit de canal, voir Ecriture de programmes d'exit de canal.

Sécurité au niveau de la liaison à l'aide d'un exit de sécurité

Les exits de sécurité fonctionnent normalement par paires, une à chaque extrémité d'un canal. Ils sont appelés immédiatement après la fin de la négociation de données initiale au démarrage du canal.

Les exits de sécurité peuvent être utilisés pour fournir l'identification et l'authentification, le contrôle d'accès et la confidentialité.

Sécurité au niveau de la liaison à l'aide d'un exit de message

Un exit de message ne peut être utilisé que sur un canal de transmission de messages et non sur un canal MQI. Il a accès à l'en-tête de la file d'attente de transmission, MQXQH, qui inclut le descripteur de message imbriqué, et aux données d'application d'un message. Il peut modifier le contenu du message et modifier sa longueur.

Un exit de message peut être utilisé à n'importe quelle fin qui nécessite l'accès à l'ensemble du message plutôt qu'à une partie de celui-ci.

Les exits de message peuvent être utilisés pour fournir l'identification et l'authentification, le contrôle d'accès, la confidentialité, l'intégrité des données et la non-répudiation, et pour des raisons autres que la sécurité.

Sécurité au niveau de la liaison à l'aide des exits d'envoi et de réception
Les exits d'envoi et de réception peuvent être utilisés sur les canaux de message et MQI. Ils sont appelés pour tous les types de données qui circulent sur un canal et pour les flux dans les deux sens.

Les exits d'émission et de réception ont accès à chaque segment de transmission. Ils peuvent modifier son contenu et sa longueur.

Sur un canal de transmission, si un MCA a besoin de fractionner un message et de l'envoyer dans plus d'un segment de transmission, une sortie d'émission est appelée pour chaque segment de transmission contenant une partie du message et, à la réception, une sortie de réception est appelée pour chaque segment de transmission. Il en est de même sur un canal MQI si les paramètres d'entrée ou de sortie d'un appel MQI sont trop grands pour être envoyés dans un segment de transmission unique.

Sur un canal MQI, l'octet 10 d'un segment de transmission identifie l'appel MQI et indique si le segment de transmission contient les paramètres d'entrée ou de sortie de l'appel. Les exits d'envoi et de réception peuvent examiner cet octet pour déterminer si l'appel MQI contient des données d'application qui peuvent avoir besoin d'être protégées.

Lorsqu'un exit d'émission est appelé pour la première fois, pour acquérir et initialiser les ressources dont il a besoin, il peut demander à l'agent MCA de réserver une quantité d'espace spécifiée dans la mémoire tampon qui contient un segment de transmission. Lorsqu'il est appelé ultérieurement pour traiter un segment de transmission, il peut utiliser cet espace pour ajouter une clé chiffrée ou une signature numérique, par example. L'exit de réception correspondent à l'autre extrémité du canal peut supprimer les données ajoutées par l'exit d'émission et les utiliser pour traiter le segment de transmission.

Les sorties d'émission et de réception sont mieux adaptées à des fins dans lesquelles elles n'ont pas besoin de comprendre la structure des données qu'elles traitent et peuvent donc traiter chaque segment de transmission comme un objet binaire.

Les exits d'envoi et de réception peuvent être utilisés pour assurer la confidentialité et l'intégrité des données, ainsi que pour des utilisations autres que la sécurité.

Tâches associées
Identification de l'appel API dans un programme d'exit d'envoi ou de réception

sécurité au niveau de l'application
La *sécurité au niveau de l'application* fait référence aux services de sécurité appelés à l'interface entre une application et un gestionnaire de files d'attente auquel elle est connectée.

Ces services sont appelés lorsque l'application émet des appels MQI au gestionnaire de files d'attente. Les services peuvent être appelés, directement ou indirectement, par l'application, le gestionnaire de files d'attente, un autre produit prenant en charge IBM MQou une combinaison de ces deux éléments. La sécurité au niveau de l'application est illustrée dans la Figure 10, à la page 112.

La sécurité au niveau de l'application est également appelée *sécurité de bout en bout* ou *sécurité au niveau des messages*.

Voici quelques exemples de services de sécurité au niveau de l'application:

- Lorsqu'une application place un message dans une file d'attente, le descripteur de message contient un ID utilisateur associé à l'application. Toutefois, il n'existe aucune donnée, telle qu'un mot de passe chiffré, qui peut être utilisée pour authentifier l'ID utilisateur. Un service de sécurité peut ajouter ces données. Lorsque le message est finalement extrait par l'application de réception, un autre composant du service peut authentifier l'ID utilisateur à l'aide des données qui ont été transmises avec le message. Il s'agit d'un exemple de service d'identification et d'authentification.
- Un message peut être chiffré lorsqu'il est placé dans une file d'attente par une application et déchiffré lorsqu'il est extrait par l'application réceptrice.Il s'agit d'un exemple de service de confidentialité.

- Un message peut être vérifié lorsqu'il est extrait par l'application de réception. Cette vérification détermine si son contenu a été délibérément modifié depuis sa première mise en file d'attente par l'application émettrice.Voici un exemple de service d'intégrité des données.

Planification de Advanced Message Security
Advanced Message Security (AMS) est un composant de IBM MQ qui offre un niveau de protection élevé pour les données sensibles transitant par le réseau IBM MQ , sans affecter les applications finales.

Si vous déplacez des informations très sensibles ou précieuses, en particulier des informations confidentielles ou liées au paiement, telles que les dossiers des patients ou les détails de la carte de crédit, vous devez accorder une attention particulière à la sécurité de l'information. S'assurer que les informations qui circulent dans l'entreprise conservent leur intégrité et sont protégées contre tout accès non autorisé constitue un défi et une responsabilité permanents. Vous êtes également susceptible d'être tenu de respecter les règles de sécurité, au risque de sanctions en cas de non-conformité.

Vous pouvez développer vos propres extensions de sécurité dans IBM MQ. Cependant, de telles solutions nécessitent des compétences spécialisées et peuvent être compliquées et coûteuses à maintenir. Advanced Message Security vous aide à relever ces défis lorsque vous déplacez des informations dans l'entreprise entre pratiquement tous les types de système informatique commercial.

Advanced Message Security étend les fonctions de sécurité de IBM MQ comme suit:

- Il fournit une protection des données de bout en bout au niveau de l'application pour votre infrastructure de messagerie point à point, à l'aide du chiffrement ou de la signature numérique des messages.
- Il fournit une sécurité complète sans écrire de code de sécurité complexe ni modifier ou recompiler les applications existantes.
- Il utilise la technologie PKI (Public Key Infrastructure) pour fournir des services d'authentification, d'autorisation, de confidentialité et d'intégrité des données pour les messages.
- Il fournit l'administration des règles de sécurité pour les grands systèmes et les serveurs distribués.
- Il prend en charge les serveurs et les clients IBM MQ .
- Il s'intègre à Managed File Transfer pour fournir une solution de messagerie sécurisée de bout en bout.

Pour plus d'informations, voir «Advanced Message Security», à la page 617.

Mise à disposition de votre propre sécurité au niveau de l'application
Vous pouvez fournir vos propres services de sécurité au niveau de l'application. Pour vous aider à implémenter la sécurité au niveau de l'application, IBM MQ fournit deux exits, l'exit d'API et l'exit de croisement d'API.

L'exit d'API et l'exit de croisement d'API peuvent fournir des services d'identification et d'authentification, de contrôle d'accès, de confidentialité, d'intégrité des données et de non-répudiation, ainsi que d'autres fonctions non liées à la sécurité.

Si l'exit d'API ou l'exit de croisement d'API n'est pas pris en charge dans votre environnement système, vous pouvez envisager d'autres moyens de fournir votre propre sécurité au niveau de l'application. L'une des méthodes consiste à développer une API de niveau supérieur qui encapsule l'interface MQI. Les programmeurs utilisent ensuite cette API, à la place de l'interface MQI, pour écrire des applications IBM MQ .

Les raisons les plus courantes de l'utilisation d'une API de niveau supérieur sont les suivantes:

- Pour masquer les fonctions plus avancées de l'interface MQI aux programmeurs.
- Pour appliquer des normes dans l'utilisation de l'interface MQI.
- Pour ajouter une fonction à l'interface MQI. Cette fonction supplémentaire peut être des services de sécurité.

Certains produits fournisseurs utilisent cette technique pour fournir une sécurité au niveau de l'application pour IBM MQ.

Si vous prévoyez de fournir des services de sécurité de cette manière, notez ce qui suit concernant la conversion des données:

- Si un jeton de sécurité, tel qu'une signature numérique, a été ajouté aux données d'application dans un message, tout code effectuant une conversion de données doit être conscient de la présence de ce jeton.
- Un jeton de sécurité peut avoir été dérivé d'une image binaire des données d'application. Par conséquent, toute vérification du jeton doit être effectuée avant la conversion des données.
- Si les données d'application d'un message ont été chiffrées, elles doivent être déchiffrées avant la conversion des données.

Programmes d'exit de canal

Les *programmes d'exit de canal* sont des programmes appelés à des endroits définis dans la séquence de traitement d'un agent MCA. Les utilisateurs et les fournisseurs peuvent écrire leurs propres programmes d'exit de canal. Certains sont fournis par IBM.

Il existe plusieurs types de programme d'exit de canal, mais seuls quatre ont un rôle à jouer pour assurer la sécurité au niveau des liens:

- Exit de sécurité
- Exit de message
- Exit d'émission
- Exit de réception

Ces quatre types de programme d'exit de canal sont illustrés dans la Figure 11, à la page 117 et sont décrits dans les rubriques suivantes.

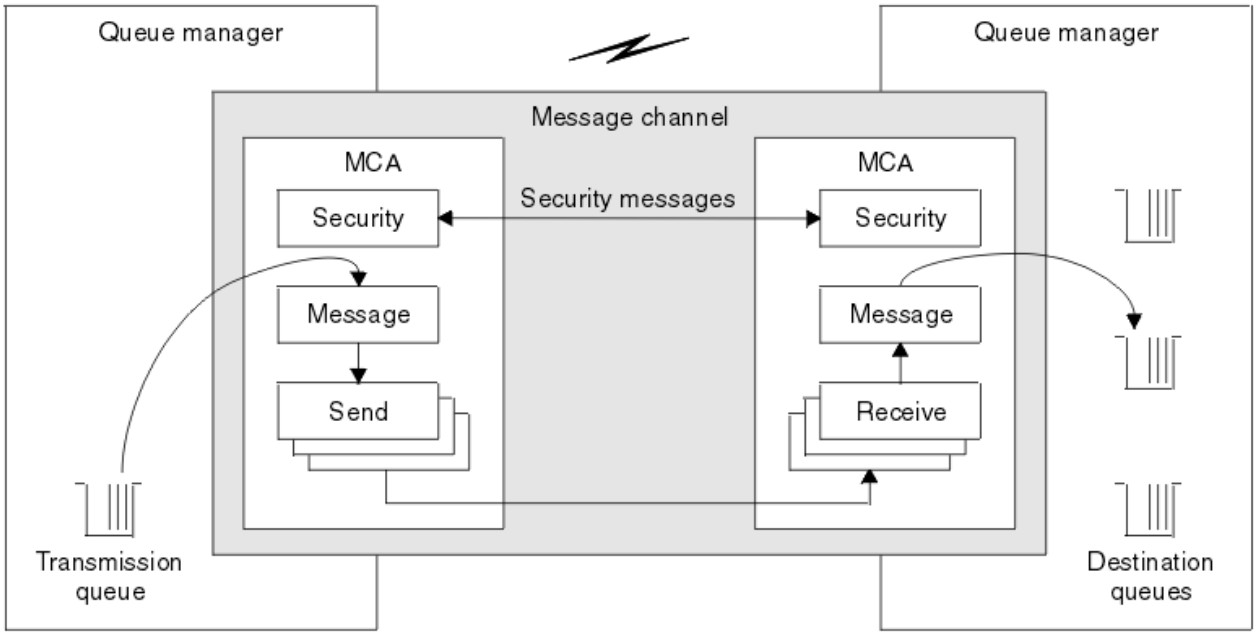


Figure 11. Exits de sécurité, de message, d'envoi et de réception sur un canal de message

Concepts associés

Programmes d'exit de canal pour les canaux de messagerie

Présentation de l'exit de sécurité

Les exits de sécurité fonctionnent normalement par paires. Ils sont appelés avant le flux de messages et leur but est de permettre à un agent MCA d'authentifier son partenaire.

Les *exits de sécurité* fonctionnent normalement par paires, une à chaque extrémité d'un canal. Ils sont appelés immédiatement après la fin de la négociation de données initiale au démarrage du canal, mais avant que les messages ne commencent à circuler. L'objectif principal de l'exit de sécurité est d'activer l'agent MCA à chaque extrémité d'un canal pour authentifier son partenaire. Cependant, rien n'empêche un exit de sécurité d'exécuter une autre fonction, même une fonction qui n'a rien à voir avec la sécurité.

Les exits de sécurité peuvent communiquer entre eux en envoyant des *messages de sécurité*. Le format du message de sécurité est défini par l'utilisateur. Un résultat possible de l'échange de messages de sécurité est que l'un des exits de sécurité peut décider de ne pas poursuivre. Dans ce cas, le canal est fermé et les messages ne circulent pas. S'il n'y a un exit de sécurité qu'à une seule extrémité d'un canal, l'exit est toujours appelé et peut choisir de continuer ou de fermer le canal.

Les exits de sécurité peuvent être appelés sur les canaux de message et MQI. Le nom d'un exit de sécurité est spécifié en tant que paramètre dans la définition de canal à chaque extrémité d'un canal.

Pour plus d'informations sur les exits de sécurité, voir «Sécurité au niveau de la liaison à l'aide d'un exit de sécurité», à la page 114.

Exit de message

Les exits de message fonctionnent uniquement sur les canaux de message et fonctionnent normalement par paires. Un exit de message peut fonctionner sur l'ensemble du message et y apporter diverses modifications.

Les exits de message aux extrémités émettrice et réceptrice d'un canal fonctionnent normalement par paires. Un exit de message à l'extrémité émettrice d'un canal est appelé une fois que l'agent MCA a reçu un message de la file d'attente de transmission. A l'extrémité réceptrice d'un canal, un exit de message est appelé avant que l'agent MCA n'insère un message dans sa file d'attente de destination.

Un exit de message a accès à l'en-tête de file d'attente de transmission, MQXQH, qui inclut le descripteur de message imbriqué, et aux données d'application d'un message. Un exit de message peut modifier le contenu du message et modifier sa longueur. Un changement de longueur peut être le résultat de la compression, de la décompression, du chiffrement ou du déchiffrement du message. Il peut également être le résultat de l'ajout de données au message ou de la suppression de données de celui-ci.

Les exits de message peuvent être utilisés à n'importe quelle fin qui nécessite l'accès à l'ensemble du message, plutôt qu'à une partie de celui-ci, et pas nécessairement pour des raisons de sécurité.

Un exit de message peut déterminer que le message qu'il est en train de traiter ne doit pas continuer vers sa destination. L'agent MCA place ensuite le message dans la file d'attente des messages non livrés. Un exit de message peut également fermer le canal.

Les exits de message peuvent être appelés uniquement sur les canaux de message et non sur les canaux MQI. En effet, l'objectif d'un canal MQI est d'activer les paramètres d'entrée et de sortie des appels MQI entre l'application IBM MQ MQI client et le gestionnaire de files d'attente.

Le nom d'un exit de message est indiqué en tant que paramètre dans la définition de canal à chaque extrémité d'un canal. Vous pouvez également spécifier une liste d'exits de message à exécuter successivement.

Pour plus d'informations sur les exits de message, voir «Sécurité au niveau de la liaison à l'aide d'un exit de message», à la page 114.

Exits d'envoi et de réception

Les exits d'envoi et de réception fonctionnent généralement par paires. Ils fonctionnent sur des segments de transmission et sont utilisés au mieux lorsque la structure des données qu'ils traitent n'est pas pertinente.

Un *exit d'émission* à une extrémité d'un canal et un *exit de réception* à l'autre extrémité fonctionnent normalement par paires. Un exit d'émission est appelé juste avant qu'un agent MCA ne lance un envoi de communications pour envoyer des données via une connexion de communication. Un exit de réception est appelé juste après qu'un agent MCA a repris le contrôle à la suite d'une réception de communications et a reçu des données d'une connexion de communication. Si le partage de conversations est en cours

d'utilisation, sur un canal MQI, une instance différente d'exit d'émission et de réception est appelée pour chaque conversation.

Les flux du protocole de canal IBM MQ entre deux agents MCA sur un canal de transmission de messages contiennent des informations de contrôle ainsi que des données de message. De même, sur un canal MQI, les flux contiennent des informations de contrôle ainsi que les paramètres des appels MQI. Les exits d'envoi et de réception sont appelés pour tous les types de données.

Les données de message ne circulent que dans une seule direction sur un canal de message mais, sur un canal MQI, les paramètres d'entrée d'un flux d'appel MQI dans une direction et les paramètres de sortie dans l'autre. Sur les canaux de message et MQI, contrôlez les flux d'informations dans les deux sens. Par conséquent, les exits d'émission et de réception peuvent être appelés aux deux extrémités d'un canal.

L'unité de données qui est transmise dans un flux unique entre deux MCM est appelée *segment de transmission*. Les exits d'émission et de réception ont accès à chaque segment de transmission. Ils peuvent modifier son contenu et sa longueur. Toutefois, un exit d'émission ne doit pas modifier les 8 premiers octets d'un segment de transmission. Ces 8 octets font partie de l'en-tête de protocole de canal IBM MQ . Il existe également des restrictions sur la mesure dans laquelle un exit d'émission peut augmenter la longueur d'un segment de transmission. En particulier, un exit d'émission ne peut pas augmenter sa longueur au-delà de la longueur maximale négociée entre les deux agents MCA au démarrage du canal.

Sur un canal de transmission, si un message est trop volumineux pour être envoyé dans un seul segment de transmission, l'agent MCA émetteur fractionne le message et l'envoie dans plusieurs segments de transmission. En conséquence, une sortie d'émission est appelée pour chaque segment de transmission contenant une partie du message et, à la réception, une sortie de réception est appelée pour chaque segment de transmission. L'agent MCA récepteur reconstitue le message des segments de transmission après qu'ils ont été traités par l'exit de réception.

De même, sur un canal MQI, les paramètres d'entrée ou de sortie d'un appel MQI sont envoyés dans plusieurs segments de transmission s'ils sont trop grands. Cela peut se produire, par exemple, sur un appel MQPUT, MQPUT1ou MQGET si les données d'application sont suffisamment volumineuses.

Compte tenu de ces considérations, il est plus approprié d'utiliser des exits d'émission et de réception à des fins dans lesquelles ils n'ont pas besoin de comprendre la structure des données qu'ils traitent et peuvent donc traiter chaque segment de transmission comme un objet binaire.

Un exit d'émission ou de réception peut fermer un canal.

Les noms d'un exit d'émission et d'un exit de réception sont spécifiés en tant que paramètres dans la définition de canal à chaque extrémité d'un canal. Vous pouvez également spécifier une liste d'exits d'émission à exécuter successivement. De même, vous pouvez spécifier une liste d'exits de réception.

Pour plus d'informations sur les exits d'envoi et de réception, voir «Sécurité au niveau de la liaison à l'aide des exits d'envoi et de réception», à la page 115.

Planification de l'intégrité des données

Planifiez la manière de préserver l'intégrité de vos données.

Vous pouvez implémenter l'intégrité des données au niveau de l'application ou du lien.

Au niveau de l'application, vous pouvez utiliser des programmes d'exit API si les fonctions standard ne répondent pas à vos besoins. Vous pouvez choisir d'utiliser Advanced Message Security (AMS) pour signer numériquement des messages afin de vous protéger contre les modifications non autorisées.

Au niveau des liens, vous pouvez choisir d'utiliser TLS, auquel cas vous devez planifier votre utilisation des certificats numériques. Vous pouvez également utiliser des programmes d'exit de canal si les fonctions standard ne répondent pas à vos besoins.

Concepts associés

«Protection des canaux avec SSL/TLS», à la page 123

La prise en charge de TLS dans IBM MQ utilise l'objet d'informations d'authentification du gestionnaire de files d'attente et diverses commandes MQSC. Vous devez également tenir compte de votre utilisation des certificats numériques.

«Intégrité des données», à la page 10
Le service d' *intégrité des données* détecte s'il y a eu une modification non autorisée des données.

«Planification de Advanced Message Security», à la page 116
Advanced Message Security (AMS) est un composant de IBM MQ qui offre un niveau de protection élevé pour les données sensibles transitant par le réseau IBM MQ , sans affecter les applications finales.

Référence associée

Référence d'exit API

Structures de données et appels d'exit de canal

Planification de l'audit

Décidez des données à auditer et de la manière dont vous allez capturer et traiter les informations d'audit. Vérifiez que votre système est correctement configuré.

La surveillance de l'activité comporte plusieurs aspects. Les aspects que vous devez prendre en compte sont souvent définis par des exigences d'auditeur, et ces exigences sont souvent dictées par des normes réglementaires telles que la loi HIPAA (Health Insurance Portability and Accountability Act) ou la loi SOX (Sarbanes-Oxley). IBM MQ fournit des fonctions destinées à faciliter la conformité à ces normes.

Déterminez si vous êtes intéressé uniquement par les exceptions ou si vous êtes intéressé par tous les comportements du système.

Certains aspects de l'audit peuvent également être considérés comme une surveillance opérationnelle ; une distinction pour l'audit est que vous examinez souvent les données historiques, et pas seulement les alertes en temps réel. La surveillance est traitée dans la section Surveillance et performances.

Données à auditer

Prenez en compte les types de données ou d'activité que vous devez auditer, comme décrit dans les sections suivantes:

Modifications apportées à IBM MQ à l'aide des interfaces IBM MQ
Configurez IBM MQ pour émettre des événements d'instrumentation, en particulier des événements de commande et des événements de configuration.

Modifications apportées à IBM MQ en dehors de son contrôle
Certaines modifications peuvent affecter le comportement de IBM MQ , mais elles ne peuvent pas être directement surveillées par IBM MQ. Par exemple, vous pouvez modifier les fichiers de configuration `mqs.ini`, `qm.ini`et `mqclient.ini`, créer et supprimer des gestionnaires de files d'attente, installer des fichiers binaires tels que des programmes d'exit utilisateur et modifier les droits d'accès aux fichiers. Pour surveiller ces activités, vous devez utiliser des outils exécutés au niveau du système d'exploitation. Différents outils sont disponibles et adaptés aux différents systèmes d'exploitation. Vous pouvez également avoir des journaux créés par des outils associés, tels que *sudo*.

Contrôle opérationnel de IBM MQ
Vous devrez peut-être utiliser les outils du système d'exploitation pour auditer les activités telles que le démarrage et l'arrêt des gestionnaires de files d'attente. Dans certains cas, IBM MQ peut être configuré pour émettre des événements d'instrumentation.

Activité d'application dans IBM MQ
Pour auditer les actions des applications, par exemple l'ouverture de files d'attente et l'insertion et l'obtention de messages, configurez IBM MQ pour émettre des événements appropriés.

Alertes d'intrus
Pour auditer les tentatives d'atteinte à la sécurité, configurez votre système pour qu'il émet des événements d'autorisation. Les événements de canal peuvent également être utiles pour afficher l'activité, en particulier si un canal se termine de manière inattendue.

Planification de la capture, de l'affichage et de l'archivage des données d'audit

La plupart des éléments dont vous avez besoin sont signalés comme des messages d'événement IBM MQ . Vous devez choisir des outils qui peuvent lire et mettre en forme ces messages. Si vous êtes intéressé par le stockage et l'analyse à long terme, vous devez les déplacer vers un mécanisme de mémoire secondaire tel qu'une base de données. Si vous ne traitez pas ces messages, ils restent dans la file d'attente d'événements, ce qui peut entraîner le remplissage de la file d'attente. Vous pouvez décider d'implémenter un outil qui exécute automatiquement des actions en fonction de certains événements ; par exemple, pour émettre une alerte lorsqu'un incident de sécurité se produit.

Vérification de la configuration correcte de votre système

Un ensemble de tests est fourni avec IBM MQ Explorer. Utilisez ces éléments pour rechercher les problèmes éventuels dans les définitions d'objet.

Vérifiez également régulièrement que la configuration du système correspond à vos attentes. Bien que les événements de commande et de configuration puissent signaler une modification, il est également utile de vider la configuration et de la comparer à une copie correcte connue.

Planification de la sécurité par topologie

Cette section traite de la sécurité dans des situations spécifiques, à savoir pour les canaux, les clusters de gestionnaires de files d'attente, les applications de publication / abonnement et de multidiffusion, et lors de l'utilisation d'un pare-feu.

Pour plus d'informations, voir les sous-rubriques suivantes:

Autorisation de canal

Lorsque vous envoyez ou recevez un message via un canal, vous devez fournir un accès à diverses ressources IBM MQ . Les agents MCA (Message Channel Agent) sont essentiellement des applications IBM MQ qui déplacent les messages entre les gestionnaires de files d'attente et qui, en tant que telles, nécessitent un accès à diverses ressources IBM MQ pour fonctionner correctement.

Pour recevoir des messages au moment de l'opération PUT pour les agents MCA, vous pouvez utiliser l'ID utilisateur associé à l'agent MCA ou l'ID utilisateur associé au message.

Au moment de la connexion, vous pouvez mapper l'ID utilisateur vérifié à un autre utilisateur, à l'aide des enregistrements d'authentification de canal **CHLAUTH** .

Dans IBM MQ, les canaux peuvent être protégés par le support TLS.

Les ID utilisateur associés aux canaux d'envoi et de réception, à l'exception du canal émetteur où l'attribut MCAUSER n'est pas utilisé, requièrent l'accès aux ressources suivantes:

- L'ID utilisateur associé à un canal émetteur requiert l'accès au gestionnaire de files d'attente, à la file d'attente de transmission, à la file d'attente de rebut et à toute autre ressource requise par les exits de canal.
- L'ID utilisateur MCAUSER d'un canal récepteur requiert les droits *+ setall* . En effet, le canal récepteur doit créer le MQMD complet, y compris toutes les zones de contexte, à l'aide des données qu'il a reçues du canal émetteur distant. Le gestionnaire de files d'attente requiert donc que l'utilisateur exécutant cette activité dispose des droits *+ setall* . Ces droits *+ setall* doivent être accordés à l'utilisateur pour:
  - Toutes les files d'attente dans lesquelles le canal récepteur insère des messages de manière valide.
  - Objet gestionnaire de files d'attente. Pour plus d'informations, voir Autorisations de contexte.
- L'ID utilisateur MCAUSER d'un canal récepteur sur lequel l'émetteur a demandé un message de rapport COA requiert le droit *+ passid* sur la file d'attente de transmission qui renvoie le message de rapport. Sans ces droits, les messages d'erreur AMQ8077 sont consignés.
- Avec l'ID utilisateur associé au canal récepteur, vous pouvez ouvrir les files d'attente cible pour y placer des messages. Cela implique l'interface MQI (Message queuing Interface), de sorte que des vérifications de contrôle d'accès supplémentaires peuvent être nécessaires si vous n'utilisez pas IBM

MQ Object Authority Manager (OAM). Vous pouvez indiquer si les vérifications d'autorisation sont effectuées sur l'ID utilisateur associé à l'agent MCA (comme décrit dans cette rubrique) ou sur l'ID utilisateur associé au message (à partir de la zone MQMD UserIdentifier).

Pour les types de canal auxquels il s'applique, le paramètre **PUTAUT** d'une définition de canal indique l'ID utilisateur utilisé pour ces vérifications.

- Par défaut, le canal utilise le compte de service du gestionnaire de files d'attente, qui dispose de droits d'administration complets et ne requiert aucune autorisation spéciale.
- Dans le cas des canaux de connexion serveur, les connexions d'administration sont bloquées par défaut par les règles CHLAUTH et nécessitent une mise à disposition explicite.
- Les canaux de type récepteur, demandeur et récepteur de cluster permettent l'administration locale par tout gestionnaire de files d'attente adjacent, sauf si l'administrateur prend des mesures pour restreindre cet accès.
- Il n'est pas nécessaire d'accorder les droits *dsp* et *ctrlx* pour l'ID utilisateur MCAUSER d'un canal récepteur.
- Avant IBM MQ 8.0.0 Fix Pack 4, si vous utilisez un ID utilisateur qui ne dispose pas de privilèges d'administration IBM MQ , vous devez accorder les droits **dsp** et **ctrlx** pour le canal à cet ID utilisateur pour que le canal fonctionne.

  Depuis la IBM MQ 8.0.0 Fix Pack 4, il n'y a pas de contrôle des droits d'accès lorsqu'un canal se resynchronise et corrige les numéros de séquence.

  Toutefois, l'émission manuelle d'une commande RESET CHANNEL requiert toujours **+dsp** et **+ctrlx** dans toutes les éditions.

  Avertissement : Lorsqu'une réinitialisation de canal est nécessaire pour la confirmation par lots de messages, IBM MQ tente d'interroger le canal, ce qui nécessite des droits d'accès **+dsp** .
- L'attribut MCAUSER n'est pas utilisé pour le type de canal SDR.
- Si vous utilisez l'ID utilisateur associé au message, il est probable que l'ID utilisateur provient d'un système distant. Cet ID utilisateur de système distant doit être reconnu par le système cible. Les commandes suivantes sont des exemples du type de commande que vous pouvez exécuter pour accorder des droits à un ID utilisateur à partir d'un système distant:

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

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

où *Profil* est un canal.

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

où *Profil* est une file d'attente de rebut, si elle est définie.

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

où *Profil* est une liste de files d'attente autorisées.

Avertissement : Soyez prudent lorsque vous autorisez un ID utilisateur à placer des messages dans la file d'attente de commandes ou dans d'autres files d'attente système sensibles.

L'ID utilisateur associé à l'agent MCA dépend du type d'agent MCA. Il existe deux types d'agent MCA:

Agent MCA appelant
Les agents MCA qui initient un canal. Les agents MCA appelants peuvent être démarrés en tant que processus individuels, en tant qu'unités d'exécution de l'initiateur de canal ou en tant qu'unités d'exécution d'un pool de processus. L'ID utilisateur utilisé est l'ID utilisateur associé au processus parent (initiateur de canal) ou l'ID utilisateur associé au processus qui démarre l'agent MCA.

Agent MCA répondeur

Les agents MCA répondeurs sont des agents MCA démarrés à la suite d'une demande d'un agent MCA appelant. Les agents MCA répondeurs peuvent être démarrés en tant que processus individuels, en tant qu'unités d'exécution du programme d'écoute ou en tant qu'unités d'exécution d'un pool de processus. L'ID utilisateur peut être l'un des types suivants (dans cet ordre de préférence):

1. Sur APPC, l'agent MCA appelant peut indiquer l'ID utilisateur à utiliser pour l'agent MCA répondeur. Cet ID est appelé ID utilisateur réseau et s'applique uniquement aux canaux démarrés en tant que processus individuels. Définissez l'ID utilisateur réseau à l'aide du paramètre USERID de la définition de canal.
2. Si le paramètre **USERID** n'est pas utilisé, la définition de canal de l'agent MCA répondeur peut indiquer l'ID utilisateur que l'agent MCA doit utiliser. Définissez l'ID utilisateur à l'aide du paramètre **MCAUSER** de la définition de canal.
3. Si l'ID utilisateur n'a été défini par aucune des deux méthodes précédentes, l'ID utilisateur du processus qui démarre l'agent MCA ou l'ID utilisateur du processus parent (le programme d'écoute) est utilisé.

Concepts associés

«Enregistrements d'authentification de canal», à la page 54
Pour exercer un contrôle plus précis sur les accès accordés aux systèmes en cours de connexion au niveau d'un canal, vous pouvez utiliser les enregistrements d'authentification de canal.

Référence associée

Propriétés de l'enregistrement d'authentification de canal

Protection des définitions d'initialisateur de canal

Seuls les membres du groupe mqm peuvent manipuler les initiateurs de canal.

Les initiateurs de canal IBM MQ ne sont pas des objets IBM MQ ; leur accès n'est pas contrôlé par la méthode d'accès aux objets (OAM). IBM MQ n'autorise pas les utilisateurs ou les applications à manipuler ces objets, sauf si leur ID utilisateur est membre du groupe mqm. Si vous disposez d'une application qui émet la commande PCF **StartChannelInitiator**, l'ID utilisateur spécifié dans le descripteur de message du message PCF doit être membre du groupe mqm sur le gestionnaire de files d'attente cible.

Un ID utilisateur doit également être membre du groupe mqm sur la machine cible pour émettre les commandes MQSC équivalentes via la commande Escape PCF ou à l'aide de `runmqsc` en mode indirect.

Files d'attente de transmission

Les gestionnaires de files d'attente placent automatiquement les messages éloignés dans une file d'attente de transmission ; aucun droit spécial n'est requis à cet effet.

Toutefois, si vous devez placer un message directement dans une file d'attente de transmission, vous devez disposer d'une autorisation spéciale ; voir Tableau 12, à la page 141.

Exits de canal

Si les enregistrements d'authentification de canal ne conviennent pas, vous pouvez utiliser des exits de canal pour une sécurité accrue. Un exit de sécurité établit une connexion sécurisée entre deux programmes d'exit de sécurité. Un programme est destiné à l'agent MCA émetteur et un programme est destiné à l'agent MCA récepteur.

Pour plus d'informations sur les exits de canal, voir «Programmes d'exit de canal», à la page 117 .

Protection des canaux avec SSL/TLS

La prise en charge de TLS dans IBM MQ utilise l'objet d'informations d'authentification du gestionnaire de files d'attente et diverses commandes MQSC. Vous devez également tenir compte de votre utilisation des certificats numériques.

Certificats numériques et référentiels de clés

Il est recommandé de définir l'attribut de label de certificat du gestionnaire de files d'attente (**CERTLABL**) au nom du certificat personnel à utiliser pour la majorité des canaux, et le remplacer pour les exceptions, en définissant le label de certificat sur les canaux qui requièrent des certificats différents.

Si vous avez besoin de plusieurs canaux avec des certificats qui diffèrent du certificat par défaut défini sur le gestionnaire de files d'attente, vous devez envisager de diviser les canaux entre plusieurs gestionnaires de files d'attente ou d'utiliser un proxy MQIPT devant le gestionnaire de files d'attente pour présenter un certificat différent.

Vous pouvez utiliser un certificat différent pour chaque canal, mais si vous stockez un trop grand nombre de certificats dans un référentiel de clés, vous pouvez vous attendre à ce que les performances soient affectées lors du démarrage des canaux TLS. Essayez de maintenir le nombre de certificats dans un référentiel de clés à moins de 50 environ et considérez que 100 est un maximum car les performances d' IBM Global Security Kit (GSKit) diminuent fortement avec les référentiels de clés plus volumineux.

L'autorisation de plusieurs certificats sur le même gestionnaire de files d'attente augmente les chances que plusieurs certificats de l'autorité de certification soient utilisés sur le même gestionnaire de files d'attente. Cela augmente les chances de conflits d'espace de nom de nom distinctif de sujet de certificat pour les certificats émis par des autorités de certification distinctes.

Alors que les autorités de certification professionnelles sont susceptibles d'être plus prudentes, les autorités de certification internes manquent souvent de conventions de dénomination claires et vous pourriez vous retrouver avec des correspondances inattendues entre une autorité de certification et une autre.

Vous devez vérifier le nom distinctif de l'émetteur du certificat en plus du nom distinctif du sujet. Pour ce faire, utilisez un enregistrement SSLPEERMAP d'authentification de canal et définissez les zones **SSLPEER** et **SSLCERTI** pour qu'elles correspondent respectivement au nom distinctif du sujet et au nom distinctif de l'émetteur.

Certificats autosignés et signés par une autorité de certification

Il est important de planifier votre utilisation des certificats numériques, à la fois lorsque vous développez et testez votre application, et pour son utilisation en production. Vous pouvez utiliser des certificats signés par une autorité de certification ou des certificats autosignés, en fonction de l'utilisation des gestionnaires de files d'attente et des applications client.

Certificats signés par une autorité de certification
Pour les systèmes de production, procurez-vous vos certificats auprès d'une autorité de certification digne de confiance. Lorsque vous obtenez un certificat d'une autorité de certification externe, vous payez pour le service.

certificats autosignés
Lors du développement de votre application, vous pouvez utiliser des certificats autosignés ou des certificats émis par une autorité de certification locale, en fonction de la plateforme:

ALW Sur les systèmes AIX, Linux, and Windows , vous pouvez utiliser des certificats autosignés. Voir «Création d'un certificat personnel autosigné sur AIX, Linux, and Windows», à la page 559 pour des instructions.

IBM i Sur les systèmes IBM i , vous pouvez utiliser des certificats signés par l'autorité de certification locale. Voir «Demande d'un certificat serveur sous IBM i», à la page 294 pour des instructions.

z/OS Sous z/OS, vous pouvez utiliser des certificats autosignés ou des certificats signés par une autorité de certification locale. Pour obtenir des instructions, voir «Creating a self-signed personal certificate on z/OS», à la page 321 ou «Requesting a personal certificate on z/OS», à la page 321 .

Les certificats autosignés ne conviennent pas à une utilisation en production pour les raisons suivantes :

- Les certificats autosignés ne peuvent pas être révoqués, ce qui peut permettre à un agresseur de usurper une identité après qu'une clé privée a été compromise. Les autorités de certification peuvent révoquer un certificat compromis, pour en empêcher toute utilisation future. Les certificats signés par une autorité de certification sont donc plus sûrs à utiliser dans un environnement de production, bien que les certificats autosignés soient plus pratiques pour un système de test.
- Les certificats autosignés n'arrivent jamais à expiration. Ce comportement est pratique et sûr dans un environnement de test, mais dans un environnement de production, les certificats restent ouverts et donc sujets à des violations de sécurité. Ce risque est aggravé du fait que les certificats autosignés ne peuvent pas être révoqués.
- Un certificat autosigné est utilisé à la fois comme certificat personnel et comme certificat d'autorité de certification racine (ou ancrage sécurisé). Un utilisateur avec un certificat personnel autosigné doit pouvoir l'utiliser pour signer d'autres certificats personnels. En général, cela n'est pas vrai des certificats personnels émis par une autorité de certification et représente un risque important.

CipherSpecs et certificats numériques

Seul un sous-ensemble des CipherSpecs pris en charge peut être utilisé avec tous les types de certificat numérique pris en charge. Il est donc nécessaire de choisir un CipherSpec approprié pour vos certificats numériques. De même, si la stratégie de sécurité de votre organisation requiert l'utilisation d'un CipherSpec particulier, vous devez obtenir des certificats numériques appropriés.

Pour plus d'informations sur la relation entre les CipherSpecs et les certificats numériques, voir «Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49

Règles de validation de certificat

La norme IETF RFC 5280 spécifie une série de règles de validation de certificat que les logiciels d'application conformes doivent implémenter afin d'éviter les attaques d'usurpation d'identité. Un ensemble de règles de validation de certificat est appelé règle de validation de certificat. Pour plus d'informations sur les règles de validation de certificat dans IBM MQ, voir «Règles de validation de certificat dans IBM MQ», à la page 47.

Planification de la vérification de la révocation de certificat

L'autorisation de plusieurs certificats provenant de différentes autorités de certification peut entraîner une vérification supplémentaire inutile de la révocation des certificats.

En particulier, si vous avez configuré explicitement l'utilisation d'un serveur de révocation d'une autorité de certification particulière, par exemple en utilisant un objet AUTHINFO ou une structure d'enregistrement d'informations d'authentification (MQAIR), une vérification de révocation échoue lorsqu'elle est présentée avec un certificat d'une autre autorité de certification.

Vous devez éviter la configuration explicite du serveur de révocation de certificat. Au lieu de cela, vous devez activer la vérification implicite lorsque chaque certificat contient son propre emplacement de serveur de révocation dans une extension de certificat, par exemple, CRL Distribution Point ou OCSP AuthorityInfoAccess.

Pour plus d'informations, voir OCSPCheckExtensions et CDPCheckExtensions.

Commandes et attributs pour la prise en charge de TLS

Le protocole TLS (Transport Layer Security) fournit une sécurité de canal, avec une protection contre les écoutes clandestines, les falsifications et les usurpations d'identité. La prise en charge de TLS par IBM MQ vous permet de spécifier, dans la définition de canal, qu'un canal particulier utilise la sécurité TLS. Vous pouvez également spécifier les détails du type de sécurité de votre choix, par exemple l'algorithme de chiffrement que vous souhaitez utiliser.

- Les commandes MQSC suivantes prennent en charge TLS:

ALTER AUTHINFO
Modifie les attributs d'un objet d'informations d'authentification.

DEFINE AUTHINFO
Crée un objet d'informations d'authentification.

DELETE AUTHINFO
Supprime un objet d'informations d'authentification.

INFORMATIONS D'AUTHENTIFICATION D'AFFICHAGE
Affiche les attributs d'un objet d'informations d'authentification spécifique.

- Les paramètres de gestionnaire de files d'attente suivants prennent en charge TLS:

CERTLABL
Définit un libellé de certificat personnel à utiliser.

Mot de passe de clé
Sur les systèmes AIX, Linux, and Windows , définit le mot de passe utilisé par IBM MQ pour accéder au référentiel de clés. Cette zone est chiffrée à l'aide du système de protection par mot de passe.

SSLCRLNL
L'attribut SSLCRLNL spécifie une liste de noms d'objets d'informations d'authentification qui sont utilisés pour fournir des emplacements de révocation de certificat afin de permettre une vérification améliorée des certificats TLS.

SSLCRYP
Sur les systèmes AIX, Linux, and Windows , définissez l'attribut de gestionnaire de files d'attente **SSLCryptoHardware** . Cet attribut est le nom de la chaîne de paramètres que vous pouvez utiliser pour configurer le matériel cryptographique que vous avez sur votre système.

SSLEV
Détermine si un message d'événement TLS est signalé si un canal utilisant TLS ne parvient pas à établir une connexion TLS.

SSLFIPS
Indique si seuls les algorithmes certifiés FIPS doivent être utilisés si la cryptographie est effectuée dans IBM MQ , plutôt que dans le matériel de cryptographie. Si le matériel de cryptographie est configuré, les modules de cryptographie fournis par le produit matériel sont utilisés et ceux-ci peuvent être certifiés FIPS à un niveau particulier. Cela dépend du produit matériel utilisé.

SSLKEYR
Sur les systèmes AIX, Linux, and Windows , associe un référentiel de clés à un gestionnaire de files d'attente. GSKit vous permet d'utiliser la sécurité TLS sur les systèmes AIX, Linux, and Windows .

SSLRKEYC
Nombre d'octets à envoyer et à recevoir dans une conversation TLS avant la renégociation de la clé secrète. Le nombre d'octets inclut les informations de contrôle envoyées par l'agent MCA.

- Les paramètres de canal suivants prennent en charge TLS:

CERTLABL
Définit un libellé de certificat personnel à utiliser.

SSLCAUTH
Indique si IBM MQ requiert et valide un certificat du client TLS.

SSLCIPH
Indique la force et la fonction de chiffrement (CipherSpec), par exemple TLS_RSA_WITH_AES_128_CBC_SHA. Le CipherSpec doit correspondre aux deux extrémités du canal.

SSLPEER
Indique le nom distinctif (identificateur unique) des partenaires autorisés.

Cette section décrit les commandes **setmqaut**, **dspmqaut**, **dmpmqaut**, **rcrmqobj**, **rcdmqimg**et **dspmqfls** permettant de prendre en charge l'objet d'informations d'authentification. Il décrit également les commandes qui peuvent être utilisées pour gérer les clés et les certificats sur AIX, Linux, and Windows. Reportez-vous aux sections suivantes :

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- «Gestion des clés et des certificats sur AIX, Linux, and Windows», à la page 557

Pour une présentation de la sécurité des canaux à l'aide de TLS, voir

- «Protocoles de sécurité TLS dans IBM MQ», à la page 25

Pour plus de détails sur les commandes MQSC associées à TLS, voir

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

Pour plus de détails sur les commandes PCF associées à TLS, voir

- Modifier, copier et créer un objet d'informations d'authentification
- Supprimer l'objet d'informations d'authentification
- Objet d'interrogation des informations d'authentification

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 “Programmes d'exit de canal” on page 117 for more information about channel exits.

Related tasks

Writing channel exit programs on z/OS

Services de sécurité SNA LU 6.2

L'unité logique SNA 6.2 offre la cryptographie au niveau de la session, l'authentification au niveau de la session et l'authentification au niveau de la conversation.

Remarque : Cette collection de rubriques suppose que vous avez une connaissance de base de l'architecture SNA (Systems Network Architecture). L'autre documentation mentionnée dans cette section contient une brève introduction aux concepts et à la terminologie pertinents. Si vous avez besoin d'une introduction technique plus complète à SNA, voir *Systems Network Architecture Technical Overview*, GC30-3073.

SNA LU 6.2 fournit trois services de sécurité:

- Cryptographie de niveau session
- Authentification au niveau de la session
- Authentification au niveau de la conversation

Pour la cryptographie au niveau de la session et l'authentification au niveau de la session, SNA utilise l'algorithme *Data Encryption Standard (DES)* . L'algorithme DES est un algorithme de chiffrement par blocs qui utilise une clé symétrique pour le chiffrement et le déchiffrement des données. La longueur du bloc et de la clé est de 8 octets.

Cryptographie de niveau session
La *cryptographie au niveau de la session* chiffre et déchiffre les données de session à l'aide de l'algorithme DES. Il peut donc être utilisé pour fournir un service de confidentialité de niveau liaison sur les canaux SNA LU 6.2 .

Les unités logiques peuvent fournir une cryptographie de données obligatoire (ou obligatoire), une cryptographie de données sélective ou aucune cryptographie de données.

Dans une *session cryptographique obligatoire*, une unité logique chiffre toutes les unités de demande de données sortantes et déchiffre toutes les unités de demande de données entrantes.

Dans une *session cryptographique sélective*, une unité logique chiffre uniquement les unités de demande de données spécifiées par le programme de transaction d'envoi (TP). L'unité logique émettrice signale que les données sont chiffrées en définissant un indicateur dans l'en-tête de demande. En vérifiant cet indicateur, la LU réceptrice peut savoir quelles unités de requête déchiffrer avant de les transmettre à la TP réceptrice.

Dans un réseau SNA, les agents IBM MQ MCA sont des programmes de transaction. Les agents MCA ne demandent pas de chiffrement pour les données qu'ils envoient. La cryptographie sélective de données n'est donc pas une option ; seule la cryptographie de données obligatoire ou aucune cryptographie de données est possible sur une session.

Pour plus d'informations sur l'implémentation de la cryptographie de données obligatoire, voir la documentation de votre sous-système SNA. Reportez-vous à la même documentation pour plus d'informations sur les formes de chiffrement renforcé pouvant être utilisées sur votre plateforme, comme le chiffrement Triple DES 24 octets sur z/OS.

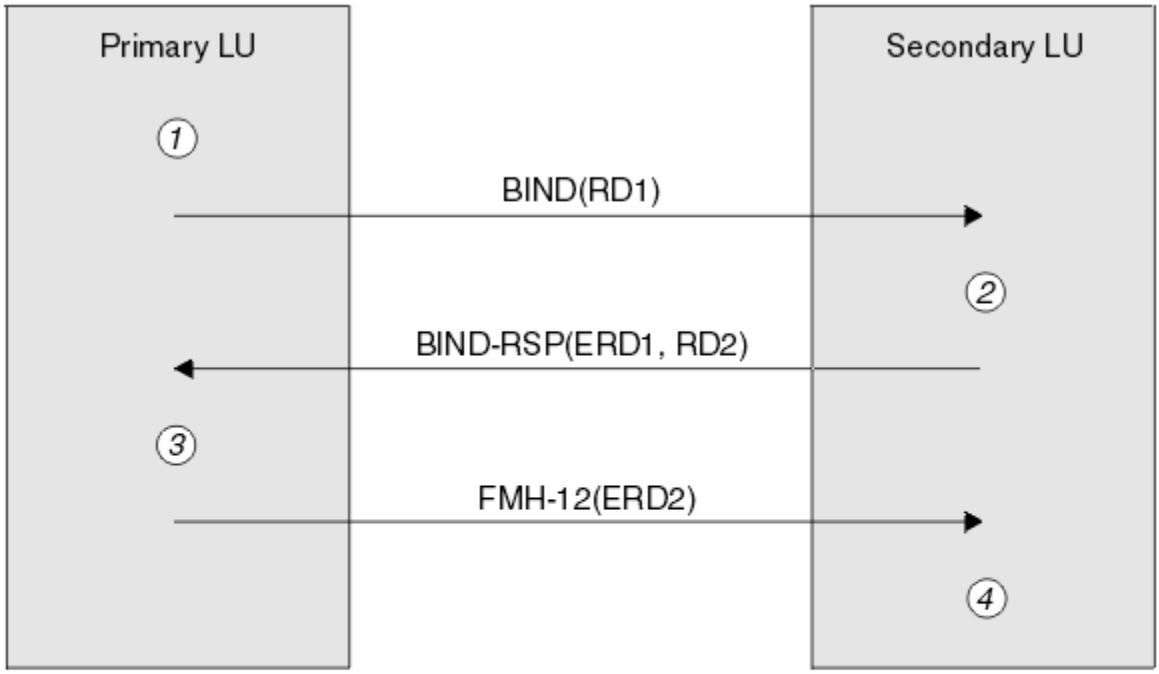
Pour plus d'informations sur la cryptographie de niveau session, voir *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

Authentification au niveau de la session
L' *authentification au niveau de la session* est un protocole de sécurité au niveau de la session qui permet à deux unités logiques de s'authentifier l'une l'autre lors de l'activation d'une session. Elle est également appelée *vérification LU-LU*.

Etant donné qu'une unité logique est effectivement la "passerelle" dans un système à partir du réseau, vous pouvez considérer que ce niveau d'authentification est suffisant dans certaines circonstances. Par exemple, si votre gestionnaire de files d'attente doit échanger des messages avec un gestionnaire de files d'attente éloignées qui s'exécute dans un environnement contrôlé et sécurisé, vous pouvez être prêt à faire confiance aux identités des autres composants du système distant une fois que l'unité logique a été authentifiée.

L'authentification au niveau de la session est effectuée par chaque unité logique qui vérifie le mot de passe de son partenaire. Le mot de passe est appelé *mot de passe LU-LU* car un mot de passe est établi entre chaque paire d'unités logiques. Le mode d'établissement d'un mot de passe LU-LU dépend de l'implémentation et est hors de la portée de SNA.

La Figure 12, à la page 130 illustre les flux d'authentification au niveau de la session.



Legend:

BIND	=	BIND request unit
BIND-RSP	=	BIND response unit
ERD	=	Encrypted random data
FMH-12	=	Function Management Header 12
RD	=	Random data

Figure 12. Flux pour l'authentification au niveau de la session

Le protocole d'authentification au niveau de la session est le suivant. Les nombres de la procédure correspondent aux nombres de Figure 12, à la page 130.

1. L'unité logique principale génère une valeur de données aléatoire (RD1) et l'envoie à l'unité logique secondaire dans la demande BIND.
2. Lorsque la LU secondaire reçoit la requête BIND avec les données aléatoires, elle chiffre les données à l'aide de l'algorithme DES avec sa copie du mot de passe LU-LU comme clé. L'unité logique secondaire génère ensuite une deuxième valeur de données aléatoires (RD2) et l'envoie, avec les données chiffrées (ERD1), à l'unité logique principale dans la réponse BIND.
3. Lorsque l'unité logique principale reçoit la réponse BIND, elle calcule sa propre version des données chiffrées à partir des données aléatoires qu'elle a générées à l'origine. Pour ce faire, il utilise l'algorithme DES avec sa copie du mot de passe LU-LU comme clé. Il compare ensuite sa version aux données chiffrées qu'il a reçues dans la réponse BIND. Si les deux valeurs sont identiques, l'unité logique principale sait que l'unité logique secondaire possède le même mot de passe et que l'unité logique secondaire est authentifiée. Si les deux valeurs ne correspondent pas, l'unité logique principale met fin à la session.

   L'unité logique principale chiffre ensuite les données aléatoires qu'elle a reçues dans la réponse BIND et envoie les données chiffrées (ERD2) à l'unité logique secondaire dans un en-tête de gestion de fonction 12 (FMH-12).
4. Lorsque l'unité logique secondaire reçoit le FMH-12, elle calcule sa propre version des données chiffrées à partir des données aléatoires qu'elle a générées. Il compare ensuite sa version aux données chiffrées qu'il a reçues dans le FMH-12. Si les deux valeurs sont identiques, l'unité logique principale est authentifiée. Si les deux valeurs ne correspondent pas, l'unité logique secondaire met fin à la session.

Dans une version améliorée du protocole, qui offre une meilleure protection contre les attaques de l'homme du milieu, la LU secondaire calcule un code d'authentification de message (MAC) DES à partir de RD1, RD2et du nom qualifié complet de la LU secondaire, en utilisant sa copie du mot de passe LU-LU comme clé. L'unité logique secondaire envoie l'adresse MAC à l'unité logique principale dans la réponse BIND au lieu de ERD1.

L'unité logique principale authentifie l'unité logique secondaire en calculant sa propre version du MAC, qu'elle compare au MAC reçu dans la réponse BIND. L'unité logique principale calcule ensuite une seconde adresse MAC à partir de RD1 et RD2, et envoie l'adresse MAC à l'unité logique secondaire dans FMH-12 au lieu de ERD2.

L'unité logique secondaire authentifie l'unité logique principale en calculant sa propre version du deuxième MAC, qu'elle compare avec le MAC reçu dans le FMH-12.

Pour plus d'informations sur la configuration de l'authentification au niveau de la session, voir la documentation de votre sous-système SNA. Pour plus d'informations sur l'authentification de niveau session, voir *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

Authentification au niveau de la conversation
Lorsqu'un programme transactionnel local tente d'allouer une conversation avec un programme transactionnel partenaire, l'unité logique locale envoie une demande de connexion à l'unité logique partenaire, en lui demandant de connecter le programme transactionnel partenaire. Dans certaines circonstances, la demande d'association peut contenir des informations de sécurité que l'unité logique partenaire peut utiliser pour authentifier le TP local. Il s'agit de l' *authentification au niveau de la conversation*ou de la *vérification de l'utilisateur final*.

Les rubriques suivantes décrivent comment IBM MQ fournit la prise en charge de l'authentification au niveau de la conversation.

Pour plus d'informations sur l'authentification au niveau de la conversation, voir *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

z/OS Pour des informations spécifiques à z/OS, voir z/OS MVS Planning: APPC/MVS Management.

Pour plus d'informations sur CPI-C, voir Utilisation des communications CPI.

Pour plus d'informations sur les services d'appels de conversation TP APPC/MVS, voir Services d'appels de conversation TP APPC/MVS.

Multi *Prise en charge de l'authentification au niveau de la conversation sur Multiplatforms*
Utilisez cette rubrique pour obtenir une vue d'ensemble du fonctionnement de l'authentification au niveau de la conversation sur Multiplatforms.

La prise en charge de l'authentification au niveau de la conversation sur Multiplatforms est illustrée dans Figure 13, à la page 132. Les numéros du diagramme correspondent aux numéros de la description qui suit.

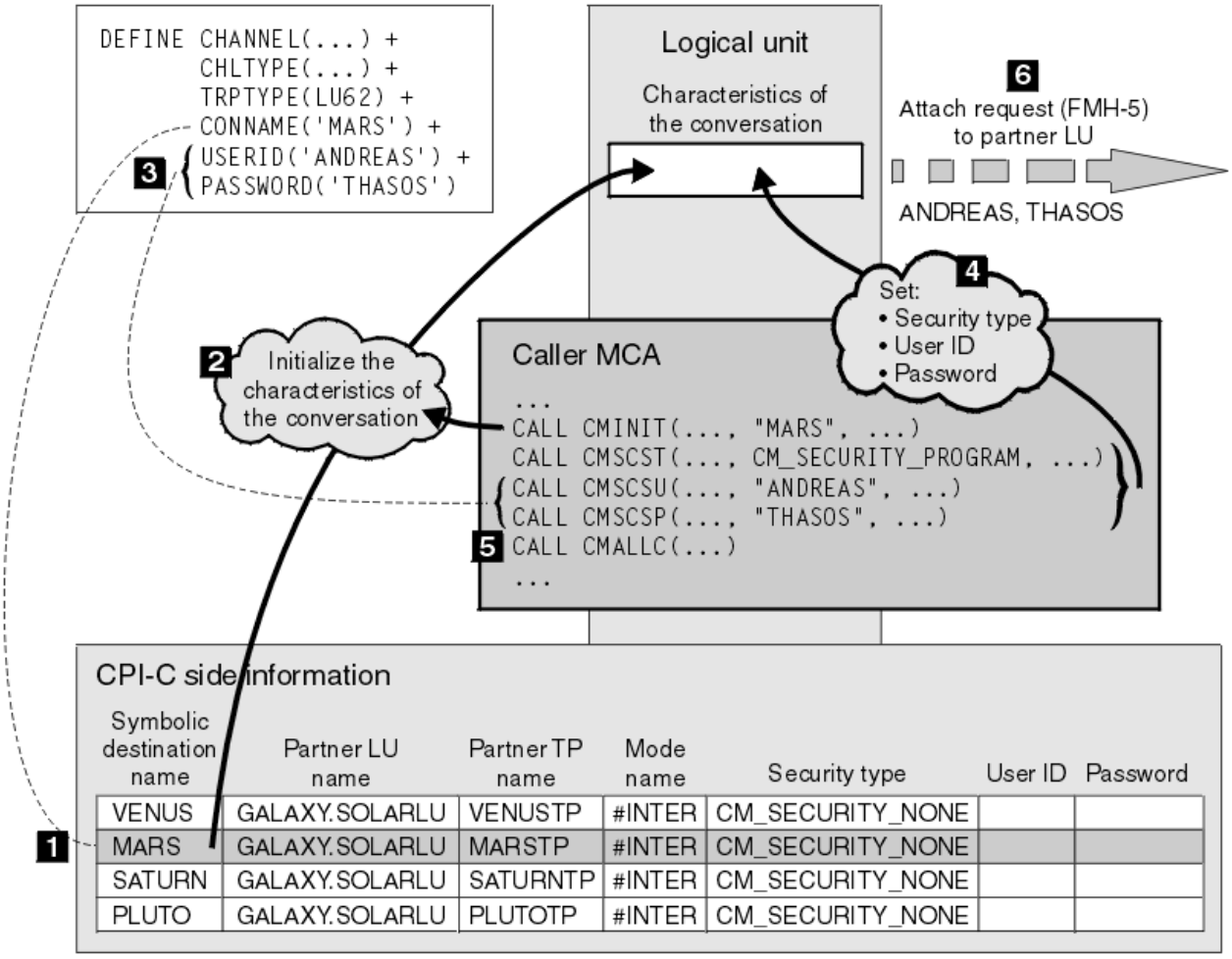


Figure 13. Prise en charge par IBM MQ de l'authentification au niveau de la conversation

Sur Multiplatforms, un agent MCA utilise des appels CPI-C (Common Programming Interface Communications) pour communiquer avec un agent MCA partenaire sur un réseau SNA. Dans la définition de canal à l'extrémité appelante d'un canal, la valeur du paramètre CONNAME est un nom de destination symbolique qui identifie une entrée d'informations côté CPI-C (1). Cette entrée indique:

- Nom de l'unité logique partenaire
- Nom du programme transactionnel partenaire, qui est un agent MCA répondeur
- Nom du mode à utiliser pour la conversation

Une entrée d'informations complémentaires peut également spécifier les informations de sécurité suivantes:

- Type de sécurité.

  Les types de sécurité couramment implémentés sont CM_SECURITY_NONE, CM_SECURITY_PROGRAM et CM_SECURITY_SAME, mais d'autres sont définis dans la spécification CPI-C.
- ID utilisateur.
- Un mot de passe.

Un agent MCA appelant se prépare à allouer une conversation avec un agent MCA répondeur en émettant l'appel CPI-C CMINIT, en utilisant la valeur de CONNAME comme l'un des paramètres de l'appel. L'appel CMINIT identifie, pour le bénéfice de l'unité logique locale, l'entrée d'informations complémentaires que l'agent MCA a l'intention d'utiliser pour la conversation. L'unité logique locale utilise les valeurs de cette entrée pour initialiser les caractéristiques de la conversation (2).

L'agent MCA appelant vérifie ensuite les valeurs des paramètres USERID et PASSWORD dans la définition de canal (3). Si USERID est défini, l'agent MCA appelant émet les appels CPI-C suivants (4):

- CMSCST, pour définir le type de sécurité de la conversation sur CM_SECURITY_PROGRAM.
- CMSCSU, pour définir l'ID utilisateur de la conversation sur la valeur USERID.
- CMSCSP, pour définir le mot de passe de la conversation sur la valeur PASSWORD. CMSCSP n'est pas appelé sauf si PASSWORD est défini.

Le type de sécurité, l'ID utilisateur et le mot de passe définis par ces appels remplacent toutes les valeurs précédemment acquises à partir de l'entrée d'informations complémentaires.

L'agent MCA appelant émet ensuite l'appel CPI-C CMALLC pour allouer la conversation (5). En réponse à cet appel, l'unité logique locale envoie une demande d'association (Function Management Header 5, ou FMH-5) à l'unité logique partenaire (6).

Si l'unité logique partenaire accepte un ID utilisateur et un mot de passe, les valeurs USERID et PASSWORD sont incluses dans la demande d'association. Si l'unité logique partenaire n'accepte pas d'ID utilisateur et de mot de passe, les valeurs ne sont pas incluses dans la demande d'association. L'unité logique locale détermine si l'unité logique partenaire accepte un ID utilisateur et un mot de passe dans le cadre d'un échange d'informations lorsque les unités logiques se lient pour former une session.

Dans une version ultérieure de la demande d'association, un remplacement de mot de passe peut se produire entre les unités logiques au lieu d'un mot de passe clair. Un remplaçant de mot de passe est un code d'authentification de message DES (MAC) ou un résumé de message SHA-1 , formé à partir du mot de passe. Les remplacements de mot de passe ne peuvent être utilisés que si les deux unités logiques les prennent en charge.

Lorsque l'unité logique partenaire reçoit une demande d'association entrante contenant un ID utilisateur et un mot de passe, elle peut utiliser l'ID utilisateur et le mot de passe à des fins d'identification et d'authentification. En faisant référence aux listes de contrôle d'accès, l'unité logique partenaire peut également déterminer si l'ID utilisateur a le droit d'allouer une conversation et de connecter l'agent MCA répondeur.

En outre, l'agent MCA répondeur peut s'exécuter sous l'ID utilisateur inclus dans la demande d'association. Dans ce cas, l'ID utilisateur devient l'ID utilisateur par défaut pour l'agent MCA répondeur et est utilisé pour les vérifications des droits d'accès lorsque l'agent MCA tente de se connecter au gestionnaire de files d'attente. Il peut également être utilisé pour les vérifications des droits d'accès ultérieures lorsque l'agent MCA tente d'accéder aux ressources du gestionnaire de files d'attente.

La manière dont un ID utilisateur et un mot de passe dans une demande de connexion peuvent être utilisés pour l'identification, l'authentification et le contrôle d'accès dépend de l'implémentation. Pour des informations spécifiques à votre sous-système SNA, reportez-vous à la documentation appropriée.

Si USERID n'est pas défini, l'agent MCA appelant n'appelle pas CMSCST, CMSCSU et CMSCSP. Dans ce cas, les informations de sécurité qui circulent dans une demande d'association sont uniquement déterminées par ce qui est spécifié dans l'entrée d'informations complémentaires et par ce que l'unité logique partenaire accepte.

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.

Sécurité des clusters de gestionnaires de files d'attente

Bien que les clusters de gestionnaires de files d'attente soient pratiques à utiliser, vous devez accorder une attention particulière à leur sécurité.

Un *cluster de gestionnaires de files d'attente* est un réseau de gestionnaires de files d'attente associés de manière logique. Un gestionnaire de files d'attente qui est membre d'un cluster est appelé *gestionnaire de files d'attente de cluster*.

Une file d'attente appartenant à un gestionnaire de files d'attente de cluster peut être rendue connue des autres gestionnaires de files d'attente du cluster. Cette file d'attente est appelée *file d'attente de cluster*. Tout gestionnaire de files d'attente d'un cluster peut envoyer des messages à des files d'attente de cluster sans avoir besoin de l'un des éléments suivants:

- Une définition de file d'attente éloignée explicite pour chaque file d'attente de cluster
- Canaux définis explicitement vers et depuis chaque gestionnaire de files d'attente éloignées
- Une file d'attente de transmission distincte pour chaque canal sortant

Vous pouvez créer un cluster dans lequel au moins deux gestionnaires de files d'attente sont des clones. Cela signifie qu'ils possèdent des instances des mêmes files d'attente locales, y compris des files d'attente locales déclarées comme files d'attente de cluster, et qu'ils peuvent prendre en charge des instances des mêmes applications serveur.

Lorsqu'une application connectée à un gestionnaire de files d'attente de cluster envoie un message à une file d'attente de cluster comportant une instance sur chacun des gestionnaires de files d'attente clonés, IBM MQ décide à quel gestionnaire de files d'attente elle doit être envoyée. Lorsque de nombreuses applications envoient des messages à la file d'attente de cluster, IBM MQ équilibre la charge de travail entre chacun des gestionnaires de files d'attente ayant une instance de la file d'attente. Si l'un des systèmes hébergeant un gestionnaire de files d'attente cloné est défaillant, IBM MQ continue d'équilibrer la charge de travail entre les gestionnaires de files d'attente restants jusqu'à ce que le système défaillant soit redémarré.

Si vous utilisez des clusters de gestionnaires de files d'attente, vous devez prendre en compte les problèmes de sécurité suivants:

- Autoriser uniquement les gestionnaires de files d'attente sélectionnés à envoyer des messages à votre gestionnaire de files d'attente
- Autoriser uniquement les utilisateurs sélectionnés d'un gestionnaire de files d'attente éloignées à envoyer des messages à une file d'attente de votre gestionnaire de files d'attente
- Autoriser les applications connectées à votre gestionnaire de files d'attente à envoyer des messages uniquement aux files d'attente éloignées sélectionnées

Ces considérations sont pertinentes même si vous n'utilisez pas de clusters, mais elles deviennent plus importantes si vous utilisez des clusters.

Si une application peut envoyer des messages à une file d'attente de cluster, elle peut envoyer des messages à n'importe quelle autre file d'attente de cluster sans avoir besoin de définitions de file d'attente éloignée, de files d'attente de transmission ou de canaux supplémentaires. Il est donc plus important de déterminer si vous devez restreindre l'accès aux files d'attente de cluster sur votre gestionnaire de files d'attente et de limiter les files d'attente de cluster auxquelles vos applications peuvent envoyer des messages.

Des considérations de sécurité supplémentaires s'appliquent uniquement si vous utilisez des clusters de gestionnaires de files d'attente:

- Autoriser uniquement les gestionnaires de files d'attente sélectionnés à rejoindre un cluster
- Forcer les gestionnaires de files d'attente indésirables à quitter un cluster

Pour plus d'informations sur toutes ces considérations, voir Maintien de la sécurité des clusters.

z/OS Pour des considérations spécifiques à IBM MQ for z/OS, voir «Security in queue manager clusters on z/OS», à la page 270.

Tâches associées

«Empêcher les gestionnaires de files d'attente de recevoir des messages», à la page 496
Vous pouvez empêcher un gestionnaire de files d'attente de cluster de recevoir des messages qu'il n'est pas autorisé à recevoir à l'aide de programmes d'exit.

Sécurité pour la publication / abonnement IBM MQ

Des considérations de sécurité supplémentaires sont à prendre en compte si vous utilisez la fonction de publication / abonnement IBM MQ .

Dans un système de publication / abonnement, il existe deux types d'application: le diffuseur de publications et l'abonné. Les *diffuseurs de publications* fournissent des informations sous la forme de messages IBM MQ . Lorsqu'un diffuseur de publications publie un message, il spécifie une *rubrique*qui identifie l'objet des informations contenues dans le message.

Les *abonnés* sont les consommateurs des informations qui sont publiées. Un abonné spécifie les rubriques qui l'intéressent en s'y abonnant.

Le *gestionnaire de files d'attente* est une application fournie avec IBM MQ Publish / Subscribe. Il reçoit les messages publiés des diffuseurs de publications et les demandes d'abonnement des abonnés, et achemine les messages publiés vers les abonnés. Un abonné reçoit des messages uniquement sur les sujets auxquels il s'est abonné.

Pour plus d'informations, voir Sécurité de publication / abonnement.

Sécurité de multidiffusion

Utilisez ces informations pour comprendre pourquoi des processus de sécurité peuvent être nécessaires avec IBM MQ Multicast.

IBM MQ Multicast ne dispose pas de sécurité intégrée. Les contrôles de sécurité sont gérés dans le gestionnaire de files d'attente au moment de l'opération MQOPEN et le paramètre de zone MQMD est géré par le client. Certaines applications du réseau peuvent ne pas être des applications IBM MQ (par exemple, les applications LLM, voir Interopérabilité multidiffusion avec IBM MQ Low Latency Messaging pour plus d'informations). Par conséquent, vous devrez peut-être implémenter vos propres procédures de sécurité car les applications de réception ne peuvent pas être certaines de la validité des zones de contexte.

Il existe trois processus de sécurité à prendre en compte:

Contrôle d'accès

Le contrôle d'accès dans IBM MQ est basé sur les ID utilisateur. Pour plus d'informations sur ce sujet, voir «Contrôle d'accès pour les clients», à la page 109.

Sécurité des réseaux

Un réseau isolé peut être une option de sécurité viable pour empêcher les faux messages. Il est possible qu'une application de l'adresse de groupe de multidiffusion publie des messages malveillants à l'aide de fonctions de communication natives, qui ne peuvent pas être distinguées des messages MQ car elles proviennent d'une application de la même adresse de groupe de multidiffusion.

Il est également possible qu'un client sur l'adresse de groupe de multidiffusion reçoive des messages destinés à d'autres clients sur la même adresse de groupe de multidiffusion.

L'isolement du réseau de multidiffusion garantit que seuls les clients et les applications valides y ont accès. Cette précaution de sécurité peut empêcher l'entrée de messages malveillants et la sortie d'informations confidentielles.

Pour plus d'informations sur les adresses réseau de groupe de multidiffusion, voir: Définition du réseau approprié pour le trafic de multidiffusion

Signatures numériques

Une signature numérique est formée par le chiffrement d'une représentation d'un message. Le chiffrement utilise la clé privée du signataire et, pour des raisons d'efficacité, opère généralement sur un résumé de message plutôt que sur le message lui-même. La signature numérique d'un message avant une opération MQPUT est une bonne précaution de sécurité, mais ce processus peut avoir un impact négatif sur les performances s'il existe un volume important de messages.

Les signatures numériques varient en fonction des données en cours de signature. Si deux messages différents sont signés numériquement par la même entité, les deux signatures diffèrent, mais les deux signatures peuvent être vérifiées avec la même clé publique, c'est-à-dire la clé publique de l'entité qui a signé les messages.

Comme indiqué précédemment dans cette section, il est possible qu'une application sur l'adresse de groupe de multidiffusion publie des messages malveillants à l'aide de fonctions de communication natives, qui ne peuvent pas être distinguées des messages MQ . Les signatures numériques fournissent une preuve de l'origine, et seul l'expéditeur connaît la clé privée, ce qui fournit des preuves solides que l'expéditeur est l'émetteur du message.

Pour plus d'informations sur ce sujet, voir «Concepts cryptographiques», à la page 11.

Pare-feux et IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru peut simplifier la communication via un pare-feu.

MQIPT permet à deux gestionnaires de files d'attente d'échanger des messages ou à une application client IBM MQ de se connecter à un gestionnaire de files d'attente, sans nécessiter de connexion TCP/IP directe. Cette architecture est utile si un pare-feu interdit une connexion TCP/IP directe entre deux systèmes. L'utilisation de MQIPT en tant que proxy peut rendre le passage des données de canal IBM MQ via un pare-feu plus simple et plus facile à gérer. MQIPT peut également protéger les données IBM MQ qui sont envoyées via Internet à l'aide du protocole TLS (Transport Layer Security) et les données IBM MQ de tunnel dans HTTP.

Pour plus d'informations, voir IBM MQ Internet Pass-Thru.

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 196.

Refer to other sections for details, in particular “Profiles used to control access to IBM MQ resources” on page 206.

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 201.
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 262.
   - **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 262.
   - **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 “Enregistrements d'authentification de canal” on page 54.
    - 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 “Protocoles de sécurité TLS dans IBM MQ” on page 25.
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 270

Configuration de la sécurité

Cette collection de rubriques contient des informations spécifiques aux différents systèmes d'exploitation et à l'utilisation des clients.

ALW Configuration de la sécurité sous AIX, Linux, and Windows

Considérations de sécurité spécifiques aux systèmes AIX, Linux, and Windows .

Les gestionnaires de files d'attente IBM MQ transfèrent des informations potentiellement utiles. Vous devez donc utiliser un système de droits d'accès pour vous assurer que les utilisateurs non autorisés ne peuvent pas accéder à vos gestionnaires de files d'attente. Prenez en compte les types de contrôle de sécurité suivants:

Qui peut administrer IBM MQ
Vous pouvez définir l'ensemble des utilisateurs qui peuvent émettre des commandes pour administrer IBM MQ.

Qui peut utiliser les objets IBM MQ
Vous pouvez définir les utilisateurs (généralement des applications) qui peuvent utiliser des appels MQI et des commandes PCF pour effectuer les opérations suivantes:

- Qui peut se connecter à un gestionnaire de files d'attente.
- Qui peut accéder aux objets (files d'attente, définitions de processus, listes de noms, canaux, canaux de connexion client, programmes d'écoute, services et objets d'informations d'authentification) et quel type d'accès ils ont à ces objets.
- Qui peut accéder aux messages IBM MQ .
- Qui peut accéder aux informations de contexte associées à un message.

Sécurité des canaux
Vous devez vous assurer que les canaux utilisés pour envoyer des messages aux systèmes distants peuvent accéder aux ressources requises.

Vous pouvez utiliser les fonctions d'exploitation standard pour accorder l'accès aux bibliothèques de programmes, aux bibliothèques de liens MQI et aux commandes. Toutefois, le répertoire contenant les files d'attente et les autres données du gestionnaire de files d'attente est privé pour IBM MQ; n'utilisez pas les commandes du système d'exploitation standard pour accorder ou révoquer des autorisations sur les ressources MQI.

ALW Fonctionnement des autorisations sur AIX, Linux, and Windows

Les tables de spécification d'autorisation dans les rubriques de cette section définissent précisément le fonctionnement des autorisations et les restrictions qui s'appliquent.

Les tableaux s'appliquent aux situations suivantes:

- Applications qui émettent des appels MQI
- Programmes d'administration qui émettent des commandes MQSC sous forme de fichiers PCF d'échappement

- Programmes d'administration qui émettent des commandes PCF

Dans cette section, les informations sont présentées sous la forme d'un ensemble de tables qui spécifient les éléments suivants:

Action à exécuter
Option MQI, commande MQSC ou commande PCF.

Objet de contrôle d'accès
File d'attente, processus, gestionnaire de files d'attente, liste de noms, informations d'authentification, canal, canal de connexion client, programme d'écoute ou service.

Autorisation requise
Exprimée sous la forme d'une constante MQZAO_.

Dans les tableaux, les constantes préfixées par MQZAO_ correspondent aux mots clés de la liste d'autorisation de la commande `setmqaut` pour l'entité particulière. Par exemple, MQZAO_BROWSE correspond au mot clé `+browse`, MQZAO_SET_ALL_CONTEXT correspond au mot clé `+setall`, etc. Ces constantes sont définies dans le fichier d'en-tête cmqzc.h, fourni avec le produit.

ALW *Autorisations pour les appels MQI*

MQCONN, **MQOPEN**, **MQPUT1**et **MQCLOSE** peuvent nécessiter des vérifications d'autorisation. Les tableaux de cette rubrique récapitulent les autorisations requises pour chaque appel.

Une application est autorisée à émettre des appels et des options MQI spécifiques uniquement si l'identificateur utilisateur sous lequel elle s'exécute (ou dont elle peut assumer les autorisations) a reçu l'autorisation appropriée.

Quatre appels MQI peuvent nécessiter des vérifications d'autorisation: **MQCONN**, **MQOPEN**, **MQPUT1**et **MQCLOSE**.

Pour **MQOPEN** et **MQPUT1**, la vérification des droits d'accès est effectuée sur le nom de l'objet en cours d'ouverture et non sur le ou les noms, ce qui se produit après la résolution d'un nom. Par exemple, une application peut être autorisée à ouvrir une file d'attente alias sans avoir le droit d'ouvrir la file d'attente de base dans laquelle l'alias est résolu. La règle est que la vérification est effectuée sur la première définition rencontrée lors du processus de résolution d'un nom qui n'est pas un alias de gestionnaire de files d'attente, sauf si la définition d'alias de gestionnaire de files d'attente est ouverte directement ; c'est-à-dire que son nom est affiché dans la zone *ObjectName* du descripteur d'objet. Des droits sont toujours nécessaires pour l'objet en cours d'ouverture. Dans certains cas, des droits d'accès supplémentaires indépendants de la file d'attente, obtenus via une autorisation pour l'objet gestionnaire de files d'attente, sont requis.

Tableau 10, à la page 141, Tableau 11, à la page 141, Tableau 12, à la page 141et Tableau 13, à la page 142 récapitulent les autorisations requises pour chaque appel. Dans les tableaux *Non applicable* , le contrôle d'autorisation n'est pas pertinent pour cette opération ; *Pas de contrôle* signifie qu'aucun contrôle d'autorisation n'est effectué.

Remarque : Vous ne trouverez aucune mention des listes de noms, des canaux, des canaux de connexion client, des programmes d'écoute, des services ou des objets d'informations d'authentification dans ces tables. En effet, aucune des autorisations ne s'applique à ces objets, à l'exception de MQOO_INQUIRE, pour lequel les mêmes autorisations s'appliquent que pour les autres objets.

L'autorisation spéciale MQZAO_ALL_MQI inclut toutes les autorisations dans les tables qui sont pertinentes pour le type d'objet, à l'exception de MQZAO_DELETE et MQZAO_DISPLAY, qui sont classées comme autorisations d'administration.

Pour modifier l'une des options de contexte de message, vous devez disposer des autorisations appropriées pour émettre l'appel. Par exemple, pour utiliser MQOO_SET_IDENTITY_CONTEXT ou MQPMO_SET_IDENTITY_CONTEXT, vous devez disposer du droit `+setid` .

Tableau 10. Autorisation de sécurité requise pour les appels MQCONN			
Autorisation requise pour:	**Objet de file d'attente («1», à la page 142)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQCONN	Non applicable	Non applicable	MQZAO_CONNECT

Tableau 11. Autorisation de sécurité requise pour les appels MQOPEN			
Autorisation requise pour:	**Objet de file d'attente («1», à la page 142)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQOO_INTERROGATION	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_BROWSE	MQZAO_PARCOURIR	Non applicable	Aucune vérification
MQOO_ENTRÉE_ *	MQZAO_ENTREE	Non applicable	Aucune vérification
MQOO_SAVE_ ALL_CONTEXT **(«2», à la page 142)**	MQZAO_ENTREE	Non applicable	Non applicable
MQOO_OUTPUT (file d'attente normale) **(«3», à la page 142)**	MQZAO_OUTPUT	Non applicable	Non applicable
MQOO_PASS_IDENTITY_CONTEXT **(«4», à la page 142)**	MQZAO_PASS_ IDENTITY_CONTEXT	Non applicable	Aucune vérification
MQOO_PASS_ALL_ CONTEXT **(«4», à la page 142, «5», à la page 142)**	MQZAO_PASS _ALL_CONTEXT	Non applicable	Aucune vérification
MQOO_SET_IDENTITY_CONTEXT **(«4», à la page 142, «5», à la page 142)**	MQZAO_SET_IDENTITY_CONTEXT	Non applicable	MQZAO_SET_ IDENTITY_CONTEXT **(«6», à la page 142)**
MQOO_SET_ ALL_CONTEXT **(«4», à la page 142, «7», à la page 142)**	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ ALL_CONTEXT **(«6», à la page 142)**
MQOO_OUTPUT (file d'attente de transmission) **(«8», à la page 142)**	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ ALL_CONTEXT **(«6», à la page 142)**
MQOO_SET	MQZAO_SET	Non applicable	Aucune vérification
MQOO_ALTERNATE_ AUTORITE UTILISATEUR	**(«9», à la page 143)**	**(«9», à la page 143)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(«9», à la page 143, «10», à la page 143)**

Tableau 12. Autorisation de sécurité requise pour les appels MQPUT1			
Autorisation requise pour:	**Objet de file d'attente («1», à la page 142)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(«11», à la page 143)**	Non applicable	Aucune vérification

Tableau 12. Autorisation de sécurité requise pour les appels MQPUT1 (suite)			
Autorisation requise pour:	**Objet de file d'attente («1», à la page 142)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(«11», à la page 143)**	Non applicable	Aucune vérification
MQPMO_SET_IDENTITY_C ONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(«11», à la page 143)**	Non applicable	MQZAO_SET_ IDENTITY_CONTEXT **(«6», à la page 142)**
MQPMO_SET_ALL_CONTE XT	MQZAO_SET_ ALL_CONTEXT **(«11», à la page 143)**	Non applicable	MQZAO_SET_ ALL_CONTEXT **(«6», à la page 142)**
(File d'attente de transmission) **(«8», à la page 142)**	MQZAO_SET_ALL_CONTE XT	Non applicable	MQZAO_SET_ ALL_CONTEXT **(«6», à la page 142)**
MQPMO_ALTERNATE_USE R_AUTHORITY	**(«12», à la page 143)**	Non applicable	MQZAO_ALTERNATE_ USER_AUTHORITY **(«10», à la page 143)**

Tableau 13. Autorisation de sécurité requise pour les appels MQCLOSE			
Autorisation requise pour:	**Objet de file d'attente («1», à la page 142)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQCO_DELETE	MQZAO_DELETE **(«13», à la page 143)**	Non applicable	Non applicable
MQCO_DELETE _PURGE	MQZAO_DELETE **(«13», à la page 143)**	Non applicable	Non applicable

Remarques relatives aux tableaux:

1. Si vous ouvrez une file d'attente modèle:
   - Le droit MQZAO_DISPLAY est requis pour la file d'attente modèle, en plus du droit d'ouverture de la file d'attente modèle pour le type d'accès pour lequel vous l'ouvrez.
   - Les droits MQZAO_CREATE ne sont pas nécessaires pour créer la file d'attente dynamique.
   - L'ID utilisateur utilisé pour ouvrir la file d'attente modèle reçoit automatiquement tous les droits spécifiques à la file d'attente (équivalents à MQZAO_ALL) pour la file d'attente dynamique créée.
2. MQOO_INPUT_ * doit également être spécifié. Valide pour une file d'attente locale, modèle ou alias.
3. Cette vérification est effectuée pour tous les cas de sortie, à l'exception des files d'attente de transmission (voir la remarque «8», à la page 142).
4. MQOO_OUTPUT doit également être spécifié.
5. MQOO_PASS_IDENTITY_CONTEXT est également impliqué par cette option.
6. Ce droit est requis pour l'objet gestionnaire de files d'attente et la file d'attente particulière.
7. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT et MQOO_SET_IDENTITY_CONTEXT sont également impliquées par cette option.
8. Cette vérification est effectuée pour une file d'attente locale ou modèle dont l'attribut de file d'attente *Utilisation* est MQUS_TRANSMISSION et qui est ouverte directement pour la sortie. Elle ne s'applique pas si une file d'attente éloignée est ouverte (soit en spécifiant les noms du gestionnaire de files d'attente éloignées et de la file d'attente éloignée, soit en indiquant le nom d'une définition locale de la file d'attente éloignée).

9. Au moins l'une des options MQOO_INQUIRE (pour tout type d'objet) ou MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT ou MQOO_SET (pour les files d'attente) doit également être spécifiée. La vérification effectuée est la même que pour les autres options spécifiées, à l'aide de l'identificateur d'utilisateur de remplacement fourni pour les droits sur les objets nommés spécifiques, et des droits d'application en cours pour la vérification MQZAO_ALTERNATE_USER_IDENTIFIER.
10. Cette autorisation permet de spécifier tout ID *AlternateUser* .
11. Une vérification MQZAO_OUTPUT est également effectuée si la file d'attente ne possède pas l'attribut de file d'attente *Usage* MQUS_TRANSMISSION.
12. La vérification effectuée est la même que pour les autres options spécifiées, à l'aide de l'identificateur d'utilisateur de remplacement fourni pour le droit de file d'attente nommé spécifique et du droit d'application en cours pour la vérification MQZAO_ALTERNATE_USER_IDENTIFIER.
13. La vérification est effectuée uniquement si les deux affirmations suivantes sont vraies:
    - Une file d'attente dynamique permanente est en cours de fermeture et de suppression.
    - La file d'attente n'a pas été créée par l'appel MQOPEN qui a renvoyé le descripteur d'objet utilisé.

    Sinon, il n'y a pas de contrôle.

ALW *Autorisations pour les commandes MQSC dans les fichiers PCF d'échappement*

Ces informations récapitulent les autorisations requises pour chaque commande MQSC contenue dans Escape PCF.

Non applicable signifie que cette opération n'est pas pertinente pour ce type d'objet.

L'ID utilisateur sous lequel le programme qui soumet la commande s'exécute doit également disposer des droits suivants:

- Droits MQZAO_CONNECT sur le gestionnaire de files d'attente
- Droit MQZAO_DISPLAY sur le gestionnaire de files d'attente afin d'exécuter des commandes PCF
- Droit d'émettre la commande MQSC dans le texte de la commande Escape PCF

ALTER *objet*

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE
Information de communication	MODIFICATION MQZAO_DE

CLEAR *objet*

Objet	Autorisation requise
File d'attente	MQZAO_CLEAR
Topic	MQZAO_CLEAR
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	Non applicable
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable
Information de communication	Non applicable

DEFINE *objet* NOREPLACE («1», à la page 148)

Objet	Autorisation requise
File d'attente	MQZAO_CREATE **(«2», à la page 148)**
Topic	MQZAO_CREATE **(«2», à la page 148)**
Processus	MQZAO_CREATE **(«2», à la page 148)**
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_CREATE **(«2», à la page 148)**
Informations d'authentification	MQZAO_CREATE **(«2», à la page 148)**
Canal	MQZAO_CREATE **(«2», à la page 148)**
Canal de connexion client	MQZAO_CREATE **(«2», à la page 148)**
Programme d'écoute	MQZAO_CREATE **(«2», à la page 148)**
Service	MQZAO_CREATE **(«2», à la page 148)**
Information de communication	MQZAO_CREATE **(«2», à la page 148)**

DEFINE *objet* REPLACE («1», à la page 148, «3», à la page 148)

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE

Objet	Autorisation requise
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE
Information de communication	MODIFICATION MQZAO_DE

DELETE *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DELETE
Topic	MQZAO_DELETE
Processus	MQZAO_DELETE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_DELETE
Informations d'authentification	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de connexion client	MQZAO_DELETE
Programme d'écoute	MQZAO_DELETE
Service	MQZAO_DELETE
Information de communication	MQZAO_DELETE

DISPLAY *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DISPLAY
Topic	MQZAO_DISPLAY
Processus	MQZAO_DISPLAY
Gestionnaire de files d'attente	MQZAO_DISPLAY
Liste de noms	MQZAO_DISPLAY
Informations d'authentification	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de connexion client	MQZAO_DISPLAY
Programme d'écoute	MQZAO_DISPLAY
Service	MQZAO_DISPLAY
Information de communication	MQZAO_DISPLAY

START *objet*

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable

Objet	Autorisation requise
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	CONTROLE MQZ
Service	CONTROLE MQZ
Information de communication	Non applicable

STOP *objet*

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	CONTROLE MQZ
Service	CONTROLE MQZ
Information de communication	Non applicable

Commandes relatives aux canaux

Commande	Objet	Autorisation requise
Envoyer une commande PING à un canal	Canal	CONTROLE MQZ
Réinitialisation du canal	Canal	MQZAO_CONTRÔLE_ÉTENDU
Résolution du canal	Canal	MQZAO_CONTRÔLE_ÉTENDU

Commandes d'abonnement

Commande	Objet	Autorisation requise
ALTER SUB	Topic	CONTROLE MQZ
DEFINE SUB	Topic	CONTROLE MQZ
SUPPRIMER DES SOUS	Topic	CONTROLE MQZ
DISPLAY SUB	Topic	MQZAO_DISPLAY

Commandes de sécurité

Commande	Objet	Autorisation requise
SET AUTHREC	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
DELETE AUTHREC	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Paramètre DISPLAY AUTHREC	Gestionnaire de files d'attente	MQZAO_DISPLAY
DISPLAY AUTHSERV	Gestionnaire de files d'attente	MQZAO_DISPLAY
AFFICHER ENTAUTH	Gestionnaire de files d'attente	MQZAO_DISPLAY
SET CHLAUTH	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
AFFICHER CHLAUTH	Gestionnaire de files d'attente	MQZAO_DISPLAY
REFRESH SECURITY	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE

Affichages de statut

Commande	Objet	Autorisation requise
DISPLAY CHSTATUS	Gestionnaire de files d'attente	MQZAO_DISPLAY Notez que les droits `+inq` (ou de manière équivalente MQZAO_INQUIRE) sont requis sur la file d'attente de transmission si le type de canal est CLUSSDR.
DISPLAY LSSTATUS	Gestionnaire de files d'attente	MQZAO_DISPLAY
AFFICHAGE DE PUBSUB	Gestionnaire de files d'attente	MQZAO_DISPLAY
STATUT DU JEU DE CARACTÈRES D'AFFICHAGE	Gestionnaire de files d'attente	MQZAO_DISPLAY
STATUT DE L'AFFICHAGE	Gestionnaire de files d'attente	MQZAO_DISPLAY
DISPLAY TPSTATUS	Gestionnaire de files d'attente	MQZAO_DISPLAY

Commandes relatives aux clusters

Commande	Objet	Autorisation requise
DISPLAY CLUSQMGR	Gestionnaire de files d'attente	MQZAO_DISPLAY
Actualiser le cluster	Appartenance au groupe'mqm'requise	
Réinitialisation d'un cluster	Appartenance au groupe'mqm'requise	
SUSPEND QMGR	Appartenance au groupe'mqm'requise	
RESUME QMGR	Appartenance au groupe'mqm'requise	

Autres commandes d'administration

Commande	Objet	Autorisation requise
PING QMGR	Gestionnaire de files d'attente	MQZAO_DISPLAY
ACTUALISEZ LE GESTIONNAIRE DE FILES D'ATTENTE	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE

Commande	Objet	Autorisation requise
RESET QMGR	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
DISPLAY CONN	Gestionnaire de files d'attente	MQZAO_DISPLAY
ARRETER CONN	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE

Remarque :

1. Pour les commandes DEFINE, le droit MQZAO_DISPLAY est également requis pour l'objet LIKE si un tel droit est spécifié, ou sur le système SYSTEM.DEFAULT.xxx si LIKE est omis.
2. Les droits MQZAO_CREATE ne sont pas spécifiques à un objet ou à un type d'objet particulier. Le droit de création est accordé pour tous les objets d'un gestionnaire de files d'attente spécifié, en spécifiant un type d'objet QMGR dans la commande `setmqaut` .
3. Ceci s'applique si l'objet à remplacer existe déjà. Si tel n'est pas le cas, la vérification est celle de l'objet DEFINE NOREPLACE.

Information associée

Mise en cluster : meilleures pratiques d'utilisation REFRESH CLUSTER

ALW *Autorisations pour les commandes PCF*

Cette section récapitule les autorisations requises pour chaque commande PCF.

Aucune vérification signifie qu'aucune vérification d'autorisation n'est effectuée ; *Non applicable* signifie que cette opération n'est pas pertinente pour ce type d'objet.

L'ID utilisateur sous lequel le programme qui soumet la commande s'exécute doit également disposer des droits suivants:

- Droits MQZAO_CONNECT sur le gestionnaire de files d'attente
- Droit MQZAO_DISPLAY sur le gestionnaire de files d'attente afin d'exécuter des commandes PCF

L'autorisation spéciale MQZAO_ALL_ADMIN inclut toutes les autorisations de la liste suivante qui sont pertinentes pour le type d'objet, à l'exception de MQZAO_CREATE, qui n'est pas spécifique à un objet ou à un type d'objet particulier.

Modifier *objet*

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Rubrique	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE
Information de communication	MODIFICATION MQZAO_DE

Effacer *objet*

Objet	Autorisation requise
File d'attente	MQZAO_CLEAR
Rubrique	MQZAO_CLEAR
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	Non applicable
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable
Information de communication	Non applicable

Copier *objet* (sans remplacement) (1)

Objet	Autorisation requise
File d'attente	MQZAO_CREATE **(2)**
Rubrique	MQZAO_CREATE **(2)**
Processus	MQZAO_CREATE **(2)**
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_CREATE **(2)**
Informations d'authentification	MQZAO_CREATE **(2)**
Canal	MQZAO_CREATE **(2)**
Canal de connexion client	MQZAO_CREATE **(2)**
Programme d'écoute	MQZAO_CREATE **(2)**
Service	MQZAO_CREATE **(2)**
Information de communication	MQZAO_CREATE **(«2», à la page 154)**

Copiez l'objet (avec remplacement) (1, 4)

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Rubrique	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE

Objet	Autorisation requise
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE
Information de communication	MODIFICATION MQZAO_DE

Créer un *objet* (sans remplacement) (3)

Objet	Autorisation requise
File d'attente	MQZAO_CREATE **(2)**
Rubrique	MQZAO_CREATE **(2)**
Processus	MQZAO_CREATE **(2)**
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_CREATE **(2)**
Informations d'authentification	MQZAO_CREATE **(2)**
Canal	MQZAO_CREATE **(2)**
Canal de connexion client	MQZAO_CREATE **(2)**
Programme d'écoute	MQZAO_CREATE **(2)**
Service	MQZAO_CREATE **(2)**
Information de communication	MQZAO_CREATE **(2)**

Créer *objet* (avec remplacement) (3, 4)

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Rubrique	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE
Information de communication	MODIFICATION MQZAO_DE

Supprimer *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DELETE
Rubrique	MQZAO_DELETE

Objet	Autorisation requise
Processus	MQZAO_DELETE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_DELETE
Informations d'authentification	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de connexion client	MQZAO_DELETE
Programme d'écoute	MQZAO_DELETE
Service	MQZAO_DELETE
Information de communication	MQZAO_DELETE

Interroger *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DISPLAY
Rubrique	MQZAO_DISPLAY
Processus	MQZAO_DISPLAY
Gestionnaire de files d'attente	MQZAO_DISPLAY
Liste de noms	MQZAO_DISPLAY
Informations d'authentification	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de connexion client	MQZAO_DISPLAY
Programme d'écoute	MQZAO_DISPLAY
Service	MQZAO_DISPLAY
Information de communication	MQZAO_DISPLAY

Consulter les noms d' *objet*

Objet	Autorisation requise
File d'attente	Aucune vérification
Topic	Aucune vérification
Processus	Aucune vérification
Gestionnaire de files d'attente	Aucune vérification
Liste de noms	Aucune vérification
Informations d'authentification	Aucune vérification
Canal	Aucune vérification
Canal de connexion client	Aucune vérification
Programme d'écoute	Aucune vérification
Service	Aucune vérification

Objet	Autorisation requise
Information de communication	Aucune vérification

Démarrez *objet*

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	CONTROLE MQZ
Service	CONTROLE MQZ
Information de communication	Non applicable

Arrêter *objet*

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	CONTROLE MQZ
Service	CONTROLE MQZ
Information de communication	Non applicable

Commandes relatives aux canaux

Commande	Objet	Autorisation requise
Envoyer une commande PING à un canal	Canal	CONTROLE MQZ
Réinitialisation du canal	Canal	MQZAO_CONTRÔLE_ÉTENDU
Résolution du canal	Canal	MQZAO_CONTRÔLE_ÉTENDU

Commandes d'abonnement

Commande	Objet	Autorisation requise
Modifier un abonnement	Topic	CONTROLE MQZ
Créer un abonnement	Topic	CONTROLE MQZ
Supprimer l'abonnement	Topic	CONTROLE MQZ
Consulter un abonnement	Topic	MQZAO_DISPLAY

Commandes de sécurité

Commande	Objet	Autorisation requise
Définition de l'enregistrement de droits d'accès	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Supprimer l'enregistrement de droits d'accès	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Consulter des enregistrements de droits	Gestionnaire de files d'attente	MQZAO_DISPLAY
Consulter un service de droits d'accès	Gestionnaire de files d'attente	MQZAO_DISPLAY
Interroger les droits d'accès de l'entité	Gestionnaire de files d'attente	MQZAO_DISPLAY
Définir l'enregistrement d'authentification de canal	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Consulter les enregistrements d'authentification de canal	Gestionnaire de files d'attente	MQZAO_DISPLAY
Régénérer la sécurité	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE

Affichages de statut

Commande	Objet	Autorisation requise
Consulter le statut d'un canal	Gestionnaire de files d'attente	MQZAO_DISPLAY Notez que les droits `+inq` (ou de manière équivalente MQZAO_INQUIRE) sont requis sur la file d'attente de transmission si le type de canal est CLUSSDR.
Interroger le statut du programme d'écoute de canal	Gestionnaire de files d'attente	MQZAO_DISPLAY
Interroger le statut de publication / d'abonnement	Gestionnaire de files d'attente	MQZAO_DISPLAY
Interroger le statut de l'abonnement	Gestionnaire de files d'attente	MQZAO_DISPLAY
Consulter le statut d'un service	Gestionnaire de files d'attente	MQZAO_DISPLAY
Consulter le statut d'une rubrique	Gestionnaire de files d'attente	MQZAO_DISPLAY

Commandes relatives aux clusters

Commande	Objet	Autorisation requise
Consulter un gestionnaire de files d'attente de cluster	Gestionnaire de files d'attente	MQZAO_DISPLAY
Régénérer un cluster	Appartenance au groupe'mqm'requise	Appartenance au groupe'mqm'requise
Réinitialisation d'un cluster	Appartenance au groupe'mqm'requise	Appartenance au groupe'mqm'requise
Interrompre un cluster de gestionnaire de files d'attente	Appartenance au groupe'mqm'requise	Appartenance au groupe'mqm'requise
Reprendre un cluster de gestionnaire de files d'attente	Appartenance au groupe'mqm'requise	Appartenance au groupe'mqm'requise

Autres commandes d'administration

Commande	Objet	Autorisation requise
Envoyer une commande Ping à un gestionnaire de files d'attente	Gestionnaire de files d'attente	MQZAO_DISPLAY
Régénérer un gestionnaire de files d'attente	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Réinitialiser un gestionnaire de files d'attente	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Réinitialiser les statistiques de file d'attente	File d'attente	MQZAO_DISPLAY et MQZAO_CHANGE
Consulter une connexion	Gestionnaire de files d'attente	MQZAO_DISPLAY
Arrêter une connexion	Gestionnaire de files d'attente	MODIFICATION MQZAO_DE

Remarque :

1. Pour les commandes de copie, le droit MQZAO_DISPLAY est également requis pour l'objet From.
2. Les droits MQZAO_CREATE ne sont pas spécifiques à un objet ou à un type d'objet particulier. Le droit de création est accordé pour tous les objets d'un gestionnaire de files d'attente spécifié, en spécifiant un type d'objet QMGR dans la commande `setmqaut` .
3. Pour les commandes de création, les droits MQZAO_DISPLAY sont également requis pour le système SYSTEM.DEFAULT.* .
4. Ceci s'applique si l'objet à remplacer existe déjà. Si ce n'est pas le cas, la vérification est la même que pour la copie ou la création sans remplacement.

AIX Création et gestion de groupes sur AIX

Sous AIX, si vous n'utilisez pas NIS ou NIS +, utilisez SMITTY pour gérer les groupes.

Pourquoi et quand exécuter cette tâche

Sous AIX, vous pouvez utiliser SMITTY pour créer un groupe, ajouter un utilisateur à un groupe, afficher la liste des utilisateurs du groupe et supprimer un utilisateur d'un groupe.

Procédure

1. Dans SMITTY, sélectionnez **Sécurité et utilisateurs** et appuyez sur Entrée.
2. Sélectionnez **Groupes** et appuyez sur Entrée.
3. Pour créer un groupe, procédez comme suit:
   a) Sélectionnez **Ajouter un groupe** et appuyez sur Entrée.
   b) Entrez le nom du groupe et les noms des utilisateurs que vous souhaitez ajouter au groupe, séparés par des virgules.
   c) Appuyez sur Entrée pour créer le groupe.
4. Pour ajouter un utilisateur à un groupe, procédez comme suit:
   a) Sélectionnez **Modifier / Afficher les caractéristiques des groupes** et appuyez sur Entrée.
   b) Entrez le nom du groupe pour afficher la liste des membres du groupe.
   c) Ajoutez les noms des utilisateurs que vous souhaitez ajouter au groupe, séparés par des virgules.
   d) Appuyez sur Entrée pour ajouter les noms au groupe.
5. Pour afficher les membres d'un groupe, procédez comme suit:
   a) Sélectionnez **Modifier / Afficher les caractéristiques des groupes** et appuyez sur Entrée.
   b) Entrez le nom du groupe pour afficher la liste des membres du groupe.
6. Pour supprimer un utilisateur d'un groupe, procédez comme suit:
   a) Sélectionnez **Modifier / Afficher les caractéristiques des groupes** et appuyez sur Entrée.
   b) Entrez le nom du groupe pour afficher la liste des membres du groupe.
   c) Supprimez le nom de l'utilisateur que vous souhaitez supprimer du groupe.
   d) Appuyez sur Entrée pour supprimer le nom du groupe.

Linux Création et gestion de groupes sur Linux

Sous Linux, si vous n'utilisez pas NIS ou NIS +, utilisez le fichier `/etc/group` pour gérer les groupes.

Pourquoi et quand exécuter cette tâche

Sous Linux, les informations de groupe sont conservées dans le fichier `/etc/group` . Vous pouvez utiliser des commandes pour créer un groupe, ajouter un utilisateur à un groupe, afficher la liste des utilisateurs du groupe et supprimer un utilisateur d'un groupe.

Procédure

1. Pour créer un groupe, utilisez la commande **groupadd** .
   Entrez la commande suivante :

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

   où *group-ID* est l'identificateur numérique du groupe et *group-name* est le nom du groupe.
2. Pour ajouter un membre à un groupe supplémentaire, utilisez la commande **usermod** pour répertorier les groupes supplémentaires dont l'utilisateur est actuellement membre, ainsi que les groupes supplémentaires dont l'utilisateur doit devenir membre.
   Par exemple, si l'utilisateur est déjà membre du groupe `groupa`et qu'il doit devenir membre de `groupb`, utilisez la commande suivante:

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

   où *user-name* est le nom d'utilisateur.
3. Pour afficher les membres d'un groupe, utilisez la commande **getent** .
   Entrez la commande suivante :

```
getent group group-name
```

où *group-name* est le nom du groupe.

4. Pour supprimer un membre d'un groupe supplémentaire, utilisez la commande **usermod** pour répertorier les groupes supplémentaires dont vous souhaitez que l'utilisateur reste membre. Par exemple, si le groupe principal de l'utilisateur est `users` et que l'utilisateur est également membre des groupes `mqm`, `groupa` et `groupb`, pour supprimer l'utilisateur du groupe `mqm` , utilisez la commande suivante:

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

où *user-name* est le nom d'utilisateur.

Windows Création et gestion de groupes sur Windows

Sous Windows, vous utilisez la fonction Gestion de l'ordinateur pour administrer des groupes sur un poste de travail ou une machine serveur membre.

Pourquoi et quand exécuter cette tâche

Pour les contrôleurs de domaine, les utilisateurs et les groupes sont administrés via Active Directory. Pour plus de détails sur l'utilisation d' Active Directory , reportez-vous aux instructions appropriées du système d'exploitation.

Les modifications apportées à l'appartenance à un groupe d'un principal ne sont pas reconnues tant que le gestionnaire de files d'attente n'est pas redémarré ou que vous n'émettez pas la commande MQSC **REFRESH SECURITY** (ou l'équivalent PCF).

Utilisez le panneau Gestion de l'ordinateur Windows pour gérer les utilisateurs et les groupes. Toute modification apportée à l'utilisateur actuellement connecté risque de ne pas être effective tant que l'utilisateur ne se reconnecte pas.

Windows *Création d'un groupe sous Windows*

Créez un groupe à l'aide du panneau de commande.

Procédure

1. Ouvrir le panneau de commande
2. Cliquez deux fois sur **Outils d'administration**.
   Le panneau Outils d'administration s'ouvre.
3. Cliquez deux fois sur **Gestion de l'ordinateur**.
   Le panneau Gestion de l'ordinateur s'ouvre.
4. Développez **Utilisateurs et groupes locaux**.
5. Cliquez avec le bouton droit de la souris sur **Groupes**et sélectionnez **Nouveau groupe ...**.
   Le panneau Nouveau groupe s'affiche.
6. Entrez un nom approprié dans la zone Nom du groupe, puis cliquez sur **Créer**.
7. Cliquez sur **Fermer**.

Windows *Ajout d'un utilisateur à un groupe sous Windows*

Ajoutez un utilisateur à un groupe à l'aide du panneau de commande.

Procédure

1. Ouvrir le panneau de commande
2. Cliquez deux fois sur **Outils d'administration**.

Le panneau Outils d'administration s'ouvre.

3. Cliquez deux fois sur **Gestion de l'ordinateur**.
   Le panneau Gestion de l'ordinateur s'ouvre.
4. Dans le panneau Gestion de l'ordinateur, développez **Utilisateurs et groupes locaux**.
5. Sélectionnez **Utilisateurs**
6. Cliquez deux fois sur l'utilisateur que vous souhaitez ajouter à un groupe.
   Le panneau des propriétés utilisateur s'affiche.
7. Sélectionnez l'onglet **Membre de** .
8. Sélectionnez le groupe auquel vous souhaitez ajouter l'utilisateur. Si le groupe de votre choix n'est pas visible:
   a) Cliquez sur **Ajouter...**.
      Le panneau Sélectionner des groupes s'affiche.
   b) Cliquez sur **Emplacements ...**.
      Le panneau Emplacements s'affiche.
   c) Sélectionnez l'emplacement du groupe auquel vous souhaitez ajouter l'utilisateur dans la liste et cliquez sur **OK**.
   d) Entrez le nom du groupe dans la zone fournie.

      Vous pouvez également cliquer sur **Avancé ...** puis **Rechercher maintenant** pour répertorier les groupes disponibles dans l'emplacement actuellement sélectionné. A partir d'ici, sélectionnez le groupe auquel vous souhaitez ajouter l'utilisateur et cliquez sur **OK**.
   e) Cliquez sur **OK**.
      Le panneau des propriétés utilisateur s'affiche avec le groupe que vous avez ajouté.
   f) Sélectionnez le groupe.
9. Cliquez sur **OK**.
   Le panneau Gestion de l'ordinateur s'affiche.

Windows *Affichage des personnes faisant partie d'un groupe sur Windows*

Affichez les membres d'un groupe à l'aide du panneau de commande.

Procédure

1. Ouvrir le panneau de commande
2. Cliquez deux fois sur **Outils d'administration**.
   Le panneau Outils d'administration s'ouvre.
3. Cliquez deux fois sur **Gestion de l'ordinateur**.
   Le panneau Gestion de l'ordinateur s'ouvre.
4. Dans le panneau Gestion de l'ordinateur, développez **Utilisateurs et groupes locaux**.
5. Sélectionnez **Groupes**.
6. Cliquez deux fois sur un groupe. Le panneau des propriétés de groupe s'affiche.
   Le panneau des propriétés de groupe s'affiche.

Résultats

Les membres du groupe s'affichent.

Windows *Suppression d'un utilisateur d'un groupe sous Windows*

Supprimez un utilisateur d'un groupe à l'aide du panneau de commande.

Procédure

1. Ouvrir le panneau de commande
2. Cliquez deux fois sur **Outils d'administration**.
   Le panneau Outils d'administration s'ouvre.
3. Cliquez deux fois sur **Gestion de l'ordinateur**.
   Le panneau Gestion de l'ordinateur s'ouvre.
4. Dans le panneau Gestion de l'ordinateur, développez **Utilisateurs et groupes locaux**.
5. Sélectionnez **Utilisateurs**.
6. Cliquez deux fois sur l'utilisateur que vous souhaitez ajouter à un groupe.
   Le panneau des propriétés utilisateur s'affiche.
7. Sélectionnez l'onglet **Membre de** .
8. Sélectionnez le groupe dont vous souhaitez supprimer l'utilisateur, puis cliquez sur **Supprimer**.
9. Cliquez sur **OK**.
   Le panneau Gestion de l'ordinateur s'affiche.

Résultats

Vous venez de supprimer l'utilisateur du groupe.

Windows Remarques spéciales relatives à la sécurité sous Windows

Certaines fonctions de sécurité se comportent différemment sur les différentes versions de Windows.

La sécurité IBM MQ s'appuie sur les appels à l'API du système d'exploitation pour obtenir des informations sur les autorisations utilisateur et les appartenances à des groupes. Certaines fonctions ne se comportent pas de manière identique sur les systèmes Windows . Cette collection de rubriques comprend des descriptions de la manière dont ces différences peuvent affecter la sécurité IBM MQ lorsque vous exécutez IBM MQ dans un environnement Windows .

Windows *Comptes utilisateur locaux et de domaine pour le service IBM MQ Windows*

Lorsqu'IBM MQ s'exécute, il doit vérifier que seuls les utilisateurs autorisés peuvent accéder aux gestionnaires de files d'attente ou aux files d'attente. Cela nécessite un compte utilisateur spécial que IBM MQ peut utiliser pour demander des informations sur tout utilisateur qui tente d'accéder à ce type d'accès.

- «Configuration de comptes utilisateur spéciaux avec Prepare IBM MQ Wizard», à la page 158
- «Utilisation de IBM MQ avec Active Directory», à la page 159
- «Droits utilisateur requis pour un service IBM MQ Windows», à la page 159

Configuration de comptes utilisateur spéciaux avec Prepare IBM MQ Wizard

Le Prepare IBM MQ Wizard crée un compte utilisateur spécial afin que le service Windows puisse être partagé par les processus qui doivent l'utiliser (voir Configuration d' IBM MQ avec le PPrepare IBM MQ Wizard).

Un service Windows est partagé entre les processus client pour une installation IBM MQ . Un service est créé pour chaque installation. Chaque service est nommé `MQ_`*`InstallationName`*et possède le nom d'affichage `IBM MQ(`*`InstallationName`*`)`.

Etant donné que chaque service doit être partagé entre des sessions de connexion non interactives et interactives, vous devez le lancer sous un compte utilisateur spécial. Vous pouvez utiliser un compte utilisateur spécial pour tous les services ou créer des comptes utilisateur spéciaux différents. Chaque compte utilisateur spécial doit avoir le droit de `se connecter en tant que service`. Pour plus d'informations, voir Tableau 14, à la page 159. Si l'ID utilisateur ne dispose pas des droits pour exécuter le service, ce dernier ne démarre pas et renvoie une erreur dans le journal des événements du système

Windows. En règle générale, vous avez exécuté le Prepare IBM MQ Wizardet configuré l'ID utilisateur correctement. Toutefois, si vous avez configuré l'ID utilisateur manuellement, il se peut que vous ayez un problème à résoudre.

Lorsque vous installez IBM MQ et que vous exécutez Prepare IBM MQ Wizard pour la première fois, il crée un compte utilisateur local pour le service appelé MUSR_MQADMIN avec les paramètres et les droits requis, y compris `Connexion en tant que service`.

Pour les installations suivantes, Prepare IBM MQ Wizard crée un compte utilisateur nommé MUSR_MQADMIN*x*, où *x* est le prochain nombre disponible représentant un ID utilisateur qui n'existe pas. Le mot de passe de MUSR_MQADMIN*x* est généré de manière aléatoire lorsque le compte est créé et utilisé pour configurer l'environnement de connexion pour le service. Le mot de passe généré n'expire pas.

Ce compte IBM MQ n'est affecté par aucune règle de compte configurée sur le système pour exiger que les mots de passe de compte soient modifiés après une certaine période.

Le mot de passe n'est pas connu en dehors de ce traitement à utilisation unique et est stocké par le système d'exploitation Windows dans une partie sécurisée du registre.

Utilisation de IBM MQ avec Active Directory

Dans certaines configurations réseau, où les comptes utilisateur sont définis sur les contrôleurs de domaine qui utilisent le service d'annuaire Active Directory , le compte utilisateur local sous lequel IBM MQ s'exécute peut ne pas disposer des droits requis pour interroger l'appartenance à un groupe d'autres comptes utilisateur de domaine. Lorsque vous installez IBM MQ, Prepare IBM MQ Wizard identifie si tel est le cas en effectuant des tests et en vous posant des questions sur la configuration réseau.

Si le compte utilisateur local sous lequel IBM MQ s'exécute ne dispose pas des droits requis, le Prepare IBM MQ Wizard vous invite à indiquer les détails du compte d'un compte utilisateur de domaine avec des droits utilisateur particuliers. Pour plus d'informations sur la création et la configuration d'un compte de domaine Windows , voir Création et configuration de comptes de domaine Windows pour IBM MQ. Pour connaître les droits utilisateur requis par le compte utilisateur de domaine, voir Tableau 14, à la page 159.

Une fois que vous avez entré des détails de compte valides pour le compte utilisateur de domaine dans le Prepare IBM MQ Wizard, l'assistant configure un service IBM MQ Windows à exécuter sous le nouveau compte. Les détails du compte sont conservés dans la partie sécurisée du registre et ne peuvent pas être lus par les utilisateurs.

Lorsque le service est en cours d'exécution, un service IBM MQ Windows est lancé et reste en cours d'exécution tant que le service est en cours d'exécution. Un administrateur IBM MQ qui se connecte au serveur après le lancement du service Windows peut utiliser IBM MQ Explorer pour administrer les gestionnaires de files d'attente sur le serveur. Le IBM MQ Explorer est ainsi connecté au processus de service Windows existant. Ces deux actions nécessitent des niveaux d'autorisation différents pour pouvoir fonctionner:

- Le processus de lancement requiert un droit de lancement.
- L'administrateur IBM MQ requiert des droits d'accès.

Droits utilisateur requis pour un service IBM MQ Windows

Le tableau suivant répertorie les droits utilisateur requis pour les comptes utilisateur locaux et de domaine sous lesquels s'exécute le service Windows pour une installation IBM MQ .

Tableau 14. Droits utilisateur requis pour un service Windows IBM MQ

Droits	**Description**
Se connecter en tant que travail par lots	Active un service IBM MQ Windows à exécuter sous ce compte utilisateur.
Ouvrir une session en tant que service	Permet aux utilisateurs de définir le service IBM MQ Windows pour se connecter à l'aide du compte configuré.

Tableau 14. Droits utilisateur requis pour un service Windows IBM MQ (suite)	
Droits	**Description**
Arrêter le système	Permet au service IBM MQ Windows de redémarrer le serveur s'il est configuré pour le faire en cas d'échec de la reprise d'un service.
Augmenter les quotas	Obligatoire pour l'appel `CreateProcessAsUser` du système d'exploitation.
Agir comme partie intégrante du système d'exploitation	Obligatoire pour l'appel `LogonUser` du système d'exploitation.
Ignorer le contrôle transversal	Obligatoire pour l'appel `LogonUser` du système d'exploitation.
Remplacer un jeton de processus	Obligatoire pour l'appel `LogonUser` du système d'exploitation.

Remarque : Les droits des programmes de débogage peuvent être nécessaires dans les environnements exécutant des applications ASP et IIS.

Votre compte utilisateur de domaine doit avoir ces droits d'utilisateur Windows définis comme droits d'utilisateur effectifs, comme indiqué dans l'application Stratégie de sécurité locale. Si ce n'est pas le cas, définissez-les à l'aide de l'application Stratégie de sécurité locale en local sur le serveur ou à l'aide du domaine d'application de sécurité du domaine.

Windows *Droits de sécurité du serveur Windows*

L'installation de IBM MQ se comporte différemment sur Windows Server, selon qu'un utilisateur local ou un utilisateur de domaine effectue l'installation.

Si un utilisateur *local* installe IBM MQ, Prepare IBM MQ Wizard détecte que l'utilisateur local créé pour le service IBM MQ Windows peut extraire les informations d'appartenance à un groupe de l'utilisateur installant. Le Prepare IBM MQ Wizard pose des questions à l'utilisateur sur la configuration réseau afin de déterminer si d'autres comptes utilisateur sont définis sur les contrôleurs de domaine s'exécutant sous Windows 2000 ou version ultérieure. Si tel est le cas, le service IBM MQ Windows doit s'exécuter sous un compte utilisateur de domaine avec des paramètres et des droits spécifiques. Le Prepare IBM MQ Wizard invite l'utilisateur à entrer les détails du compte de cet utilisateur, comme décrit dans Configuration d' IBM MQ avec Prepare IBM MQ Wizard.

Si un utilisateur *domain* installe IBM MQ, Prepare IBM MQ Wizard détecte que l'utilisateur local créé pour le service IBM MQ Windows ne peut pas extraire les informations d'appartenance à un groupe de l'utilisateur installant. Dans ce cas, Prepare IBM MQ Wizard invite toujours l'utilisateur à indiquer les détails du compte utilisateur de domaine à utiliser par le service IBM MQ Windows .

Lorsque le service IBM MQ Windows doit utiliser un compte utilisateur de domaine, IBM MQ ne peut pas fonctionner correctement tant qu'il n'a pas été configuré à l'aide de Prepare IBM MQ Wizard. Prepare IBM MQ Wizard ne permet pas à l'utilisateur de continuer avec d'autres tâches tant que le service Windows n'a pas été configuré avec un compte approprié.

Pour plus d'informations, voir Création et configuration de comptes de domaine pour IBM MQ.

Windows *Modification du nom d'utilisateur associé au service IBM MQ*

Vous pouvez modifier le nom d'utilisateur associé au service IBM MQ en créant un nouveau compte et en entrant ses détails à l'aide du Prepare IBM MQ Wizard.

Pourquoi et quand exécuter cette tâche

Lorsque vous installez IBM MQ et exécutez Prepare IBM MQ Wizard pour la première fois, il crée un compte utilisateur local pour le service appelé MUSR_MQADMIN. Pour les installations suivantes, Prepare

IBM MQ Wizard crée un compte utilisateur nommé MUSR_MQADMIN*x*, où *x* est le prochain nombre disponible représentant un ID utilisateur qui n'existe pas.

Vous devrez peut-être changer le nom d'utilisateur associé au service IBM MQ de MUSR_MQADMIN ou MUSR_MQADMIN*x* à autre chose. Par exemple, vous pouvez être amené à effectuer cette opération si votre gestionnaire de files d'attente est associé à Db2, qui n'accepte pas les noms d'utilisateur de plus de 8 caractères.

Procédure

1. Créer un nouveau compte utilisateur (par exemple, **NEW_NAME**)
2. Utilisez le Prepare IBM MQ Wizard pour entrer les détails du nouveau compte utilisateur.

Tâches associées

Configuration d' IBM MQ avec Prepare IBM MQ Wizard

Windows *Modification du mot de passe du compte utilisateur local du service IBM MQ Windows*

Vous pouvez modifier le mot de passe du compte utilisateur local du service IBM MQ Windows à l'aide du panneau Gestion de l'ordinateur.

Pourquoi et quand exécuter cette tâche

Pour modifier le mot de passe du compte utilisateur local du service IBM MQ Windows , procédez comme suit:

Procédure

1. Identifiez l'utilisateur sous lequel le service s'exécute.
2. Arrêtez le service IBM MQ à partir du panneau Gestion de l'ordinateur.
3. Modifiez le mot de passe requis de la même manière que vous modifiez le mot de passe d'une personne.
4. Accédez aux propriétés du service IBM MQ à partir du panneau Gestion de l'ordinateur.
5. Sélectionnez la page **Connexion** .
6. Vérifiez que le nom de compte spécifié correspond à l'utilisateur pour lequel le mot de passe a été modifié.
7. Entrez le mot de passe dans les zones **Mot de passe** et **Confirmer le mot de passe** , puis cliquez sur **OK**.

Windows *Modification du mot de passe d'un service IBM MQ Windows pour une installation exécutée sous un compte utilisateur de domaine*

Comme alternative à l'utilisation du Prepare IBM MQ Wizard pour entrer les détails du compte utilisateur de domaine, vous pouvez utiliser le panneau Gestion de l'ordinateur pour modifier les détails de la **connexion** pour le service IBM MQ spécifique à l'installation.

Pourquoi et quand exécuter cette tâche

Si le service IBM MQ Windows d'une installation s'exécute sous un compte utilisateur de domaine, vous pouvez modifier le mot de passe du compte comme suit:

Procédure

1. Modifiez le mot de passe du compte de domaine sur le contrôleur de domaine. Vous devrez peut-être demander à votre administrateur de domaine de le faire pour vous.
2. Procédez comme suit pour modifier la page **Connexion** du service IBM MQ .
   a) Identifiez l'utilisateur sous lequel le service s'exécute.
   b) Arrêtez le service IBM MQ à partir du panneau Gestion de l'ordinateur.

c) Modifiez le mot de passe requis de la même manière que vous modifiez le mot de passe d'une personne.

d) Accédez aux propriétés du service IBM MQ à partir du panneau Gestion de l'ordinateur.

e) Sélectionnez la page **Connexion** .

f) Vérifiez que le nom de compte spécifié correspond à l'utilisateur pour lequel le mot de passe a été modifié.

g) Entrez le mot de passe dans les zones **Mot de passe** et **Confirmer le mot de passe** , puis cliquez sur **OK**.

Le compte utilisateur sous lequel le service IBM MQ Windows s'exécute exécute exécute toutes les commandes MQSC qui sont émises par les applications de l'interface utilisateur ou qui sont exécutées automatiquement lors du démarrage, de l'arrêt ou de la reprise du service du système. Ce compte utilisateur doit donc disposer des droits d'administration IBM MQ . Par défaut, il est ajouté au groupe mqm local sur le serveur. Si cette appartenance est supprimée, le service IBM MQ Windows ne fonctionne pas. Pour plus d'informations sur les droits utilisateur, voir «Droits utilisateur requis pour un service IBM MQ Windows», à la page 159.

Si un problème de sécurité survient avec le compte utilisateur sous lequel le service IBM MQ Windows s'exécute, des messages d'erreur et des descriptions apparaissent dans le journal des événements système.

Tâches associées

Configuration d' IBM MQ avec Prepare IBM MQ Wizard

Windows *Remarques à prendre en compte lors de la promotion de serveurs Windows vers des contrôleurs de domaine*

Lors de la promotion d'un serveur Windows sur un contrôleur de domaine, vous devez déterminer si le paramètre de sécurité relatif aux droits d'accès des utilisateurs et des groupes est approprié. Lorsque vous modifiez l'état d'une machine Windows entre le serveur et le contrôleur de domaine, vous devez tenir compte du fait que cela peut affecter le fonctionnement de IBM MQ car IBM MQ utilise un groupe mqm défini en local.

Paramètres de sécurité relatifs aux droits d'utilisateur de domaine et de groupe

IBM MQ s'appuie sur les informations d'appartenance à un groupe pour implémenter sa stratégie de sécurité, ce qui signifie qu'il est important que l'ID utilisateur qui exécute des opérations IBM MQ puisse déterminer les appartenances à des groupes d'autres utilisateurs.

Lorsque vous promouvez un serveur Windows sur un contrôleur de domaine, une option s'affiche pour le paramètre de sécurité relatif aux droits d'accès des utilisateurs et des groupes. Cette option contrôle si des utilisateurs arbitraires peuvent extraire des appartenances à des groupes à partir du répertoire actif. Si un contrôleur de domaine est configuré de sorte que les comptes locaux soient autorisés à interroger l'appartenance à un groupe des comptes utilisateur de domaine, l'ID utilisateur par défaut créé par IBM MQ lors du processus d'installation peut obtenir des appartenances à des groupes pour d'autres utilisateurs, selon les besoins. Toutefois, si un contrôleur de domaine est configuré de sorte que les comptes locaux ne soient pas autorisés à interroger l'appartenance à un groupe des comptes utilisateur de domaine, cela empêche IBM MQ de vérifier que les utilisateurs définis sur le domaine sont autorisés à accéder aux gestionnaires de files d'attente ou aux files d'attente et que l'accès échoue. Si vous utilisez Windows sur un contrôleur de domaine qui a été configuré de cette manière, un compte utilisateur de domaine spécial avec les droits requis doit être utilisé.

Dans ce cas, vous devez connaître:

- Comment se comportent les droits de sécurité pour votre version de Windows .
- Comment autoriser les membres du groupe mqm de domaine à lire l'appartenance à un groupe.
- Comment configurer un service IBM MQ Windows pour qu'il s'exécute sous un utilisateur de domaine.

Pour plus d'informations, voir Configuration de comptes utilisateur pour IBM MQ.

Accès IBM MQ au groupe mqm local

Lorsque des serveurs Windows sont promus ou rétrogradés à partir de contrôleurs de domaine, IBM MQ perd l'accès au groupe mqm local.

Lorsqu'un serveur est promu en tant que contrôleur de domaine, la portée passe de locale à locale. Lorsque la machine est rétrogradée sur le serveur, tous les groupes locaux de domaine sont supprimés. Cela signifie que le passage d'une machine d'un serveur à un contrôleur de domaine et le retour au serveur perdent l'accès à un groupe mqm local. Le symptôme est une erreur indiquant l'absence d'un groupe mqm local, par exemple:

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

Pour résoudre ce problème, recréez le groupe mqm local à l'aide des outils de gestion Windows standard. Etant donné que toutes les informations d'appartenance à un groupe sont perdues, vous devez rétablir les utilisateurs IBM MQ privilégiés dans le groupe mqm local nouvellement créé. Si la machine est un membre de domaine, vous devez également ajouter le groupe mqm de domaine au groupe mqm local pour accorder aux ID utilisateur IBM MQ de domaine privilégié le niveau de droits requis.

Windows *Restrictions sur les groupes imbriqués sous Windows*

L'utilisation de groupes imbriqués est soumise à des restrictions. Elles résultent en partie du niveau fonctionnel du domaine et en partie des restrictions IBM MQ .

Active Directory peut prendre en charge différents types de groupe dans un contexte de domaine en fonction du niveau fonctionnel du domaine. Par défaut, les domaines Windows 2003 se trouvent dans le répertoire " Niveau fonctionnel Windows 2000 mixte. (Windows Server 2008 et Windows Server 2012 suivent le modèle de domaine Windows 2003 .) Le niveau fonctionnel de domaine détermine les types de groupe pris en charge et le niveau d'imbrication autorisé lors de la configuration des ID utilisateur dans un environnement de domaine. Reportez-vous à la documentation Active Directory pour plus de détails sur la portée du groupe et les critères d'inclusion.

Outre les exigences relatives à Active Directory , d'autres restrictions s'appliquent aux ID utilisés par IBM MQ. Les API réseau utilisées par IBM MQ ne prennent pas en charge toutes les configurations prises en charge par le niveau fonctionnel de domaine. Par conséquent, IBM MQ ne peut pas interroger les appartenances aux groupes des ID de domaine présents dans un groupe local de domaine qui est ensuite imbriqué dans un groupe local. En outre, l'imbrication multiple de groupes globaux et universels n'est pas prise en charge. Toutefois, les groupes globaux ou universels immédiatement imbriqués sont pris en charge.

Windows *Autorisation des utilisateurs à utiliser IBM MQ à distance*

Si vous devez créer et démarrer des gestionnaires de files d'attente lorsque vous êtes connecté à IBM MQ à distance, vous devez disposer de l'accès utilisateur `Créer des objets globaux` .

Pourquoi et quand exécuter cette tâche

Remarque : Les administrateurs disposent de l'accès de `création d'objets globaux` par défaut. Par conséquent, si vous en êtes un, vous pourrez créer et démarrer des gestionnaires de files d'attente en étant connecté à distance sans modifier vos droits utilisateur.

Si vous vous connectez à une machine Windows à l'aide de Terminal Services ou d'une connexion Bureau à distance et que vous rencontrez des problèmes lors de la création, du démarrage ou de la suppression d'un gestionnaire de files d'attente, cela peut être dû au fait que vous ne disposez pas de l'accès utilisateur `Créer des objets globaux`.

L'accès utilisateur de `création d'objets globaux` limite les utilisateurs autorisés à créer des objets dans l'espace de nom global. Pour qu'une application crée un objet global, elle doit s'exécuter dans un espace de nom global ou l'utilisateur sous l'ID duquel s'exécute l'application doit disposer de l'accès utilisateur de `création d'objets globaux`.

Lorsque vous vous connectez à distance à une machine Windows à l'aide des services Terminal Services ou d'une connexion de bureau distante, les applications s'exécutent dans leur propre espace de nom local. Si vous tentez de créer ou de supprimer un gestionnaire de files d'attente à l'aide d'IBM MQ Explorer ou avec la commande **crtmqm** ou **dltmqm**, ou si vous tentez de démarrer un gestionnaire de files d'attente avec la commande **strmqm**, un incident d'autorisation survient. Un FDC IBM MQ avec l'ID sonde XY132002 est créé.

Le démarrage d'un gestionnaire de files d'attente via IBM MQ Explorer ou avec la commande **amqmdain qmgr start** fonctionne car ces commandes ne démarrent pas le gestionnaire directement. En effet, ces commandes envoient la demande de démarrage du gestionnaire de files d'attente à un processus distinct s'exécutant dans l'espace de nom global.

Si les différentes méthodes d'administration de IBM MQ ne fonctionnent pas lorsque vous utilisez les services de terminal, essayez de définir le droit utilisateur `Créer des objets globaux` .

Procédure

1. Ouvrez le panneau Outils d'administration:

   Windows Server 2008 et Windows Server 2012
   Accédez à ce panneau à l'aide du **Panneau de configuration** > **Système et maintenance** > **Outils d'administration**.

   Windows 8.1
   Accédez à ce panneau à l'aide de **Outils d'administration** > **Gestion de l'ordinateur**
2. Cliquez deux fois sur **Stratégie de sécurité locale**.
3. Développez **Stratégies locales**.
4. Cliquez sur **Affectation des droits utilisateur**.
5. Ajoutez le nouvel utilisateur ou le nouveau groupe à la règle `Créer des objets globaux` .

Windows *Programme d'exit de canal SSPI sous Windows*

IBM MQ for Windows fournit un programme d'exit de sécurité, qui peut être utilisé sur les canaux de message et MQI. L'exit est fourni en tant que code source et code objet et fournit une authentification unidirectionnelle et bidirectionnelle.

L'exit de sécurité utilise l'interface SSPI (Security Support Provider Interface), qui fournit les fonctions de sécurité intégrées des plateformes Windows .

L'exit de sécurité fournit les services d'identification et d'authentification suivants:

authentification unidirectionnelle
Cela utilise la prise en charge de l'authentification Windows NT LAN Manager (NTLM). NTLM permet aux serveurs d'authentifier leurs clients. Il ne permet pas à un client d'authentifier un serveur ou à un serveur d'en authentifier un autre. NTLM a été conçu pour un environnement réseau dans lequel les serveurs sont supposés être authentiques. NTLM est pris en charge sur toutes les plateformes Windows prises en charge par IBM WebSphere MQ 7.0.

Ce service est généralement utilisé sur un canal MQI pour permettre à un gestionnaire de files d'attente serveur d'authentifier une application IBM MQ MQI client . Une application client est identifiée par l'ID utilisateur associé au processus en cours d'exécution.

Pour effectuer l'authentification, l'exit de sécurité à l'extrémité client d'un canal acquiert un jeton d'authentification auprès de NTLM et envoie le jeton dans un message de sécurité à son partenaire à l'autre extrémité du canal. L'exit de sécurité partenaire transmet le jeton à NTLM, qui vérifie que le jeton est authentique. Si l'exit de sécurité partenaire n'est pas satisfait de l'authenticité du jeton, il demande à l'agent MCA de fermer le canal.

Authentification bidirectionnelle ou mutuelle
Cela utilise les services d'authentification Kerberos . Le protocole Kerberos ne suppose pas que les serveurs d'un environnement réseau sont authentiques. Les serveurs peuvent authentifier les clients

et d'autres serveurs, et les clients peuvent authentifier les serveurs. Kerberos est pris en charge sur toutes les plateformes Windows prises en charge par IBM WebSphere MQ 7.0.

Ce service peut être utilisé sur les canaux de message et MQI. Sur un canal de transmission de messages, il fournit une authentification mutuelle des deux gestionnaires de files d'attente. Sur un canal MQI, il permet au gestionnaire de files d'attente du serveur et à l'application IBM MQ MQI client de s'authentifier mutuellement. Un gestionnaire de files d'attente est identifié par son nom préfixé par la chaîne `ibmMQSeries/`. Une application client est identifiée par l'ID utilisateur associé au processus en cours d'exécution.

Pour effectuer l'authentification mutuelle, l'exit de sécurité initiateur acquiert un jeton d'authentification auprès du serveur de sécurité Kerberos et envoie le jeton dans un message de sécurité à son partenaire. L'exit de sécurité partenaire transmet le jeton au serveur Kerberos , qui vérifie qu'il est authentique. Le serveur de sécurité Kerberos génère un second jeton que le partenaire envoie dans un message de sécurité à l'exit de sécurité initiateur. L'exit de sécurité initiateur demande ensuite au serveur Kerberos de vérifier que le deuxième jeton est authentique. Lors de cet échange, si l'un des exits de sécurité n'est pas satisfait de l'authenticité du jeton envoyé par l'autre, il demande à l'agent MCA de fermer le canal.

L'exit de sécurité est fourni au format source et au format objet. Vous pouvez utiliser le code source comme point de départ pour l'écriture de vos propres programmes d'exit de canal ou vous pouvez utiliser le module d'objet tel qu'il est fourni. Le module d'objet possède deux points d'entrée, l'un pour l'authentification unidirectionnelle à l'aide de la prise en charge de l'authentification NTLM et l'autre pour l'authentification bidirectionnelle à l'aide des services d'authentification Kerberos .

Pour plus d'informations sur le fonctionnement du programme d'exit de canal SSPI et pour savoir comment l'implémenter, voir Utilisation de l'exit de sécurité SSPI sur les systèmes Windows.

Windows *Application des fichiers de modèle de sécurité sous Windows*

L'application d'un modèle peut affecter les paramètres de sécurité appliqués aux fichiers et répertoires IBM MQ . Si vous utilisez le modèle hautement sécurisé, appliquez-le avant d'installer IBM MQ.

Windows prend en charge les fichiers de modèle de sécurité textuelle que vous pouvez utiliser pour appliquer des paramètres de sécurité uniformes à un ou plusieurs ordinateurs avec le composant logiciel enfichable MMC Configuration et analyse de la sécurité. En particulier, Windows fournit plusieurs modèles qui incluent une série de paramètres de sécurité dans le but de fournir des niveaux de sécurité spécifiques. Ces modèles incluent Compatible, Secure et Hautement Secure.

L'application de l'un de ces modèles peut affecter les paramètres de sécurité appliqués aux fichiers et répertoires IBM MQ . Si vous souhaitez utiliser le modèle hautement sécurisé, configurez votre machine avant d'installer IBM MQ.

Si vous appliquez le modèle hautement sécurisé à une machine sur laquelle IBM MQ est déjà installé, tous les droits que vous avez définis sur les fichiers et répertoires IBM MQ sont supprimés. Etant donné que ces droits sont supprimés, vous perdez *Administrator*, *mqm*et, le cas échéant, l'accès du groupe *Everyone* à partir des répertoires d'erreurs.

Windows *Configuration de droits d'accès supplémentaires pour les applications Windows se connectant à IBM MQ*

Le compte sous lequel les processus IBM MQ s'exécutent peut nécessiter une autorisation supplémentaire pour que l'accès à SYNCHRONISER aux processus d'application puisse être accordé.

Pourquoi et quand exécuter cette tâche

Vous pouvez rencontrer des problèmes si vous avez des applications Windows , par exemple des pages ASP, qui se connectent à IBM MQ et qui sont configurées pour s'exécuter à un niveau de sécurité supérieur à la normale.

IBM MQ requiert l'accès SYNCHRONISER aux processus d'application afin de coordonner certaines actions. Lorsqu'une application serveur tente pour la première fois de se connecter à un gestionnaire de files d'attente, IBM MQ modifie le processus pour accorder le droit SYNCHRONISER aux administrateurs

IBM MQ . Toutefois, le compte sous lequel les processus IBM MQ s'exécutent peut nécessiter une autorisation supplémentaire pour que l'accès demandé puisse être accordé.

Pour configurer des droits supplémentaires sur l'ID utilisateur sous lequel les processus IBM MQ s'exécutent, procédez comme suit:

Procédure

1. Démarrez l'outil Stratégie de sécurité locale, cliquez sur **Paramètres de sécurité**->**Stratégies locales**->**Affectations de droits utilisateur**, puis sur **Déboguer les programmes**.
2. Cliquez deux fois sur **Déboguer les programmes**, puis ajoutez votre ID utilisateur IBM MQ à la liste

   Si le système se trouve dans un domaine Windows et que le paramètre de stratégie effectif n'est toujours pas défini, même si le paramètre de stratégie locale est défini, l'ID utilisateur doit être autorisé de la même manière au niveau du domaine, à l'aide de l'outil de stratégie de sécurité du domaine.

IBM i Configuration de la sécurité sous IBM i

La sécurité sur IBM i est implémentée à l'aide de IBM MQ Object Authority Manager (OAM) et de la sécurité au niveau de l'objet IBM i .

Considérations de sécurité à prendre en compte lors de la détermination des droits d'accès aux objets IBM MQ .

Vous devez prendre en compte les points suivants lors de la configuration des droits des utilisateurs de votre entreprise:

1. Accordez et révoquez les droits sur les commandes IBM MQ for IBM i à l'aide des commandes IBM i GRTOBJAUT et RVKOBJAUT .

   Dans la bibliothèque QMQM , certains objets noncommand (* cmd) sont définis pour disposer des droits ***PUBLIC** sur ***USE**. Ne modifiez pas les droits de ces objets et n'utilisez pas de liste d'autorisation pour fournir des droits. Toute autorité incorrecte peut compromettre la fonctionnalité IBM MQ .
2. Lors de l'installation de IBM MQ for IBM i, les profils utilisateur spéciaux suivants sont créés:

   QMQM
   Est utilisé principalement pour les fonctions internes du produit uniquement. Toutefois, il peut être utilisé pour exécuter des applications sécurisées à l'aide de MQCNO_FASTPATH_BINDINGS. Voir Connexion à un gestionnaire de files d'attente à l'aide de l'appel MQCONNX.

   QMQMADM
   Est utilisé comme profil de groupe pour les administrateurs de IBM MQ. Le profil de groupe donne accès aux commandes CL et aux ressources IBM MQ .

   Lors de l'utilisation de SBMJOB pour soumettre des programmes qui appellent des commandes IBM MQ , USER ne doit pas être défini explicitement sur QMQMADM. A la place, définissez USER sur QMQM ou sur un autre profil utilisateur pour lequel QMQMADM est spécifié en tant que groupe.
3. Si vous envoyez des commandes de canal à des gestionnaires de files d'attente éloignées, vérifiez que votre profil utilisateur est membre du groupe QMQMADM sur le système cible. Pour obtenir la liste des commandes de canal PCF et MQSC, voir Commandes CLIBM MQ for IBM i.
4. L'ensemble de groupes associé à un utilisateur est mis en cache lorsque les autorisations de groupe sont calculées par la méthode d'accès aux objets (OAM).

   Les modifications apportées aux appartenances à un groupe d'utilisateurs après la mise en cache de l'ensemble de groupes ne sont pas reconnues tant que vous n'avez pas redémarré le gestionnaire de files d'attente ou exécuté RFRMQMAUT pour actualiser la sécurité.
5. Limitez le nombre d'utilisateurs autorisés à utiliser des commandes particulièrement sensibles. Ces commandes incluent:

   - Création d'un gestionnaire de files d'attente de messages (CRTMQM)

- Supprimer le gestionnaire de files d'attente de messages (DLTMQM)
- Démarrage du gestionnaire de files d'attente de messages (STRMQM)
- Arrêt du gestionnaire de files d'attente de messages (ENDMQM)
- Démarrer le serveur de commandes (STRMQMCSVR)
- Arrêt du serveur de commandes (ENDMQMCSVR)

6. Les définitions de canal contiennent une spécification de programme d'exit de sécurité. La création et la modification de canaux nécessitent des considérations particulières. Les détails des exits de sécurité sont fournis dans «Présentation de l'exit de sécurité», à la page 117.
7. Les programmes d'exit de canal et de moniteur de déclenchement peuvent être remplacés. La sécurité de ces remplacements est de la responsabilité du programmeur.

IBM i Gestionnaire des droits d'accès aux objets sous IBM i

Le gestionnaire des droits d'accès aux objets (OAM) gère les autorisations des utilisateurs pour manipuler les objets IBM MQ , y compris les files d'attente et les définitions de processus. Il fournit également une interface de commande grâce à laquelle vous pouvez accorder ou révoquer des droits d'accès à un objet pour un groupe spécifique d'utilisateurs. La décision d'autoriser l'accès à une ressource est prise par la méthode d'accès aux objets (OAM) et le gestionnaire de files d'attente suit cette décision. Si la méthode d'accès aux objets (OAM) ne peut pas prendre de décision, le gestionnaire de files d'attente empêche l'accès à cette ressource.

Grâce à la méthode d'accès aux objets (OAM), vous pouvez contrôler:

- Accès aux objets IBM MQ via l'interface MQI. Lorsqu'un programme d'application tente d'accéder à un objet, la méthode d'accès aux objets (OAM) vérifie que le profil utilisateur à l'origine de la demande dispose de l'autorisation pour l'opération demandée.

  En particulier, cela signifie que les files d'attente et les messages des files d'attente peuvent être protégés contre les accès non autorisés.
- Droit d'utilisation des commandes PCF et MQSC.

Différents groupes d'utilisateurs peuvent disposer de droits d'accès différents sur le même objet. Par exemple, pour une file d'attente spécifique, un groupe peut effectuer à la fois des opérations d'insertion et d'extraction ; un autre groupe peut être autorisé uniquement à parcourir la file d'attente (MQGET avec l'option de navigation). De même, certains groupes peuvent avoir des droits d'extraction et d'insertion sur une file d'attente, mais ils ne sont pas autorisés à modifier ou à supprimer la file d'attente.

Commandes IBM MQ for IBM i et opérations sur les objets IBM MQ for IBM i

IBM i Droits IBM MQ sur IBM i

Pour accéder aux objets IBM MQ , vous devez disposer des droits permettant d'émettre la commande et d'accéder à l'objet référencé. Les administrateurs ont accès à toutes les ressources IBM MQ .

L'accès aux objets IBM MQ est contrôlé par les droits d'accès à:

1. Exécutez la commande IBM MQ
2. Accès aux objets IBM MQ référencés par la commande

Toutes les commandes CL IBM MQ for IBM i sont fournies avec un propriétaire de QMQM et le profil d'administration (QMQMADM) possède les droits *USE avec l'accès *PUBLIC défini sur *EXCLUDE.

Remarque : Le programme QSRDUPER est utilisé par le programme d'installation de logiciel sous licence IBM MQ for IBM i pour dupliquer des objets Commande (*CMD) dans QSYS. Dans IBM i V5R4 et les versions ultérieures, le programme QSRDUPER a été modifié de sorte que le comportement par défaut consiste à créer une commande proxy plutôt qu'un doublon de la commande d'origine. Une commande proxy redirige l'exécution de la commande vers une autre commande et possède un attribut PRX. Si une commande proxy portant le même nom que la commande en cours de copie existe dans la bibliothèque QSYS, les droits privés sur la commande proxy ne sont pas accordés à la commande dans la bibliothèque

du produit. Les tentatives d'invite ou d'exécution de la commande proxy dans QSYS vérifient les droits de la commande cible dans la bibliothèque du produit. Toute modification des droits sur les objets *CMD doit donc être effectuée dans la bibliothèque de produit (QMQM) et celles de QSYS n'ont pas besoin d'être modifiées. Exemple :

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

Les modifications apportées à la structure des droits d'accès de certaines des commandes CL du produit permettent l'utilisation publique de ces commandes, si vous disposez des droits d'accès OAM requis sur les objets IBM MQ pour effectuer ces modifications.

Pour être un administrateur IBM MQ sous IBM i, vous devez être membre du *groupe QMQMADM*. Ce groupe possède des propriétés telles que les propriétés du groupe mqm sur les systèmes AIX, Linux, and Windows . En particulier, le groupe QMQMADM est créé lorsque vous installez IBM MQ for IBM iet les membres du groupe QMQMADM ont accès à toutes les ressources IBM MQ sur le système. Vous avez également accès à toutes les ressources IBM MQ si vous disposez des droits *ALLOBJ.

Les administrateurs peuvent utiliser des commandes CL pour administrer IBM MQ. L'une de ces commandes est GRTMQMAUT, qui permet d'accorder des droits à d'autres utilisateurs. Une autre commande, STRMQMMQSC, permet à un administrateur d'émettre des commandes MQSC vers un gestionnaire de files d'attente local.

Concepts associés

«Droit d'administration de IBM MQ sur IBM i», à la page 96

IBM i *Droits d'accès pour les objets IBM MQ sous IBM i*

Droits d'accès requis pour l'exécution des commandes CL IBM MQ .

IBM MQ for IBM i catégorise les commandes CL du produit en deux groupes:

Groupe 1

Les utilisateurs doivent appartenir au groupe d'utilisateurs QMQMADM ou disposer des droits *ALLOBJ pour pouvoir traiter ces commandes. Les utilisateurs disposant de l'un ou l'autre de ces droits peuvent traiter toutes les commandes de toutes les catégories sans avoir besoin de droits supplémentaires.

Remarque : Ces droits remplacent les droits OAM.

Ces commandes peuvent être regroupées comme suit:

- Commandes relatives au serveur de commandes
  - ENDMQMCSVR, arrêt du serveur de commandes IBM MQ
  - STRMQMCSVR, démarrage du serveur de commandes IBM MQ
- Commande du gestionnaire de files d'attente de messages non livrés
  - STRMQMDLQ, démarrage du gestionnaire de file d'attente des messages non livrés IBM MQ
- Commande du programme d'écoute
  - ENDMQMLSR, Arrêt du programme d'écoute IBM MQ
  - STRMQMLSR, démarrage d'un programme d'écoute non-objet
- Commandes relatives à la reprise sur incident lié au support
  - RCDMQMIMG, Enregistrement d'image d'objet IBM MQ
  - RCRMQMOBJ, recréation d'objet IBM MQ
  - WRKMQMTRN, Utilisation des transactions IBM MQ Q
- Commandes relatives au gestionnaire de files d'attente
  - CRTMQM, Création d'un gestionnaire de files d'attente de messages
  - DLTMQM, Suppression d'un gestionnaire de files d'attente de messages

  - ENDMQM, Arrêt du gestionnaire de files d'attente de messages
  - STRMQM, démarrage du gestionnaire de files d'attente de messages
- Commandes de sécurité
  - GRTMQMAUT, octroi de droits sur les objets IBM MQ
  - RVKMQMAUT, révocation des droits sur les objets IBM MQ
- Commande relative aux traces
  - TRCMQM, Traçage du travail IBM MQ
- Commandes de transaction
  - RSVMQMTRN, Résolution de la transaction IBM MQ
- Commandes moniteur de déclenchement
  - STRMQMTRM, démarrage du moniteur de déclenchement
- Commandes IBM MQSC
  - RUNMQSC, Exécution des commandes IBM MQSC
  - STRMQMMQSC, commandes Démarrer IBM MQSC

Groupe 2

Le reste des commandes, pour lesquelles deux niveaux de droits sont requis:

1. Droits IBM i pour l'exécution de la commande. Un administrateur IBM MQ le définit à l'aide de la commande **GRTOBJAUT** pour remplacer la restriction *PUBLIC (*EXCLUDE) pour un utilisateur ou un groupe d'utilisateurs.

   Exemple :

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

2. Droits IBM MQ permettant de manipuler les objets IBM MQ associés à la ou aux commandes, avec les droits IBM i appropriés à l'étape 1.

   Ces droits sont contrôlés par l'utilisateur disposant des droits OAM appropriés pour l'action requise, définis par un administrateur IBM MQ à l'aide de la commande **GRTMQMAUT**

   Exemple :

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

Les commandes peuvent être regroupées comme suit:

- Commandes relatives aux canaux
  - CHGMQMCHL, modification du canal IBM MQ

    Cela nécessite des droits de connexion * au gestionnaire de files d'attente et des droits d'accès * admchg au canal.
  - CPYMQMCHL, Copie du canal IBM MQ

    Pour cela, vous devez disposer des droits * connect et * admcrt sur le gestionnaire de files d'attente, des droits * admdsp sur le type de canal par défaut à copier et des droits * admcrt sur la classe d'objets du canal.

    Par exemple, la copie d'un canal émetteur nécessite le droit * admdsp sur SYSTEM.DEF.SENDER DEF.SENDER
  - CRTMQMCHL, Créer un canal IBM MQ

Pour cela, vous devez disposer des droits * connect et * admcrt sur le gestionnaire de files d'attente, * admdsp sur le type de canal par défaut à créer et * admcrt sur la classe d'objets du canal.

Par exemple, la création d'un canal émetteur requiert le droit * admdsp sur SYSTEM.DEF.SENDER DEF.SENDER

– DLTMQMCHL, Supprimer le canal IBM MQ

Cela nécessite * le droit de connexion au gestionnaire de files d'attente et * le droit d'admdlt au canal.

– RSVMQMCHL, Résolution du canal IBM MQ

Cela nécessite * l'autorisation de connexion au gestionnaire de files d'attente et * l'autorisation ctrlx au canal.

- Afficher les commandes

Pour traiter les commandes DSP, vous devez accorder à l'utilisateur les droits `*connect` et `*admdsp` sur le gestionnaire de files d'attente, ainsi que toute option spécifique répertoriée:

– DSPMQM, Affichage du gestionnaire de files d'attente de messages

– DSPMQMAUT, Affichage des droits sur les objets IBM MQ

– DSPMQMAUTI, Affichage des IBM MQ informations d'authentification- `*admdsp` à l'objet d'informations d'authentification

– DSPMQMCHL, Affichage du canal IBM MQ - `*admdsp` vers le canal

– DSPMQMCSVR, Affichage du serveur de commandes IBM MQ

– DSPMQMNL, Affichage de la IBM MQ liste de noms- `*admdsp` dans la liste de noms

– DSPMQMOBJN, Affichage des noms d'objet IBM MQ

– DSPMQMPRC, Affichage du processus IBM MQ - `*admdsp` au processus

– DSPMQMQ, Affichage de la file d'attente IBM MQ - `*admdsp` dans la file d'attente

– DSPMQMTOP, Affichage de la IBM MQ rubrique- `*admdsp` à la rubrique

- Gestion des commandes

Pour traiter les commandes WRK et afficher le panneau des options, vous devez accorder à l'utilisateur les droits `*connect` et `*admdsp` sur le gestionnaire de files d'attente, ainsi que toute option spécifique répertoriée:

– WRKMQM, Gestion des gestionnaires de files d'attente de messages

– WRKMQMAUT, Gestion des droits sur les objets IBM MQ

– WRKMQMAUTD, Gestion des données de droits sur les objets IBM MQ

– WRKMQMAUTI, Utilisation des informations d'authentification IBM MQ

- `*admchg` pour la commande Change IBM MQ Authentication Information Object.
- `*admcrt` pour la commande Create and Copy IBM MQ Authentication Information Object.
- `*admdlt` pour la commande Delete IBM MQ Authentication Information Object.
- `*admdsp` pour la commande Display IBM MQ Authentication Information Object.

– WRKMQMCHL, Utiliser le canal IBM MQ

Pour cela, vous devez disposer des droits suivants:

- `*admchg` pour la commande Change IBM MQ Channel.
- `*admclr` pour la commande Clear IBM MQ Channel.
- `*admcrt` pour la commande Créer et copier IBM MQ Channel.
- `*admdlt` pour la commande Delete IBM MQ Channel.
- `*admdsp` pour la commande Display IBM MQ Channel.

- *ctrl pour la commande Start IBM MQ Channel.
- *ctrl pour la commande End IBM MQ Channel.
- *ctrl pour la commande de canal IBM MQ Ping.
- *ctrlx pour la commande Reset IBM MQ Channel.
- *ctrlx pour la commande Resolve IBM MQ Channel.

– WRKMQMTCSPS, Gestion de l'état des canaux IBM MQ

  Pour cela, vous devez disposer des droits *admdsp sur le canal.

– WRKMQMCL, Gestion des clusters IBM MQ
– WRKMQMCLQ, Gestion des files d'attente de cluster IBM MQ
– WRKMQMCLQM, Utilisation du gestionnaire de files d'attente de cluster IBM MQ
– WRKMQMLSR, Utilisation du programme d'écoute IBM MQ
– WRKMQMMSG, Gestion des messages IBM MQ

  Pour cela, vous devez disposer des droits *browse sur la file d'attente.

– WRKMQMNL, Gestion des listes de noms IBM MQ

  Pour cela, vous devez disposer des droits suivants:

  - *admchg pour la commande Change IBM MQ Namelist.
  - *admcrt pour la commande Create and Copy IBM MQ Namelist.
  - *admdlt pour la commande Delete IBM MQ Namelist.
  - *admdsp pour la commande Display IBM MQ Namelist.

– WRKMQMPRC, Gestion des processus IBM MQ

  Pour cela, vous devez disposer des droits suivants:

  - *admchg pour la commande Change IBM MQ Process.
  - *admcrt pour la commande Créer et copier IBM MQ Process.
  - *admdlt pour la commande Delete IBM MQ Process.
  - *admdsp pour la commande Display IBM MQ Process.

– WRKMQMQ, Gestion des files d'attente IBM MQ

  Pour cela, vous devez disposer des droits suivants:

  - *admchg pour la commande Modifier la file d'attente IBM MQ .
  - *admclr pour la commande Clear IBM MQ Queue.
  - *admcrt pour la commande Create and Copy IBM MQ Queue.
  - *admdlt pour la commande Delete IBM MQ Queue.
  - *admdsp pour la commande Display IBM MQ Queue.

– WRKMQMQSTS, Gestion de l'état de la file d'attente IBM MQ
– WRKMQMTOP, Gestion des rubriques IBM MQ

  Pour cela, vous devez disposer des droits suivants:

  - *admchg pour la commande Change IBM MQ Topic.
  - *admcrt pour la commande Create and Copy IBM MQ Topic.
  - *admdlt pour la commande Delete IBM MQ Topic.
  - *admdsp pour la commande Display IBM MQ Topic.

– WRKMQMSUB, Gestion des abonnements IBM MQ

• Autres commandes de canal

Pour traiter les commandes de canal, vous devez accorder à l'utilisateur les droits spécifiques répertoriés:

- ENDMQMCHL, Arrêt du canal IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*allmqi` sur la file d'attente de transmission associée au canal.
- ENDMQMLSR, Fin du programme d'écoute IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*ctrl` sur l'objet programme d'écoute nommé.
- PNGMQMCHL, canal IBM MQ Ping

  Pour cela, vous devez disposer des droits `*connect` et `*inq` sur le gestionnaire de files d'attente et des droits `*ctrl` sur l'objet canal.
- RSTMQMCHL, réinitialisation du canal IBM MQ

  Pour cela, vous devez disposer des droits `*connect` sur le gestionnaire de files d'attente.
- STRMQMCHL, démarrage du canal IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*ctrl` sur l'objet canal.
- STRMQMCHLI, démarrage de l'initialisateur de canal IBM MQ

  Cela requiert des droits `*connect` et `*inq` sur le gestionnaire de files d'attente et des droits `*allmqi` sur la file d'attente d'initialisation associée à la file d'attente de transmission du canal.
- STRMQMLSR, démarrage du programme d'écoute IBM MQ

  Pour cela, vous devez disposer du droit de connexion * au gestionnaire de files d'attente et du droit de contrôle * ctrl sur l'objet programme d'écoute nommé.

- Autres commandes:

  Pour traiter les commandes suivantes, vous devez accorder à l'utilisateur les droits spécifiques répertoriés:

  - CCTMQM, Connexion au gestionnaire de files d'attente de messages

    Cela ne nécessite aucun droit sur les objets IBM MQ .
  - CHGMQM, modification du gestionnaire de files d'attente de messages

    Pour cela, vous devez disposer des droits `*connect` et `*admchg` sur le gestionnaire de files d'attente.
  - CHGMQMAUTI, modification des informations d'authentification IBM MQ

    Cela requiert des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admchg` et `*admdsp` sur l'objet d'informations d'authentification.
  - CHGMQMNL, modification de la liste de noms IBM MQ

    Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admchg` sur la liste de noms.
  - CHGMQMPRC, modification du processus IBM MQ

    Cela requiert le droit `*connect` sur le gestionnaire de files d'attente et le droit `*admchg` sur le processus.
  - CHGMQMQ, Modification de la file d'attente IBM MQ

    Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admchg` sur la file d'attente.
  - CLRMQMQ, Effacement de la file d'attente IBM MQ

    Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admclr` sur la file d'attente.

– CPYMQMAUTI, Copie des informations d'authentification IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admdsp` sur l'objet d'informations d'authentification et des droits `*admcrt` sur la classe d'objets d'informations d'authentification.

– CPYMQMNL, Copie de la liste de noms IBM MQ

  Pour cela, vous devez disposer des droits `*connect` et `*admcrt` sur le gestionnaire de files d'attente.

– CPYMQMPRC, Copie du processus IBM MQ

  Pour cela, vous devez disposer des droits `*connect` et `*admcrt` sur le gestionnaire de files d'attente.

– CPYMQMQ, Copie de la file d'attente IBM MQ

  Pour cela, vous devez disposer des droits `*connect` et `*admcrt` sur le gestionnaire de files d'attente.

– CRTMQMAUTI, Création des informations d'authentification IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admdsp` sur l'objet d'informations d'authentification et des droits `*admcrt` sur la classe d'objets d'informations d'authentification.

– CRTMQMNL, Création d'une liste de noms IBM MQ

  Cela requiert des droits `*connect` et `*admcrt` sur le gestionnaire de files d'attente et des droits `*admdsp` sur la liste de noms par défaut.

– CRTMQMPRC, création d'un processus IBM MQ

  Cela nécessite des droits `*connect` et `*admcrt` sur le gestionnaire de files d'attente et des droits `*admdsp` sur le processus par défaut.

– CRTMQMQ, Création d'une file d'attente IBM MQ

  Cela nécessite des droits `*connect` et `*admcrt` sur le gestionnaire de files d'attente et des droits `*admdsp` sur la file d'attente par défaut.

– CVTMQMDTA, Commande de conversion de type de données IBM MQ

  Cela ne nécessite aucun droit sur les objets IBM MQ .

– DLTMQMAUTI, Suppression des informations d'authentification IBM MQ

  Cela requiert des droits `*connect` sur le gestionnaire de files d'attente et des droits `*ctrlx` sur l'objet d'informations d'authentification.

– DLTMQMNL, Suppression de la liste de noms IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admdlt` sur la liste de noms.

– DLTMQMPRC, Suppression du processus IBM MQ

  Cela requiert le droit `*connect` sur le gestionnaire de files d'attente et le droit `*admdlt` sur le processus.

– DLTMQMQ, Suppression de la file d'attente IBM MQ

  Cela nécessite des droits `*connect` sur le gestionnaire de files d'attente et des droits `*admdlt` sur la file d'attente.

– DSCMQM, Déconnexion du gestionnaire de files d'attente de messages

  Cela ne nécessite aucun droit sur les objets IBM MQ .

– RFRMQMAUT, Actualiser la sécurité

  Pour cela, vous devez disposer des droits `*connect` sur le gestionnaire de files d'attente.

– RFRMQMCL, Actualiser le cluster

Pour cela, vous devez disposer des droits *connect sur le gestionnaire de files d'attente.

– RSMMQMCLQM, Reprise du gestionnaire de files d'attente de cluster

  Pour cela, vous devez disposer des droits *connect sur le gestionnaire de files d'attente.

– RSTMQMCL, réinitialisation du cluster

  Pour cela, vous devez disposer des droits *connect sur le gestionnaire de files d'attente.

– SPDMQMCLQM, Interrompre le gestionnaire de files d'attente de cluster

  Pour cela, vous devez disposer des droits *connect sur le gestionnaire de files d'attente.

IBM i *Autorisations d'accès sur IBM i*

Utilisez ces informations pour comprendre les commandes d'autorisation d'accès.

Les autorisations définies par le mot clé AUT sur les commandes GRTMQMAUT et RVKMQMAUT peuvent être catégorisées comme suit:

- Autorisations liées aux appels MQI
- Commandes d'administration liées aux autorisations
- Autorisations de contexte
- Autorisations générales, c'est-à-dire pour les appels MQI, pour les commandes ou les deux

Les tableaux suivants répertorient les différents droits, à l'aide du paramètre AUT pour les appels MQI, les appels de contexte, les commandes MQSC et PCF et les opérations génériques.

Tableau 15. Autorisations pour les appels MQI

AUT	Description
*ALTUSR	Autorisez l'utilisation des droits d'un autre utilisateur pour les appels MQOPEN et MQPUT1 .
*BROWSE	Extrayez un message d'une file d'attente en émettant un appel MQGET avec l'option BROWSE.
*CONNECT	Connectez l'application au gestionnaire de files d'attente spécifié en émettant un appel MQCONN.
*GET	Extrayez un message d'une file d'attente en émettant un appel MQGET.
*INQ	Effectuez une interrogation sur une file d'attente spécifique en émettant un appel MQINQ.
*PUB	Ouvrez une rubrique pour publier un message à l'aide d'un appel MQPUT.
*PUT	Insérez un message dans une file d'attente spécifique en émettant un appel MQPUT.
*RESUME	Reprenez un abonnement à l'aide d'un appel MQSUB.
*SET	Définissez les attributs d'une file d'attente à partir de l'interface MQI en émettant un appel MQSET. Si vous ouvrez une file d'attente pour plusieurs options, vous devez être autorisé pour chacune d'elles.
*SUB	Créer, modifier ou reprendre un abonnement à une rubrique à l'aide d'un appel MQSUB.

Tableau 16. Autorisations pour les appels de contexte

AUT	Description
*PASSALL	Transmettez tout le contexte à la file d'attente spécifiée. Toutes les zones de contexte sont copiées à partir de la demande d'origine.

Tableau 16. Autorisations pour les appels de contexte (suite)	
AUT	**Description**
*PASSID	Transmettez le contexte d'identité dans la file d'attente spécifiée. Le contexte d'identité est le même que celui de la demande.
*SETALL	Définit tous les contextes dans la file d'attente spécifiée. Il est utilisé par des utilitaires système spéciaux.
*SETID	Définit le contexte d'identité sur la file d'attente spécifiée. Il est utilisé par des utilitaires système spéciaux.

Tableau 17. Autorisations pour les appels MQSC et PCF	
AUT	**Description**
*ADMCHG	Modifiez les attributs de l'objet indiqué.
*ADMCLR	Mettez à blanc l'objet indiqué (commande PCF Mettre à blanc l'objet uniquement).
*ADMCRT	Créer des objets du type spécifié.
*ADMDLT	Supprimez l'objet spécifié.
*ADMDSP	Affiche les attributs de l'objet spécifié.

Tableau 18. Autorisations pour les opérations génériques	
AUT	**Description**
*ALL	Utilisez toutes les opérations applicables à l'objet. Les droits `all` sont équivalents à l'union des droits `alladm`, `allmqi`et `system` appropriés au type d'objet.
*ALLADM	Effectuez toutes les opérations d'administration applicables à l'objet.
*ALLMQI	Utilisez tous les appels MQI applicables à l'objet.
*CTRL	Contrôle le démarrage et l'arrêt des canaux, des programmes d'écoute et des services.
*CTRLX	Réinitialisez le numéro de séquence et résolvez les canaux en attente de validation.

IBM i *Utilisation des commandes d'autorisation d'accès sous IBM i*

Utilisez ces informations pour en savoir plus sur les commandes d'autorisation d'accès et utilisez les exemples de commande.

Utilisation de la commande GRTMQMAUT

Si vous disposez des droits requis, vous pouvez utiliser la commande GRTMQMAUT pour accorder à un profil utilisateur ou à un groupe d'utilisateurs l'autorisation d'accéder à un objet particulier. Les exemples suivants illustrent l'utilisation de la commande GRTMQMAUT :

1. ```
   GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
             AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
   ```

   Dans cet exemple :

   - `RED.LOCAL.QUEUE` est le nom de l'objet.
   - `*LCLQ` (file d'attente locale) est le type d'objet.
   ```

- GROUPA est le nom d'un profil utilisateur sur le système pour lequel les autorisations doivent être modifiées. Ce profil peut être utilisé comme profil de groupe pour d'autres utilisateurs.
- *BROWSE et *PUT sont les autorisations accordées à la file d'attente spécifiée.

  *BROWSE ajoute l'autorisation de parcourir les messages dans la file d'attente (pour émettre une commande MQGET avec l'option de navigation).

  *PUT ajoute une autorisation d'insertion (MQPUT) de messages dans la file d'attente.
- `saturn.queue.manager` est le nom du gestionnaire de files d'attente.

2. La commande suivante accorde aux utilisateurs JACK et JILL toutes les autorisations applicables, à toutes les définitions de processus, pour le gestionnaire de files d'attente par défaut.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```

3. La commande suivante accorde à l'utilisateur GEORGE le droit d'insérer un message dans la file d'attente ORDERS, sur le gestionnaire de files d'attente TRENT.

```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

Utilisation de la commande RVKMQMAUT

Si vous disposez de l'autorisation requise, vous pouvez utiliser la commande RVKMQMAUT pour supprimer l'autorisation accordée précédemment à un profil utilisateur ou à un groupe d'utilisateurs pour accéder à un objet particulier. Les exemples suivants illustrent l'utilisation de la commande RVKMQMAUT :

1.
```
RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
AUT(*PUT) MQMNAME('saturn.queue.manager')
```

   Les droits d'insertion de messages dans la file d'attente spécifiée, qui ont été accordés dans l'exemple précédent, sont supprimés pour GROUPA.

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```

   Le droit d'obtenir des messages à partir de n'importe quelle file d'attente dont le nom commence par les caractères PAY, appartenant au gestionnaire de files d'attente PAYROLLQM, est supprimé de tous les utilisateurs du système, sauf s'ils, ou un groupe auquel ils appartiennent, ont été autorisés séparément.

Utilisation de la commande DSPMQMAUT

L'affichage des droits MQM (DSPMQMAUT) affiche, pour l'objet et l'utilisateur spécifiés, la liste des autorisations dont dispose l'utilisateur pour l'objet. L'exemple suivant illustre l'utilisation de la commande:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

Utilisation de la commande RFRMQMAUT

L'actualisation de la sécurité MQM (RFRMQMAUT) vous permet de mettre à jour immédiatement les informations du groupe d'autorisations de la méthode d'accès aux objets (OAM), en reflétant les

modifications apportées au niveau du système d'exploitation, sans qu'il soit nécessaire d'arrêter et de redémarrer le gestionnaire de files d'attente. L'exemple suivant illustre l'utilisation de la commande:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i Tables de spécifications d'autorisation sous IBM i

Utilisez ces informations pour déterminer l'autorisation requise pour utiliser des appels d'API particuliers, ainsi que des options spécifiques de ces appels, sur des objets de file d'attente, des objets de processus et des objets de gestionnaire de files d'attente.

Les tables de spécification d'autorisation démarrant dans Tableau 19, à la page 178 définissent précisément le fonctionnement des autorisations et les restrictions qui s'appliquent. Les tableaux s'appliquent aux situations suivantes:

- Applications qui émettent des appels MQI
- Programmes d'administration qui émettent des commandes MQSC sous forme de fichiers PCF d'échappement
- Programmes d'administration qui émettent des commandes PCF

Dans cette section, les informations sont présentées sous la forme d'un ensemble de tables qui spécifient les données suivantes:

Action à exécuter
Option MQI, commande MQSC ou commande PCF.

Objet de contrôle d'accès
File d'attente, définition de processus, gestionnaire de files d'attente, liste de noms, canal, canal de connexion client, programme d'écoute, service ou objet d'informations d'authentification.

Autorisation requise
Exprimée sous la forme d'une constante MQZAO_.

Dans les tableaux, les constantes préfixées par MQZAO_ correspondent aux mots clés de la liste d'autorisation pour les commandes **GRTMQMAUT** et **RVKMQMAUT** pour l'entité particulière. Par exemple, MQZAO_BROWSE correspond au mot clé *BROWSE ; De même, le mot clé MQZAO_SET_ALL_CONTEXT correspond au mot clé *SETALL, etc. Ces constantes sont définies dans le fichier d'en-tête cmqzc.h, qui est fourni avec le produit.

Autorisations MQI

Une application est autorisée à émettre des appels et des options MQI spécifiques uniquement si l'identificateur utilisateur sous lequel elle s'exécute (ou dont elle peut assumer les autorisations) a reçu l'autorisation appropriée.

Quatre appels MQI requièrent des vérifications d'autorisation: MQCONN, MQOPEN, MQPUT1et MQCLOSE.

Pour MQOPEN et MQPUT1, la vérification des droits est effectuée sur le nom de l'objet en cours d'ouverture et non sur le ou les noms, ce qui se produit après la résolution d'un nom. Par exemple, une application peut être autorisée à ouvrir une file d'attente alias sans avoir le droit d'ouvrir la file d'attente de base dans laquelle l'alias est résolu. La règle est que la vérification est effectuée sur la première définition rencontrée lors du processus de résolution de nom qui n'est pas un alias de gestionnaire de files d'attente, sauf si la définition d'alias de gestionnaire de files d'attente est ouverte directement ; c'est-à-dire que son nom apparaît dans la zone *ObjectName* du descripteur d'objet. Des droits sont toujours nécessaires pour l'objet en cours d'ouverture ; dans certains cas, des droits supplémentaires indépendants de la file d'attente, obtenus via une autorisation pour l'objet gestionnaire de files d'attente, sont requis.

Tableau 19, à la page 178, Tableau 20, à la page 178, Tableau 21, à la page 179et Tableau 22, à la page 179 récapitulent les autorisations requises pour chaque appel.

Remarque : Ces tables ne mentionnent pas les listes de noms, les canaux, les canaux de connexion client, les programmes d'écoute, les services ou les objets d'informations d'authentification. En effet, aucune des autorisations ne s'applique à ces objets, à l'exception de MQOO_INQUIRE, pour lequel les mêmes autorisations s'appliquent que pour les autres objets.

Tableau 19. Autorisation de sécurité requise pour les appels MQCONN

Autorisation requise pour:	**Objet de file d'attente («1», à la page 179)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQCONN, option	Non applicable	Non applicable	MQZAO_CONNECT

Tableau 20. Autorisation de sécurité requise pour les appels MQOPEN

Autorisation requise pour:	**Objet de file d'attente («1», à la page 179)**	**Objet Processus**	**Objet gestionnaire de files d'attente**
MQOO_INTERROGATION	MQZAO_INQUIRE **(«2», à la page 179)**	MQZAO_INQUIRE **(«2», à la page 179)**	MQZAO_INQUIRE **(«2», à la page 179)**
MQOO_BROWSE	MQZAO_PARCOURIR	Non applicable	Aucune vérification
MQOO_ENTRÉE_ *	MQZAO_ENTREE	Non applicable	Aucune vérification
MQOO_SAVE_ ALL_CONTEXT **(«3», à la page 179)**	MQZAO_ENTREE	Non applicable	Non applicable
MQOO_OUTPUT (file d'attente normale) **(«4», à la page 179)**	MQZAO_OUTPUT	Non applicable	Non applicable
MQOO_PASS_IDENTITY_CONTEXT **(«5», à la page 179)**	MQZAO_PASS_ IDENTITY_CONTEXT	Non applicable	Aucune vérification
MQOO_PASS_ALL_ CONTEXT **(«5», à la page 179, «6», à la page 179)**	MQZAO_PASS _ALL_CONTEXT	Non applicable	Aucune vérification
MQOO_SET_IDENTITY_CONTEXT **(«5», à la page 179, «6», à la page 179)**	MQZAO_SET_IDENTITY_CONTEXT	Non applicable	MQZAO_SET_ IDENTITY_CONTEXT **(«7», à la page 179)**
MQOO_SET_ ALL_CONTEXT **(«5», à la page 179, «8», à la page 180)**	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ ALL_CONTEXT **(«7», à la page 179)**
MQOO_OUTPUT (file d'attente de transmission) **(«9», à la page 180)**	MQZAO_SET_ALL_CONTEXT	Non applicable	MQZAO_SET_ ALL_CONTEXT **(«7», à la page 179)**
MQOO_SET	MQZAO_SET	Non applicable	Aucune vérification
MQOO_ALTERNATE_ AUTORITE UTILISATEUR	**(«10», à la page 180)**	**(«10», à la page 180)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(«10», à la page 180, «11», à la page 180)**

Tableau 21. Autorisation de sécurité requise pour les appels MQPUT1

Autorisation requise pour:	Objet de file d'attente («1», à la page 179)	Objet Processus	Objet gestionnaire de files d'attente
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT («12», à la page 180)	Non applicable	Aucune vérification
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT («12», à la page 180)	Non applicable	Aucune vérification
MQPMO_SET_IDENTITY_C ONTEXT	MQZAO_SET_ IDENTITY_CONTEXT («12», à la page 180)	Non applicable	MQZAO_SET_ IDENTITY_CONTEXT («7», à la page 179)
MQPMO_SET_ALL_CONTE XT	MQZAO_SET_ ALL_CONTEXT («12», à la page 180)	Non applicable	MQZAO_SET_ ALL_CONTEXT («7», à la page 179)
(File d'attente de transmission) («9», à la page 180)	MQZAO_SET_ALL_CONTE XT	Non applicable	MQZAO_SET_ ALL_CONTEXT («7», à la page 179)
MQPMO_ALTERNATE_USE R_AUTHORITY	(«13», à la page 180)	Non applicable	MQZAO_ALTERNATE_ USER_AUTHORITY («11», à la page 180)

Tableau 22. Autorisation de sécurité requise pour les appels MQCLOSE

Autorisation requise pour:	Objet de file d'attente («1», à la page 179)	Objet Processus	Objet gestionnaire de files d'attente
MQCO_DELETE	MQZAO_DELETE («14», à la page 180)	Non applicable	Non applicable
MQCO_DELETE _PURGE	MQZAO_DELETE («14», à la page 180)	Non applicable	Non applicable

Remarques relatives aux tableaux:

1. Si une file d'attente modèle est ouverte:
   - Le droit MQZAO_DISPLAY est requis pour la file d'attente modèle, en plus du droit d'ouverture de la file d'attente modèle pour le type d'accès pour lequel vous l'ouvrez.
   - Les droits MQZAO_CREATE ne sont pas nécessaires pour créer la file d'attente dynamique.
   - L'ID utilisateur utilisé pour ouvrir la file d'attente modèle reçoit automatiquement tous les droits spécifiques à la file d'attente (équivalents à MQZAO_ALL) pour la file d'attente dynamique créée.
2. L'objet file d'attente, processus, liste de noms ou gestionnaire de files d'attente est vérifié, en fonction du type d'objet ouvert.
3. MQOO_INPUT_ * doit également être spécifié. Cette option est valide pour une file d'attente locale, modèle ou alias.
4. Cette vérification est effectuée pour toutes les observations de sortie, à l'exception de la casse spécifiée dans la remarque «9», à la page 180.
5. MQOO_OUTPUT doit également être spécifié.
6. MQOO_PASS_IDENTITY_CONTEXT est également impliqué par cette option.
7. Ce droit est requis pour l'objet gestionnaire de files d'attente et la file d'attente particulière.

8. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT et MQOO_SET_IDENTITY_CONTEXT sont également impliquées par cette option.
9. Cette vérification est effectuée pour une file d'attente locale ou modèle dont l'attribut de file d'attente *Utilisation* est MQUS_TRANSMISSION et qui est ouverte directement pour la sortie. Elle ne s'applique pas si une file d'attente éloignée est ouverte (soit en spécifiant les noms du gestionnaire de files d'attente éloignées et de la file d'attente éloignée, soit en indiquant le nom d'une définition locale de la file d'attente éloignée).
10. Au moins l'une des options MQOO_INQUIRE (pour tout type d'objet) ou (pour les files d'attente) MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT ou MQOO_SET doit également être spécifiée. La vérification effectuée est la même que pour les autres options spécifiées, à l'aide de l'identificateur d'utilisateur de remplacement fourni pour les droits sur les objets nommés spécifiques, et des droits d'application en cours pour la vérification MQZAO_ALTERNATE_USER_IDENTIFIER.
11. Cette autorisation permet de spécifier tout ID *AlternateUser* .
12. Une vérification MQZAO_OUTPUT est également effectuée si la file d'attente ne possède pas l'attribut de file d'attente *Usage* MQUS_TRANSMISSION.
13. La vérification effectuée est la même que pour les autres options spécifiées, à l'aide de l'identificateur d'utilisateur de remplacement fourni pour l'autorité de file d'attente nommée et de l'autorité d'application en cours pour la vérification MQZAO_ALTERNATE_USER_IDENTIFIER.
14. La vérification est effectuée uniquement si les deux affirmations suivantes sont vraies:
    - Une file d'attente dynamique permanente est en cours de fermeture et de suppression.
    - La file d'attente n'a pas été créée par le MQOPEN qui a renvoyé le descripteur d'objet utilisé.

    Sinon, il n'y a pas de contrôle.

Remarques générales:

1. L'autorisation spéciale MQZAO_ALL_MQI inclut toutes les autorisations suivantes qui sont pertinentes pour le type d'objet:
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_SET
   - MQZAO_PARCOURIR
   - MQZAO_ENTREE
   - 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 (voir la remarque «14», à la page 180) et MQZAO_DISPLAY sont classés en tant qu'autorisations d'administration. Ils ne sont donc pas inclus dans MQZAO_ALL_MQI.
3. *Aucune vérification* signifie qu'aucune vérification d'autorisation n'est effectuée.
4. *Non applicable* signifie que le contrôle d'autorisation n'est pas pertinent pour cette opération. Par exemple, vous ne pouvez pas émettre un appel MQPUT à un objet de processus.

IBM i *Autorisations pour les commandes MQSC dans les fichiers PCF d'échappement sous IBM i*

Ces autorisations permettent à un utilisateur d'émettre des commandes d'administration en tant que message PCF d'arrêt programme. Ces méthodes permettent à un programme d'envoyer une commande d'administration sous forme de message à un gestionnaire de files d'attente, pour exécution pour le compte de cet utilisateur.

Cette section récapitule les autorisations nécessaires pour chaque commande MQSC contenue dans Escape PCF.

Non applicable signifie que le contrôle d'autorisation n'est pas pertinent pour cette opération.

L'ID utilisateur sous lequel le programme qui soumet la commande s'exécute doit également disposer des droits suivants:

- Droits MQZAO_CONNECT sur le gestionnaire de files d'attente
- Droits DISPLAY sur le gestionnaire de files d'attente afin d'exécuter des commandes PCF
- Droit d'émettre les commandes MQSC dans le texte de la commande Escape PCF

ALTER *objet*

Objet	**Autorisation requise**
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE

CLEAR *objet*

Objet	**Autorisation requise**
File d'attente	MQZAO_CLEAR
Topic	MQZAO_CLEAR
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	Non applicable
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

DEFINE *objet* NOREPLACE («1», à la page 185)

Objet	**Autorisation requise**
File d'attente	MQZAO_CREATE **(«2», à la page 185)**
Topic	MQZAO_CREATE **(«2», à la page 185)**
Processus	MQZAO_CREATE **(«2», à la page 185)**

Objet	Autorisation requise
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_CREATE **(«2», à la page 185)**
Informations d'authentification	MQZAO_CREATE **(«2», à la page 185)**
Canal	MQZAO_CREATE **(«2», à la page 185)**
Canal de connexion client	MQZAO_CREATE **(«2», à la page 185)**
Programme d'écoute	MQZAO_CREATE **(«2», à la page 185)**
Service	MQZAO_CREATE **(«2», à la page 185)**

DEFINE *objet* REPLACE («1», à la page 185, «3», à la page 185)

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE

DELETE *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DELETE
Topic	MQZAO_DELETE
Processus	MQZAO_DELETE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_DELETE
Informations d'authentification	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de connexion client	MQZAO_DELETE
Programme d'écoute	MQZAO_DELETE
Service	MQZAO_DELETE

DISPLAY *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DISPLAY

Objet	Autorisation requise
Topic	MQZAO_DISPLAY
Processus	MQZAO_DISPLAY
Gestionnaire de files d'attente	MQZAO_DISPLAY
Liste de noms	MQZAO_DISPLAY
Informations d'authentification	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de connexion client	MQZAO_DISPLAY
Programme d'écoute	
Service	

Envoyer une commande PING à un canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Réinitialisation du canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	MQZAO_CONTRÔLE_ÉTENDU
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Résolution du canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	MQZAO_CONTRÔLE_ÉTENDU
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

START *objet*

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	CONTROLE MQZ
Service	CONTROLE MQZ

STOP *objet*

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	CONTROLE MQZ

Objet	Autorisation requise
Service	CONTROLE MQZ

Remarque :

1. Pour les commandes DEFINE, le droit MQZAO_DISPLAY est également requis pour l'objet LIKE si un tel droit est spécifié, ou sur le système SYSTEM.DEFAULT.xxx si LIKE est omis.
2. Les droits MQZAO_CREATE ne sont pas spécifiques à un objet ou à un type d'objet particulier. Le droit de création est accordé pour tous les objets d'un gestionnaire de files d'attente spécifié, en spécifiant un type d'objet QMGR dans la commande GRTMQMAUT .
3. Cette option s'applique si l'objet à remplacer existe déjà. Si tel n'est pas le cas, la vérification est celle de l'objet DEFINE NOREPLACE.

IBM i *Autorisations pour les commandes PCF sous IBM i*

Ces autorisations permettent à un utilisateur d'émettre des commandes d'administration en tant que commandes PCF. Ces méthodes permettent à un programme d'envoyer une commande d'administration sous forme de message à un gestionnaire de files d'attente, pour exécution pour le compte de cet utilisateur.

Cette section récapitule les autorisations requises pour chaque commande PCF.

Aucune vérification signifie qu'aucune vérification d'autorisation n'est effectuée ; *Non applicable* signifie que la vérification d'autorisation n'est pas pertinente pour cette opération.

L'ID utilisateur sous lequel le programme qui soumet la commande s'exécute doit également disposer des droits suivants:

- Droits MQZAO_CONNECT sur le gestionnaire de files d'attente
- Droits DISPLAY sur le gestionnaire de files d'attente afin d'exécuter des commandes PCF

L'autorisation spéciale MQZAO_ALL_ADMIN inclut les autorisations suivantes:

- MODIFICATION MQZAO_DE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY
- CONTROLE MQZ
- MQZAO_CONTRÔLE_ÉTENDU

MQZAO_CREATE n'est pas inclus car il n'est pas spécifique à un objet ou à un type d'objet particulier

Modifier *objet*

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	MODIFICATION MQZAO_DE
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE

Objet	Autorisation requise
Service	MODIFICATION MQZAO_DE

Effacer *objet*

Objet	Autorisation requise
File d'attente	MQZAO_CLEAR
Topic	MQZAO_CLEAR
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	Non applicable
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Copier l'objet (sans remplacement) («1», à la page 191)

Objet	Autorisation requise
File d'attente	MQZAO_CREATE **(«2», à la page 191)**
Topic	MQZAO_CREATE **(«2», à la page 191)**
Processus	MQZAO_CREATE **(«2», à la page 191)**
Gestionnaire de files d'attente	Non applicable
NomelistMQZAO_CREATE	MQZAO_CREATE **(«2», à la page 191)**
Informations d'authentification	MQZAO_CREATE **(«2», à la page 191)**
Canal	MQZAO_CREATE **(«2», à la page 191)**
Canal de connexion client	MQZAO_CREATE **(«2», à la page 191)**
Programme d'écoute	MQZAO_CREATE **(«2», à la page 191)**
Service	MQZAO_CREATE **(«2», à la page 191)**

Copiez l'objet (avec remplacement) («1», à la page 191, «4», à la page 191)

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE

Objet	Autorisation requise
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE

Créer un objet (sans remplacement) («3», à la page 191)

Objet	Autorisation requise
File d'attente	MQZAO_CREATE **(«2», à la page 191)**
Topic	MQZAO_CREATE **(«2», à la page 191)**
Processus	MQZAO_CREATE **(«2», à la page 191)**
Gestionnaire de files d'attente	Non applicable
Liste de noms	MQZAO_CREATE **(«2», à la page 191)**
Informations d'authentification	MQZAO_CREATE **(«2», à la page 191)**
Canal	MQZAO_CREATE **(«2», à la page 191)**
Canal de connexion client	MQZAO_CREATE **(«2», à la page 191)**
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE

Créer un objet (avec remplacement) («3», à la page 191, «4», à la page 191)

Objet	Autorisation requise
File d'attente	MODIFICATION MQZAO_DE
Topic	MODIFICATION MQZAO_DE
Processus	MODIFICATION MQZAO_DE
Gestionnaire de files d'attente	Non applicable
Liste de noms	MODIFICATION MQZAO_DE
Informations d'authentification	MODIFICATION MQZAO_DE
Canal	MODIFICATION MQZAO_DE
Canal de connexion client	MODIFICATION MQZAO_DE
Programme d'écoute	MODIFICATION MQZAO_DE
Service	MODIFICATION MQZAO_DE

Supprimer *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DELETE
Topic	MQZAO_DELETE
Processus	MQZAO_DELETE
Gestionnaire de files d'attente	MQZAO_DELETE
Liste de noms	MQZAO_DELETE

Objet	Autorisation requise
Informations d'authentification	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de connexion client	MQZAO_DELETE
Programme d'écoute	MQZAO_DELETE
Service	MQZAO_DELETE

Interroger *objet*

Objet	Autorisation requise
File d'attente	MQZAO_DISPLAY
Topic	MQZAO_DISPLAY
Processus	MQZAO_DISPLAY
Gestionnaire de files d'attente	MQZAO_DISPLAY
Liste de noms	MQZAO_DISPLAY
Informations d'authentification	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de connexion client	MQZAO_DISPLAY
Programme d'écoute	MQZAO_DISPLAY
Service	MQZAO_DISPLAY

Consulter les noms d' *objet*

Objet	Autorisation requise
File d'attente	Aucune vérification
Topic	Aucune vérification
Processus	Aucune vérification
Gestionnaire de files d'attente	Aucune vérification
Liste de noms	Aucune vérification
Informations d'authentification	Aucune vérification
Canal	Aucune vérification
Canal de connexion client	Aucune vérification
Programme d'écoute	Aucune vérification
Service	Aucune vérification

Envoyer une commande PING à un canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable

Objet	Autorisation requise
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Réinitialisation du canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	MQZAO_CONTRÔLE_ÉTENDU
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Réinitialiser les statistiques de file d'attente

Objet	Autorisation requise
File d'attente	MQZAO_DISPLAY et MQZAO_CHANGE
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	Non applicable
Canal de connexion client	Non applicable
Programme d'écoute	
Service	

Résolution du canal

Objet	Autorisation requise
File d'attente	Non applicable

Objet	Autorisation requise
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	MQZAO_CONTRÔLE_ÉTENDU
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Démarrer un canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Arrêter le canal

Objet	Autorisation requise
File d'attente	Non applicable
Topic	Non applicable
Processus	Non applicable
Gestionnaire de files d'attente	Non applicable
Liste de noms	Non applicable
Informations d'authentification	Non applicable
Canal	CONTROLE MQZ
Canal de connexion client	Non applicable
Programme d'écoute	Non applicable
Service	Non applicable

Remarque :

1. Pour les commandes de copie, le droit MQZAO_DISPLAY est également requis pour l'objet From.
2. Les droits MQZAO_CREATE ne sont pas spécifiques à un objet ou à un type d'objet particulier. Le droit de création est accordé pour tous les objets d'un gestionnaire de files d'attente spécifié, en spécifiant un type d'objet QMGR dans la commande GRTMQMAUT .
3. Pour les commandes de création, les droits MQZAO_DISPLAY sont également requis pour le système SYSTEM.DEFAULT.* .
4. Cette option s'applique si l'objet à remplacer existe déjà. Si ce n'est pas le cas, la vérification est la même que pour la copie ou la création sans remplacement.

IBM i Profils OAM génériques sous IBM i

Les profils génériques du gestionnaire des droits d'accès aux objets (OAM) vous permettent de définir les droits d'accès d'un utilisateur à de nombreux objets à la fois, au lieu d'avoir à émettre des commandes **GRTMQMAUT** distinctes pour chaque objet individuel lors de sa création. L'utilisation de profils génériques dans la commande **GRTMQMAUT** vous permet de définir un droit générique pour tous les futurs objets créés qui correspondent à ce profil.

Le reste de cette section décrit plus en détail l'utilisation des profils génériques:

- «Utilisation des caractères génériques», à la page 191
- «Priorités de profil», à la page 192

Utilisation des caractères génériques

Ce qui rend un profil générique, c'est l'utilisation de caractères spéciaux (caractères génériques) dans le nom du profil. Par exemple, le caractère générique point d'interrogation (?) correspond à n'importe quel caractère unique dans un nom. Par conséquent, si vous spécifiez ABC.?EF, l'autorisation que vous accordez à ce profil s'applique à tous les objets créés avec les noms ABC.DEF, ABC.CEF, ABC.BEF, etc.

Les caractères génériques disponibles sont les suivants:

?

Utilisez le point d'interrogation (?) au lieu de n'importe quel caractère. Par exemple, AB.?D s'applique aux objets AB.CD, AB.EDet AB.FD.

*****

Utilisez l'astérisque (*) comme suit:

- Un *qualificateur* dans un nom de profil pour correspondre à un qualificateur dans un nom d'objet. Le qualificatif est une partie de nom d'objet délimitée par un point. Par exemple, dans ABC.DEF.GHI, les qualificatifs sont ABC, DEF et GHI.

  Par exemple, ABC.*.JKL s'applique aux objets ABC.DEF.JKLet ABC.GHI.JKL. (Notez que cela ne s'applique **pas** à ABC.JKL ; * utilisé dans ce contexte indique toujours un qualificateur.)
- Caractère dans un qualificatif d'un nom de profil qui doit correspondre à zéro ou plusieurs caractères dans le qualificatif d'un nom d'objet.

  Par exemple, ABC.DE*.JKL s'applique aux objets ABC.DE.JKL, ABC.DEF.JKLet ABC.DEGH.JKL.

******

Utilisez le double astérisque (**) ***une fois*** dans un nom de profil comme suit:

- Nom de profil complet correspondant à tous les noms d'objet. Par exemple, si vous utilisez le mot clé OBJTYPE (*PRC) pour identifier les processus, puis utilisez ** comme nom de profil, vous modifiez les autorisations pour tous les processus.
- Comme qualificatif de début, de milieu ou de fin dans un nom de profil pour correspondre à zéro ou plusieurs qualificatifs dans un nom d'objet. Par exemple, **.ABC identifie tous les objets avec le qualificateur final ABC.

Priorités de profil

Un point important à comprendre lors de l'utilisation de profils génériques est la priorité donnée aux profils lors de la détermination des droits à appliquer à un objet en cours de création. Par exemple, supposons que vous ayez émis les commandes suivantes:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

Le premier accorde le droit d'insertion à toutes les files d'attente pour le principal FRED dont les noms correspondent au profil AB. * ; La seconde permet d'obtenir des droits sur les mêmes types de file d'attente qui correspondent au profil AB.C*.

Supposons que vous créiez maintenant une file d'attente appelée AB.CD. Selon les règles de correspondance des caractères génériques, GRTMQMAUT peut s'appliquer à cette file d'attente. Alors, a-t-elle mis ou obtenu l'autorité?

Pour trouver la réponse, vous appliquez la règle selon laquelle, chaque fois que plusieurs profils peuvent s'appliquer à un objet, **seuls les plus spécifiques s'appliquent**. La façon dont vous appliquez cette règle consiste à comparer les noms de profil de gauche à droite. Lorsqu'ils diffèrent, un caractère non générique est plus spécifique qu'un caractère générique. Ainsi, dans l'exemple précédent, la file d'attente AB.CD dispose du droit **get** (AB.C* est plus spécifique que AB. *).

Lorsque vous comparez des caractères génériques, l'ordre de la *spécificité* est le suivant:

1. ?
2. *
3. **

IBM i Spécification du service d'autorisation installé sous IBM i

Vous pouvez spécifier le composant de service d'autorisation à utiliser.

Le paramètre **`Service Component name`** sous **GRTMQMAUT** et **RVKMQMAUT** vous permet de spécifier le nom du composant de service d'autorisation installé.

La sélection de **F24** dans le panneau initial, suivie de **F9=All parameters** dans le panneau suivant de l'une des commandes, vous permet de spécifier le composant d'autorisation installé (*DFT) ou le nom du composant de service d'autorisation requis spécifié dans la section Service du fichier qm.ini du gestionnaire de files d'attente.

DSPMQMAUT dispose également de ce paramètre supplémentaire. Ce paramètre permet de rechercher tous les composants d'autorisation installés (*DFT), ou le nom de composant de service d'autorisation indiqué, pour le nom d'objet, le type d'objet et l'utilisateur indiqués.

IBM i Utilisation avec et sans profils de droits d'accès sous IBM i

Utilisez ces informations pour apprendre à utiliser des profils de droits d'accès et à travailler sans profils de droits d'accès.

Vous pouvez utiliser des profils d'autorité, comme expliqué dans «Utilisation des profils de droits d'accès», à la page 192, ou sans eux, comme expliqué ici:

Pour travailler sans profils de droits d'accès, utilisez *NONE comme paramètre de droits d'accès sur **GRTMQMAUT** pour créer des profils sans droits d'accès. Tous les profils existants restent inchangés.

Sous **RVKMQMAUT**, utilisez *REMOVE comme paramètre de droits d'accès pour supprimer un profil de droits d'accès existant.

Utilisation des profils de droits d'accès

Deux commandes sont associées au profilage des droits:

- **WRKMQMAUT**
- **WRKMQMAUTD**

Vous pouvez accéder à ces commandes directement à partir de la ligne de commande ou à partir du panneau WRKMQM en:

1. Entrez le nom du gestionnaire de files d'attente et appuyez sur la touche `Enter` pour accéder au panneau des résultats **WRKMQM** .
2. Sélection de `F23=More options` sur ce panneau.

L'option 24 sélectionne le panneau de résultats pour la **WRKMQMAUT** commande et l'option 25 sélectionne la commande **WRKMQMAUTI** , qui est utilisée avec la couche de liaisons SSL.

WRKMQMAUT

Cette commande permet de gérer les données de droits d'accès contenues dans la file d'attente des droits d'accès.

Remarque : Pour exécuter cette commande, vous devez disposer des droits `*connect` et `*admdsp` sur le gestionnaire de files d'attente. Toutefois, pour créer ou supprimer un profil, vous devez disposer des droits QMQMADM.

Si vous affichez les informations à l'écran, une liste des noms de profil de droits d'accès, ainsi que leurs types, s'affiche. Si vous imprimez la sortie, vous recevez une liste détaillée de toutes les données de droits d'accès, des utilisateurs enregistrés et de leurs droits d'accès.

Lorsque vous entrez un nom d'objet ou de profil dans ce panneau et que vous appuyez sur la touche Entrée, vous accédez au panneau de résultats pour **WRKMQMAUT** .

Si vous sélectionnez `4=Delete`, vous accédez à un nouveau panneau à partir duquel vous pouvez confirmer que vous souhaitez supprimer tous les noms d'utilisateur enregistrés dans le nom de profil de droits d'accès générique que vous spécifiez. Cette option exécute **RVKMQMAUT** avec l'option *REMOVE pour tous les utilisateurs et applique **uniquement** aux noms de profil génériques.

Si vous sélectionnez `12=Work with profile` , vous accédez au panneau des résultats de la commande **WRKMQMAUTD** , comme expliqué dans «WRKMQMAUTD», à la page 193.

WRKMQMAUTD

Cette commande permet d'afficher tous les utilisateurs enregistrés avec un nom de profil de droits et un type d'objet particuliers. Pour exécuter cette commande, vous devez disposer des droits `*connect` et `*admdsp` sur le gestionnaire de files d'attente. Toutefois, pour accorder, exécuter, créer ou supprimer un profil, vous devez disposer des droits QMQMADM.

La sélection de `F24=More keys` dans le panneau d'entrée initial, suivie de l'option `F9=All Parameters` , affiche le nom du composant de service comme pour **GRTMQMAUT** et **RVKMQMAUT**.

Remarque : La touche `F11=Display Object Authorizations` permet de basculer entre les types de droits suivants:

- Autorisations d'objet
- Autorisations de contexte
- Autorisations MQI

Les options de l'écran sont les suivantes:

2=Grant
Permet d'accéder au panneau **GRTMQMAUT** pour ajouter des droits aux droits en cours.

3=Revoke
Permet d'accéder au panneau **RVKMQMAUT** pour supprimer certaines des définitions en cours

4=Delete
Permet d'accéder à un panneau qui vous permet de supprimer les données de droits d'accès pour les utilisateurs spécifiés. **RVKMQMAUT** est exécuté avec l'option *REMOVE.

5=Display
Vous permet d'accéder à la commande **DSPMQMAUT** existante

F6=Create
Vous permet d'accéder au panneau **GRTMQMAUT** qui vous permet de créer un enregistrement de droits d'accès de profil.

IBM i Instructions relatives à Object Authority Manager sous IBM i

Conseils et astuces supplémentaires pour l'utilisation du gestionnaire des droits d'accès aux objets (OAM)

Limiter l'accès aux opérations sensibles

Certaines opérations sont sensibles ; limitez-les aux utilisateurs privilégiés. Exemple :

- Accès à certaines files d'attente spéciales, telles que les files d'attente de transmission ou la file d'attente de commandes SYSTEM.ADMIN.COMMAND.QUEUE
- Exécution de programmes qui utilisent des options de contexte MQI complètes
- Création et copie de files d'attente d'application

Répertoires du gestionnaire de files d'attente

Les répertoires et les bibliothèques contenant les files d'attente et les autres données du gestionnaire de files d'attente sont privés pour le produit. N'utilisez pas de commandes de système d'exploitation standard pour accorder ou révoquer des autorisations sur les ressources MQI.

Files d'attente

Les droits d'accès à une file d'attente dynamique sont basés sur, mais ne sont pas nécessairement les mêmes que ceux de la file d'attente modèle dont elle est dérivée.

Pour les files d'attente d'alias et les files d'attente éloignées, l'autorisation est celle de l'objet lui-même, et non celle de la file d'attente dans laquelle l'alias ou la file d'attente éloignée est résolue. Il est possible d'autoriser un profil utilisateur à accéder à une file d'attente alias qui se résout en une file d'attente locale pour laquelle le profil utilisateur ne dispose pas de droits d'accès.

Limitez les droits de création de files d'attente aux utilisateurs privilégiés. Si vous ne le faites pas, les utilisateurs peuvent ignorer le contrôle d'accès normal en créant un alias.

Droits d'utilisateur de remplacement

Les droits d'utilisateur de remplacement contrôlent si un profil utilisateur peut utiliser les droits d'un autre profil utilisateur lors de l'accès à un objet IBM MQ . Cette technique est essentielle lorsqu'un serveur reçoit des demandes d'un programme et que le serveur souhaite s'assurer que le programme dispose des droits requis pour la demande. Le serveur peut disposer des droits requis, mais il doit savoir si le programme dispose des droits requis pour les actions qu'il a demandées.

Exemple :

- Un programme serveur s'exécutant sous le profil utilisateur PAYSERV extrait un message de demande d'une file d'attente qui a été placée dans la file d'attente par le profil utilisateur USER1.
- Lorsque le programme serveur obtient le message de demande, il traite la demande et insère la réponse dans la file d'attente de réponse spécifiée dans le message de demande.
- Au lieu d'utiliser son propre profil utilisateur (PAYSERV) pour autoriser l'ouverture de la file d'attente de réponse, le serveur peut spécifier un autre profil utilisateur, dans ce cas, USER1. Dans cet exemple,

vous pouvez utiliser les droits d'utilisateur de remplacement pour contrôler si PAYSERV est autorisé à spécifier USER1 comme profil d'utilisateur de remplacement lorsqu'il ouvre la file d'attente de réponse.

Le profil utilisateur de remplacement est spécifié dans la zone *AlternateUserId* du descripteur d'objet.

Remarque : Vous pouvez utiliser des profils utilisateur de remplacement sur n'importe quel objet IBM MQ . L'utilisation d'un profil utilisateur de remplacement n'affecte pas le profil utilisateur utilisé par les autres gestionnaires de ressources.

Droits d'accès au contexte

Le contexte est une information qui s'applique à un message particulier et qui est contenue dans le descripteur de message, MQMD, qui fait partie du message.

Pour la description des zones de descripteur de message relatives au contexte, voir MQMD-Descripteur de message.

Pour plus d'informations sur les options de contexte, voir Contexte de message.

Remarques relatives à la sécurité à distance

Pour la sécurité à distance, tenez compte des points suivants:

Droit d'insertion
Pour la sécurité des gestionnaires de files d'attente, vous pouvez spécifier les droits d'insertion utilisés lorsqu'un canal reçoit un message envoyé par un autre gestionnaire de files d'attente.

Ce paramètre est valide uniquement pour les types de canal RCVR, RQSTR ou CLUSRCVR. Indiquez l'attribut de canal PUTAUT comme suit:

infrastructure d"évaluation de déploiement
Profil utilisateur par défaut. Il s'agit du profil utilisateur QMQM sous lequel l'agent MCA s'exécute.

CTX
Profil utilisateur dans le contexte de message.

Files d'attente de transmission
Les gestionnaires de files d'attente placent automatiquement les messages éloignés dans une file d'attente de transmission ; aucun droit spécial n'est requis. Toutefois, l'insertion d'un message directement dans une file d'attente de transmission nécessite une autorisation spéciale.

Exits de canal
Les exits de canal peuvent être utilisés pour une sécurité accrue.

Enregistrements d'authentification de canal
Permet d'exercer un contrôle plus précis sur l'accès accordé aux systèmes de connexion au niveau d'un canal.

Pour plus d'informations sur la sécurité à distance, voir «Autorisation de canal», à la page 121.

Protection des canaux avec SSL/TLS

Le protocole TLS (Transport Layer Security) offre une sécurité de canal, avec une protection contre les écoutes clandestines, les falsifications et les usurpations d'identité. La prise en charge de TLS par IBM MQ vous permet de spécifier, dans la définition de canal, qu'un canal particulier utilise la sécurité TLS. Vous pouvez également spécifier les détails de la sécurité de votre choix, tels que l'algorithme de chiffrement que vous souhaitez utiliser.

La prise en charge de TLS dans IBM MQ utilise l' *objet d'informations d'authentification* du gestionnaire de files d'attente et diverses commandes CL et MQSC, ainsi que des paramètres de gestionnaire de files d'attente et de canal qui définissent la prise en charge de TLS requise en détail.

Les commandes CL suivantes prennent en charge TLS:

WRKMQMAUTI
Gestion des attributs d'un objet d'informations d'authentification.

CHGMQMAUTI
Modifier les attributs d'un objet d'informations d'authentification.

CRTMQMAUTI
Créez un objet d'informations d'authentification.

CPYMQMAUTI
Créez un objet d'informations d'authentification en copiant un objet existant.

DLTMQMAUTI
Supprimez un objet d'informations d'authentification.

DSPMQMAUTI
Affiche les attributs d'un objet d'informations d'authentification spécifique.

Pour une présentation de la sécurité des canaux à l'aide de TLS, voir

- Protection des canaux avec TLS

Pour plus de détails sur les commandes PCF associées à TLS, voir

- Modifier, copier et créer un objet d'informations d'authentification
- Supprimer l'objet d'informations d'authentification
- Objet d'interrogation des informations d'authentification

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 196.

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.

Table 23. RACF classes used by IBM MQ (continued)		
Member class	**Group class**	**Contents**
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 275.

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 200. 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 200.

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 201 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.

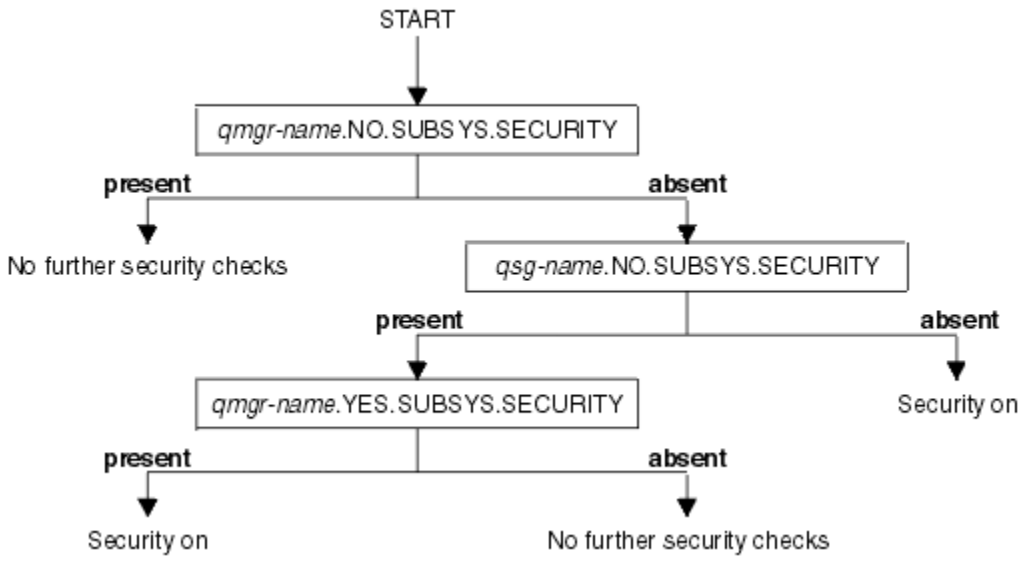


Figure 14. Checking for subsystem security

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 202 and Figure 16 on page 202 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.

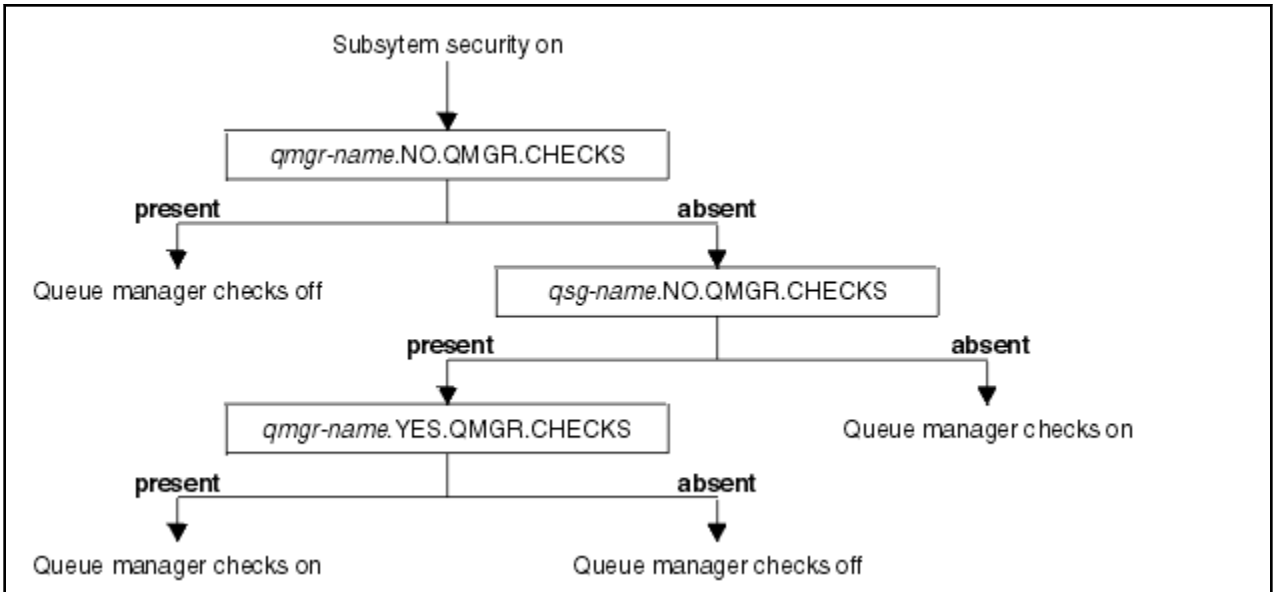


Figure 15. Checking for queue manager level security

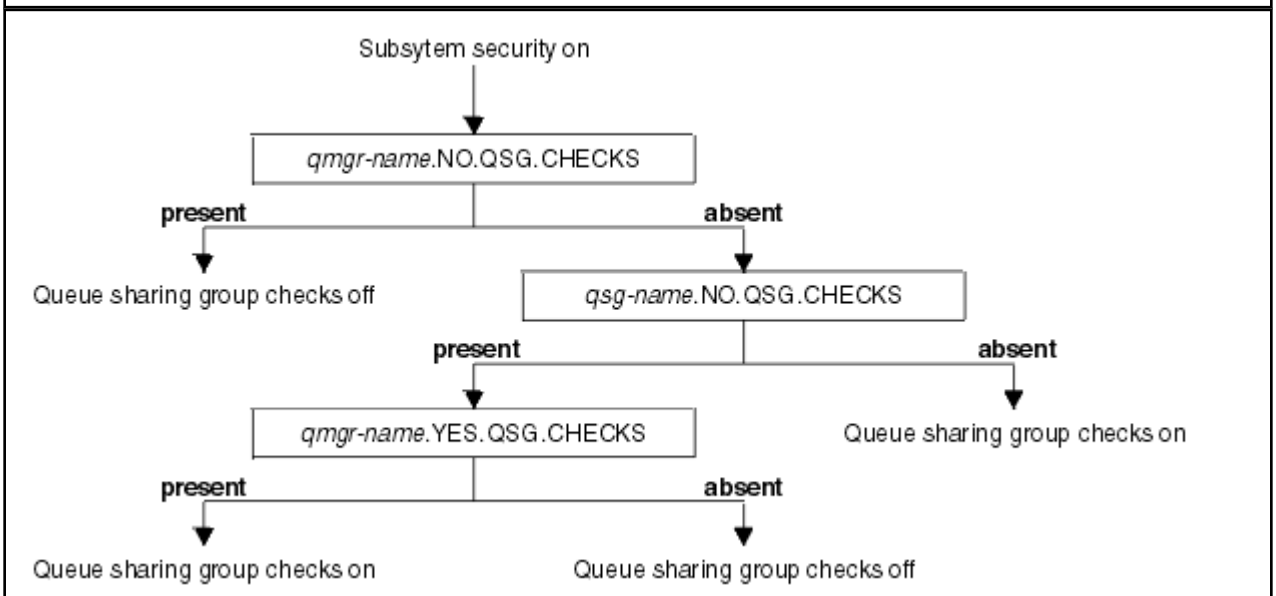


Figure 16. Checking for queue sharing group level security

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 202, Table 27 on page 203, Table 28 on page 203, and Table 29 on page 203 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

Table 26. Valid security switch combinations for queue manager level security (continued)
Combinations
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

*Table 29. Other valid security switch combinations that switch both levels of checking **on**. (continued)*
Combinations
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 204 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 257 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 246.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see “Profil de sécurité RESLEVEL” on page 240.

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 212 and “Considerations for model queues” on page 213 .

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 226 and "Profiles for alternate user security" on page 224. 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 217.

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 246 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 “Sécurité de la messagerie distante” on page 107.

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 216 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 216.

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

SYSTEM queue	CSQUTIL	CSQ0UTIL	mqweb server	Operations and control panels	Channel initiator for distributed queuing
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	UPDATE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	UPDATE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.CONTROL.QUEUE	-	-	-	-	ALTER
SYSTEM.BROKER.DEFAULT.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	-	-	-	-	UPDATE
SYSTEM.CHANNEL.INITQ	-	-	-	-	UPDATE
SYSTEM.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	-	-	ALTER

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.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 217	-	-	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				

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

	Minimum RACF access level required			
RACF class:	**MXTOPIC**	**MQQUEUE or MXQUEUE (1)**	**MQADMIN or MXADMIN**	**MQADMIN or MXADMIN**
RACF profile:	**(15 or 16)**	**(2)**	**(3)**	**(4)**
MQOO_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)**			
MQSUB option				

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)**
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 222.

When you consider alias queues used for destination queues for PUT or GET restrictions, see “Considerations for alias queues” on page 212.

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 222.

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 224.

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 246. 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 217.

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 246 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 254.

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 249 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 210), and alternate user security (see "Profiles for alternate user security" on page 224). For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 217.

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 228.

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 229 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 234 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 234	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	No check	-
ALTER QMODEL“5” on page 234	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 233	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 234	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL“5” on page 234	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 233	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 233	hlq.DISPLAY.CONN	READ	No check	-
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	No check	-
DISPLAY LOG “1” on page 233	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 233	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 233	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 233	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 233	-	-	-
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 "Sécurité de publication / abonnement" on page 500
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 237	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 237	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 237	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 “Sécurité de publication / abonnement” on page 500
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 237 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 238 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 Profil de sécurité RESLEVEL

Vous pouvez définir un profil spécial dans la classe MQADMIN ou MXADMIN pour contrôler le nombre d'ID utilisateur vérifiés pour la sécurité des ressources d'API. Ce profil est appelé profil RESLEVEL. La manière dont ce profil affecte la sécurité des ressources d'API dépend de la manière dont vous accédez à IBM MQ.

Lorsqu'une application tente de se connecter à IBM MQ, IBM MQ vérifie l'accès que l'ID utilisateur associé à la connexion a à un profil dans la classe MQADMIN ou MXADMIN appelé:

```
hlq.RESLEVEL
```

Où `hlq` peut être `ssid` (ID de sous-système) ou `qsg` (ID de groupe de partage de files d'attente).

Les ID utilisateur associés à chaque type de connexion sont les suivants:

- ID utilisateur de la tâche de connexion pour les connexions par lots
- ID utilisateur de l'espace adresse CICS pour les connexions CICS
- ID utilisateur de l'espace adresse de la région IMS pour les connexions IMS
- ID utilisateur de l'espace adresse de l'initiateur de canal pour les connexions de l'initiateur de canal

Avertissement : RESLEVEL est une option très puissante ; elle peut entraîner le contournement de toutes les vérifications de la sécurité des ressources pour une connexion particulière.

Si aucun profil RESLEVEL n'est défini, vous devez veiller à ce qu'aucun autre profil de la classe MQADMIN ne corresponde à hlq.RESLEVEL. Par exemple, si vous avez un profil dans MQADMIN appelé hlq. * * et pas de profil hlq.RESLEVEL , méfiez-vous des conséquences de hlq. * * car il est utilisé pour la vérification RESLEVEL.

Définissez un profil hlq.RESLEVEL et définissez UACC sur NONE, au lieu de n'avoir aucun profil RESLEVEL. Avoir le moins d'utilisateurs ou de groupes possible dans la liste d'accès. Pour plus de détails sur l'audit de l'accès RESLEVEL, voir «Auditing considerations on z/OS», à la page 265.

Si vous utilisez uniquement la sécurité au niveau du gestionnaire de files d'attente, IBM MQ effectue des vérifications RESLEVEL sur le profil `qmgr-name.RESLEVEL` . Si vous utilisez uniquement la sécurité au niveau du groupe de partage de files d'attente, IBM MQ effectue des vérifications RESLEVEL sur le profil `qsg-name.RESLEVEL` . Si vous utilisez une combinaison de la sécurité au niveau du gestionnaire de files d'attente et du groupe de partage de files d'attente, IBM MQ vérifie d'abord l'existence d'un profil RESLEVEL au niveau du gestionnaire de files d'attente. S'il n'en trouve pas, il recherche un profil RESLEVEL au niveau du groupe de partage de files d'attente.

S'il ne trouve pas de profil RESLEVEL, IBM MQ active la vérification de l'ID du travail et de la tâche (ou de l'utilisateur de remplacement) pour une connexion CICS ou IMS . Pour une connexion par lots, IBM MQ active la vérification de l'ID utilisateur du travail (ou d'un autre ID utilisateur). Pour l'initiateur de canal, IBM MQ active la vérification de l'ID utilisateur du canal et de l'ID utilisateur MCA (ou autre).

S'il existe un profil RESLEVEL, le niveau de vérification dépend de l'environnement et du niveau d'accès du profil.

N'oubliez pas que si votre gestionnaire de files d'attente est membre d'un groupe de partage de files d'attente et que vous ne définissez pas ce profil au niveau du gestionnaire de files d'attente, il se peut qu'un profil défini au niveau du groupe de partage de files d'attente affecte le niveau de vérification. Pour activer la vérification de deux ID utilisateur, vous devez définir un profil RESLEVEL (précédé du nom de gestionnaire de files d'attente du nom de groupe de partage de files d'attente) avec un UACC (NONE) et vous assurer que les utilisateurs concernés ne disposent pas des droits d'accès à ce profil.

Lorsque vous prenez en compte l'accès de l'ID utilisateur de l'initiateur de canal à RESLEVEL, n'oubliez pas que la connexion établie par l'initiateur de canal est également la connexion utilisée par les canaux. Un paramètre qui provoque le contournement de tous les contrôles de sécurité de ressource pour l'ID utilisateur de l'initiateur de canal ignore les contrôles de sécurité pour tous les canaux. Si l'accès de l'ID utilisateur de l'initiateur de canal à RESLEVEL est différent de NONE, un seul ID utilisateur (pour un niveau d'accès READ ou UPDATE) ou aucun ID utilisateur (pour un niveau d'accès CONTROL ou ALTER) est vérifié pour l'accès. Si vous accordez à l'ID utilisateur de l'initiateur de canal un niveau d'accès autre que NONE à RESLEVEL, assurez-vous que vous comprenez l'effet de ce paramètre sur les contrôles de sécurité effectués pour les canaux.

L'utilisation du profil RESLEVEL signifie que les enregistrements d'audit de sécurité normaux ne sont pas utilisés. Par exemple, si vous placez UAUDIT sur un utilisateur, l'accès au profil hlq.RESLEVEL dans MQADMIN n'est pas audité.

Si vous utilisez l'option RACF WARNING sur le profil hlq.RESLEVEL , aucun message d'avertissement RACF n'est généré pour les profils de la classe RESLEVEL.

La vérification de la sécurité pour les messages de rapport tels que les COD est contrôlée par le profil RESLEVEL associé à l'application d'origine. Par exemple, si l'ID utilisateur d'un travail par lots dispose des droits CONTROL ou ALTER sur un profil RESLEVEL, toutes les vérifications de ressources effectuées par le travail par lots sont ignorées, y compris la vérification de la sécurité des messages de rapport.

Si vous modifiez le profil RESLEVEL, les utilisateurs doivent se déconnecter et se reconnecter avant que la modification n'ait lieu. (Cela inclut l'arrêt et le redémarrage de l'initiateur de canal si l'accès de l'ID utilisateur de l'espace adresse de mise en file d'attente répartie au profil RESLEVEL est modifié.)

Pour désactiver l'audit RESLEVEL, utilisez le paramètre système RESAUDIT.

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 242. 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 216, 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 243 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 249 for more information.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested, and how many are checked.

The following table shows the checks made for the channel initiator's connection, and for all channels since they use this connection.

Table 55. Checks made at different RACF access levels for channel initiator connections

RACF access level	Level of checking
NONE	Check two user IDs.
READ	Check one user ID.
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
Note: See “User IDs used by the channel initiator” on page 249 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 253 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 253 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.
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.

Issued from...	User ID contents
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 243 in topic “RESLEVEL and CICS connections” on page 242, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 248 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.queuenam e profile	hlq.resourcename profile
No, 1 check	-	REG	REG
No, 2 checks	-	REG+SEC	REG+SEC
Yes, 1 check	REG	REG	REG

Table 59. User ID checking against profile name for IMS-type user IDs (continued)			
Alternate user ID specified on open?	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
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 249.

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

Table 61. User IDs checked against profile name for TCP/IP channels (continued)			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space of the receiver or requester end is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

Note: The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the receiver or requester is used. This also applies to TCP/IP channels using TLS.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an **MQOPEN** or **MQPUT1** call is issued for the target destination queue.

z/OS *Receiving channels using LU 6.2*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 62. User IDs checked against profile name for LU 6.2 channels			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT

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**
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 “Contrôle d'accès pour les clients” on page 109 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 244 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 249 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 258 summarizes the situations in which security information is cached and in which cached information is used.

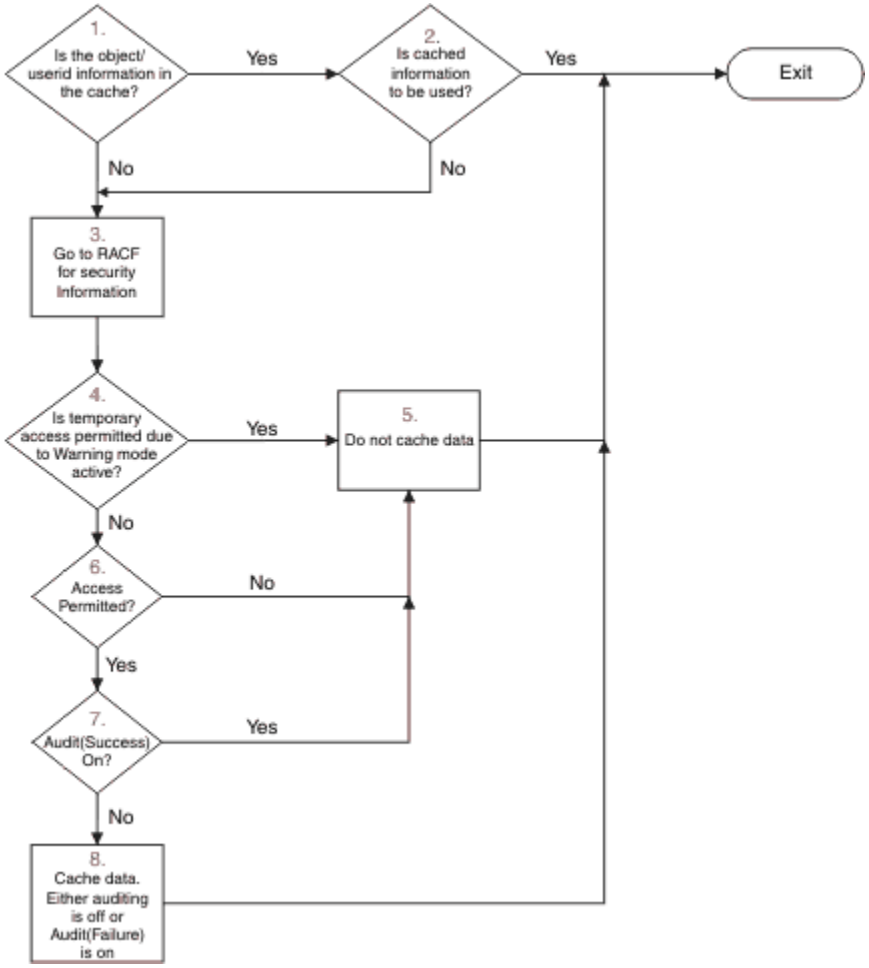


Figure 17. Logic flow for IBM MQ security caching

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 261 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 261 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 268, 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

Pour exercer un contrôle plus précis sur les accès accordés aux systèmes en cours de connexion au niveau d'un canal, vous pouvez utiliser les enregistrements d'authentification de canal.

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 266.

```
      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 226.
- 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 216, and to all the user destination queues and the dead-letter queue (but see “Dead-letter queue security” on page 215).

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 226 "RESLEVEL and the channel initiator connection" on page 244 and "User IDs for security checking on z/OS" on page 246 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 209.

Data sets
The channel initiator address space needs appropriate access to queue manager data sets, see "Authorizing access to data sets" on page 261.

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 229.

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 246 for more information.

You can also use the Transport Layer Security (TLS) protocol to provide security on channels. See "Protocoles de sécurité TLS dans IBM MQ" on page 25 for more information about using TLS with IBM MQ.

See also "Contrôle d'accès pour les clients" on page 109 for information about server-connection security.

User IDs
The user IDs described in "User IDs used by the channel initiator" on page 249 and "User IDs used by the intra-group queuing agent" on page 253 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 268:

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 229) 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 274.
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.

Configuration de la sécurité IBM MQ MQI client

Vous devez prendre en compte la sécurité IBM MQ MQI client pour que les applications client n'aient pas un accès illimité aux ressources sur le serveur.

Lors de l'exécution d'une application client, n'exécutez pas l'application à l'aide d'un ID utilisateur disposant de droits d'accès plus nombreux que nécessaire ; par exemple, un utilisateur du groupe `mqm` ou même l'utilisateur `mqm` lui-même.

En exécutant une application en tant qu'utilisateur disposant de trop de droits d'accès, vous risquez que l'application accède à des parties du gestionnaire de files d'attente et les modifie, soit par accident, soit par malveillance.

Il existe deux aspects de la sécurité entre une application client et son serveur de gestionnaire de files d'attente: l'authentification et le contrôle d'accès.

- L'authentification peut être utilisée pour s'assurer que l'application client, exécutée en tant qu'utilisateur spécifique, est bien celle qu'elle dit être. En utilisant l'authentification, vous pouvez empêcher un agresseur d'accéder à votre gestionnaire de files d'attente en empruntant l'une de vos applications.

  L'authentification est fournie par l'une des deux options suivantes:

  – La fonction d'authentification de connexion.

    Pour plus d'informations sur l'authentification de connexion, voir «Authentification de connexion», à la page 75.

  – Utilisation de l'authentification mutuelle dans TLS.

    Pour plus d'informations sur TLS, voir «Utilisation de SSL/TLS», à la page 283.

- Le contrôle d'accès peut être utilisé pour accorder ou supprimer des droits d'accès pour un utilisateur ou un groupe d'utilisateurs spécifique. En exécutant une application client avec un utilisateur créé spécifiquement (ou un utilisateur appartenant à un groupe spécifique), vous pouvez ensuite utiliser des contrôles d'accès pour vous assurer que l'application ne peut pas accéder à des parties de votre gestionnaire de files d'attente auxquelles l'application n'est pas censée accéder.

  Lors de la configuration du contrôle d'accès, vous devez tenir compte des règles d'authentification de canal et de la zone MCAUSER sur un canal. Ces deux fonctions permettent de modifier l'ID utilisateur utilisé pour vérifier les droits de contrôle d'accès.

  Pour plus d'informations sur le contrôle d'accès, voir «Autorisation de l'accès aux objets», à la page 363.

Si vous avez configuré une application client pour qu'elle se connecte à un canal spécifique avec un ID restreint, mais que le canal possède un ID administrateur défini dans sa zone MCAUSER, à condition que l'application client se connecte correctement, l'ID administrateur est utilisé pour les vérifications de contrôle d'accès. Par conséquent, l'application client dispose de droits d'accès complets à votre gestionnaire de files d'attente.

Pour plus d'informations sur l'attribut MCAUSER, voir «Mappage d'un ID utilisateur client à un ID utilisateur MCAUSER», à la page 400.

Les règles d'authentification de canal peuvent également être utilisées comme méthode de contrôle de l'accès à un gestionnaire de files d'attente, en définissant des règles et des critères spécifiques pour l'acceptation d'une connexion.

Pour plus d'informations sur les règles d'authentification de canal, voir: «Enregistrements d'authentification de canal», à la page 54.

Comment indiquer que seuls les CipherSpecs certifiés FIPS sont utilisés lors de l'exécution sur MQI Client

Créez vos référentiels de clés à l'aide d'un logiciel compatible FIPS, puis indiquez que le canal doit utiliser des CipherSpecscertifiés FIPS.

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

Pour être conformes à la norme FIPS lors de l'exécution, les référentiels de clés doivent avoir été créés et gérés à l'aide de logiciels conformes à la norme FIPS tels que **runmqakm** avec l'option `-fips` .

Vous pouvez spécifier qu'un canal TLS doit utiliser uniquement des CipherSpecs certifiés FIPS de trois manières, répertoriées par ordre de priorité:

1. Définissez la zone FipsRequired dans la structure MQSCO sur `MQSSL_FIPS_YES`.
2. Définissez la variable d'environnement **MQSSLFIPS** sur YES.
3. Définissez l'attribut **SSLFipsRequired** dans la section SSL du fichier de configuration du client sur YES.

Par défaut, les CipherSpecs certifiés FIPS ne sont pas requis.

Ces valeurs ont la même signification que les valeurs de paramètre équivalentes sous **ALTER QMGR SSLFIPS** (voir **ALTER QMGR** (modification des paramètres du gestionnaire de files d'attente)). Si le processus client ne possède actuellement aucune connexion TLS active et qu'une valeur FipsRequired est correctement spécifiée sur un MQCONNX SSL, toutes les connexions TLS suivantes associées à ce processus doivent utiliser uniquement les CipherSpecs associées à cette valeur. Cela s'applique jusqu'à l'arrêt de cette connexion et de toutes les autres connexions TLS, auquel cas une connexion MQCONNX ultérieure peut fournir une nouvelle valeur pour FipsRequired.

Si du matériel de cryptographie est présent, les modules de cryptographie utilisés par IBM MQ peuvent être configurés pour être ceux fournis par le produit matériel, et ceux-ci peuvent être certifiés FIPS à un niveau particulier. Les modules configurables et leur certification FIPS dépendent du produit matériel utilisé.

Dans la mesure du possible, si des CipherSpecs FIPS uniquement sont configurés, le client MQI rejette les connexions qui spécifient un CipherSpec non FIPS avec `MQRC_SSL_INITIALIZATION_ERROR`. IBM MQ ne garantit pas le rejet de toutes ces connexions et il vous incombe de déterminer si votre configuration IBM MQ est conforme à la norme FIPS.

Concepts associés

«FIPS (Federal Information Processing Standards) pour AIX, Linux, and Windows», à la page 36
Lorsque la cryptographie est requise sur un canal SSL/TLS sur des systèmes AIX, Linux, and Windows , IBM MQ utilise un package de cryptographie appelé IBM Crypto for C (ICC). Sur les plateformes AIX, Linux, and Windows , le logiciel ICC a transmis le programme de validation Cryptomodule FIPS (Federal Information Processing Standards) de l'Institut national des normes et de la technologie des Etats-Unis, au niveau 140-2.

AIX Exécution d'applications client TLS avec plusieurs installations de GSKit 8.0 sur AIX

Les applications client TLS sous AIX peuvent rencontrer des `MQRC_CHANNEL_CONFIG_ERROR` et des erreurs `AMQ6175` lors de l'exécution sur des systèmes AIX avec plusieurs installations IBM Global Security Kit (GSKit) 8.0 .

Lors de l'exécution d'applications client sur un système AIX avec plusieurs installations GSKit 8.0 , les appels de connexion client peuvent renvoyer `MQRC_CHANNEL_CONFIG_ERROR` lors de l'utilisation de TLS. `/var/mqm/errors` consigne l'erreur d'enregistrement `AMQ6175` et `AMQ9220` pour l'application client défaillante, par exemple:

```
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 ----------------------------------------------------------
```

Une cause courante de cette erreur est que le paramètre de la variable d'environnement LIBPATH ou LD_LIBRARY_PATH a entraîné le client IBM MQ à charger un ensemble mixte de bibliothèques à partir de deux installations GSKit 8.0 différentes. L'exécution d'une application client IBM MQ dans un environnement Db2 peut provoquer cette erreur.

Pour éviter cette erreur, incluez les répertoires de la bibliothèque IBM MQ à l'avant du chemin de la bibliothèque afin que les bibliothèques IBM MQ soient prioritaires. Pour ce faire, utilisez la commande **setmqenv** avec le paramètre **-k** , par exemple:

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

Pour plus d'informations sur l'utilisation de la commande **setmqenv** , voir setmqenv (set IBM MQ environment)

Configuration des canaux TLS avec MQSC

Pour configurer des canaux TLS, utilisez les commandes **runmqsc** et ALTER CHANNEL. Si vous le souhaitez, vous pouvez configurer un canal afin qu'il n'accepte que les certificats dont les attributs dans le nom distinctif du propriétaire correspondent aux valeurs données. Vous pouvez également configurer un canal de gestionnaire de files d'attente pour que ce dernier refuse la connexion si la partie initialisante n'envoie pas son propre certificat personnel.

Pourquoi et quand exécuter cette tâche

Pour configurer des canaux dans IBM MQ Explorer, voir Configuration de canaux TLS avec IBM MQ Explorer.

Pour configurer des canaux à l'aide de **runmqsc**, procédez comme suit.

Procédure

1. Appelez la commande **runmqsc** en vous connectant au gestionnaire de files d'attente cible.
2. Identifiez le canal que vous souhaitez activer pour TLS.

Notez le nom et le type de canal.

3. La commande ALTER CHANNEL permet de modifier diverses propriétés d'un canal IBM MQ .

   Vous fournissez le nom et le type de canal en plus de la commande. Par exemple, pour modifier un canal émetteur appelé MQ.TEST exécutez la commande suivante:

   ```
   ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR)
   ```

   Il existe différents attributs de canal liés à TLS que vous pouvez ajuster sur les définitions de canal IBM MQ .

Que faire ensuite

Configuration de la sécurité des messages

La messagerie compatible avec TLS propose deux méthodes pour sécuriser les messages :

- Le chiffrement, qui assure que le message ne pourra être lu, même s'il est intercepté.
- Les fonctions de hachage qui assurent que toute modification du message sera détectée.

La combinaison de ces méthodes est appelée CipherSpec, ou spécification de chiffrement. Il faut que le même CipherSpec soit défini pour chaque extrémité d'un canal ; sinon la messagerie TLS ne peut pas fonctionner. Pour plus d'informations, voir «Sécurisation de IBM MQ», à la page 7.

Pour modifier un protocole TLS d'activation de canal IBM MQ , spécifiez une valeur dans l'attribut SSLCIPH. Cet attribut doit être défini sur un CipherSpec valide pour la plateforme de file d'attente du gestionnaire de files d'attente dans la liste «Activation des CipherSpecs», à la page 435.

Pour modifier un canal IBM MQ afin de désactiver TLS, définissez SSLCIPH sur une valeur vide. Exemple :

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

Remarque : Vous devez placer le nom du canal entre apostrophes pour vous assurer que la casse des caractères est respectée. Sans apostrophes, IBM MQ transforme la chaîne en majuscules.

Filtrage de certificats selon le nom de leur propriétaire

Les certificats contiennent le nom distinctif du propriétaire du certificat. Si vous le souhaitez, vous pouvez configurer le canal afin qu'il n'accepte que les certificats dont les attributs dans le nom distinctif du propriétaire correspondent aux valeurs données.

Les noms d'attributs qu'IBM MQ peut filtrer figurent dans le tableau ci-après :

Noms d'attributs	**Explication**
SERIALNUMBER	Numéro de série du certificat
MAIL	Adresse électronique
Deprecated E	Adresse électronique (dépréciée dans la préférence pour MAIL)
UID ou USERID	ID utilisateur
CN	Nom CN
T	Titre
OU	Nom d'unité organisationnelle
DC	Composant de domaine
O	Nom de l'organisation
STREET	Rue/Première ligne d'adresse
L	Nom du lieu

Noms d'attributs	Explication
ST (ou SP ou S)	Nom du département
ordinateur personnel	Code postal
C	Pays
UNSTRUCTUREDNAME	Nom d'hôte
UNSTRUCTUREDADDRESS	Adresse IP
DNQ	Qualificateur de nom distinctif

Vous pouvez utiliser le caractère générique (*) au début ou à la fin de la valeur d'attribut à la place d'un nombre quelconque de caractères. Par exemple, pour n'accepter de certificats qu'en provenance de tout utilisateur dont le nom se termine par `Smith` et travaillant pour `IBM` dans `GB`, tapez :

```
CN=*Smith, O=IBM, C=GB
```

Exemple :

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

Remarque : Vous devez placer la chaîne SSLPEER entre guillemets simples pour vous assurer que la casse des caractères est conservée. Sans apostrophes, IBM MQ transforme la chaîne en majuscules.

Authentification des parties initialisant des connexions à un gestionnaire de files d'attente

Lorsqu'une autre partie initialise une connexion TLS sur un gestionnaire de files d'attente, celui-ci doit lui envoyer son certificat personnel comme preuve d'identité. Vous pouvez également configurer le canal de gestionnaire de files d'attente pour que ce dernier refuse la connexion si la partie initialisante n'envoie pas son propre certificat personnel.

Pour ce faire, définissez l'attribut SSLCAUTH. Cet attribut est un attribut booléen et peut avoir les valeurs OPTIONAL ou REQUIRED:

- La valeur OPTIONAL authentifie le certificat d'un client de connexion, s'il en existe un, mais n'exige pas qu'un client en envoie un. Un client est rejeté s'il envoie un certificat non valide.
- REQUIRED rejette tous les clients qui se connectent et qui ne fournissent pas de certificat TLS valide

Exemple :

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

IBM i Configuration des communications pour SSL ou TLS sur IBM i

Les communications sécurisées qui font appel aux protocoles de sécurité cryptographiques SSL ou TLS impliquent la configuration des canaux de communication et la gestion des certificats numériques à utiliser à des fins d'authentification.

Pour configurer votre installation SSL ou TLS, vous devez définir vos canaux pour utiliser les protocoles SSL ou TLS. Vous devez également créer et gérer vos certificats numériques. Sur certains systèmes d'exploitation, vous pouvez effectuer les tests avec des certificats autosignés. Toutefois, sous IBM i, vous devez utiliser des certificats personnels signés par une autorité de certification locale.

Pour plus d'informations sur la création et la gestion des certificats, voir «Utilisation de SSL/TLS sous IBM i», à la page 283.

Cette collection de rubriques présente certaines des tâches impliquées dans la configuration des communications SSL ou TLS et fournit des conseils étape par étape sur l'exécution de ces tâches.

Vous pouvez également tester l'authentification de client SSL ou TLS, qui sont des parties facultatives des protocoles SSL et TLS. Lors de l'établissement de liaison SSL ou TLS, le client SSL ou TLS obtient toujours

un certificat numérique du serveur et le valide. Avec l'implémentation de IBM MQ, le serveur SSL ou TLS demande toujours un certificat au client.

Sous IBM i, le client SSL ou TLS envoie un certificat uniquement s'il en comporte un libellé au format IBM MQ correct:

- Pour un gestionnaire de files d'attente, `ibmwebspheremq` suivi du nom de votre gestionnaire de files d'attente est remplacé par des minuscules. Par exemple, pour `QM1`, `ibmwebspheremqqm1`.
- Pour un IBM MQ C Client for IBM i, `ibmwebspheremq` suivi de votre ID utilisateur de connexion remplacé par des minuscules, par exemple `ibmwebspheremqmyuserid`.

IBM MQ utilise le préfixe `ibmwebspheremq` sur un libellé pour éviter toute confusion avec les certificats d'autres produits. Veillez à spécifier l'intégralité du libellé de certificat en minuscules.

Le serveur SSL ou TLS valide toujours le certificat client si celui-ci est envoyé. Si le client SSL ou TLS n'envoie pas de certificat, l'authentification échoue uniquement si l'extrémité du canal agissant en tant que serveur SSL ou TLS est définie avec le paramètre SSLCAUTH défini sur REQUIRED ou une valeur de paramètre SSLPEER définie. Pour plus d'informations, voir Connexion de deux gestionnaires de files d'attente à l'aide de SSL ou TLS.

ALW Configuration des communications pour SSL ou TLS sur AIX, Linux, and Windows

Les communications sécurisées qui font appel aux protocoles de sécurité cryptographiques SSL ou TLS impliquent la configuration des canaux de communication et la gestion des certificats numériques à utiliser à des fins d'authentification.

Pour configurer votre installation SSL ou TLS, vous devez définir vos canaux pour utiliser les protocoles SSL ou TLS. Vous devez également créer et gérer vos certificats numériques. Sur les systèmes AIX, Linux, and Windows , vous pouvez effectuer les tests avec des certificats autosignés.

Avertissement : Il n'est pas possible d'utiliser un mélange de certificats signés par Elliptic Curve et de certificats signés par RSA sur les gestionnaires de files d'attente que vous souhaitez joindre à l'aide de canaux activés par TLS.

Les gestionnaires de files d'attente utilisant des canaux TLS activés doivent tous utiliser des certificats signés par RSA ou tous utiliser des certificats signés par EC, et non les deux.

Pour plus d'informations, voir «Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49.

Les certificats autosignés ne peuvent pas être révoqués, ce qui pourrait permettre à un agresseur de usurper une identité après qu'une clé privée a été compromise. Les autorités de certification peuvent révoquer un certificat compromis, pour en empêcher toute utilisation future. Les certificats signés par une autorité de certification sont donc plus sûrs à utiliser dans un environnement de production, bien que les certificats autosignés soient plus pratiques pour un système de test.

Pour plus d'informations sur la création et la gestion des certificats, voir «Utilisation de SSL/TLS sous AIX, Linux, and Windows», à la page 302.

Cette collection de rubriques présente certaines des tâches liées à la configuration des communications SSL et fournit des conseils étape par étape sur l'exécution de ces tâches.

Vous pouvez également vouloir tester l'authentification des clients SSL ou TLS, qui constitue une partie facultative des protocoles. Lors de l'établissement de liaison SSL ou TLS, le client SSL ou TLS obtient toujours un certificat numérique du serveur et le valide. Avec l'implémentation de IBM MQ, le serveur SSL ou TLS demande toujours un certificat au client.

Sous AIX, Linux, and Windows, le client SSL ou TLS envoie un certificat uniquement s'il en a un libellé au format IBM MQ correct:

- Pour un gestionnaire de files d'attente, le format est `ibmwebspheremq` suivi du nom de votre gestionnaire de files d'attente remplacé par des minuscules. Par exemple, pour `QM1`, `ibmwebspheremqqm1`

- Pour un client IBM MQ , `ibmwebspheremq` suivi de votre ID utilisateur de connexion est passé en minuscules, par exemple `ibmwebspheremqmyuserid`.

IBM MQ utilise le préfixe `ibmwebspheremq` sur un libellé pour éviter toute confusion avec les certificats d'autres produits. Veillez à spécifier l'intégralité du libellé de certificat en minuscules.

Le serveur SSL ou TLS valide toujours le certificat client si celui-ci est envoyé. Si le client n'envoie pas de certificat, l'authentification échoue uniquement si l'extrémité du canal agissant en tant que serveur SSL ou TLS est définie avec le paramètre SSLCAUTH défini sur REQUIRED ou une valeur de paramètre SSLPEER définie. Pour plus d'informations, voir Connexion de deux gestionnaires de files d'attente à l'aide de SSL ou TLS.

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 317.

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.

Utilisation de SSL/TLS

Ces rubriques fournissent des instructions pour l'exécution de tâches uniques liées à l'utilisation de TLS avec IBM MQ.

La plupart d'entre eux sont utilisés comme étapes dans les tâches de niveau supérieur décrites dans les sections suivantes:

- «Identification et authentification des utilisateurs», à la page 329
- «Autorisation de l'accès aux objets», à la page 363
- «Confidentialité des messages», à la page 434
- «Intégrité des données de messages», à la page 491
- «Maintien de la sécurité des clusters», à la page 492

IBM i Utilisation de SSL/TLS sous IBM i

Cette collection de rubriques fournit des instructions pour les tâches individuelles utilisant le protocole TLS (Transport Layer Security) dans IBM MQ for IBM i.

Pour IBM i , la prise en charge de TLS fait partie intégrante du système d'exploitation. Vérifiez que vous avez installé les prérequis répertoriés dans Configuration matérielle et logicielle requise sur IBM i.

Sous IBM i, vous gérez les clés et les certificats numériques à l'aide de l'outil Digital Certificate Manager (DCM).

Accès à DCM

Suivez ces instructions pour accéder à l'interface DCM.

Pourquoi et quand exécuter cette tâche

Effectuez les étapes suivantes dans un navigateur Web qui prend en charge les cadres.

Procédure

1. Accédez à `http://machine.domain:2001` ou à `https://machine.domain:2010`, où *machine* est le nom de votre ordinateur.
2. Entrez un profil utilisateur et un mot de passe valides lorsque vous y êtes invité.

   Vérifiez que votre profil utilisateur dispose des droits spéciaux *ALLOBJ et *SECADM pour vous permettre de créer de nouveaux magasins de certificats. Si vous ne disposez pas des droits spéciaux, vous pouvez uniquement gérer vos certificats personnels ou afficher les signatures d'objet pour les objets pour lesquels vous disposez de droits. Si vous êtes autorisé à utiliser une application de signature d'objet, vous pouvez également signer des objets à partir de DCM.
3. Sur la page Configurations Internet, cliquez sur **Gestionnaire de certificats Certificate Manager**.

   La page Certificate Manager numérique s'affiche.

Affectation d'un certificat à un gestionnaire de files d'attente sous IBM i

Utilisez DCM pour affecter un certificat à un gestionnaire de files d'attente.

Utilisez la gestion des certificats numériques IBM i pour affecter un certificat à un gestionnaire de files d'attente. Cela signifie que vous pouvez spécifier qu'un gestionnaire de files d'attente utilise le magasin de certificats de système et que le gestionnaire de files d'attente est enregistré pour être utilisé en tant qu'application avec le Certificate Managernumérique. Pour ce faire, remplacez la valeur de l'attribut **SSLKEYR** du gestionnaire de files d'attente par `*SYSTEM`.

Lorsque le paramètre **SSLKEYR** est remplacé par `*SYSTEM`, IBM MQ enregistre le gestionnaire de files d'attente en tant qu'application serveur avec un libellé d'application unique de QIBM_WEBSPHERE_MQ_QMGRNAME et un libellé avec une description de Qmgrname (WMQ). Notez que les attributs de canal **CERTLABL** ne sont pas utilisés si vous utilisez l'espace de stockage de certificats *SYSTEM. Le gestionnaire de files d'attente apparaît alors en tant qu'application serveur dans

le Certificate Managernumérique et vous pouvez affecter à cette application n'importe quel certificat serveur ou client dans le magasin système.

Etant donné que le gestionnaire de files d'attente est enregistré en tant qu'application, des fonctions avancées de DCM, telles que la définition de listes de confiance de l'autorité de certification, peuvent être exécutées.

Si le paramètre **SSLKEYR** est remplacé par une valeur autre que *SYSTEM, IBM MQ désenregistre le gestionnaire de files d'attente en tant qu'application avec Digital Certificate Manager. Si un gestionnaire de files d'attente est supprimé, il est également désenregistré de DCM. Un utilisateur disposant des droits *SECADM suffisants peut également ajouter ou supprimer manuellement des applications dans DCM.

Configuration d'un référentiel de clés sur IBM i

Un référentiel de clés doit être configuré aux deux extrémités de la connexion. Les magasins de certificats par défaut peuvent être utilisés ou vous pouvez créer les vôtres.

Une connexion TLS requiert un *référentiel de clés* à chaque extrémité de la connexion. Chaque gestionnaire de files d'attente et IBM MQ MQI client doivent avoir accès à un référentiel de clés. Si vous souhaitez accéder au référentiel de clés à l'aide d'un nom de fichier et d'un mot de passe (c'est-à-dire sans utiliser l'option *SYSTEM), vérifiez que le profil utilisateur QMQM dispose des droits suivants:

- Droit d'exécution sur le répertoire contenant le référentiel de clés
- Droits de lecture sur le fichier contenant le référentiel de clés

Pour plus d'informations, voir «Référentiel de clés SSL/TLS», à la page 26. Notez que les attributs **CERTLABL** de canal ne sont pas utilisés si vous utilisez l'espace de stockage de certificats *SYSTEM.

Sous IBM i, les certificats numériques sont stockés dans un magasin de certificats géré avec DCM. Ces certificats numériques comportent des libellés qui associent un certificat à un gestionnaire de files d'attente ou à un IBM MQ MQI client. TLS utilise les certificats à des fins d'authentification.

Le libellé est soit la valeur de l'attribut **CERTLABL** , s'il est défini, soit la valeur par défaut `ibmwebspheremq` avec le nom du gestionnaire de files d'attente ou l'ID de connexion de l'utilisateur IBM MQ MQI client ajouté, le tout en minuscules. Pour plus de détails voir Labels de certificat numérique.

Le nom du gestionnaire de files d'attente ou du magasin de certificats IBM MQ MQI client comprend un chemin et un nom de radical. Le chemin par défaut est `/QIBM/UserData/ICSS/Cert/Server/` et le nom de radical par défaut est `Default`. Sous IBM i, le magasin de certificats par défaut, `/QIBM/UserData/ICSS/Cert/Server/Default.kdb`, est également appelé *SYSTEM. Si vous le souhaitez, vous pouvez définir votre propre chemin et nom de radical.

Si vous définissez votre propre chemin ou nom de fichier, définissez les droits d'accès au fichier pour contrôler étroitement son accès.

«Modification de l'emplacement du référentiel de clés pour un gestionnaire de files d'attente sous IBM i», à la page 287 vous indique comment spécifier le nom du magasin de certificats. Vous pouvez spécifier le nom du magasin de certificats avant ou après la création du magasin de certificats.

Remarque : Les opérations que vous pouvez effectuer avec DCM peuvent être limitées par les droits de votre profil utilisateur. Par exemple, vous avez besoin des droits *ALLOBJ et *SECADM pour créer un certificat d'autorité de certification.

IBM i *Chiffrement des mots de passe du référentiel de clés sous IBM i*

Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

Les composants et fonctions IBM MQ suivants prennent en charge deux méthodes différentes de stockage des mots de passe de référentiel de clés:

- Référentiel de clés TLS du gestionnaire de files d'attente.

- IBM MQ MQI clients qui utilisent TLS.

Les mots de passe de référentiel de clés à utiliser par ces composants sont protégés à l'aide du système de protection par mot de passe IBM MQ . Le mécanisme permettant de fournir un mot de passe et de le chiffrer varie légèrement en fonction du composant:

Référentiel de clés TLS du gestionnaire de files d'attente

Le mot de passe est chiffré lorsque l'attribut de gestionnaire de files d'attente **SSLKEYRPWD** est défini à l'aide de la commande CHGMQM (Change Message Queue Manager) .

Le mot de passe est chiffré avec l'algorithme AES-128 . Les détails de cet algorithme sont connus du public et sont considérés comme sécurisés.

Le mot de passe est stocké dans un fichier de dissimulation dans un format propriétaire qui n'est pas compris par les autres logiciels pouvant accéder au référentiel de clés.

Un mot de passe chiffré par un composant IBM MQ ne peut pas être utilisé par un autre composant IBM MQ .

Une clé de chiffrement unique peut être fournie lorsque le mot de passe du référentiel de clés est chiffré. Une clé de chiffrement unique empêche toute personne qui n'a pas accès à la clé de chiffrement de pouvoir déchiffrer le mot de passe. Vous fournissez cette clé via l'attribut de gestionnaire de files d'attente **INITKEY** , qui doit être défini avant de fournir un mot de passe à chiffrer.

Pour plus d'informations sur le système de protection par mot de passe IBM MQ , voir «Protection des mots de passe dans les fichiers de configuration du composant IBM MQ», à la page 582.

IBM MQ MQI clients qui utilisent TLS

Le «IBM MQ Utilitaire de client SSL (amqrsslc) pour IBM i», à la page 299 peut stocker le mot de passe du référentiel de clés dans un fichier de dissimulation. Voir aussi Administration à l'aide de commandes MQSC sous IBM i.

Le mot de passe est chiffré avec l'algorithme AES-128 . Les détails de cet algorithme sont connus du public et sont considérés comme sécurisés.

Le mot de passe est stocké dans un fichier de dissimulation dans un format propriétaire qui n'est pas compris par les autres logiciels pouvant accéder au référentiel de clés.

Une clé de chiffrement unique peut être fournie lorsque le mot de passe du référentiel de clés est chiffré. Une clé de chiffrement unique empêche toute personne qui n'a pas accès à la clé de chiffrement de pouvoir déchiffrer le mot de passe. Vous fournissez cette clé via le paramètre **-sf** .

Le mot de passe chiffré est stocké dans un fichier de dissimulation dans le même répertoire que le fichier de référentiel de clés.

IBM MQ MQI clients prend également en charge les mots de passe fournis via d'autres mécanismes. Voir «Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur IBM i», à la page 289.

Quelle que soit la méthode que vous choisissez pour chiffrer le mot de passe du référentiel de clés, veillez à connaître les limitations du chiffrement des mots de passe stockés. Voir «Limites de la protection via le chiffrement de mot de passe», à la page 589.

Concepts associés

«Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous IBM i», à la page 288
Etant donné que le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

«Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur IBM i», à la page 289

Etant donné que le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

«Utilisation de SSL/TLS sous IBM i», à la page 283
Cette collection de rubriques fournit des instructions pour les tâches individuelles utilisant le protocole TLS (Transport Layer Security) dans IBM MQ for IBM i.

Création d'un magasin de certificats sous IBM i
Si vous ne souhaitez pas utiliser le magasin de certificats par défaut, suivez cette procédure pour créer le vôtre.

Pourquoi et quand exécuter cette tâche

Créez un nouveau magasin de certificats uniquement si vous ne souhaitez pas utiliser le magasin de certificats par défaut IBM i .

Pour indiquer que le magasin de certificats de système IBM i doit être utilisé, remplacez la valeur de l'attribut SSLKEYR du gestionnaire de files d'attente par *SYSTEM. Cette valeur indique que le gestionnaire de files d'attente utilise le magasin de certificats du système et que le gestionnaire de files d'attente est enregistré pour être utilisé en tant qu'application avec Digital Certificate Manager (DCM).

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283
2. Dans le panneau de navigation, cliquez sur **Créer un nouveau magasin de certificats**.
   La page Créer un nouveau magasin de certificats s'affiche dans le cadre de la tâche.
3. Dans le cadre de la tâche, sélectionnez **Autre magasin de certificats de système** et cliquez sur **Continuer**.
   La page Créer un certificat dans le nouveau magasin de certificats s'affiche dans le cadre de la tâche.
4. Sélectionnez **Non-Ne pas créer de certificat dans le magasin de certificats** et cliquez sur **Continuer**.
   La page Nom et mot de passe du magasin de certificats s'affiche dans le cadre de la tâche.
5. Dans la zone **Chemin d'accès au magasin de certificats et nom de fichier** , entrez un chemin d'accès au système de fichiers intégré et un nom de fichier, par exemple `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`
6. Entrez un mot de passe dans la zone **Mot de passe** et entrez-le à nouveau dans la zone **Confirmer le mot de passe** . Cliquez sur **Continue**.
   Notez le mot de passe (qui est sensible à la casse) car vous en avez besoin lorsque vous stockez la clé de référentiel.
7. Pour quitter DCM, fermez la fenêtre de votre navigateur.

Que faire ensuite

Une fois que vous avez créé le magasin de certificats à l'aide de DCM, veillez à stocker le mot de passe, comme décrit dans «Stockage du mot de passe du magasin de certificats sur les systèmes IBM i», à la page 286

Tâches associées

«Importation d'un certificat dans un référentiel de clés sous IBM i», à la page 297
Suivez cette procédure pour importer un certificat.

Stockage du mot de passe du magasin de certificats sur les systèmes IBM i
Stockez le mot de passe du magasin de certificats à l'aide de commandes CL.

Les instructions suivantes s'appliquent au stockage du mot de passe du magasin de certificats sur IBM i pour un gestionnaire de files d'attente. Sinon, pour un IBM MQ MQI client, si vous n'utilisez pas l'espace de stockage de certificats *SYSTEM (c'est-à-dire que l'environnement MQSSLKEYR est défini sur une valeur autre que *SYSTEM), suivez la procédure décrite dans la section «Stocker le mot de passe du

magasin de certificats», à la page 300 de «IBM MQ Utilitaire de client SSL (amqrsslc) pour IBM i», à la page 299.

Si vous avez indiqué que l'espace de stockage de certificats *SYSTEM doit être utilisé (en remplaçant la valeur de l'attribut SSLKEYR du gestionnaire de files d'attente par *SYSTEM), vous ne devez pas suivre ces étapes.

Une fois que vous avez créé le magasin de certificats à l'aide de DCM, utilisez les commandes suivantes pour stocker le mot de passe:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

Le mot de passe est sensible à la casse. Il doit être entré entre apostrophes exactement comme vous l'avez entré à l'étape 6 de la section «Création d'un magasin de certificats sous IBM i», à la page 286.

Remarque : Si vous n'utilisez pas le magasin de certificats de système par défaut et que vous ne stockez pas le mot de passe, les tentatives de démarrage des canaux TLS échouent car ils ne peuvent pas obtenir le mot de passe requis pour accéder au magasin de certificats.

Protection par mot de passe

Lorsqu'un mot de passe de référentiel de clés est spécifié, IBM MQ chiffre le mot de passe à l'aide du système IBM MQ Password Protection. Pour chiffrer le mot de passe, une clé initiale est utilisée ; si elle n'est pas fournie au gestionnaire de files d'attente, une clé par défaut est utilisée à la place.

Avant de fournir le mot de passe du référentiel de clés, vous devez définir une clé initiale unique pour le gestionnaire de files d'attente. Pour ce faire, utilisez l'attribut **INITKEY** de la commande **ALTER QMGR** MQSC:

```
ALTER QMGR INITKEY('value')
```

Localisation du référentiel de clés d'un gestionnaire de files d'attente sous IBM i

Utilisez cette procédure pour obtenir l'emplacement du magasin de certificats de votre gestionnaire de files d'attente.

Procédure

1. Affichez les attributs de votre gestionnaire de files d'attente à l'aide de la commande suivante:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. Examinez le résultat de la commande pour connaître le chemin et le nom de la racine du magasin de certificats.

   Par exemple: `/QIBM/UserData/ICSS/Cert/Server/Default`, où `/QIBM/UserData/ICSS/Cert/Server` est le chemin et `Default` le nom de la racine.

Modification de l'emplacement du référentiel de clés pour un gestionnaire de files d'attente sous IBM i

Modifiez l'emplacement du magasin de certificats de votre gestionnaire de files d'attente à l'aide de la commande CHGMQM ou ALTER QMGR.

Procédure

Utilisez la commande CHGMQM ou ALTER QMGR MQSC pour définir l'attribut de référentiel de clés de votre gestionnaire de files d'attente.

a) Utilisation de CHGMQM: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) Utilisation de ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

Dans les deux cas, le magasin de certificats possède le nom de fichier complet: `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb`

Que faire ensuite

Lorsque vous modifiez l'emplacement du magasin de certificats d'un gestionnaire de files d'attente, les certificats ne sont pas transférés à partir de l'ancien emplacement. Si les certificats de l'autorité de certification préinstallés lorsque vous créez le magasin de certificats sont insuffisants, vous devez remplir le nouveau magasin de certificats avec des certificats, comme décrit dans «Importation d'un certificat dans un référentiel de clés sous IBM i», à la page 297. Vous devez également stocker le mot de passe pour le nouvel emplacement, comme décrit dans «Stockage du mot de passe du magasin de certificats sur les systèmes IBM i», à la page 286.

IBM i *Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous IBM i*

Etant donné que le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

IBM MQ fournit un mécanisme permettant de fournir le mot de passe du référentiel de clés à un gestionnaire de files d'attente:

- Paramètre **SSLKEYRPWD** de la commande **CHGMQM**

Le mot de passe du référentiel de clés est chiffré à l'aide du système de protection par mot de passe IBM MQ . Pour plus d'informations sur les méthodes de protection du mot de passe du référentiel de clés, voir «Chiffrement des mots de passe du référentiel de clés sous IBM i», à la page 284.

Voir aussi Administration à l'aide de commandes MQSC sous IBM i.

L'attribut SSLKEYRPWD

Pour fournir un mot de passe de référentiel de clés directement au gestionnaire de files d'attente, exécutez la commande **CHGMQM** suivante, en remplaçant *queue_manager* par le nom de votre gestionnaire de files d'attente et *password* par le mot de passe de votre référentiel de clés.

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

Avertissement : Veillez à entourer le nom et le mot de passe du gestionnaire de files d'attente de guillemets simples, sinon IBM MQ convertit les caractères en majuscules.

Lorsqu'un mot de passe de référentiel de clés est spécifié à l'aide de cette méthode, le mot de passe est chiffré à l'aide du système de protection par mot de passe IBM MQ avant d'être stocké.

Une clé de chiffrement, appelée clé initiale, est utilisée pour chiffrer le mot de passe. Définissez le gestionnaire de files d'attente pour qu'il utilise une clé initiale unique afin de protéger le mot de passe de manière sécurisée. Si vous ne fournissez pas de clé initiale, la clé par défaut est utilisée.

Vérifiez que le gestionnaire de files d'attente est configuré avec une clé initiale unique avant de définir le mot de passe du référentiel de clés. Vous pouvez modifier la clé initiale à l'aide de l'attribut **INITKEY** de la commande **ALTER QMGR** . Exemple :

```
ALTER QMGR INITKEY('mykey')
```

Avertissement : Si vous modifiez la clé initiale après avoir défini le mot de passe du référentiel de clés, le mot de passe du référentiel de clés n'est pas chiffré avec la nouvelle clé initiale. Si vous modifiez la clé initiale, vous devez également réinitialiser le mot de passe du référentiel de clés. Sinon, IBM MQ ne peut pas déchiffrer le mot de passe du référentiel de clés et ne peut donc pas accéder au référentiel de clés.

Pour plus d'informations sur l'attribut **SSLKEYRPWD** , voir Le paramètre **SSLKEYRPWD** de la commande **CHGMQM**.

Concepts associés

«Chiffrement des mots de passe du référentiel de clés sous IBM i», à la page 284
Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

«Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur IBM i», à la page 289
Etant donné que le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

IBM i *Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur IBM i*

Etant donné que le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

IBM MQ fournit quatre mécanismes pour fournir le mot de passe du référentiel de clés à un IBM MQ MQI client:

- «Les zones KeyRepoPassword de MQSCO », à la page 289
- «Variable d'environnement MQKEYRPWD», à la page 290
- «Attribut SSLKeyRepositoryPassword du fichier de configuration du client», à la page 290
- «Fichier de dissimulation du référentiel de clés», à la page 290

Si vous n'utilisez pas de fichier de dissimulation de référentiel de clés, vous pouvez fournir le mot de passe du référentiel de clés sous la forme d'une chaîne de texte en clair ou d'une chaîne chiffrée à l'aide du système de protection par mot de passe IBM MQ . Pour plus d'informations sur les méthodes de protection du mot de passe du référentiel de clés, voir «Chiffrement des mots de passe du référentiel de clés sous IBM i», à la page 284.

Les zones KeyRepoPassword de MQSCO

Pour fournir un mot de passe de référentiel de clés à l'aide de la structure MQSCO, vous devez utiliser une combinaison des trois zones de chaîne de variable suivantes:

KeyRepoPasswordLength
Longueur du mot de passe.

KeyRepoPasswordPtr
Pointeur vers l'emplacement en mémoire qui contient le mot de passe.

KeyRepoPasswordOffset
Emplacement du mot de passe en mémoire, représenté en nombre d'octets depuis le début de la structure MQSCO.

Remarque : Vous ne pouvez indiquer qu'un seul **KeyRepoPasswordPtr** ou **KeyRepoPasswordOffset**.

Exemple :

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Avertissement : Si vous fournissez le mot de passe à l'aide de cette méthode, chiffrez le mot de passe avant qu'il ne soit fourni à l'application IBM MQ client . Pour plus d'informations, voir «Chiffrement du mot de passe du référentiel de clés», à la page 291.

Pour plus d'informations sur la structure MQCSO, voir MQSCO-Options de configuration SSL/TLS.

Variable d'environnement *MQKEYRPWD*

Si un mot de passe de référentiel de clés n'est pas fourni au client à l'aide de la structure MQSCO, vous pouvez spécifier le mot de passe de référentiel de clés à l'aide de la variable d'environnement *MQKEYRPWD* . Exemple :

```
export MQKEYRPWD=passw0rd
```

ou

```
set MQKEYRPWD=passw0rd
```

où *passw0rd* est votre mot de passe.

Avertissement : Si vous fournissez le mot de passe à l'aide de cette méthode, chiffrez-le avant de définir la valeur de la variable d'environnement. Pour plus d'informations, voir «Chiffrement du mot de passe du référentiel de clés», à la page 291.

Attribut SSLKeyRepositoryPassword du fichier de configuration du client

Si un mot de passe de référentiel de clés n'est pas fourni au client à l'aide de l'une des autres méthodes, vous pouvez spécifier le mot de passe de référentiel de clés à l'aide de l'attribut **SSLKeyRepositoryPassword** dans la section **SSL** du fichier de configuration du client. Exemple :

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Avertissement : Si vous fournissez le mot de passe à l'aide de cette méthode, chiffrez-le avant de définir la valeur de l'attribut **SSLKeyRepositoryPassword** . Pour plus d'informations, voir «Chiffrement du mot de passe du référentiel de clés», à la page 291.

Pour plus d'informations sur la strophe SSL du fichier de configuration du client, voir Strophe SSL du fichier de configuration du client.

Fichier de dissimulation du référentiel de clés

Si le mot de passe du référentiel de clés n'est pas fourni au client à l'aide de l'une des autres méthodes, IBM MQ suppose qu'un fichier de dissimulation existe dans le même répertoire que le référentiel de clés. Le fichier de dissimulation a le même nom de radical que le référentiel de clés, mais possède l'extension `.sth` .

Un fichier de dissimulation de référentiel de clés est créé à l'aide de l'outil de ligne de commande **amqrsslc** . Pour créer le fichier de dissimulation, exécutez la commande suivante:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

Cette commande vous invite à indiquer le mot de passe à chiffrer. Le mot de passe est chiffré par le système de protection par mot de passe IBM MQ , avec une clé de chiffrement par défaut, sauf si celle-ci est fournie à l'aide du paramètre **-sf** .

Pour plus d'informations, reportez-vous aux sections «IBM MQ Utilitaire de client SSL (amqrsslc) pour IBM i», à la page 299 et «Chiffrement du mot de passe du référentiel de clés», à la page 291.

Chiffrement du mot de passe du référentiel de clés

Si vous fournissez le mot de passe du référentiel de clés à l'aide d'une méthode autre qu'un fichier de dissimulation, chiffrez le mot de passe à l'aide du système de protection par mot de passe IBM MQ . Pour chiffrer le mot de passe, exécutez la commande **runmqicred** . Entrez le mot de passe du référentiel de clés lorsque vous y êtes invité. La commande génère le mot de passe chiffré. Le mot de passe chiffré peut être fourni à IBM MQ MQI client à la place du mot de passe en texte en clair à l'aide de l'une des méthodes décrites.

Une clé de chiffrement, appelée clé initiale, est utilisée pour chiffrer le mot de passe. Lorsque vous chiffrez le mot de passe, utilisez une clé initiale unique pour protéger le mot de passe de manière sécurisée. Pour fournir votre propre clé initiale, utilisez le paramètre **-sf** dans la commande **runmqicred** . Si vous ne fournissez pas de clé initiale, la clé par défaut est utilisée.

Pour plus d'informations, voir runmqicred (protection des mots de passe du client IBM MQ).

Si vous fournissez votre propre clé initiale lorsque le mot de passe du référentiel de clés est chiffré et que vous fournissez le mot de passe chiffré à IBM MQ MQI client, vous devez également vous assurer que vous fournissez la même clé initiale à IBM MQ MQI client. Pour plus d'informations sur la façon de fournir la clé initiale à un IBM MQ MQI client, voir «Fourniture d'une clé initiale pour un IBM MQ MQI client sous IBM i», à la page 291.

Concepts associés

«Chiffrement des mots de passe du référentiel de clés sous IBM i», à la page 284
Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

«Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous IBM i», à la page 288
Etant donné que le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

IBM i *Fourniture d'une clé initiale pour un IBM MQ MQI client sous IBM i*

Si vous fournissez des variables à un IBM MQ MQI client qui ont été chiffrées à l'aide du système de protection par mot de passe IBM MQ , vous devrez peut-être fournir la clé initiale correspondante qui a été utilisée pour chiffrer la valeur.

Si vous n'avez pas spécifié de clé initiale lors du chiffrement de la valeur, vous n'avez pas besoin de fournir de valeur de clé initiale à IBM MQ client. Toutefois, si vous avez utilisé une clé initiale unique, vous pouvez la fournir à IBM MQ client à l'aide des méthodes suivantes:

- «Fourniture de la clé initiale à l'aide de la structure MQCSP», à la page 291
- «Fourniture de la clé initiale à l'aide de la variable d'environnement MQS_MQI_KEYFILE», à la page 292
- «Fourniture de la clé initiale à l'aide du fichier de configuration du client», à la page 292

Fourniture de la clé initiale à l'aide de la structure MQCSP

Pour fournir la clé initiale à l'aide de la structure MQCSP, vous devez utiliser une combinaison des trois zones de chaîne de variable suivantes:

InitialKeyLength
Longueur de la clé initiale

InitialKeyPtr
Un pointeur vers l'emplacement en mémoire contenant la clé initiale

InitialKeyOffset
Emplacement de la clé initiale en mémoire, représenté en nombre d'octets depuis le début de la structure MQCSP.

Remarque : Vous ne pouvez indiquer qu'un seul **InitialKeyPtr** ou **InitialKeyOffset**.

Exemple :

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

Fourniture de la clé initiale à l'aide de la variable d'environnement MQS_MQI_KEYFILE

Si aucune clé initiale n'est fournie au client à l'aide de la structure MQCSP, IBM MQ vérifie la variable d'environnement *MQS_MQI_KEYFILE* . Vous devez définir cette variable d'environnement à l'emplacement d'un fichier contenant une seule ligne de texte, constituée de la clé initiale que vous souhaitez utiliser.

Par exemple, si un fichier appelé `mykey.key` existe dans le répertoire racine et qu'il contient la clé initiale, vous devez définir la variable d'environnement comme suit:

```
export MQS_MQI_KEYFILE=/mykey.key
```

ou

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

Fourniture de la clé initiale à l'aide du fichier de configuration du client

Si aucune clé initiale n'est fournie au client à l'aide d'un mécanisme précédent, IBM MQ vérifie l'attribut **MQIInitialKeyFile** de la strophe de sécurité du fichier `mqclient.ini` . Vous devez définir cet attribut sur l'emplacement d'un fichier contenant une seule ligne de texte, constituée de la clé initiale que vous souhaitez utiliser.

Par exemple, si un fichier appelé `mykey.key` existe dans le répertoire racine et qu'il contient la clé initiale, le fichier de configuration du client doit contenir ce qui suit:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

Concepts associés

«Chiffrement des mots de passe du référentiel de clés sous IBM i», à la page 284
Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

«Utilisation de SSL/TLS sous IBM i», à la page 283
Cette collection de rubriques fournit des instructions pour les tâches individuelles utilisant le protocole TLS (Transport Layer Security) dans IBM MQ for IBM i.

Création d'une autorité de certification et d'un certificat à des fins de test sur IBM i

Utilisez cette procédure pour créer un certificat d'autorité de certification local afin de signer des demandes de certificat et pour créer et installer le certificat d'autorité de certification.

Avant de commencer

Les instructions de cette rubrique supposent qu'une autorité de certification locale n'existe pas. S'il existe une autorité de certification locale, accédez à «Demande d'un certificat serveur sous IBM i», à la page 294.

Pourquoi et quand exécuter cette tâche

Les certificats de l'autorité de certification fournis lors de l'installation de TLS sont signés par l'autorité de certification émettrice. Sous IBM i, vous pouvez générer une autorité de certification locale qui peut signer des certificats de serveur pour tester les communications TLS sur votre système. Procédez comme suit dans un navigateur Web pour créer un certificat d'autorité de certification local:

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans le panneau de navigation, cliquez sur **Créer une autorité de certification**.

   La page Créer une autorité de certification s'affiche dans le cadre de la tâche.
3. Entrez un mot de passe dans la zone **Mot de passe du magasin de certificats** et entrez-le à nouveau dans la zone **Confirmer le mot de passe** .
4. Entrez un nom dans la zone **Nom de l'autorité de certification** , par exemple `TLS Test Certificate Authority`.
5. Entrez les valeurs appropriées dans les zones **Nom usuel** et **Organisation** et sélectionnez un pays. Pour les autres zones facultatives, entrez les valeurs requises.
6. Entrez une période de validité pour l'autorité de certification locale dans la zone **Période de validité** .

   La valeur par défaut est 1095 jours.
7. Cliquez sur **Continue**.

   L'autorité de certification est créée et DCM crée un magasin de certificats et un certificat d'autorité de certification pour votre autorité de certification locale.
8. Cliquez sur **Installer le certificat**.

   La boîte de dialogue du gestionnaire de téléchargement s'affiche.
9. Entrez le nom de chemin d'accès complet du fichier temporaire dans lequel vous souhaitez stocker le certificat de l'autorité de certification et cliquez sur **Sauvegarder**.
10. Une fois le téléchargement terminé, cliquez sur **Ouvrir**.

    La fenêtre Certificat s'affiche.
11. Cliquez sur **Installer le certificat**.

    L'assistant d'importation de certificat s'affiche.
12. Cliquez sur **Suivant**.
13. Sélectionnez **Sélectionner automatiquement le magasin de certificats en fonction du type de certificat** et cliquez sur **Suivant**.
14. Cliquez sur **Terminer**.

    Une fenêtre de confirmation s'affiche.
15. Cliquez sur **OK**.
16. Dans la fenêtre Certificat, cliquez sur **OK**.
17. Cliquez sur **Continue**.

    La page Politique de l'autorité de certification s'affiche dans le cadre de la tâche.
18. Dans la zone **Autoriser la création de certificats d'utilisateur** , sélectionnez **Oui**.
19. Dans la zone **Période de validité** , entrez la période de validité des certificats émis par votre autorité de certification locale.

    La valeur par défaut est 365 jours.
20. Cliquez sur **Continue**.

    La page Créer un certificat dans le nouveau magasin de certificats s'affiche dans le cadre de la tâche.

21. Vérifiez qu'aucune des applications n'est sélectionnée.
22. Cliquez sur **Continuer** pour terminer la configuration de l'autorité de certification locale.

Que faire ensuite

Si vous devez renouveler un certificat existant, voir Renouvellement d'un certificat existant dans la documentation IBM i .

Demande d'un certificat serveur sous IBM i

Les certificats numériques constituent une protection contre l'usurpation d'identité en certifiant qu'une clé publique appartient à une entité spécifiée. Un nouveau certificat serveur peut être demandé auprès d'une autorité de certification à l'aide du Certificate Manager numérique (DCM).

Pourquoi et quand exécuter cette tâche

Effectuez les étapes suivantes dans un navigateur Web:

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans le panneau de navigation, cliquez sur **Sélectionner un magasin de certificats**.

   La page Sélectionner un magasin de certificats s'affiche dans le cadre de la tâche.
3. Sélectionnez le magasin de certificats que vous souhaitez utiliser et cliquez sur **Continuer**.
4. Facultatif : Si vous avez sélectionné ***SYSTEM** à l'étape 3, entrez le mot de passe du magasin système et cliquez sur **Continuer**.
5. Facultatif : Si vous avez sélectionné **Autre magasin de certificats système** à l'étape 3, dans la zone **Chemin d'accès au magasin de certificats et nom de fichier** , entrez le chemin et le nom de fichier IFS que vous avez définis lors de la création de votre magasin de certificats. Entrez également un mot de passe dans la zone **Certificate Store Password** . Cliquez ensuite sur **Continuer**
6. Dans le panneau de navigation, cliquez sur **Créer un certificat**.
7. Dans le cadre de la tâche, sélectionnez le bouton d'option **Serveur ou certificat client** et cliquez sur **Continuer**.

   La page Sélectionner une autorité de certification s'affiche dans le cadre de la tâche.
8. Si vous disposez d'une autorité de certification locale sur votre poste de travail, choisissez l'autorité de certification locale ou une autorité de certification commerciale pour signer le certificat. Sélectionnez le bouton d'option correspondant à l'autorité de certification de votre choix et cliquez sur **Continuer**.

   La page Créer un certificat s'affiche dans le cadre de la tâche.
9. Facultatif : Pour un gestionnaire de files d'attente, dans la zone **Libellé du certificat** , entrez le libellé du certificat.

   Le libellé est soit la valeur de l'attribut **CERTLABL** , s'il est défini, soit la valeur par défaut `ibmwebspheremq` avec le nom du gestionnaire de files d'attente ajouté, le tout en minuscules. Pour plus de détails voir Labels de certificat numérique.

   Par exemple, pour le gestionnaire de files d'attente QM1, entrez `ibmwebspheremqqm1` pour utiliser la valeur par défaut.
10. Facultatif : Pour un IBM MQ MQI client, dans la zone **Libellé du certificat** , entrez `ibmwebspheremq` suivi de votre ID utilisateur de connexion en minuscules.
    Par exemple, saisissez : `ibmwebspheremqmyuserID`
11. Entrez les valeurs appropriées dans les zones **Nom usuel** et **Organisation** et sélectionnez un pays. Pour les autres zones facultatives, entrez les valeurs requises.

Résultats

Si vous avez sélectionné une autorité de certification commerciale pour signer votre certificat, DCM crée une demande de certificat au format PEM (Privacy-Enhanced Mail). Transmettez la demande à l'autorité de certification choisie.

Si vous avez sélectionné l'autorité de certification locale pour signer votre certificat, DCM vous informe que le certificat a été créé dans le magasin de certificats et qu'il peut être utilisé.

Demander un certificat de serveur pour un système distant surIBM i

Suivez cette procédure pour créer un certificat signé par votre autorité de certification (CA) locale ou pour demander un certificat de serveur signé par une autorité de certification commerciale pour l'importation dans un référentiel de clés sur d'autres plates-formes.

Pourquoi et quand exécuter cette tâche

Un certificat d'utilisateur doit être utilisé lorsque le Certificate Manager (DCM) numérique sert de gestionnaire de certificats pour IBM MQ sur plusieurs plateformes. Pour les certificats personnels distribués sur d'autres plateformes et importés dans un référentiel de clés, effectuez les étapes suivantes dans un navigateur Web :

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans le panneau de **navigation** , cliquez sur **Créer un certificat**.

   La page **Créer un certificat** s'affiche dans le cadre de la tâche.
3. Dans le panneau **Créer un certificat** , sélectionnez le bouton d'option **Certificat d'utilisateur** et cliquez sur **Continuer**.

   La page **Créer un certificat d'utilisateur** s'affiche.
4. Dans le panneau **Créer un certificat d'utilisateur** , renseignez les zones obligatoires sous Informations sur le certificat pour **Nom de l'organisation**, **Etat** ou province , **Pays** ou **région**. Vous pouvez éventuellement insérer des valeurs dans les zones **Unité organisationnelle** et **Localité** ou **ville** . Cliquez sur **Continue**.

   Le **nom usuel** est automatiquement défini sur l'ID utilisateur avec lequel vous êtes connecté au système iSeries .
5. Dans le panneau **Create User Certificate** suivant, cliquez sur **Install certificate** , puis sur **Continue**.

   Un message s'affiche pour indiquer que `votre certificat personnel a été installé. Vous devez conserver une copie de sauvegarde de ce certificat.`
6. Cliquez sur **OK**.
7. En fonction du navigateur Web que vous avez utilisé pour accéder à DCM, effectuez l'une des étapes suivantes :
   - Pour Microsoft Edge, choisissez: **Outils > Options Internet > Onglet Contenu > Bouton Certificats > Onglet Personnel >**. Sélectionnez le certificat et cliquez sur **Exporter**.
   - Pour Mozilla Firefox , sélectionnez: **Tools> Options > Advanced> Encryption tab > View Certificates button > Your Certificates tab >**. Sélectionnez le certificat et cliquez sur **Sauvegarder**. Sélectionnez le chemin et le nom de fichier et cliquez sur **OK**.
8. Transférez le certificat exporté vers le système distant à l'aide de FTP au format binaire.
9. Importez le certificat qui a été exporté à l'étape«7», à la page 295 au référentiel de clés sur le système distant.
   - Si le certificat a été enregistré à l'aideMicrosoft Edge, suivez les instructions décrites dans«Importer un certificat personnel depuis unMicrosoft Fichier .pfx», à la page 572 déposer.
   - Si le certificat a été sauvegardé à l'aide de Mozilla Firefox, utilisez les instructions décrites dans Importation d'un certificat personnel dans un référentiel de clés.

Lors de l'importation, assurez-vous que le nom de l'étiquette du certificat personnel et du certificat de signataire est remplacé par la valeur quiIBM MQ attend. L'étiquette doit être soit la valeur duIBM MQ gestionnaire de files d'attente**CERTLABL** attribut, s'il est défini, ou la valeur par défaut de`ibmwebspheremq` avec le nom du gestionnaire de files d'attente ajouté, le tout en minuscules. Pour plus d'informations, voir Étiquettes de certificat numérique .

Ajout de certificats serveur à un référentiel de clés sur IBM i

Suivez cette procédure pour ajouter un certificat demandé au référentiel de clés.

Pourquoi et quand exécuter cette tâche

Une fois que l'autorité de certification vous a envoyé un nouveau certificat serveur, vous l'ajoutez au magasin de certificats à partir duquel vous avez généré la demande. Si l'autorité de certification envoie le certificat dans le cadre d'un message électronique, copiez le certificat dans un fichier distinct.

Remarque :

- Vous n'avez pas besoin d'exécuter cette procédure si le certificat serveur est signé par votre autorité de certification locale.
- Avant d'importer un certificat serveur au format PKCS #12 dans DCM, vous devez d'abord importer le certificat de l'autorité de certification correspondant.

Utilisez la procédure suivante pour recevoir un certificat de serveur dans le magasin de certificats du gestionnaire de files d'attente:

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans la catégorie de tâche **Gérer les certificats** du panneau de navigation, cliquez sur **Importer un certificat**.

   La page Importer un certificat s'affiche dans le cadre de la tâche.
3. Sélectionnez le bouton d'option correspondant à votre type de certificat et cliquez sur **Continuer**.

   La page Importation d'un serveur ou d'un certificat client ou la page Importation d'un certificat d'autorité de certification s'affiche dans le cadre de la tâche.
4. Dans la zone **Importer un fichier** , entrez le nom de fichier du certificat à importer et cliquez sur **Continuer**.

   DCM détermine automatiquement le format du fichier.
5. Si le certificat est un certificat **serveur ou client** , entrez le mot de passe dans le cadre de la tâche et cliquez sur **Continuer**.

   DCM vous informe que le certificat a été importé.

Exportation d'un certificat à partir d'un référentiel de clés sous IBM i

L'exportation d'un certificat exporte à la fois la clé publique et la clé privée. Cette action doit être effectuée avec une extrême prudence, car la transmission d'une clé privée compromettrait complètement votre sécurité.

Avant de commencer

Lorsque vous partagez le certificat d'un utilisateur avec un autre utilisateur, vous échangez des clés publiques. Ce processus est décrit dans la tâche 5 de **. Partage de certificats** dans la section Partage de certificats de «Guide de démarrage rapide pour AMS sur AIX and Linux», à la page 633. Lorsque vous exportez un certificat comme décrit ici, vous exportez à la fois la clé publique et la clé privée. Cette action doit être effectuée avec une extrême prudence, car la transmission d'une clé privée compromettrait complètement votre sécurité.

Pourquoi et quand exécuter cette tâche

Effectuez les étapes suivantes sur l'ordinateur à partir duquel vous souhaitez exporter le certificat:

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans le panneau de navigation, cliquez sur **Sélectionner un magasin de certificats**.

   La page Sélectionner un magasin de certificats s'affiche dans le cadre de la tâche.
3. Sélectionnez le magasin de certificats que vous souhaitez utiliser et cliquez sur **Continuer**.
4. Facultatif : Si vous avez sélectionné ***SYSTEM** à l'étape 3, entrez le mot de passe du magasin système et cliquez sur **Continuer**.
5. Facultatif : Si vous avez sélectionné **Autre magasin de certificats système** à l'étape 3, dans la zone **Chemin et nom du magasin de certificats** , entrez le chemin et le nom de fichier IFS que vous avez définis lors de la création de votre magasin de certificats et entrez un mot de passe dans la zone **Mot de passe du magasin de certificats** . Cliquez ensuite sur **Continuer**
6. Dans la catégorie de tâche **Gérer les certificats** du panneau de navigation, cliquez sur **Exporter le certificat**.

   La page Exporter un certificat s'affiche dans le cadre de la tâche.
7. Sélectionnez le bouton d'option correspondant à votre type de certificat et cliquez sur **Continuer**.

   La page Exporter un certificat serveur ou client ou la page Exporter un certificat d'autorité de certification s'affiche dans le cadre de la tâche.
8. Sélectionnez le certificat à exporter.
9. Sélectionnez le bouton d'option pour indiquer si vous souhaitez exporter le certificat dans un fichier ou directement dans un autre magasin de certificats.
10. Si vous avez choisi d'exporter un certificat serveur ou client dans un fichier, fournissez les informations suivantes:
    - Chemin et nom de fichier de l'emplacement où vous souhaitez stocker le certificat exporté.
    - Pour un certificat personnel, mot de passe utilisé pour chiffrer le certificat exporté et l'édition cible. Pour les certificats de l'autorité de certification, vous n'avez pas besoin de spécifier le mot de passe.
11. Si vous avez choisi d'exporter un certificat directement dans un autre magasin de certificats, indiquez le magasin de certificats cible et son mot de passe.
12. Cliquez sur **Continue**.

Importation d'un certificat dans un référentiel de clés sous IBM i

Suivez cette procédure pour importer un certificat.

Avant de commencer

Avant d'importer un certificat personnel au format PKCS #12 dans DCM, vous devez d'abord importer le certificat de l'autorité de certification correspondant.

Pourquoi et quand exécuter cette tâche

Effectuez ces étapes sur la machine sur laquelle vous souhaitez importer le certificat.

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans le panneau de navigation, cliquez sur **Sélectionner un magasin de certificats**.

   La page Sélectionner un magasin de certificats s'affiche dans le cadre de la tâche.
3. Sélectionnez le magasin de certificats que vous souhaitez utiliser et cliquez sur **Continuer**.

4. Facultatif : Si vous avez sélectionné ***SYSTEM** à l'étape 3, entrez le mot de passe du magasin système et cliquez sur **Continuer**.
5. Facultatif : Si vous avez sélectionné **Autre magasin de certificats système** à l'étape 3, dans la zone **Chemin et nom du magasin de certificats** , entrez le chemin et le nom de fichier IFS que vous avez définis lors de la création de votre magasin de certificats et entrez un mot de passe dans la zone **Mot de passe du magasin de certificats** . Cliquez ensuite sur **Continuer**
6. Dans la catégorie de tâche **Gérer les certificats** du panneau de navigation, cliquez sur **Importer un certificat**.

   La page Importer un certificat s'affiche dans le cadre de la tâche.
7. Sélectionnez le bouton d'option correspondant à votre type de certificat et cliquez sur **Continuer**.

   La page d'importation du serveur ou du certificat client ou la page d'importation du certificat de l'autorité de certification s'affiche dans le cadre de la tâche.
8. Dans la zone **Importer un fichier** , entrez le nom de fichier du certificat à importer et cliquez sur **Continuer**.

   DCM détermine automatiquement le format du fichier.
9. Si le certificat est un certificat **serveur ou client** , entrez le mot de passe dans le cadre de la tâche et cliquez sur **Continuer**. DCM vous informe que le certificat a été importé.

Suppression de certificats dans IBM i

Utilisez cette procédure pour supprimer des certificats personnels.

Procédure

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans le panneau de navigation, cliquez sur **Sélectionner un magasin de certificats**.

   La page Sélectionner un magasin de certificats s'affiche dans le cadre de la tâche.
3. Cochez la case **Autre magasin de certificats système** et cliquez sur **Continuer**.

   La page Magasin de certificats et mot de passe s'affiche.
4. Dans la zone **Chemin d'accès au magasin de certificats et nom de fichier** , entrez le chemin d'accès au système de fichiers intégré et le nom de fichier que vous avez définis lors de la création du magasin de certificats.
5. Entrez un mot de passe dans la zone **Certificate Store Password** . Cliquez sur **Continue**.

   La page Magasin de certificats en cours s'affiche dans le cadre de la tâche.
6. Dans la catégorie de tâche **Gérer les certificats** du panneau de navigation, cliquez sur **Supprimer le certificat**.

   La page Confirmation de suppression de certificat s'affiche dans le cadre de la tâche.
7. Sélectionnez le certificat à supprimer. Cliquez sur **Supprimer**.
8. Cliquez sur **Oui** pour confirmer la suppression du certificat. Sinon, cliquez sur **Non**.

   DCM vous informe si le certificat a été supprimé.

*Utilisation de l'espace de stockage de certificats *SYSTEM pour l'authentification unidirectionnelle sous IBM i*

Suivez ces instructions pour configurer l'authentification unidirectionnelle.

Avant de commencer

- Créez un gestionnaire de files d'attente, des canaux et des files d'attente de transmission.
- Créez un certificat serveur ou client sur le gestionnaire de files d'attente du serveur.
- Transférez le certificat de l'autorité de certification au gestionnaire de files d'attente client et importez-le dans le référentiel de clés.
- Démarrez un programme d'écoute sur le serveur et les gestionnaires de files d'attente client.

Pourquoi et quand exécuter cette tâche

Pour utiliser l'authentification unidirectionnelle, en utilisant un ordinateur exécutant IBM i comme serveur TLS, définissez le paramètre SSL Key Repository (SSLKEYR) sur *SYSTEM. Ce paramètre enregistre le gestionnaire de files d'attente IBM MQ en tant qu'application. Vous pouvez ensuite affecter un certificat au gestionnaire de files d'attente pour activer l'authentification unidirectionnelle.

Vous pouvez également utiliser des magasins de clés privées pour implémenter l'authentification unidirectionnelle en créant un certificat factice pour le gestionnaire de files d'attente client dans le référentiel de clés.

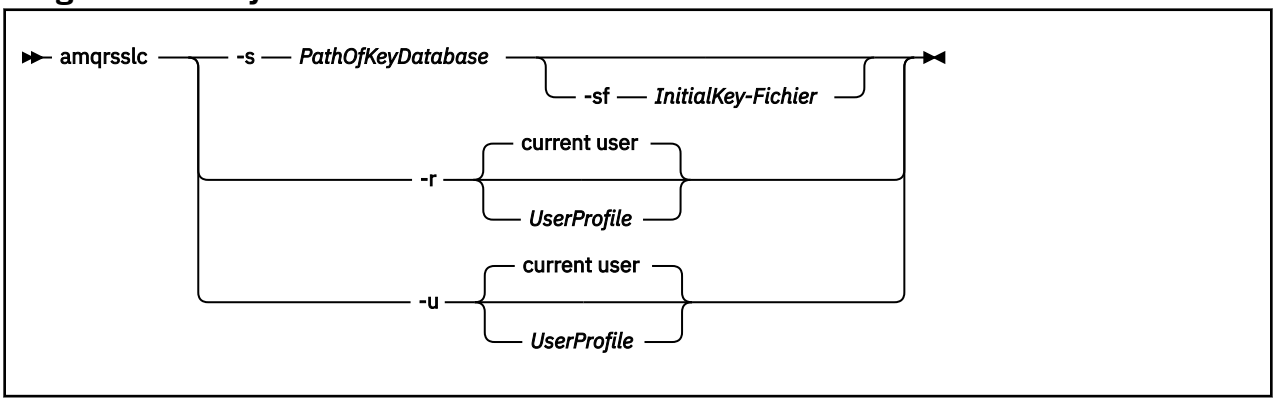
Procédure

1. Effectuez les étapes suivantes sur les gestionnaires de files d'attente du serveur et du client:
   a) Modifiez le gestionnaire de files d'attente pour définir le paramètre SSLKEYR en exécutant la commande `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)`.
   b) Stockez le mot de passe du référentiel de clés par défaut en exécutant la commande `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')`.
      Le mot de passe doit être entre apostrophes.
   c) Modifiez les canaux pour qu'ils aient le CipherSpec correct dans le paramètre SSLCIPHER.
   d) Actualisez la sécurité TLS en émettant la commande `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)`.
2. Affectez le certificat au gestionnaire de files d'attente du serveur à l'aide de DCM, comme suit:
   a) Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
   b) Dans le panneau de navigation, cliquez sur **Sélectionner un magasin de certificats**.
      La page Sélectionner un magasin de certificats s'affiche dans le cadre de la tâche.
   c) Sélectionnez l'espace de stockage de certificats *SYSTEM et cliquez sur **Continuer**.
   d) Dans le panneau de gauche, développez **Gérer les applications**.
   e) Sélectionnez la définition **Afficher l'application** pour vérifier que le gestionnaire de files d'attente a été enregistré en tant qu'application.
      `SSL (WMQ)` est répertorié dans le tableau.
   f) Sélectionnez **Mettre à jour l'affectation de certificat**.
   g) Sélectionnez **Serveur** et cliquez sur **Continuer**.
   h) Sélectionnez QMGRNAME (WMQ) et cliquez sur **Mettre à jour l'affectation de certificat**.
   i) Sélectionnez le certificat et cliquez sur **Affecter un nouveau certificat**. Une fenêtre s'ouvre pour indiquer que le certificat a été affecté à l'application.

IBM MQ Utilitaire de client SSL (amqrsslc) pour IBM i

L'utilitaire IBM MQ SSL Client (amqrsslc) for IBM i est utilisé par IBM MQ MQI client sur les systèmes IBM i pour enregistrer ou désenregistrer le profil utilisateur du client ou pour stocker le mot de passe du magasin de certificats. L'utilitaire ne peut être exécuté que par un utilisateur disposant des droits spéciaux *ALLOBJ ou par un membre de QMQMADM doté d'options permettant de créer ou de supprimer des enregistrements d'application dans le Certificate Manager numérique (DCM).

Diagramme de syntaxe



Enregistrer le profil utilisateur du client

Si IBM MQ MQI client utilise l'espace de stockage de certificats *SYSTEM, vous devez enregistrer le profil utilisateur du client (utilisateur connecté) pour l'utiliser en tant qu'application avec Digital Certificate Manager (DCM).

Si vous souhaitez enregistrer le profil utilisateur du client, exécutez le programme **amqrsslc** avec l'option `-r` avec *UserProfile*. Le profil utilisateur utilisé lors de l'appel de **amqrsslc** doit disposer du droit *USE. Si vous indiquez *UserProfile* avec l'option `-r` , l'option *UserProfile* est enregistrée en tant qu'application serveur avec un libellé d'application unique de QIBM_WEBSPHERE_MQ_*UserProfile* et un libellé avec une description de *UserProfile* (WMQ). Cette application serveur est ensuite affichée dans DCM et vous pouvez lui affecter n'importe quel certificat serveur ou client dans le magasin système.

Remarque : Si aucun profil utilisateur n'est indiqué avec l'option `-r` , le profil utilisateur de l'utilisateur exécutant l'outil **amqrsslc** est enregistré.

Le code suivant utilise **amqrsslc** pour enregistrer un profil utilisateur. Dans le premier exemple, le profil utilisateur spécifié est enregistré ; dans le second, il s'agit du profil de l'utilisateur connecté:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

Annulation de l'enregistrement du profil utilisateur du client

Pour annuler l'enregistrement du profil de client, exécutez le programme **amqrsslc** avec l'option `-u` avec *UserProfile*. Le profil utilisateur utilisé lors de l'appel de **amqrsslc** doit disposer du droit *USE. La fourniture de l'option *UserProfile* avec l'option `-u` désenregistre *UserProfile* avec le libellé QIBM_WEBSPHERE_MQ_*UserProfile* dans DCM.

Remarque : Si aucun profil utilisateur n'est spécifié avec l'option `-u` , le profil utilisateur de l'utilisateur exécutant l'outil **amqrsslc** est désenregistré.

Le code suivant utilise **amqrsslc** pour annuler l'enregistrement d'un profil utilisateur. Dans le premier exemple, le profil utilisateur spécifié est désenregistré ; dans le second, il s'agit du profil de l'utilisateur connecté:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

Stocker le mot de passe du magasin de certificats

Si le IBM MQ MQI client n'utilise pas l'espace de stockage de certificats *SYSTEM et qu'il utilise un autre espace de stockage de certificats (c'est-à-dire que MQSSLKEYR est défini sur une valeur autre que *SYSTEM), le mot de passe de la base de données de clés peut être stocké de sorte qu'il n'ait pas besoin d'être spécifié par l'application client lors de son exécution.

Utilisez l'option `-s` pour stocker le mot de passe de la base de données de clés. Indiquez le chemin d'accès complet et le nom de la base de données de clés. Si l'extension de fichier n'est pas fournie, elle est supposée être `.kdb`.

Dans le code suivant, le nom de fichier complet du magasin de certificats est `/Path/Of/KeyDatabase/MyKey.kdb`:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

L'exécution de ce code entraîne une demande de mot de passe pour cette base de données de clés. Ce mot de passe est stocké dans un fichier portant le même nom que la base de données de clés avec une extension `.sth` .

En outre, la clé initiale permettant de chiffrer le mot de passe peut être spécifiée. La clé initiale doit être stockée dans un fichier sous la forme d'une seule ligne de texte, puis l'emplacement de ce fichier est fourni au programme via l'indicateur **`-sf`** . Si aucun fichier de clés initial n'est fourni, une clé par défaut est utilisée pour chiffrer le mot de passe.

Le fichier de dissimulation est stocké dans le même chemin que la base de données de clés. L'exemple de code génère un fichier de dissimulation `/Path/Of/KeyDatabase/MyKey.sth`.

QMQM est le propriétaire de l'utilisateur et QMQMADM le propriétaire du groupe pour ce fichier. QMQM et QMQMADM ont des droits de lecture, d'écriture et d'autres profils ont uniquement des droits de lecture.

Lorsque les modifications apportées aux certificats ou au magasin de certificats prennent effet sur IBM i

Lorsque vous modifiez les certificats dans un magasin de certificats ou l'emplacement du magasin de certificats, les modifications sont prises en compte en fonction du type de canal et de la manière dont le canal est en cours d'exécution.

Les modifications apportées aux certificats dans le magasin de certificats et à l'attribut de référentiel de clés prennent effet dans les situations suivantes:

- Lorsqu'un nouveau processus de canal unique sortant exécute pour la première fois un canal TLS.
- Lorsqu'un nouveau processus de canal unique TCP/IP entrant reçoit pour la première fois une demande de démarrage d'un canal TLS.
- Lorsque la commande MQSC REFRESH SECURITY TYPE (SSL) est émise pour actualiser l'environnement TLS IBM MQ .
- Pour les processus d'application client, lorsque la dernière connexion TLS du processus est fermée. La connexion TLS suivante récupère les changements de certificat.
- Pour les canaux qui s'exécutent en tant qu'unités d'exécution d'un processus de regroupement de processus (amqrmppa), lorsque le processus de regroupement de processus est démarré ou redémarré et exécute d'abord un canal TLS. Si le processus de regroupement de processus a déjà exécuté un canal TLS et que vous souhaitez que la modification soit prise en compte immédiatement, exécutez la commande MQSC REFRESH SECURITY TYPE (SSL).
- Pour les canaux qui s'exécutent en tant qu'unités d'exécution de l'initiateur de canal, lorsque l'initiateur de canal est démarré ou redémarré et qu'il exécute d'abord un canal TLS. Si le processus initiateur de canal a déjà exécuté un canal TLS et que vous souhaitez que la modification soit prise en compte immédiatement, exécutez la commande MQSC REFRESH SECURITY TYPE (SSL).
- Pour les canaux qui s'exécutent en tant qu'unités d'exécution d'un programme d'écoute TCP/IP, lorsque le programme d'écoute est démarré ou redémarré et qu'il reçoit d'abord une demande de démarrage d'un canal TLS. Si le programme d'écoute a déjà exécuté un canal TLS et que vous souhaitez que la modification soit prise en compte immédiatement, exécutez la commande MQSC REFRESH SECURITY TYPE (SSL).

Configuration du matériel de cryptographie sous IBM i

Utilisez cette procédure pour configurer le coprocesseur de cryptographie sous IBM i

Avant de commencer

Vérifiez que votre profil utilisateur dispose des droits spéciaux *ALLOBJ et *SECADM pour vous permettre de configurer le matériel de coprocesseur.

Procédure

1. Accédez à `http://machine.domain:2001` ou à `https://machine.domain:2010`, où *machine* est le nom de votre ordinateur.

   Une boîte de dialogue s'affiche pour demander un nom d'utilisateur et un mot de passe.
2. Entrez un profil utilisateur et un mot de passe IBM i valides.
3. Accédez à Cryptographie et suivez les liens appropriés pour plus d'informations.

Que faire ensuite

Pour plus d'informations sur la configuration du 4767 Cryptographic Coprocessor, voir 4767 Cryptographic Coprocessor.

ALW Utilisation de SSL/TLS sous AIX, Linux, and Windows

Sur les systèmes AIX, Linux, and Windows , la prise en charge de TLS (Transport Layer Security) est installée avec IBM MQ.

Remarque : Deprecated V 9.4.0 Depuis la IBM MQ 9.4.0, l'utilisation des référentiels de clés et des fichiers de dissimulation CMS avec les applications IBM MQ Java est obsolète. Migrez vers l'utilisation des référentiels de clés PKCS #12 et protégez les mots de passe des référentiels de clés à l'aide du système de protection par mot de passe IBM MQ .

Important : V 9.4.0 V 9.4.0 Depuis la IBM MQ 9.4.0, les référentiels de clés et les fichiers de dissimulation CMS ne sont pas pris en charge avec les canaux AMQP et MQTT qui utilisent SSL/TLS. Utilisez les référentiels de clés PKCS #12 et protégez les mots de passe des référentiels de clés à l'aide du système de protection par mot de passe IBM MQ .

Pour plus d'informations sur les règles de validation de certificat, voir Validation de certificat et conception de règles de confiance.

Pour plus d'informations sur les commandes utilisées pour gérer les référentiels de clés et les certificats sur AIX, Linux, and Windows, voir «Commandes runmqakm et runmqktool sous AIX, Linux, and Windows», à la page 558.

ALW *Configuration d'un référentiel de clés sur AIX, Linux, and Windows*

Procédez comme suit pour créer un nouveau référentiel de clés.

Avant de commencer

Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Avant de créer le référentiel de clés, passez en revue les options fournies par IBM MQ pour stocker de manière sécurisée le mot de passe du référentiel de clés. Pour plus d'informations, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Remarque : Deprecated V 9.4.0 Depuis la IBM MQ 9.4.0, l'utilisation des référentiels de clés et des fichiers de dissimulation CMS avec les applications IBM MQ Java est obsolète. Migrez vers l'utilisation des référentiels de clés PKCS #12 et protégez les mots de passe des référentiels de clés à l'aide du système de protection par mot de passe IBM MQ .

Important : V 9.4.0 V 9.4.0 Depuis la IBM MQ 9.4.0, les référentiels de clés et les fichiers de dissimulation CMS ne sont pas pris en charge avec les canaux AMQP et MQTT qui utilisent SSL/TLS. Utilisez les référentiels de clés PKCS #12 et protégez les mots de passe des référentiels de clés à l'aide

du système de protection par mot de passe IBM MQ . Vous pouvez créer un référentiel de clés PKCS #12 à l'aide de la commande suivante:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

Cette commande crée un fichier de référentiel de clés PKCS #12 nommé *filename*.p12 qui est sécurisé avec le mot de passe spécifié.

Pourquoi et quand exécuter cette tâche

Une connexion TLS requiert un *référentiel de clés* à chaque extrémité de la connexion. Chaque gestionnaire de files d'attente IBM MQ et IBM MQ MQI client doivent avoir accès à un référentiel de clés. Pour plus d'informations, voir «Référentiel de clés SSL/TLS», à la page 26.

Les certificats numériques sont stockés dans le référentiel de clés. Ces certificats numériques comportent des libellés. Le libellé de certificat associe un certificat personnel à un gestionnaire de files d'attente spécifique ou à IBM MQ MQI client. TLS utilise ce certificat à des fins d'authentification. Sur les systèmes AIX, Linux, and Windows , IBM MQ utilise l'une des valeurs suivantes pour le libellé de certificat:

- Valeur de l'attribut de canal ou de gestionnaire de files d'attente **CERTLABL** , s'il est défini.
- La valeur par défaut de ibmwebspheremq, avec le nom du gestionnaire de files d'attente ou l'ID de connexion de l'utilisateur IBM MQ MQI client ajouté, le tout en minuscules.

Pour plus d'informations, voir Labels de certificat numérique.

Le nom du fichier de référentiel de clés comprend un chemin et un nom de radical:

- Sur les systèmes AIX and Linux , le chemin par défaut d'un gestionnaire de files d'attente (défini lors de la création du gestionnaire de files d'attente) est /var/mqm/qmgrs/*queue_manager_name*/ssl.

  Sur les systèmes Windows , le chemin par défaut est *MQ_DATA_PATH*\qmgrs*queue_manager_name*\ssl, où *MQ_DATA_PATH* est le chemin de données sélectionné lors de l'installation de IBM MQ. Par exemple, C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl.

  Le nom de fichier par défaut est key.kdb. Vous pouvez également utiliser votre propre chemin et nom de fichier.

  Si vous choisissez votre propre chemin ou nom de fichier, définissez les droits d'accès au fichier pour contrôler étroitement l'accès à ce dernier.
- Pour un client IBM MQ , il n'existe pas de chemin ou de nom de fichier par défaut. Contrôler étroitement l'accès à ce fichier.

Ne créez pas de référentiels de clés sur un système de fichiers qui ne prend pas en charge les verrous de niveau fichier, par exemple NFS version 2 sur les systèmes Linux .

Pour plus d'informations sur la vérification et la spécification du nom de fichier de la base de données de clés, voir «Modification de l'emplacement du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows», à la page 309 . Vous pouvez spécifier le nom du fichier de la base de données de clés avant ou après la création du référentiel de clés.

Vous pouvez utiliser les commandes **runmqakm** (GSKCapiCmd) ou V 9.4.0 V 9.4.0 **runmqktool** (keytool) pour gérer les référentiels de clés utilisés par IBM MQ. Pour plus d'informations, voir «Commandes runmqakm et runmqktool sous AIX, Linux, and Windows», à la page 558.

L'ID utilisateur qui exécute les commandes de gestion du référentiel de clés doit disposer d'un droit d'accès en écriture sur le répertoire dans lequel le fichier de référentiel de clés est créé ou mis à jour. Pour un gestionnaire de files d'attente qui utilise le répertoire ssl par défaut, l'ID utilisateur qui exécute la commande **runmqakm** ou **runmqktool** doit être membre du groupe mqm. Pour un IBM MQ MQI client, si vous exécutez **runmqakm** ou **runmqktool** à partir d'un ID utilisateur différent de l'ID utilisateur qui exécute le client, vous devez modifier les droits d'accès aux fichiers pour permettre à IBM MQ MQI client d'accéder au référentiel de clés. Pour plus d'informations, voir «Accès et sécurisation de vos fichiers de la

base de données clé sous Windows», à la page 307 ou «Accès et sécurisation de vos fichiers de la base de données clé sous les systèmes AIX and Linux», à la page 307.

Vous pouvez créer un référentiel de clés vide à l'aide de la commande **runmqakm** .

V 9.4.0 V 9.4.0 Si vous utilisez la commande **runmqktool** à la place, le référentiel de clés est créé lorsqu'une commande est émise pour créer ou importer un certificat.

Remarque : Si vous devez gérer les certificats TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Procédure

1. Exécutez la commande suivante pour créer un référentiel de clés avec la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-type *type*
V 9.4.0 V 9.4.0 Indique le type de référentiel de clés. Pour un référentiel de clés utilisé par IBM MQ, les valeurs possibles sont les suivantes:

- pkcs12
- Deprecated cms

  Remarque : Depuis la IBM MQ 9.4.0, l'utilisation des référentiels de clés et des fichiers de dissimulation CMS est obsolète pour les applications IBM MQ Java et n'est pas prise en charge pour les canaux AMQP et MQTT qui utilisent SSL/TLS.

-stash
Facultatif. Spécifiez cette option pour stocker le mot de passe du référentiel de clés dans un fichier de dissimulation. Vous n'avez pas besoin de stocker le mot de passe dans un fichier de dissimulation si vous chiffrez le mot de passe à l'aide du système de protection par mot de passe IBM MQ .

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

-forte
Vérifie que le mot de passe entré répond aux exigences minimales de puissance de mot de passe. Les exigences minimales pour un mot de passe sont les suivantes:

- Le mot de passe doit avoir une longueur minimale de 14 caractères.
- Le mot de passe doit contenir au moins un caractère minuscule, un caractère majuscule et un chiffre ou un caractère spécial. Les caractères spéciaux incluent l'astérisque (*), le signe dollar ($), le signe nombre (#) et le signe pourcentage (%). Un espace est classé comme un caractère spécial.
- Chaque caractère peut apparaître au maximum trois fois dans un mot de passe.
- Un maximum de deux caractères consécutifs dans le mot de passe peut être identique.
- Tous les caractères se trouvent dans le jeu de caractères ASCII imprimables standard, compris entre 0x20 et 0x7E.

2. Définissez les droits d'accès pour les fichiers de référentiel de clés comme décrit dans «Accès et sécurisation de vos fichiers de la base de données clé sous Windows», à la page 307 ou «Accès et sécurisation de vos fichiers de la base de données clé sous les systèmes AIX and Linux», à la page 307.

   Sous Windows, par défaut, seul l'ID utilisateur qui a exécuté la commande de création du référentiel de clés est autorisé à lire le fichier de dissimulation (`.sth`). Une fois qu'un fichier de dissimulation a été créé à l'aide de la commande **runmqakm** , vérifiez les droits d'accès aux fichiers et accordez des droits d'accès au compte de service exécutant le gestionnaire de files d'attente ou à un groupe tel que `mqmlocal`.
3. Si vous n'utilisez pas de fichier de dissimulation, indiquez le mot de passe du magasin de clés au gestionnaire de files d'attente ou à l'application client en suivant les instructions de la rubrique «Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows», à la page 309 ou «Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows», à la page 311.

Que faire ensuite

Ajoutez des certificats d'autorité de certification (CA) par défaut au référentiel de clés vide, si nécessaire. Pour plus d'informations, voir «Ajout de certificats d'autorité de certification par défaut dans un référentiel de clés vide sous AIX, Linux, and Windows», à la page 308.

ALW *Génération de mots de passe fiables pour la protection des référentiels de clés sous AIX, Linux, and Windows*

Vous pouvez générer des mots de passe fiables pour la protection du référentiel de clés à l'aide de la commande **runmqakm** (GSKCapiCmd).

Vous pouvez utiliser la commande **runmqakm** avec les paramètres suivants pour générer un mot de passe fiable:

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

où *password_length* est la longueur du mot de passe à générer. La longueur minimale du mot de passe pouvant être indiquée est 14.

Lorsque vous utilisez le mot de passe généré dans le paramètre **-pw** des commandes d'administration de certificat suivantes, placez toujours le mot de passe entre guillemets. Sur les systèmes AIX and Linux , vous devez également utiliser une barre oblique inversée pour échapper les caractères suivants s'ils apparaissent dans la chaîne de mot de passe:

```
!  \  "  '
```

Lorsque vous entrez un mot de passe de référentiel de clés en réponse à une invite de la commande **runmqakm** ou V 9.4.0 V 9.4.0 **runmqktool** , il n'est pas nécessaire de placer le mot de passe entre guillemets ou en échappement car l'interpréteur de commandes du système d'exploitation n'affecte pas la saisie de données dans ces cas.

ALW *Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows*

Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

Les composants et fonctions IBM MQ suivants prennent en charge deux méthodes différentes de stockage des mots de passe de référentiel de clés:

- Référentiel de clés TLS du gestionnaire de files d'attente.
- IBM MQ MQI clients qui utilisent TLS.

- V 9.4.0 La configuration Native HA dans la section **NativeHALocalInstance** du fichier `qm.ini` .
- V 9.4.0 Configuration de l'authentification par jeton dans la section **AuthToken** du fichier `qm.ini` .

Les mots de passe de référentiel de clés à utiliser par ces composants peuvent être chiffrés et stockés à l'aide de l'une des méthodes suivantes:

Système de protection par mot de passe IBM MQ .

Chaque composant IBM MQ fournit une commande permettant de chiffrer le mot de passe du référentiel de clés. La commande chiffrée générée par la commande est stockée dans un fichier.

Pour le référentiel de clés TLS du gestionnaire de files d'attente, le mot de passe est chiffré lorsque l'attribut de gestionnaire de files d'attente **SSLKEYRPWD** est défini.

Le mot de passe est chiffré avec l'algorithme AES-128 . Les détails de cet algorithme sont connus du public et sont considérés comme sécurisés.

Le mot de passe est stocké dans un format propriétaire qui n'est pas compris par les autres logiciels pouvant accéder au référentiel de clés.

Un mot de passe chiffré par un composant IBM MQ ne peut pas être utilisé par un autre composant IBM MQ .

Une clé de chiffrement unique peut être fournie lorsque le mot de passe du référentiel de clés est chiffré. Une clé de chiffrement unique empêche toute personne qui n'a pas accès à la clé de chiffrement de pouvoir déchiffrer le mot de passe.

Le mot de passe du référentiel de clés en texte clair est nécessaire pour gérer les certificats qui se trouvent dans le référentiel de clés. En plus de chiffrer le mot de passe du référentiel de clés à l'aide du système de protection par mot de passe IBM MQ , vous devez également stocker le mot de passe du référentiel de clés dans un emplacement sécurisé où il est accessible à cette fin.

Pour plus d'informations sur le système de protection par mot de passe IBM MQ , voir «Protection des mots de passe dans les fichiers de configuration du composant IBM MQ», à la page 582.

Un fichier de dissimulation de référentiel de clés.

La commande **runmqakm** peut stocker le mot de passe du référentiel de clés dans un fichier de dissimulation.

Le mot de passe est chiffré à l'aide d'une méthode propriétaire spécifique au fournisseur cryptographique de IBM MQ, IBM Global Security Kit (GSKit).

Une clé de chiffrement unique ne peut pas être fournie.

Le mot de passe chiffré est stocké dans un fichier de dissimulation dans le même répertoire que le fichier de référentiel de clés.

Toute personne disposant d'un accès en lecture au référentiel de clés et au fichier de dissimulation peut accéder au contenu du référentiel de clés et le gérer.

Remarque : Deprecated V 9.4.0 Depuis la IBM MQ 9.4.0, l'utilisation des fichiers de dissimulation avec les applications IBM MQ Java est obsolète.

Important : V 9.4.0 V 9.4.0 Depuis la IBM MQ 9.4.0, les fichiers de dissimulation ne sont pas pris en charge par les canaux AMQP et MQTT qui utilisent TLS.

Quelle que soit la méthode que vous choisissez pour chiffrer le mot de passe du référentiel de clés, veillez à connaître les limitations du chiffrement des mots de passe stockés. Pour plus d'informations, voir «Limites de la protection via le chiffrement de mot de passe», à la page 589.

Concepts associés

«Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows», à la page 309

Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

«Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows», à la page 311
Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

«Utilisation de SSL/TLS sous AIX, Linux, and Windows», à la page 302
Sur les systèmes AIX, Linux, and Windows , la prise en charge de TLS (Transport Layer Security) est installée avec IBM MQ.

Windows *Accès et sécurisation de vos fichiers de la base de données clé sous Windows*

Il se peut que les fichiers de la base de données de clés ne disposent pas des droits d'accès appropriés. Vous devez définir l'accès approprié à ces fichiers.

Définissez le contrôle d'accès aux fichiers *key*.p12, *key*.kdb, *key*.sth, *key*.crlet *key*.rdb, où *key* est le nom du radical de votre base de données de clés, pour accorder des droits à un ensemble restreint d'utilisateurs.

Si vous avez utilisé une extension de référentiel de clés autre que .p12 ou .kdb, vous devez également vous assurer que les droits de ce fichier sont définis.

Envisagez d'accorder l'accès comme suit:

entièrement habilité
BUILTIN\Administrators, NT AUTHORITY\SYSTEM et l'utilisateur qui a créé les fichiers base de données.

droit de lecture
Pour un gestionnaire de files d'attente, le groupe mqm local uniquement. Cela suppose que l'agent MCA s'exécute sous un ID utilisateur dans le groupe mqm.

Pour un client, ID utilisateur sous lequel le processus client est exécuté.

Linux AIX *Accès et sécurisation de vos fichiers de la base de données clé sous les systèmes AIX and Linux*

Il se peut que les fichiers de la base de données de clés ne disposent pas des droits d'accès appropriés. Vous devez définir l'accès approprié à ces fichiers.

Pour un gestionnaire de files d'attente, définissez les droits d'accès aux fichiers de la base de données de clés de sorte que les processus de gestionnaire de files d'attente et de canal puissent les lire si nécessaire, mais que les autres utilisateurs ne puissent pas les lire ou les modifier. Normalement, l'utilisateur mqm a besoin de droits d'accès en lecture. Si vous avez créé le fichier de la base de données de clés en vous connectant en tant qu'utilisateur mqm, les droits sont probablement suffisants ; si vous n'étiez pas l'utilisateur mqm, mais un autre utilisateur du groupe mqm, vous devrez probablement accorder des droits de lecture à d'autres utilisateurs du groupe mqm.

De même pour un client, définissez des droits sur les fichiers de la base de données de clés afin que les processus de l'application client puissent les lire lorsque cela est nécessaire, mais les autres utilisateurs ne peuvent pas les lire ou les modifier. Normalement, l'utilisateur sous lequel le processus client s'exécute a besoin de droits de lecture. Si vous avez créé le fichier de la base de données de clés en vous connectant en tant que cet utilisateur, les droits sont probablement suffisants ; si vous n'étiez pas l'utilisateur du processus client, mais un autre utilisateur de ce groupe, vous devrez probablement accorder des droits de lecture à d'autres utilisateurs du groupe.

Définissez les droits d'accès aux fichiers *key*.p12, *key*.kdb, *key*.sth, *key*.crlet *key*.rdb, où *key* est le nom de racine de votre base de données de clés, sur read et write pour le propriétaire du fichier, et sur read pour le groupe d'utilisateurs mqm ou client (-rw-r -----).

Si vous avez utilisé une extension de référentiel de clés autre que .p12 ou .kdb, vous devez également vous assurer que les droits de ce fichier sont définis.

ALW *Ajout de certificats d'autorité de certification par défaut dans un référentiel de clés vide sous AIX, Linux, and Windows*

Suivez cette procédure pour ajouter un ou plusieurs certificats de l'autorité de certification par défaut à un référentiel de clés vide.

Lorsque vous créez un référentiel de clés, il est vide. Vous pouvez ajouter des certificats d'autorité de certification par défaut à un référentiel de clés à l'aide de la commande **runmqakm** .

Utilisation runmqakm

Exécutez la commande suivante pour ajouter des certificats d'autorité de certification par défaut à un référentiel de clés à l'aide de la commande **runmqakm** :

```
runmqakm -cert -populate -db filename -pw password
```

où :

-db ***nom_fichier***
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw ***mot_de_passe***
Indique le mot de passe du référentiel de clés.

Remarque : IBM MQ fait confiance à tous les certificats signés par les certificats de l'autorité de certification dans votre référentiel de clés. Réfléchissez soigneusement aux autorités de certification que vous souhaitez accréditer et ajoutez uniquement les certificats de l'autorité de certification nécessaires pour authentifier vos clients et vos gestionnaires de files d'attente. Il n'est pas recommandé d'ajouter l'ensemble complet de certificats de l'autorité de certification par défaut à un référentiel de clés.

ALW ***Localisation du référentiel de clés d'un gestionnaire de files d'attente sous AIX, Linux, and Windows***

Utilisez cette procédure pour obtenir l'emplacement du fichier de base de données de clés de votre gestionnaire de files d'attente

Procédure

1. Affichez les attributs de votre gestionnaire de files d'attente à l'aide de l'une des commandes MQSC suivantes:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

   Vous pouvez également afficher les attributs de votre gestionnaire de files d'attente à l'aide des commandes IBM MQ Explorer ou PCF.
2. Recherchez dans le résultat de la commande le chemin et le nom de la racine du fichier de la base de données de clés.
   Exemple :

   a. sous AIX and Linux: `/var/mqm/qmgrs/QM1/ssl/key`, où `/var/mqm/qmgrs/QM1/ssl` est le chemin d'accès et `key` est le nom de la racine

   b. sous Windows: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`, où *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` est le chemin et `key` est le nom de la racine. *MQ_INSTALLATION_PATH* représente le répertoire de haut niveau dans lequel IBM MQ est installé.

   Remarque : Depuis IBM MQ 9.3.0 , la zone SSLKEYR prend en charge à la fois un nom de fichier complet (y compris l'extension) et un nom de radical (sans extension). Si un nom de radical est défini, IBM MQ ajoute automatiquement `.kdb` et utilise ce référentiel de clés.

ALW *Modification de l'emplacement du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows*

Vous pouvez modifier l'emplacement du fichier de la base de données de clés de votre gestionnaire de files d'attente en utilisant divers moyens, notamment la commande MQSC ALTER QMGR.

Vous pouvez modifier l'emplacement du fichier de base de données de clés de votre gestionnaire de files d'attente à l'aide de la commande MQSC ALTER QMGR pour définir l'attribut de référentiel de clés de votre gestionnaire de files d'attente. Par exemple, sous AIX and Linux:

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

Sous Windows :

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

Avertissement : Sous Windows et Linux, si des canaux TLS AMQP sont utilisés, le suffixe du fichier de référentiel de clés doit être l'un des suivants:

- `.kdb`, pour un référentiel de clés CMS
- `.p12` ou `.pkcs12`, pour un référentiel de clés PKCS #12 .

Vous pouvez également modifier les attributs de votre gestionnaire de files d'attente à l'aide des commandes IBM MQ Explorer ou PCF.

Lorsque vous modifiez l'emplacement du fichier de base de données de clés d'un gestionnaire de files d'attente, les certificats ne sont pas transférés à partir de l'ancien emplacement. Si le fichier de base de données de clés auquel vous accédez est un nouveau fichier de base de données de clés, vous devez le remplir avec les certificats de l'autorité de certification et les certificats personnels dont vous avez besoin, comme décrit dans «Importation d'un certificat personnel dans un référentiel de clés sous AIX, Linux, and Windows», à la page 570.

Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows

Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

IBM MQ fournit deux mécanismes pour fournir le mot de passe du référentiel de clés à un gestionnaire de files d'attente:

- «Attribut KEYRPWD», à la page 309
- «Fichier de dissimulation du référentiel de clés», à la page 310

Si vous n'utilisez pas de fichier de dissimulation de référentiel de clés, le mot de passe du référentiel de clés est chiffré à l'aide du système de protection par mot de passe IBM MQ . Pour plus d'informations sur les méthodes de protection du mot de passe du référentiel de clés, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Attribut KEYRPWD

Pour fournir un mot de passe de référentiel de clés directement au gestionnaire de files d'attente, exécutez la commande MQSC suivante en remplaçant *password* par votre mot de passe de référentiel de clés:

```
ALTER QMGR KEYRPWD('password')
```

Avertissement : Veillez à placer le mot de passe entre apostrophes, sinon IBM MQ convertit les caractères en majuscules.

Lorsqu'un mot de passe de référentiel de clés est spécifié à l'aide de cette méthode, le mot de passe est chiffré à l'aide du système de protection par mot de passe IBM MQ avant d'être stocké.

Une clé de chiffrement, appelée clé initiale, est utilisée pour chiffrer le mot de passe. Définissez le gestionnaire de files d'attente pour qu'il utilise une clé initiale unique afin de protéger le mot de passe de manière sécurisée. Si vous ne fournissez pas de clé initiale, la clé par défaut est utilisée.

Vérifiez que le gestionnaire de files d'attente est configuré avec une clé initiale unique avant de définir le mot de passe du référentiel de clés. Vous pouvez modifier la clé initiale à l'aide de l'attribut **INITKEY** de la commande **ALTER QMGR** . Exemple :

```
ALTER QMGR INITKEY('mykey')
```

Avertissement : La modification de la clé initiale après la définition du mot de passe du référentiel de clés n'entraîne pas le chiffrement du mot de passe du référentiel de clés avec la nouvelle clé initiale. La modification de la clé initiale sans réinitialisation du mot de passe du référentiel de clés empêche IBM MQ de déchiffrer le mot de passe du référentiel de clés et, par conséquent, d'accéder au référentiel de clés.

Pour plus d'informations sur l'attribut **KEYRPWD** , voir KEYRPWD.

Fichier de dissimulation du référentiel de clés

Si aucun mot de passe de référentiel de clés n'est fourni au gestionnaire de files d'attente à l'aide de l'attribut **KEYRPWD** , IBM MQ suppose qu'un fichier de dissimulation existe dans le même répertoire que le référentiel de clés. Le fichier de dissimulation a le même nom de radical que le référentiel de clés, mais possède l'extension `.sth` .

Un fichier de dissimulation de référentiel de clés est créé en même temps que le référentiel de clés, ou version ultérieure, en tant que commande **runmqakm** distincte.

Avertissement : Le format du fichier de dissimulation est spécifique au IBM MQ fournisseur cryptographique IBM Global Security Kit (GSKit)et n'est pas disponible sur les plateformes qui utilisent un fournisseur cryptographique différent.

Pour créer un fichier de dissimulation lors de la création du référentiel de clés, spécifiez le paramètre **-stash** . Exemple :

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

où *passw0rd* est le mot de passe du référentiel de clés.

Pour créer un fichier de dissimulation ultérieurement, exécutez la commande suivante:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

où *passw0rd* est le mot de passe du référentiel de clés.

Concepts associés

«Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305
Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

«Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows», à la page 311
Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

ALW *Localisation du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows*

L'emplacement du référentiel de clés est indiqué par la variable MQSSLKEYR ou spécifié dans l'appel MQCONNX.

Examinez la variable d'environnement MQSSLKEYR pour trouver l'emplacement du fichier de la base de données de clés pour votre IBM MQ MQI client. Exemple :

```
echo $MQSSLKEYR
```

Vérifiez également votre application, car le nom de fichier de la base de données de clés peut également être défini dans un appel MQCONNX, comme décrit dans«Spécification de l'emplacement du référentiel de clés pour un IBM MQ MQI client sous AIX, Linux, and Windows», à la page 311. La valeur définie dans un appel MQCONNX remplace la valeur de MQSSLKEYR.

ALW *Spécification de l'emplacement du référentiel de clés pour un IBM MQ MQI client sous AIX, Linux, and Windows*

Il n'existe pas de référentiel de clés par défaut pour un IBM MQ MQI client. Vous pouvez spécifier son emplacement de deux manières. Assurez-vous que le fichier de la base de données de clés est accessible uniquement par les utilisateurs ou les administrateurs prévus afin d'empêcher toute copie non autorisée vers d'autres systèmes.

Vous pouvez spécifier l'emplacement du fichier de base de données de clés pour votre IBM MQ MQI client de deux manières:

- Définition de la variable d'environnement MQSSLKEYR. Par exemple, sous AIX and Linux:

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

  Sous Windows :

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

- Indiquez le chemin et le nom de la racine du fichier de la base de données de clés dans la zone *KeyRepository* de la structure MQSCO lorsqu'une application effectue un appel MQCONNX. Pour plus d'informations sur l'utilisation de la structure MQSCO dans MQCONNX, voir Présentation de MQSCO.

Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows

Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

IBM MQ fournit quatre mécanismes pour fournir le mot de passe du référentiel de clés à un IBM MQ MQI client:

- «Les zones KeyRepoPassword de MQSCO », à la page 312
- «Variable d'environnement MQKEYRPWD», à la page 312
- «Attribut SSLKeyRepositoryPassword du fichier de configuration du client», à la page 312
- «Fichier de dissimulation du référentiel de clés», à la page 313

Si vous n'utilisez pas de fichier de dissimulation de référentiel de clés, vous pouvez fournir le mot de passe de référentiel de clés sous la forme d'une chaîne de texte en clair ou d'une chaîne chiffrée à l'aide du système de protection par mot de passe IBM MQ . Pour plus d'informations sur les méthodes de protection du mot de passe du référentiel de clés, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Les zones KeyRepoPassword de MQSCO

Pour fournir un mot de passe de référentiel de clés à l'aide de la structure MQSCO, vous devez utiliser une combinaison des trois zones de chaîne de variable suivantes:

KeyRepoPasswordLength
Longueur du mot de passe.

KeyRepoPasswordPtr
Pointeur vers l'emplacement en mémoire qui contient le mot de passe.

KeyRepoPasswordOffset
Emplacement du mot de passe en mémoire, représenté en nombre d'octets depuis le début de la structure MQSCO.

Remarque : Vous ne pouvez indiquer qu'un seul **KeyRepoPasswordPtr** ou **KeyRepoPasswordOffset**.

Exemple :

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Avertissement : Si vous fournissez le mot de passe à l'aide de cette méthode, chiffrez le mot de passe avant qu'il ne soit fourni à l'application IBM MQ client . Pour plus d'informations, voir «Chiffrement du mot de passe du référentiel de clés», à la page 313.

Pour plus d'informations sur la structure MQCSO, voir MQSCO-Options de configuration SSL/TLS.

Variable d'environnement *MQKEYRPWD*

Si un mot de passe de référentiel de clés n'est pas fourni au client à l'aide de la structure MQSCO, vous pouvez spécifier le mot de passe de référentiel de clés à l'aide de la variable d'environnement *MQKEYRPWD* . Exemple :

```
export MQKEYRPWD=passw0rd
```

ou

```
set MQKEYRPWD=passw0rd
```

où `passw0rd` est votre mot de passe.

Avertissement : Si vous fournissez le mot de passe à l'aide de cette méthode, chiffrez-le avant de définir la valeur de la variable d'environnement. Pour plus d'informations, voir «Chiffrement du mot de passe du référentiel de clés», à la page 313.

Attribut SSLKeyRepositoryPassword du fichier de configuration du client

Si un mot de passe de référentiel de clés n'est pas fourni au client à l'aide de l'une des autres méthodes, vous pouvez spécifier le mot de passe de référentiel de clés à l'aide de l'attribut **SSLKeyRepositoryPassword** dans la section **SSL** du fichier de configuration du client. Exemple :

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Avertissement : Si vous fournissez le mot de passe à l'aide de cette méthode, chiffrez-le avant de définir la valeur de l'attribut **SSLKeyRepositoryPassword** . Pour plus d'informations, voir «Chiffrement du mot de passe du référentiel de clés», à la page 313.

Pour plus d'informations sur la strophe SSL du fichier de configuration du client, voir Strophe SSL du fichier de configuration du client.

Fichier de dissimulation du référentiel de clés

Si le mot de passe du référentiel de clés n'est pas fourni au client à l'aide de l'une des autres méthodes, IBM MQ suppose qu'un fichier de dissimulation existe dans le même répertoire que le référentiel de clés. Le fichier de dissimulation a le même nom de radical que le référentiel de clés, mais possède l'extension `.sth` .

Un fichier de dissimulation de référentiel de clés est créé en même temps que le référentiel de clés, ou version ultérieure, à l'aide d'une commande **runmqakm** distincte.

Avertissement : Le format du fichier de dissimulation est spécifique au IBM MQ fournisseur cryptographique IBM Global Security Kit (GSKit)et n'est pas disponible sur les plateformes qui utilisent un fournisseur cryptographique différent.

Pour créer un fichier de dissimulation lors de la création du référentiel de clés, spécifiez le paramètre **-stash** . Exemple :

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

où *passw0rd* est le mot de passe du référentiel de clés.

Pour créer un fichier de dissimulation ultérieurement, exécutez la commande suivante:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

où *passw0rd* est le mot de passe du référentiel de clés.

Chiffrement du mot de passe du référentiel de clés

Si vous fournissez le mot de passe du référentiel de clés à l'aide d'une méthode autre qu'un fichier de dissimulation, chiffrez le mot de passe à l'aide du système de protection par mot de passe IBM MQ . Pour chiffrer le mot de passe, exécutez la commande **runmqicred** . Entrez le mot de passe du référentiel de clés lorsque vous y êtes invité. La commande génère le mot de passe chiffré. Le mot de passe chiffré peut être fourni à IBM MQ MQI client à la place du mot de passe en texte en clair à l'aide de l'une des méthodes décrites.

Une clé de chiffrement, appelée clé initiale, est utilisée pour chiffrer le mot de passe. Lorsque vous chiffrez le mot de passe, utilisez une clé initiale unique pour protéger le mot de passe de manière sécurisée. Pour fournir votre propre clé initiale, utilisez le paramètre **-sf** dans la commande **runmqicred** . Si vous ne fournissez pas de clé initiale, la clé par défaut est utilisée.

Pour plus d'informations, voir runmqicred (protection des mots de passe du client IBM MQ).

Si vous fournissez votre propre clé initiale lorsque le mot de passe du référentiel de clés est chiffré et que vous fournissez le mot de passe chiffré au IBM MQ MQI client, vous devez également vous assurer que vous fournissez la même clé initiale au IBM MQ MQI client. Pour plus d'informations sur la façon de fournir la clé initiale à un IBM MQ MQI client, voir «Fourniture d'une clé initiale pour un IBM MQ MQI client sous AIX, Linux, and Windows», à la page 314.

Concepts associés

«Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305
Plusieurs composants IBM MQ doivent accéder à un référentiel de clés qui contient des certificats numériques ou des clés symétriques. Un référentiel de clés est sécurisé avec un mot de passe car il contient des informations sensibles. Le mot de passe du référentiel de clés doit être stocké dans un emplacement où IBM MQ peut le lire lors de l'accès au référentiel de clés. Le mot de passe doit également être chiffré pour réduire les risques d'accès non autorisé au référentiel de clés.

«Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows», à la page 309

Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

ALW *Fourniture d'une clé initiale pour un IBM MQ MQI client sous AIX, Linux, and Windows*
Si vous fournissez des variables à un IBM MQ MQI client qui ont été chiffrées à l'aide du système de protection par mot de passe IBM MQ , vous devrez peut-être fournir la clé initiale correspondante qui a été utilisée pour chiffrer la valeur.

Si vous n'avez pas spécifié de clé initiale lors du chiffrement de la valeur, vous n'avez pas besoin de fournir de valeur de clé initiale à IBM MQ client. Toutefois, si vous avez utilisé une clé initiale unique, vous pouvez la fournir à IBM MQ client à l'aide des méthodes suivantes:

- «Fourniture de la clé initiale à l'aide de la structure MQCSP», à la page 314
- «Fourniture de la clé initiale à l'aide de la variable d'environnement MQS_MQI_KEYFILE», à la page 314
- «Fourniture de la clé initiale à l'aide du fichier de configuration du client», à la page 315

Fourniture de la clé initiale à l'aide de la structure MQCSP

Pour fournir la clé initiale à l'aide de la structure MQCSP, vous devez utiliser une combinaison des trois zones de chaîne de variable suivantes:

InitialKeyLength
Longueur de la clé initiale

InitialKeyPtr
Un pointeur vers l'emplacement en mémoire contenant la clé initiale

InitialKeyOffset
Emplacement de la clé initiale en mémoire, représenté en nombre d'octets à partir du début de la structure MQCSP.

Remarque : Vous ne pouvez indiquer qu'un seul **InitialKeyPtr** ou **InitialKeyOffset**.

Exemple :

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

Fourniture de la clé initiale à l'aide de la variable d'environnement MQS_MQI_KEYFILE

Si aucune clé initiale n'est fournie au client à l'aide de la structure MQCSP, IBM MQ vérifie la variable d'environnement *MQS_MQI_KEYFILE* . Vous devez définir cette variable d'environnement à l'emplacement d'un fichier contenant une seule ligne de texte, constituée de la clé initiale que vous souhaitez utiliser.

Par exemple, si un fichier appelé `mykey.key` existe dans le répertoire racine et qu'il contient la clé initiale, vous devez définir la variable d'environnement comme suit:

```
export MQS_MQI_KEYFILE=/mykey.key
```

ou

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

Fourniture de la clé initiale à l'aide du fichier de configuration du client

Si aucune clé initiale n'est fournie au client à l'aide d'un mécanisme précédent, IBM MQ vérifie l'attribut **MQIInitialKeyFile** de la section Sécurité du fichier `mqclient.ini` . Vous devez définir cet attribut sur l'emplacement d'un fichier contenant une seule ligne de texte, constituée de la clé initiale que vous souhaitez utiliser.

Par exemple, si un fichier nommé `mykey.key` existe dans le répertoire racine et qu'il contient la clé initiale, le fichier de configuration du client doit contenir ce qui suit:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

Concepts associés

«Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows», à la page 311
Comme le référentiel de clés contient des informations sensibles, il est sécurisé avec un mot de passe. Pour pouvoir accéder au contenu du référentiel de clés afin d'effectuer des opérations TLS, IBM MQ doit pouvoir extraire le mot de passe du référentiel de clés.

«Utilisation de SSL/TLS», à la page 283
Ces rubriques fournissent des instructions pour l'exécution de tâches uniques liées à l'utilisation de TLS avec IBM MQ.

ALW *Lorsque les modifications apportées aux certificats ou au référentiel de clés prennent effet sur AIX, Linux, and Windows*

Lorsque vous modifiez les certificats dans un référentiel de clés ou l'emplacement du référentiel de clés, les modifications prennent effet à un moment qui dépend du type de canal et de la façon dont le canal s'exécute.

Les modifications apportées aux certificats dans le référentiel de clés ou à l'emplacement du référentiel de clés prennent effet dans les cas suivants:

- Lorsqu'un nouveau processus de canal unique sortant exécute pour la première fois un canal TLS.
- Lorsqu'un nouveau processus de canal unique TCP/IP entrant reçoit pour la première fois une demande de démarrage d'un canal TLS.
- Lorsque la commande MQSC **REFRESH SECURITY TYPE(SSL)** est émise pour actualiser l'environnement TLS.
- Pour les processus d'application client, lorsque la dernière connexion TLS du processus est fermée. La prochaine connexion TLS prendra en compte les modifications apportées au certificat.
- Pour les canaux qui s'exécutent en tant qu'unités d'exécution d'un processus de regroupement de processus (amqrmppa), lorsque le processus de regroupement de processus est démarré ou redémarré et exécute d'abord un canal TLS. Si le processus de regroupement de processus a déjà exécuté un canal TLS et que vous souhaitez que la modification soit prise en compte immédiatement, exécutez la commande MQSC **REFRESH SECURITY TYPE(SSL)**.
- Pour les canaux qui s'exécutent en tant qu'unités d'exécution de l'initiateur de canal, lorsque l'initiateur de canal est démarré ou redémarré et qu'il exécute d'abord un canal TLS. Si le processus initiateur de canal a déjà exécuté un canal TLS et que vous souhaitez que la modification soit prise en compte immédiatement, exécutez la commande MQSC **REFRESH SECURITY TYPE(SSL)**.
- Pour les canaux qui s'exécutent en tant qu'unités d'exécution d'un programme d'écoute TCP/IP, lorsque le programme d'écoute est démarré ou redémarré et qu'il reçoit d'abord une demande de démarrage d'un canal TLS. Si le programme d'écoute a déjà exécuté un canal TLS et que vous souhaitez que la modification soit prise en compte immédiatement, exécutez la commande MQSC **REFRESH SECURITY TYPE(SSL)**.

Vous pouvez également actualiser l'environnement TLS IBM MQ à l'aide des commandes IBM MQ Explorer ou PCF.

Important : Les modifications apportées au fichier de configuration du magasin de clés ou au magasin de clés utilisé par un intercepteur MCA Advanced Message Security (AMS) ou un client AMS prennent effet au redémarrage du gestionnaire de files d'attente ou de l'application.

ALW *Configuration du matériel de cryptographie sous AIX, Linux, and Windows*

Vous pouvez configurer le matériel de cryptographie pour un gestionnaire de files d'attente ou un client de plusieurs manières.

Vous pouvez configurer le matériel de cryptographie pour un gestionnaire de files d'attente sur AIX, Linux, and Windows à l'aide de l'une des méthodes suivantes:

- Utilisez la commande **ALTER QMGR** MQSC avec le paramètre SSLCRYP , comme décrit dans ALTER QMGR.
- Utilisez IBM MQ Explorer pour configurer le matériel de cryptographie sur votre système AIX, Linux, and Windows . Pour plus d'informations, reportez-vous à l'aide en ligne.

Vous pouvez configurer le matériel de cryptographie pour un client IBM MQ sous AIX, Linux, and Windows à l'aide de l'une des méthodes suivantes:

- Définissez la variable d'environnement **MQSSLCRYP** . Les valeurs admises pour **MQSSLCRYP** sont les mêmes que pour le paramètre **SSLCRYP** , comme décrit dans ALTER QMGR. Pour définir cette variable d'environnement, utilisez l'une des commandes suivantes:
  - Linux AIX Sur les systèmes AIX and Linux :

    ```
    export MQSSLCRYP=string
    ```

  - Windows Sur les systèmes Windows :

    ```
    SET MQSSLCRYP=string
    ```

  où *string* représente la chaîne de paramètres à utiliser pour configurer le matériel de cryptographie présent sur le système.

  Si vous utilisez la version GSK_PKCS11 du paramètre **SSLCRYP** , le libellé de jeton PKCS #11 doit correspondre à celui avec lequel vous avez configuré votre matériel.
- Définissez l'attribut **SSLCryptoHardware** dans la section SSL du fichier de configuration IBM MQ client . Les valeurs admises sont les mêmes que pour le paramètre **SSLCRYP** , comme décrit dans **ALTER QMGR**.

  Si vous utilisez la version GSK_PKCS11 du paramètre **SSLCRYP** , le libellé de jeton PKCS #11 doit correspondre à celui avec lequel vous avez configuré votre matériel.
- Définissez la zone **CryptoHardware** de la structure des options de configuration SSL, MQSCO, sur un appel MQCONNX. Pour plus d'informations, voir Présentation de MQSCO.

Avertissement : >Lorsque vous fournissez la configuration du matériel de cryptographie via la variable d'environnement **MQSSLCRYP** ou l'attribut **SSLCryptoHardware** , vous devez protéger le mot de passe avant de le stocker. Pour plus d'informations, voir «IBM MQ clients qui utilisent du matériel de cryptographie», à la page 586.

Si vous avez configuré du matériel cryptographique qui utilise l'interface PKCS #11 à l'aide de l'une de ces méthodes, vous devez stocker le certificat personnel à utiliser sur vos canaux dans le fichier de la base de données de clés pour le jeton cryptographique que vous avez configuré. Ceci est décrit dans «Gestion des certificats sur le matériel PKCS #11», à la page 579.

MQ Appliance Utilisation de SSL/TLS sous IBM MQ Appliance

IBM MQ Appliance prend en charge le protocole TLS (Transport Layer Security).

Le IBM MQ Appliance comporte des commandes distinctes pour la gestion des certificats. Pour des informations détaillées sur la gestion des certificats, voir la documentation IBM MQ Appliance , Gestion des certificats TLS

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 318.

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 *Exigences d'ID utilisateur supplémentaires pour TLS sur z/OS*

Ces informations décrivent les exigences supplémentaires dont votre ID utilisateur a besoin pour configurer et utiliser TLS sur z/OS.

Vérifiez que vous disposez de toutes les mises à jour HIPER (High Impact or ???) appropriées sur votre système.

Si le référentiel de clés appartient à l'ID utilisateur CHINIT, cet ID utilisateur doit disposer d'un accès en lecture à l'IRR IRR.DIGTCERT.LISTRING dans la classe FACILITY, et mise à jour des droits d'accès dans le cas contraire, et accès en lecture à l'IRR IRR.DIGTCERT.LIST . Accordez l'accès à l'aide de la commande PERMIT avec ACCESS(UPDATE) ou ACCESS(READ), selon le cas.

Vérifiez que vous avez configuré les prérequis suivants:

- L'ID utilisateur *ssidCHIN* est correctement défini dans RACFet l'ID utilisateur *ssidCHIN* dispose des droits d'accès appropriés aux profils suivants.
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  Ces variables sont définies dans la classe RACF FACILITY.
- L'ID utilisateur *ssidCHIN* est le propriétaire du fichier de clés.
- Le certificat personnel du gestionnaire de files d'attente, s'il est créé par la commande RACDCERT, est créé avec un ID utilisateur de type de certificat qui est également identique à l'ID utilisateur *ssidCHIN* .
- L'initiateur de canal est recyclé ou la commande **REFRESH SECURITY TYPE(SSL)** est émise pour récupérer les modifications apportées au fichier de clés.
- La procédure de l'initialisateur de canal IBM MQ a accès à la bibliothèque d'exécution SSL système *pdsname*.SIEALNKE via la liste de liens, LPA ou une instruction de définition de données STEPLIB. Cette bibliothèque doit disposer de droits APF.
- L'ID utilisateur sous l'autorité duquel l'initiateur du canal est exécuté est configuré pour utiliserz/OS UNIX System Services (z/OS UNIX), comme décrit dans lez/OS UNIX System Services Planification Documentation.

  Les utilisateurs qui ne souhaitent pas que l'initiateur de canal appelle z/OS UNIX à l'aide de l'ID utilisateur par défaut et du segment OMVS, n'ont besoin que de modéliser un nouveau segment OMVS basé sur le segment par défaut car l'initiateur de canal ne requiert pas de droits spéciaux et ne s'exécute pas dans UNIX en tant que superutilisateur.

Voir les commandes PERMIT dans«Giving the channel initiator the correct access rights on z/OS», à la page 319 pour quelques exemples sur la façon dont vous donnez à l'initiateur du canal l'accès correct.

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 “Référentiel de clés SSL/TLS” on page 26 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 “Certificats numériques” 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 270

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 318.
- *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 321. 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 “Labels de certificat numérique, compréhension des exigences” on page 28 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 322.

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 318.
- *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 318.
- *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 323. 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 “Enregistrements d'authentification de canal” on page 54.

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 322.
- 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 324.

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 “CipherSpecs et CipherSuites dans IBM MQ” on page 43 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 326, 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.

Identification et authentification des utilisateurs

Vous pouvez identifier et authentifier les utilisateurs à l'aide de certificats X.509 , de la structure MQCSP ou de plusieurs types de programme d'exit utilisateur.

Utilisation de certificats X.509

Vous pouvez identifier et authentifier les utilisateurs à l'aide de certificats X.509 avec la commande **SET CHLAUTH** et le paramètre **SSLPEER** . Le paramètre **SSLPEER** spécifie un filtre à utiliser pour la comparaison avec le nom distinctif du sujet du certificat provenant du gestionnaire de files d'attente ou du client homologue à l'autre extrémité du canal.

Pour plus d'informations sur l'utilisation de la commande **SET CHLAUTH** et du paramètre **SSLPEER** , voir SET CHLAUTH.

Les certificats numériques peuvent être révoqués par les autorités de certification. Vous pouvez vérifier le statut de révocation des certificats à l'aide d'OCSP ou de CRL sur les serveurs LDAP, en fonction de la plateforme. Pour plus d'informations, voir «Utilisation des certificats révoqués», à la page 350.

Utilisation de la structure MQCSP

La structure des paramètres de sécurité de connexion MQCSP est spécifiée sur un appel MQCONNX. Cette structure peut contenir des données d'identification fournies par l'application. L'application peut fournir un ID utilisateur et un mot de passe dans la structure MQCSP. Depuis IBM MQ 9.3.4, les applications peuvent également fournir un jeton d'authentification. Si nécessaire, le MQCSP peut être modifié dans un exit de sécurité.

Avertissement : Les données d'identification d'une structure MQCSP sont parfois envoyées sur le réseau en texte en clair. Pour vous assurer que les données d'identification de l'application client sont protégées, voir «Protection par mot de passe MQCSP», à la page 32.

Pour plus d'informations, voir «Identification et authentification des utilisateurs à l'aide de la structure MQCSP», à la page 331 et «Utilisation des jetons d'authentification», à la page 335.

Linux AIX Sous AIX et Linux, l'ID utilisateur et le mot de passe spécifiés dans la structure MQCSP peuvent être authentifiés à l'aide du système d'exploitation ou de la méthode PAM (Pluggable Authentication Method). Le module PAM fournit un mécanisme général d'authentification des utilisateurs qui masque les détails des services. Pour plus d'informations, voir «Utilisation de la méthode PAM (Pluggable Authentication Method)», à la page 362.

Implémentation de l'identification et de l'authentification dans les exits

Vous pouvez identifier et authentifier les utilisateurs à l'aide de plusieurs types de programme d'exit utilisateur. Pour plus d'informations, voir «Implémentation de l'identification et de l'authentification dans les exits de sécurité», à la page 332, «Mappage d'identité dans les exits de message», à la page 333 et «Mappage d'identité dans l'exit d'API et l'exit de croisement d'API», à la page 334.

Utilisateurs privilégiés

Un utilisateur privilégié est un utilisateur disposant de droits d'administration complets pour IBM MQ.

Outre les utilisateurs répertoriés dans le tableau suivant, il existe certains objets et autorisations pour lesquels une attention particulière doit être accordée lors de l'octroi de l'accès, afin de garantir l'intégrité et la sécurité du gestionnaire de files d'attente. Un examen supplémentaire doit être effectué lors de l'octroi de l'une des autorisations suivantes:

- Toute autorisation sur les objets SYSTEM
- Autorisations d'administration pour créer, modifier et supprimer des objets.

  z/OS Sous z/OS, cette autorisation correspond aux droits de sécurité de la commande et de la ressource de commande permettant d'exécuter les commandes DEFINE, ALTER et DELETE.

Multi Sur toutes les autres plateformes, ces autorisations sont des autorisations d'administration telles que `+crt`, `+chg` et `+dlt`.

- Autorisation d'administration pour effacer les files d'attente.

  z/OS Sous z/OS, cette autorisation correspond à la sécurité des commandes et aux droits de sécurité des ressources de commandes permettant d'émettre des commandes CLEAR.

  Multi Sur toutes les autres plateformes, cette autorisation est `+clr`.
- Autorisations d'administration permettant d'arrêter des canaux, d'annuler ou de valider des messages.

  z/OS Sous z/OS, cette autorisation est une autorisation de sécurité de commande et de sécurité de ressource de commande permettant d'émettre des commandes telles que RESET CHANNEL, START CHANNEL et STOP CHANNEL.

  Multi Sur toutes les autres plateformes, ces autorisations sont `+ctrl` et `+ctrlx`.
- Autorisation MQI de l'utilisateur de remplacement qui permet aux applications d'augmenter les privilèges pour les vérifications d'autorisation.

  z/OS Sous z/OS, cette autorisation correspond à tout droit accordé aux autres profils de sécurité utilisateur.

  Multi Sur toutes les autres plateformes, cette autorisation est `+altusr`.
- Autorisations de contexte qui permettent aux applications de modifier le contexte de sécurité des messages.

  z/OS Sous z/OS, cette autorisation correspond à toute autorisation accordée aux profils de sécurité de contexte.

  Multi Sur toutes les autres plateformes, ces autorisations sont `+setall` et `+setid`.

En tant que principe général, les applications de messagerie ne doivent recevoir que les autorisations MQI de base pour les files d'attente ou les rubriques nécessaires. Les canaux MCA qui s'exécutent sous un MCAUSER non privilégié et certains autres types d'applications spéciaux, tels que les gestionnaires de files d'attente de rebut, peuvent nécessiter des autorisations supplémentaires qui ne sont normalement pas accordées aux applications pour fonctionner correctement.

Tableau 67. Utilisateurs privilégiés par plateforme

Plateforme	**Utilisateurs privilégiés**
Systèmes Windows	• SYSTEME • Membres du groupe mqm • Membres du groupe Administrateurs
Systèmes AIX and Linux	• Membres du groupe mqm
Systèmes IBM i	• Les profils qmqm et qmqmadm • Tous les membres du groupe qmqmadm • Tout utilisateur défini avec le paramètre *ALLOBJ

Tableau 67. Utilisateurs privilégiés par plateforme (suite)	
Plateforme	**Utilisateurs privilégiés**
z/OS	ID utilisateur sous lequel s'exécutent les espaces adresse de l'initiateur de canal, du gestionnaire de files d'attente et de la sécurité avancée des messages. Ces ID utilisateur ne disposent pas automatiquement de droits d'administration complets pour IBM MQ, mais ils sont considérés comme privilégiés en raison du niveau d'accès généralement accordé à ces ID utilisateur.

Identification et authentification des utilisateurs à l'aide de la structure MQCSP

Vous pouvez spécifier la structure des paramètres de sécurité de connexion MQCSP sur un appel MQCONNX. La structure MQCSP est le principal moyen utilisé par les applications qui utilisent l'interface de file d'attente de messages (MQI) pour contrôler les données d'identification utilisées pour l'authentification.

La structure MQCSP contient les données d'identification que le service d'autorisation peut utiliser pour identifier et authentifier l'utilisateur.

La structure MQCSP peut être modifiée par des exits de sécurité côté client ou côté serveur, même si l'application ne fournit pas explicitement la structure MQCSP. Un exemple d'application qui ne fournit pas explicitement de structure MQCSP est une application qui utilise IBM MQ classes for JMS. Pour un exemple d'exit de sécurité côté client qui insère un ID utilisateur et un mot de passe dans la structure MQCSP, voir «Exit de sécurité côté client pour l'insertion d'un ID utilisateur et d'un mot de passe (mqccred)», à la page 85.

V 9.4.0 La structure MQCSP contient un ID utilisateur et un mot de passe ou un jeton d'authentification. Les restrictions suivantes s'appliquent aux données d'identification fournies dans la structure MQCSP:

- Une application ou un exit doit fournir un ID utilisateur et un mot de passe ou un jeton d'authentification, mais pas les deux.
- Seuls les jetons d'authentification qui répondent à des formats et des exigences spécifiques peuvent être utilisés pour accéder à IBM MQ. Pour plus d'informations sur les conditions requises pour les jetons d'authentification dans IBM MQ, voir «Conditions requises pour les jetons d'authentification», à la page 338.
- Si l'identité dans le jeton d'authentification doit être adoptée comme contexte de l'application, le jeton doit fournir une revendication utilisateur appropriée et la valeur de la revendication doit être un ID utilisateur IBM MQ valide. Par exemple, le nom d'utilisateur doit respecter les restrictions de longueur maximale et de caractères spéciaux. Pour plus d'informations sur l'adoption d'un ID utilisateur, voir «Relation entre les paramètres MQCSP et ADOPTCTX», à la page 331.

Pour plus d'informations sur la structure MQCSP, voir MQCSP-Paramètres de sécurité.

Avertissement : Les données d'identification d'une structure MQCSP pour une application client sont parfois envoyées sur le réseau en texte en clair. Pour vous assurer que les données d'identification de l'application client sont protégées, voir «Protection par mot de passe MQCSP», à la page 32.

Relation entre les paramètres MQCSP et ADOPTCTX

IBM MQ authentifie toujours les données d'identification qui sont transmises dans la structure MQCSP si la fonction d'authentification de connexion est activée. Une fois les données d'identification authentifiées, IBM MQ peut adopter l'ID utilisateur pour les vérifications d'autorisation ultérieures sur les opérations effectuées par l'application connectée. L'ID utilisateur dans les données d'identification MQCSP est

adopté si l'objet d'informations d'authentification (AUTHINFO) référencé par l'attribut **CONNAUTH** du gestionnaire de files d'attente est défini avec **ADOPTCTX(YES)**.

IBM MQ a une limite sur la longueur des ID utilisateur qu'il peut utiliser pour les vérifications d'autorisation. Pour plus d'informations sur ces limites, voir «ID utilisateur», à la page 94. Lorsqu'un ID utilisateur transmis dans la structure MQCSP est adopté, IBM MQ se comporte différemment, en fonction des autres options de configuration:

- Lors de l'utilisation de l'authentification de connexion LDAP, IBM MQ adopte l'ID utilisateur qui se trouve dans l'attribut de nom d'utilisateur abrégé de l'enregistrement LDAP de l'utilisateur. L'attribut de nom d'utilisateur abrégé est défini à l'aide de l'attribut **SHORTUSR** de l'objet AUTHINFO.

  Par exemple, si **SHORTUSR** est défini sur `'CN'` et que l'enregistrement LDAP répertorie l'utilisateur en tant que `'CN=Test,SN=MQ,O=IBM,C=UK'`, l'ID utilisateur `Test` est utilisé.
- Lors de l'utilisation de l'authentification de connexion au système d'exploitation ou de l'authentification PAM, si ADOPTCTX est défini sur YES, l'ID utilisateur transmis dans la structure MQCSP est tronqué afin de respecter la limite de 12 caractères de l'ID utilisateur IBM MQ lorsqu'il est adopté comme contexte de connexion.

  Si **ChlAuthEarlyAdopt** est activé, la troncature se produit une fois que les données d'identification de l'utilisateur ont été authentifiées.

  Si **ChlAuthEarlyAdopt** n'est pas activé, la troncature est effectuée avant l'adoption. Sous Windows, si l'utilisateur est indiqué au format `user@domain`, cela signifie que la troncature peut entraîner une spécification de domaine qui n'est pas valide lorsque l'utilisateur comporte moins de 12 caractères.

  Par exemple, si un utilisateur `` `ibmmq@windowsdomain` `` est fourni via MQCSP, il est tronqué à `` `ibmmq@window` `` dans ce scénario. Il en résulte l'erreur suivante:

  ```
  AMQ8074W: L'autorisation a échoué car le SID'SID'ne correspond pas à l'entité'ibmmq@window'
  ```

  Dans ce cas, si vous transmettez un ID utilisateur de plus de 12 caractères, tel qu'un ID utilisateur de domaine Windows au format `user@domain`, via le MQCSP, vous devez configurer **ChlAuthEarlyAdopt**=*Y* dans le fichier qm.ini pour éviter cette erreur.

  Vous pouvez également utiliser ADOPTCTX (NO) dans la configuration CONNAUTH AUTHINFO et utiliser une autre approche, telle qu'une règle CHLAUTH USERMAP, un exit de sécurité ou le paramètre MCAUSER de l'objet de canal pour définir l'ID utilisateur du canal.

Implémentation de l'identification et de l'authentification dans les exits de sécurité

Vous pouvez utiliser un exit de sécurité pour implémenter l'authentification unidirectionnelle ou mutuelle.

L'objectif principal d'un exit de sécurité est d'activer l'agent MCA à chaque extrémité d'un canal pour authentifier son partenaire. A chaque extrémité d'un canal de transmission de messages et à l'extrémité serveur d'un canal MQI, un agent MCA agit généralement pour le compte du gestionnaire de files d'attente auquel il est connecté. A l'extrémité client d'un canal MQI, un agent MCA agit généralement pour le compte de l'utilisateur de l'application IBM MQ MQI client . Dans cette situation, l'authentification mutuelle a lieu entre deux gestionnaires de files d'attente ou entre un gestionnaire de files d'attente et l'utilisateur d'une application IBM MQ MQI client .

L'exit de sécurité fourni (l'exit de canal SSPI) illustre comment l'authentification mutuelle peut être implémentée en échangeant des jetons d'authentification qui sont générés, puis vérifiés, par un serveur d'authentification sécurisé tel que Kerberos. Pour plus de détails, voir «Programme d'exit de canal SSPI sous Windows», à la page 164.

L'authentification mutuelle peut également être mise en oeuvre à l'aide de la technologie PKI (Public Key Infrastructure). Chaque exit de sécurité génère des données aléatoires, les signe à l'aide de la clé privée du gestionnaire de files d'attente ou de l'utilisateur qu'il représente et envoie les données signées à son partenaire dans un message de sécurité. L'exit de sécurité partenaire effectue l'authentification en vérifiant la signature numérique à l'aide de la clé publique du gestionnaire de files d'attente ou de l'utilisateur. Avant d'échanger des signatures numériques, les exits de sécurité peuvent avoir besoin de

convenir de l'algorithme de génération d'un résumé de message, si plusieurs algorithmes sont disponibles pour être utilisés.

Lorsqu'un exit de sécurité envoie les données signées à son partenaire, il doit également envoyer un moyen d'identifier le gestionnaire de files d'attente ou l'utilisateur qu'il représente. Il peut s'agir d'un nom distinctif ou même d'un certificat numérique. Si un certificat numérique est envoyé, l'exit de sécurité partenaire peut le valider en utilisant la chaîne de certificats pour le certificat de l'autorité de certification racine. Cela garantit la propriété de la clé publique utilisée pour vérifier la signature numérique.

L'exit de sécurité partenaire ne peut valider un certificat numérique que s'il a accès à un référentiel de clés qui contient les certificats restants dans la chaîne de certificats. Si un certificat numérique pour le gestionnaire de files d'attente ou l'utilisateur n'est pas envoyé, il doit être disponible dans le référentiel de clés auquel l'exit de sécurité partenaire a accès. L'exit de sécurité partenaire ne peut pas vérifier la signature numérique sauf s'il peut trouver la clé publique du signataire.

Le protocole TLS (Transport Layer Security) utilise des techniques PKI telles que celles qui viennent d'être décrites. Pour plus d'informations sur la manière dont la couche Secure Sockets Layer effectue l'authentification, voir «Concepts TLS (Transport Layer Security)», à la page 19.

Si un serveur d'authentification sécurisé ou une prise en charge de l'infrastructure PKI n'est pas disponible, d'autres techniques peuvent être utilisées. Une technique commune, qui peut être implémentée dans les exits de sécurité, utilise un algorithme de clé symétrique.

L'un des exits de sécurité, l'exit A, génère un nombre aléatoire et l'envoie dans un message de sécurité à son exit de sécurité partenaire, l'exit B. L'exit B chiffre le nombre à l'aide de sa copie d'une clé connue uniquement des deux exits de sécurité. L'exit B envoie le numéro chiffré à l'exit A dans un message de sécurité avec un deuxième nombre aléatoire que l'exit B a généré. L'exit A vérifie que le premier nombre aléatoire a été chiffré correctement, chiffre le deuxième nombre aléatoire à l'aide de sa copie de la clé et envoie le nombre chiffré à l'exit B dans un message de sécurité. La sortie B vérifie alors que le deuxième nombre aléatoire a été chiffré correctement. Lors de cet échange, si l'un des exits de sécurité n'est pas satisfait de l'authenticité d'un autre, il peut demander à l'agent MCA de fermer le canal.

Un avantage de cette technique est qu'aucune clé ou mot de passe n'est envoyé sur la connexion de communication lors de l'échange. Un inconvénient est qu'il ne permet pas de résoudre le problème de la répartition sécurisée de la clé partagée. Une solution à ce problème est décrite dans «Implémentation de la confidentialité dans les programmes d'exit utilisateur», à la page 482. Une technique similaire est utilisée dans SNA pour l'authentification mutuelle de deux unités logiques lorsqu'elles se lient pour former une session. Cette technique est décrite dans «Authentification au niveau de la session», à la page 129.

Toutes les techniques d'authentification mutuelle précédentes peuvent être adaptées pour fournir une authentification unidirectionnelle.

Mappage d'identité dans les exits de message

Vous pouvez utiliser des exits de message pour traiter des informations afin d'authentifier un ID utilisateur, mais il peut être préférable d'implémenter l'authentification au niveau de l'application.

Lorsqu'une application place un message dans une file d'attente, la zone *UserIdentifier* du descripteur de message contient un ID utilisateur associé à l'application. Toutefois, il n'existe aucune donnée pouvant être utilisée pour authentifier l'ID utilisateur. Ces données peuvent être ajoutées par un exit de message à l'extrémité émettrice d'un canal et vérifiées par un exit de message à l'extrémité réceptrice du canal. Les données d'authentification peuvent être par example un mot de passe chiffré ou une signature numérique.

Ce service peut être plus efficace s'il est implémenté au niveau de l'application. La condition de base est que l'utilisateur de l'application qui reçoit le message puisse identifier et authentifier l'utilisateur de l'application qui a envoyé le message. Il est donc naturel d'envisager la mise en oeuvre de ce service au niveau de l'application. Pour plus d'informations, voir «Mappage d'identité dans l'exit d'API et l'exit de croisement d'API», à la page 334.

Mappage d'identité dans l'exit d'API et l'exit de croisement d'API

Une application qui reçoit un message doit être en mesure d'identifier et d'authentifier l'utilisateur de l'application qui a envoyé le message. Ce service est généralement mieux implémenté au niveau de l'application. Les exits API peuvent implémenter le service de différentes manières.

Au niveau d'un message individuel, l'identification et l'authentification sont un service qui implique deux utilisateurs, l'expéditeur et le destinataire du message. La condition de base est que l'utilisateur de l'application qui reçoit le message puisse identifier et authentifier l'utilisateur de l'application qui a envoyé le message. Notez que l'exigence concerne l'authentification unidirectionnelle et non bidirectionnelle.

Selon la façon dont il est implémenté, les utilisateurs et leurs applications peuvent avoir besoin d'interfacer, voire d'interagir, avec le service. En outre, le moment et la manière dont le service est utilisé peuvent dépendre de l'emplacement des utilisateurs et de leurs applications, ainsi que de la nature des applications elles-mêmes. Il est donc naturel d'envisager d'implémenter le service au niveau de l'application plutôt qu'au niveau de la liaison.

Si vous envisagez d'implémenter ce service au niveau de la liaison, vous devrez peut-être résoudre les problèmes suivants:

- Sur un canal de transmission de messages, comment appliquer le service uniquement aux messages qui en ont besoin?
- Comment autorisez-vous les utilisateurs et leurs applications à interagir avec le service, si c'est une exigence?
- Dans une situation à plusieurs tronçons, où un message est envoyé via plusieurs canaux de transmission de messages sur le chemin de sa destination, où appelez-vous les composants du service?

Voici quelques exemples de la façon dont le service d'identification et d'authentification peut être implémenté au niveau de l'application. Le terme *exit d'API* signifie un exit d'API ou un exit de croisement d'API.

- Lorsqu'une application place un message dans une file d'attente, un exit API peut acquérir un jeton d'authentification à partir d'un serveur d'authentification sécurisé tel que Kerberos. L'exit API peut ajouter ce jeton aux données d'application dans le message. Lorsque le message est extrait par l'application de réception, un deuxième exit API peut demander au serveur d'authentification d'authentifier l'expéditeur en vérifiant le jeton.
- Lorsqu'une application place un message dans une file d'attente, un exit API peut ajouter les éléments suivants aux données d'application du message:
  - Certificat numérique de l'expéditeur
  - Signature numérique de l'expéditeur

  Si des algorithmes différents sont disponibles pour la génération d'un résumé de message, l'exit d'API peut inclure le nom de l'algorithme qu'il a utilisé.

  Lorsque le message est extrait par l'application de réception, un deuxième exit d'API peut effectuer les vérifications suivantes:

  - L'exit API peut valider le certificat numérique en utilisant la chaîne de certificats pour le certificat de l'autorité de certification racine. Pour ce faire, l'exit API doit avoir accès à un référentiel de clés qui contient les certificats restants dans la chaîne de certificats. Cette vérification garantit que l'expéditeur, identifié par le nom distinctif, est le véritable propriétaire de la clé publique contenue dans le certificat.
  - L'exit API peut vérifier la signature numérique à l'aide de la clé publique contenue dans le certificat. Cette vérification authentifie l'expéditeur.

  Le nom distinctif de l'expéditeur peut être envoyé à la place du certificat numérique complet. Dans ce cas, le référentiel de clés doit contenir le certificat de l'expéditeur pour que le deuxième exit d'API puisse trouver la clé publique de l'expéditeur. Une autre possibilité consiste à envoyer tous les certificats de la chaîne de certificats.

- Lorsqu'une application place un message dans une file d'attente, la zone *UserIdentifier* du descripteur de message contient un ID utilisateur associé à l'application. L'ID utilisateur peut être utilisé pour identifier l'expéditeur. Pour activer l'authentification, un exit d'API peut ajouter des données, telles qu'un mot de passe chiffré, aux données d'application du message. Lorsque le message est extrait par l'application de réception, un deuxième exit API peut authentifier l'ID utilisateur à l'aide des données qui ont été transmises avec le message.

  Cette technique peut être considérée comme suffisante pour les messages provenant d'un environnement contrôlé et sécurisé, et dans les cas où un serveur d'authentification sécurisé ou une prise en charge de l'infrastructure PKI n'est pas disponible.

V 9.4.0 Linux AIX Utilisation des jetons d'authentification

Depuis IBM MQ 9.4.0, les applications client peuvent fournir des jetons pour l'authentification auprès d'un gestionnaire de files d'attente s'exécutant sous AIX ou Linux. L'ID utilisateur dans le jeton peut également être utilisé pour l'autorisation d'accès aux ressources IBM MQ .

Les jetons JWT (JSON Web Tokens) adoptent un modèle d'identité basé sur les revendications. L'identité et le contrôle d'accès sont résumés dans des idées de revendications et d'émetteurs de jetons.

- Une réclamation est une paire nom-valeur qui contient des informations sur un utilisateur et qui détermine qui est l'utilisateur, et non ce qu'il peut faire.
- L'émetteur de jeton est un tiers de confiance ou un serveur qui émet un jeton pour un utilisateur en fonction uniquement de l'identité de l'utilisateur. L'émetteur du jeton ne se préoccupe pas de ce que l'utilisateur peut faire.

Un jeton est une structure simple qui contient des réclamations et qui peut facilement être transférée entre les parties sur Internet. L'utilisation de jetons pour l'authentification offre l'avantage d'une gestion centralisée des identités. Vous pouvez utiliser un émetteur de jeton sécurisé pour que vos applications puissent s'authentifier auprès de nombreux services sans s'enregistrer séparément auprès de chaque service. Les jetons offrent une sécurité accrue car les données d'identification ne sont pas envoyées à chaque service, uniquement à l'émetteur de confiance.

Un jeton JWT est défini via la norme Internet proposée RFC7519.

Fonctionnement des jetons avec IBM MQ

Les jetons utilisés avec IBM MQ doivent être des jetons JWT valides qui ont été signés avec un algorithme pris en charge par IBM MQ . Le jeton JWT doit être signé conformément à la norme JWS (JSON Web Signature). Les jetons qui utilisent les technologies JWE (JSON Web Encryption) et JWK (JSON Web Key) JOSE ne peuvent pas être utilisés avec IBM MQ. Pour plus d'informations, voir «Conditions requises pour les jetons d'authentification», à la page 338.

L'application qui fournit le jeton d'authentification peut s'exécuter sur n'importe quelle plateforme prenant en charge IBM MQ clients. L'application doit être écrite en C ou en Java, et se connecter au gestionnaire de files d'attente à l'aide de liaisons client. Toutefois, le gestionnaire de files d'attente doit s'exécuter sous AIX ou Linux.

Le gestionnaire de files d'attente valide la signature du jeton par rapport à la clé publique de l'émetteur digne de confiance ou à la clé symétrique dans le référentiel de clés. Pour configurer le gestionnaire de files d'attente, suivez les étapes de la rubrique «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un noeud final JWKS», à la page 341 ou Configuration d'un gestionnaire de files d'attente pour qu'il accepte des jetons d'authentification à l'aide d'un magasin de clés local.

L'émetteur du jeton est la partie accréditée qui dispose de l'accès de sécurité délégué, ce qui signifie qu'elle vérifie l'identité de l'utilisateur de l'application. Le gestionnaire de files d'attente vérifie qu'un jeton d'authentification est valide et que l'utilisateur authentifié est autorisé à accéder aux objets IBM MQ . Le gestionnaire de files d'attente peut, mais n'a pas besoin de connaître les utilisateurs avant de se connecter avec un jeton. L'administrateur IBM MQ doit configurer l'authentification et l'autorisation pour

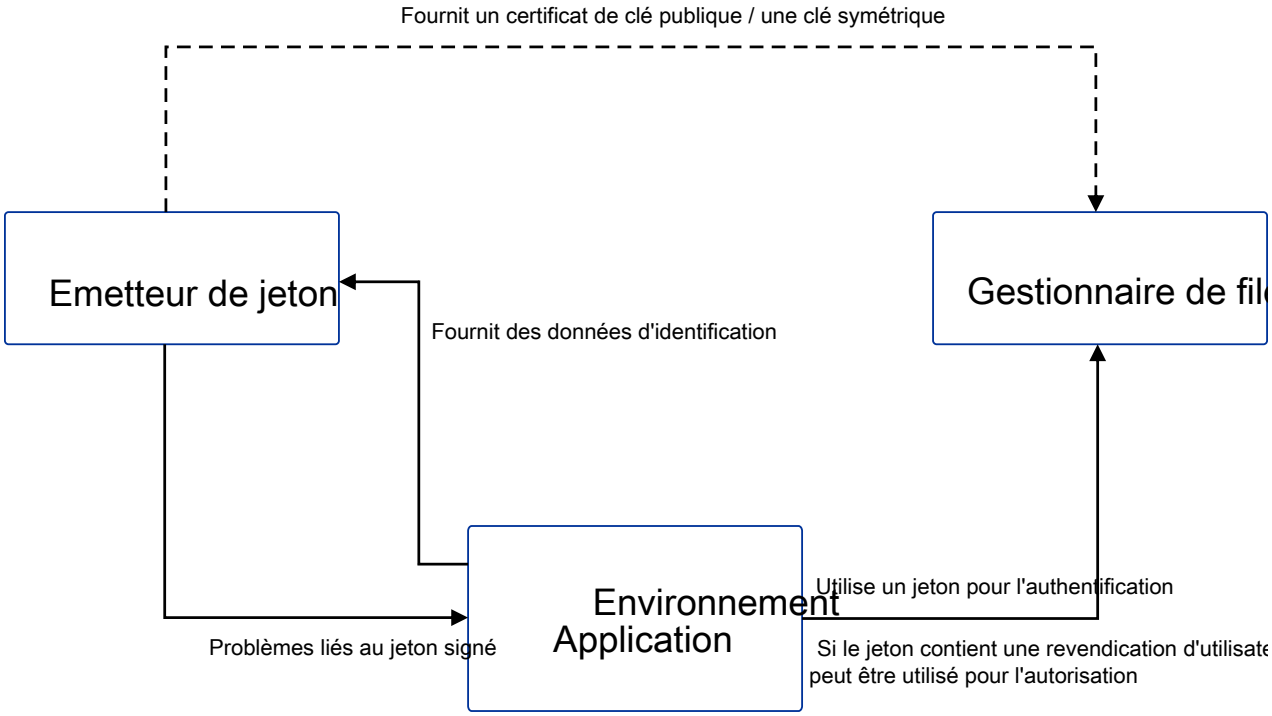
les applications qui se connectent au gestionnaire de files d'attente et définir les exigences relatives à ce que doivent contenir les jetons.

L'application client peut demander dynamiquement un jeton à l'émetteur qu'elle utilise pour l'authentification lorsqu'elle se connecte à IBM MQ. L'application utilise ensuite la structure MQCSP, ou l'équivalent dans l'API choisie, pour transmettre le jeton au gestionnaire de files d'attente lorsqu'elle se connecte.

Si l'application ne peut pas être modifiée pour demander un jeton d'authentification et présenter le jeton au gestionnaire de files d'attente lorsqu'elle se connecte, un exit de sécurité peut également être utilisé pour fournir un jeton dans la structure MQCSP.

Si le jeton remplit les conditions requises pour les jetons d'authentification et que la signature du jeton est valide, la connexion est établie. Le gestionnaire de files d'attente peut également utiliser l'ID utilisateur contenu dans le jeton pour les vérifications d'autorisation afin d'accéder aux ressources IBM MQ si la revendication utilisateur facultative est contenue dans le jeton. La revendication utilisateur est la revendication dans le jeton qui contient l'ID utilisateur que le gestionnaire de files d'attente adopte pour les vérifications d'autorisation. Ce nom de la réclamation utilisateur est spécifié avec l'attribut **UserClaim** dans la section **AuthToken** du fichier `qm.ini` .

Pour plus d'informations, voir «Utilisation de jetons d'authentification dans une application», à la page 346 et MQCSP-Paramètres de sécurité.



Le diagramme présente un exemple de base du flux attendu pour l'utilisation de jetons avec IBM MQ. Le cycle de vie attendu est le suivant:

- Le jeton est émis vers une application par l'émetteur digne de confiance. Pour plus d'informations, voir Conditions requises pour les jetons d'authentification.
- L'application transmet le jeton au gestionnaire de files d'attente lors de la connexion. Pour plus d'informations, voir Utilisation de jetons d'authentification dans une application.
- Le gestionnaire de files d'attente valide la signature du jeton par rapport à la clé publique de l'émetteur digne de confiance ou à la clé symétrique dans le référentiel de clés. Pour configurer le gestionnaire de files d'attente, suivez les étapes de la rubrique «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un noeud final JWKS», à la page 341.
- Si le jeton d'authentification contient une revendication utilisateur valide, l'utilisateur dans le jeton peut être adopté pour les vérifications d'autorisation permettant d'accéder aux ressources IBM MQ . Pour plus d'informations, voir Adoption d'utilisateurs pour l'autorisation.
- L'administrateur IBM MQ gère les certificats d'émetteur de jeton de confiance. Lorsque le certificat arrive à expiration, un nouveau certificat doit être obtenu auprès de l'émetteur du jeton et ajouté au référentiel de clés.
- Si vous avez configuré votre gestionnaire de files d'attente et que l'application se connecte mais rencontre des problèmes avec le jeton, voir Traitement des incidents liés au jeton d'authentification et Codes d'erreur d'authentification de jeton.

IBM MQ fonctionne avec tout émetteur de jeton qui fournit des jetons conformes aux normes JWT et JWS.

Si vous n'utilisez pas déjà de jetons mais que vous souhaitez comprendre ce qui est impliqué dans la mise en place d'un serveur de jetons, voir le guide d'initiation pour le projet gratuit et open source Keycloak.

Référence associée

Section AuthToken du fichier `qm.ini`

V 9.4.0 Linux AIX Conditions requises pour les jetons d'authentification

Exigences de validation, structure et algorithmes pour les jetons d'authentification utilisés avec IBM MQ.

Exigences

Les jetons d'authentification utilisés avec IBM MQ doivent répondre aux exigences suivantes.

- La longueur du jeton ne doit pas dépasser la longueur maximale de 8192 caractères. Pour plus d'informations, voir TokenLength (MQLONG) for MQCSP.
- La structure et le codage du jeton sont valides, comme défini par la spécification JWT (JSON Web Token) dans RFC7519, et la spécification JWS (JSON Web Signature) dans RFC7515.
- Les paramètres d'en-tête de jeton requis qui sont spécifiés dans Tableau 68, à la page 339 sont présents et les valeurs des paramètres sont valides.
- Les réclamations de contenu requises spécifiées dans Tableau 69, à la page 340 sont présentes et les valeurs des réclamations sont valides.
- Le jeton est signé avec un algorithme dans Tableau 70, à la page 340 pris en charge par IBM MQ .
- La valeur de la réclamation d'expiration (**exp**) est postérieure à l'heure en cours.
- Si la réclamation Non antérieur (**nbf**) est présente, la valeur est antérieure à l'heure en cours.
- Si une réclamation utilisateur est présente, la valeur doit répondre aux exigences de «ID utilisateur dans les jetons d'authentification», à la page 341.

Structure de jeton

IBM MQ accepte les jetons JWT conformes à la norme RFC7519 . Le jeton JWT doit être signé et codé conformément à la norme JWS définie dans RFC7515.

IBM MQ s'attend à ce que le jeton sécurisé JWS contienne les trois composants suivants:

En-tête JOSE

Objet JSON contenant des paramètres qui décrivent le type de jeton et les algorithmes de cryptographie utilisés pour sécuriser son contenu.

L'exemple d'en-tête suivant déclare que l'objet codé est un jeton JWT et que l'en-tête et le contenu sont sécurisés à l'aide de l'algorithme HMAC SHA-256 .

```
{
 "typ":"JWT",
 "alg":"HS256"
}
```

Contenu JWS

Objet JSON qui contient des réclamations telles que spécifiées dans la norme JWT. Chaque membre de l'objet JSON est une réclamation. Les revendications peuvent affirmer l'identité de l'émetteur du jeton ou l'ID utilisateur du porteur.

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

Signature JWS

Permet de vérifier que le jeton est émis par un émetteur digne de confiance.

Ces composants sont représentés dans le jeton sécurisé JWS sous forme de chaînes base64url-encoded séparées par un point ('.').

Un jeton d'authentification conforme à la norme JWS est signé pour permettre la validation de l'authenticité du jeton, mais il n'est pas chiffré. Par conséquent, il peut être lu, et éventuellement réutilisé, par toute personne ayant accès au jeton. Configurez la connexion au gestionnaire de files d'attente pour vous assurer que l'authentification est protégée à l'aide du chiffrement lorsqu'elle est envoyée sur le réseau, par exemple à l'aide de TLS. Pour plus d'informations sur les options de protection des données d'identification fournies par une application, voir Protection par mot de passe MQCSP.

IBM MQ prend en charge les paramètres et les revendications suivants dans l'en-tête et le contenu des jetons d'authentification. Tous les paramètres ou revendications supplémentaires d'un jeton sont ignorés. Si un jeton contient plusieurs paramètres ou revendications portant le même nom, le dernier paramètre ou la revendication portant le même nom est utilisé.

Tableau 68. Description des paramètres d'en-tête de jeton

Partie de jeton	Nom du paramètre :	Type de données	Obligatoire	Description
En-tête	**typ**	String	Oui	Type de jeton. La valeur de ce paramètre doit être "JWT".
	alg	String	Oui	Algorithme utilisé pour sécuriser l'en-tête et le contenu. La valeur de ce paramètre doit être l'un des algorithmes de Tableau 70, à la page 340.

Tableau 69. Descriptions des réclamations de contenu de jeton				
Partie de jeton	**Nom du paramètre :**	**Type de données**	**Obligatoire**	**Description**
Contenu utile	**exp**	Entier	Oui	Heure d'expiration du jeton, exprimée en nombre de secondes écoulées depuis le 1er janvier 1979, à 00:00 UTC. Le jeton n'est plus accepté après ce délai.
	nbf	Entier	Non	Heure, exprimée en nombre de secondes écoulées depuis le 1er janvier 1979, à 00:00 UTC avant laquelle le jeton n'est pas accepté.
	Le nom de réclamation de l'utilisateur indiqué dans la zone **UserClaim** de la section **AuthToken** du fichier `qm.ini` .	String	Obligatoire uniquement si la revendication de l'utilisateur dans le jeton est utilisée pour l'autorisation.	Nom de la réclamation contenant l'ID utilisateur adopté pour les vérifications d'autorisation. Par exemple, si votre jeton comporte la revendication utilisateur `"AppUser": "MyUserName"`, vous devez spécifier **UserClaim**=`AppUser` dans la section **AuthToken** du fichier `qm.ini` .

Pour un bon exemple de jeton codé et décodé, voir la page debugger sur le site Web `jwt.io` .

Algorithmes

IBM MQ prend en charge un sous-ensemble d'algorithmes inclus dans la spécification JWA (JSON Web Algorithms) pour les jetons sécurisés JWS .

Tableau 70. Algorithmes Web JSON (JWA) pris en charge par IBM MQ pour les jetons sécurisés JWS	
alg valeur du paramètre	**Signature numérique ou algorithme MAC**
HS256	HMAC utilisant SHA-256
HS384	HMAC utilisant SHA-384
HS512	HMAC utilisant SHA-512
RS256	RSASSA-PKCS1-v1_5 avec SHA-256
RS384	RSASSA-PKCS1-v1_5 avec SHA-384
RS512	RSASSA-PKCS1-v1_5 avec SHA-512

Exigences relatives au certificat de clé asymétrique

Si un jeton est signé avec une clé asymétrique, le certificat de clé publique de l'émetteur de jeton doit se trouver dans le référentiel de clés utilisé par le gestionnaire de files d'attente pour l'authentification par jeton. Lorsque le jeton d'authentification est reçu, le certificat doit être dans sa période de validité. Aucune vérification n'est effectuée pour s'assurer que le certificat de l'émetteur du jeton n'a pas été révoqué.

ID utilisateur dans les jetons d'authentification

Si le gestionnaire de files d'attente est configuré pour adopter l'ID utilisateur contenu dans la revendication utilisateur d'un jeton d'authentification comme contexte de l'application, l'ID utilisateur adopté doit répondre aux exigences suivantes:

- Il peut contenir jusqu'à 12 caractères.
- Il doit commencer par l'un des caractères suivants:

  A-Z a-z
- Il peut contenir l'un des caractères suivants:

  0-9 A-Z a-z +,-. : = _
- Il ne doit pas s'agir de l'un des ID utilisateur réservés UNKNOWN et NOBODY.

Tâches associées

Configuration d'un gestionnaire de files d'attente pour qu'il accepte **AuthTokens**

Référence associée

Section AuthToken du fichier `qm.ini`

V 9.4.0 Linux AIX Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un noeud final JWKS

Configurez votre gestionnaire de files d'attente IBM MQ s'exécutant sous AIX ou Linux pour authentifier les utilisateurs et les applications avec des jetons d'authentification à l'aide d'un noeud final JWKS.

Avant de commencer

Pour plus d'informations sur le fonctionnement des jetons avec IBM MQ, voir Utilisation des jetons d'authentification.

Avant de configurer votre gestionnaire de files d'attente, vérifiez que l'objet AUTHINFO référencé dans l'attribut **CONNAUTH** du gestionnaire de files d'attente est de type `IDPWOS`. L'authentification par jeton est disponible uniquement lorsque le gestionnaire de files d'attente est configuré pour la vérification de l'ID utilisateur et du mot de passe du système d'exploitation.

Vérifiez que l'attribut **SecurityPolicy** de la strophe Service n'est pas défini sur `Group`. L'authentification par jeton n'est pas disponible si **SecurityPolicy** est explicitement défini sur `Groupe`. Si**SecurityPolicy** est réglé sur `Groupe` , retirer le**SecurityPolicy** de la strophe Service, puis redémarrez le gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Les applications peuvent s'authentifier auprès du gestionnaire de files d'attente à l'aide de jetons. IBM MQ accepte les jetons Web JSON (*JWT*) des émetteurs dignes de confiance qui suivent la norme Internet proposée RFC7519. Vous pouvez utiliser des jetons pour authentifier une identité, qui peut ensuite être adoptée pour des vérifications d'autorisation ultérieures.

Le moyen le plus simple de configurer votre gestionnaire de files d'attente pour qu'il accepte des jetons consiste à pointer vers un noeud final JWKS, comme décrit ci-dessous. Si votre service d'authentification ne fournit pas ce type de noeud final ou si JWKS ne convient pas pour d'autres raisons, voir «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un magasin de clés local», à la page 343.

Procédure

1. Demandez à votre administrateur de serveur d'authentification les détails suivants:
   - Noeud final JWKS correct (URL).

- Quel certificat ce serveur utilise pour chiffrer le trafic HTTP et / ou quelle autorité signe ce certificat.

  Important : Vous devez toujours fournir des informations JWKS via TLS/HTTPS et vous en avez besoin pour vous assurer que le gestionnaire de files d'attente peut faire confiance à la connexion.

2. Configurez le gestionnaire de files d'attente pour créer des connexions HTTPS sortantes en fournissant un **HTTPSKeyStore** dans le fichier qm.ini .

   Pour plus d'informations, voir

   - L'explication HTTPSKeyStore dans le fichier `qm.ini` .
   - «Création d'un référentiel de clés à utiliser comme magasin de clés de confiance TLS», à la page 349.

   Si le serveur d'authentification utilise un certificat / une autorité de certification sur mesure, vous devez vous assurer qu'il est correctement présent dans ce fichier HTTPSKeyStore.
3. Configurez le noeud final JWKS en définissant une strophe JWKS dans le fichier de configuration qm.ini .

   La strophe supplémentaire fournit les éléments suivants:

   - **`issuername`**: Cette valeur doit correspondre à la revendication'`iss`'présente dans les jetons signés par cette autorité et est souvent basée sur l'URL du service d'authentification.
   - **`endpoint`**: Il s'agit de l'adresse à partir de laquelle le gestionnaire de files d'attente interroge les clés publiques utilisées pour valider les signatures de jeton.
   - **`userclaim`**:Cette option est facultative pour l'identification d'une zone personnalisée dans les jetons qui doit être utilisée pour les vérifications des droits d'accès IBM MQ une fois qu'un jeton a été validé.

     Avertissement : Elle doit être présente si vous prévoyez d'utiliser **ADOPTCTX**(YES) pour ces connexions.
4. Une fois les modifications apportées au fichier `.ini` terminées, exécutez la commande REFRESH SECURITY TYPE(AUTHINFO) ou redémarrez le gestionnaire de files d'attente.

   Si la configuration aboutit, les applications peuvent se connecter immédiatement à l'aide de jetons signés.

   S'il y a des problèmes, par exemple, l'impossibilité de contacter le service d'authentification pour extraire des clés publiques, les problèmes sont signalés dans le fichier journal AMQERR01 du gestionnaire de files d'attente.

Résultats

Vous avez configuré un gestionnaire de files d'attente pour qu'il accepte les jetons d'authentification à l'aide d'un noeud final JWKS.

Remarque : Les clés sont régulièrement actualisées à partir du serveur d'authentification (toutes les 15 minutes), et plus fréquemment si un ID de clé inconnu est présenté par une application de connexion. En règle générale, cela signifie qu'aucune autre action de configuration IBM MQ n'est requise pour mettre à jour les certificats à mesure qu'ils expirent et qu'ils sont remplacés côté serveur. Pour forcer une actualisation immédiate, exécutez la commande REFRESH SECURITY TYPE(AUTHINFO) à tout moment.

Concepts associés

Traitement des incidents liés aux jetons d'authentification

Tâches associées

Utilisation de jetons d'authentification dans une application

Référence associée

Section AuthToken du fichier `qm.ini`

V 9.4.0 Linux AIX Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un magasin de clés local

Configurez votre gestionnaire de files d'attente IBM MQ pour authentifier les utilisateurs et les applications avec des jetons d'authentification.

Avant de commencer

Dans la mesure du possible, envisagez d'utiliser un noeud final JWKS (voir «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un noeud final JWKS», à la page 341) plutôt que de configurer manuellement vos certificats de validation de jeton. L'utilisation de JWKS simplifie généralement la configuration initiale et la maintenance continue.

Pour plus d'informations sur le fonctionnement des jetons avec IBM MQ , voir Utilisation des jetons d'authentification.

Avant de configurer votre gestionnaire de files d'attente, vérifiez que l'objet AUTHINFO référencé dans l'attribut **CONNAUTH** du gestionnaire de files d'attente est de type `IDPWOS`. L'authentification par jeton est disponible uniquement lorsque le gestionnaire de files d'attente est configuré pour la vérification de l'ID utilisateur et du mot de passe du système d'exploitation.

Vérifiez que l'attribut **SecurityPolicy** de la strophe Service n'est pas défini sur `Group`. L'authentification par jeton n'est pas disponible si **SecurityPolicy** est explicitement défini sur `Groupe`. Si **SecurityPolicy** est défini sur `Groupe`, supprimez l'attribut **SecurityPolicy** de la strophe Service, puis redémarrez le gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Depuis IBM MQ 9.3.4 , les applications peuvent s'authentifier auprès du gestionnaire de files d'attente à l'aide de jetons. IBM MQ accepte les jetons Web JSON (*JWT*) des émetteurs dignes de confiance qui suivent la norme Internet proposée RFC7519. Vous pouvez utiliser des jetons pour authentifier une identité, qui peut ensuite être adoptée pour des vérifications d'autorisation ultérieures.

Configurez votre gestionnaire de files d'attente pour qu'il accepte les jetons en sauvegardant le certificat de clé publique de l'émetteur de confiance ou la clé symétrique dans le référentiel de clés du gestionnaire de files d'attente. Ajoutez la section AuthToken au fichier `qm.ini` et actualisez la configuration de la sécurité afin que le gestionnaire de files d'attente récupère la nouvelle configuration.

Vous pouvez configurer un magasin de clés local plutôt que d'utiliser JWKS dans un environnement de test ou lorsque la connectivité directe à votre serveur d'authentification à partir de votre gestionnaire de files d'attente n'est pas possible. Vous pouvez également définir un magasin de clés local en plus des noeuds finaux JWKS.

Remarque : Lorsqu'un noeud final JWKS et un magasin de clés local fournissent un émetteur et un KID correspondants pour un jeton présenté, la clé fournie par le noeud final JWKS est utilisée de préférence.

Dans ces situations, configurez le magasin de clés local comme suit:

Procédure

1. Créez le référentiel de clés.
   a) Créez un référentiel de clés pour le certificat de clé publique ou la clé symétrique reçue de l'émetteur digne de confiance. Vous pouvez utiliser un référentiel de clés CMS avec l'extension de fichier `.kdb` ou un référentiel de clés PKCS#12 avec l'extension de fichier `.p12`.

      Exécutez la commande suivante pour créer un référentiel de clés CMS :

      ```
      runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
      -type cms
      ```

Si la**runmqakm** la commande renvoie une erreur, voir runmqakm -keydb . Si la commande aboutit, utilisez la commande `ls` pour répertorier le contenu du répertoire:

```
ls -l /var/mqm/qmgrs/qm1/tokenissuer
```

Les fichiers suivants sont affichés:

```
-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) Si nécessaire, modifiez la propriété du groupe pour les fichiers de référentiel de clés que vous avez créés afin que le groupe mqm puisse disposer d'un accès en lecture. Initialement, seul l'administrateur qui a exécuté la commande a accès aux fichiers créés.

```
chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

c) Changez le mode des fichiers de référentiel de clés pour ajouter des droits de lecture pour le groupe mqm. Par exemple, la commande suivante ajoute des droits d'accès en lecture / écriture pour le propriétaire du fichier et des droits d'accès en lecture seule pour le groupe.

```
chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

2. Chiffrez le mot de passe du référentiel de clés à l'aide de la commande **runqmcred** et sauvegardez la chaîne chiffrée dans un fichier.

a) Créez un fichier contenant la clé initiale utilisée pour chiffrer le mot de passe du référentiel de clés.

Le fichier doit contenir la clé initiale sous la forme d'une seule ligne de texte. La longueur maximale de la clé initiale est de 256 octets. Si vous avez déjà défini une clé initiale pour le gestionnaire de files d'attente à l'aide de l'attribut de gestionnaire de files d'attente **INITKEY** , copiez la valeur de l'attribut **INITKEY** dans le nouveau fichier. Si vous n'avez pas encore défini de clé initiale pour le gestionnaire de files d'attente, créez une nouvelle clé de chiffrement unique et ajoutez-la au fichier de clés initial.

Remarque : Pour plus d'informations, voir INITKEY. Si vous ne spécifiez pas la clé initiale, une clé par défaut est utilisée. Il est plus sûr d'utiliser votre propre clé initiale.

Remarque : Accordez les droits d'accès minimaux nécessaires sur le fichier de clés initial pour que le contenu du fichier reste sécurisé. Le fichier de clés initial est utilisé uniquement pour chiffrer le mot de passe du référentiel de clés. Par conséquent, seuls les administrateurs qui utilisent la clé initiale pour chiffrer les mots de passe doivent accéder au fichier de clés initial en lecture.

b) Si la clé initiale du gestionnaire de files d'attente n'est pas déjà définie, définissez la valeur de l'attribut **INITKEY** du gestionnaire de files d'attente sur la clé initiale que vous avez créée à l'étape «2.a», à la page 344. Utilisez la commande **ALTER QMGR** pour définir la clé initiale du gestionnaire de files d'attente. Exemple :

```
ALTER QMGR INITKEY('myEncrypt10nK3y')
```

c) Exécutez la commande **runqmcred** pour chiffrer le mot de passe du référentiel de clés. Utilisez le paramètre **-sf** pour spécifier le chemin d'accès au fichier qui contient la clé initiale.

```
runqmcred -sf initial.key
```

Lorsque vous y êtes invité, entrez le mot de passe du référentiel de clés. Le mot de passe chiffré est généré par la commande.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
*************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

Copiez la chaîne sur la dernière ligne et sauvegardez-la dans un fichier.

3. Utilisez l'une des méthodes suivantes pour ajouter le certificat de clé publique ou la clé symétrique de l'émetteur de jeton au référentiel de clés.
   - Pour ajouter le certificat de clé publique RSA au référentiel de clés, exécutez la commande suivante:

     ```
     runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
     -label keylabel
              -file keyfile
     ```

   - Pour ajouter une clé symétrique codée base64 au référentiel de clés, exécutez la commande suivante:

     ```
     runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
     -label keylabel
              -file keyfile -format ascii
     ```

   Où *keylabel* est le libellé à associer au certificat ou à la clé secrète, et *keyfile* est le nom du fichier qui contient le certificat ou la clé secrète codée base64 .
4. Ajoutez la section **AuthToken** et les attributs suivants au fichier qm.ini :
   - Chemin d'accès au référentiel de clés, spécifié à l'aide de l'attribut **KeyStore** .
   - Fichier contenant le mot de passe du référentiel de clés, spécifié à l'aide de l'attribut **KeyStorePwdFile** .
   - Libellé du certificat ou de la clé symétrique que vous avez ajouté à l'étape «3», à la page 345, spécifié à l'aide de l'attribut **CertLabel** .

   Exemple :

   ```
   AuthToken:
      KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
      KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
      CertLabel=rsakey
   ```

   Où key.kdb est le nom du référentiel de clés que vous avez créé à l'étape «1.a», à la page 343et key.pw est le fichier qui contient le mot de passe chiffré pour le référentiel de clés que vous avez créé à l'étape «2.c», à la page 344.

   Pour plus d'informations sur la section **AuthToken** , voir la section AuthToken du fichier qm.ini.
5. Si le gestionnaire de files d'attente est configuré pour adopter l'ID utilisateur contenu dans la demande d'utilisateur de jeton à utiliser dans les vérifications d'autorisation ultérieures, ajoutez l'attribut **UserClaim** à la section **AuthToken** .

   Pour déterminer si le gestionnaire de files d'attente est configuré pour adopter l'ID utilisateur dans le jeton, émettez la commande MQSC suivante:

   ```
   DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
   ```

   Où *authinfo_name* est la valeur de l'attribut **CONNAUTH** du gestionnaire de files d'attente. Si la valeur de l'attribut **ADOPTCTX** est YES, le gestionnaire de files d'attente est configuré pour adopter l'ID utilisateur dans le jeton et l'attribut **UserClaim** doit être spécifié dans la section **AuthToken** .

   Définissez la valeur de l'attribut **UserClaim** sur le nom de la revendication de jeton qui contient l'ID utilisateur à adopter. Par exemple, si le jeton contient la revendication "AppUser": "MyUserName", ajoutez la ligne suivante à la section **AuthToken** :

   ```
   UserClaim=AppUser
   ```
6. Actualisez la configuration de sécurité du gestionnaire de files d'attente pour qu'elle récupère la configuration de jeton dans le fichier qm.ini . Exécutez la commande suivante pour démarrer la commande **runmqsc** :

   ```
   runmqsc qm1
   ```

   puis émettez la commande MQSC suivante:

```
REFRESH SECURITY TYPE(CONNAUTH)
```

Que faire ensuite

Travaillez avec vos développeurs pour les aider à comprendre comment ils peuvent utiliser des jetons dans des applications pour s'authentifier auprès du gestionnaire de files d'attente.

Concepts associés

Traitement des incidents liés aux jetons d'authentification

Tâches associées

Utilisation de jetons d'authentification dans une application

Référence associée

Section AuthToken du fichier qm.ini

V 9.4.0 Linux AIX Obtention d'un jeton d'authentification auprès de l'émetteur de jeton que vous avez choisi

Ecrivez votre application pour obtenir un jeton d'authentification auprès de l'émetteur de jeton que vous avez choisi lorsqu'il se connecte à un gestionnaire de files d'attente IBM MQ .

Avant de commencer

Reportez-vous aux informations de la rubrique «Utilisation de jetons d'authentification dans une application», à la page 346.

Procédure

- La façon dont vous obtenez un jeton d'authentification et le contenu exact du jeton varient d'un émetteur de jeton à l'autre.

  Ecrivez votre application pour interagir avec l'émetteur de jeton que vous avez choisi afin de demander et d'obtenir le jeton d'authentification. Le jeton d'authentification doit être conforme aux exigences de IBM MQ pour les jetons d'authentification. Pour plus d'informations sur ces exigences, voir «Conditions requises pour les jetons d'authentification», à la page 338.

  Si vous avez l'intention d'adopter un ID utilisateur contenu dans une revendication de jeton comme contexte de l'application, le jeton d'authentification doit également répondre aux exigences suivantes:

  – Le jeton d'authentification doit contenir une demande qui correspond au nom de la demande d'utilisateur dans la configuration d'authentification par jeton du gestionnaire de files d'attente.

  – La valeur de la revendication utilisateur doit répondre aux exigences relatives aux ID utilisateur dans les jetons d'authentification. Pour plus d'informations, voir «ID utilisateur dans les jetons d'authentification», à la page 341.

Résultats

Vous avez maintenant obtenu un JWT correctement formaté qui peut être présenté à IBM MQ pour validation.

Tâches associées

Configuration d'un gestionnaire de files d'attente pour qu'il accepte **AuthTokens**

Référence associée

Section AuthToken du fichier qm.ini

MQCSP-Paramètres de sécurité

V 9.4.0 Linux AIX Utilisation de jetons d'authentification dans une application

Ecrivez votre application pour fournir un jeton d'authentification lorsqu'elle se connecte à un gestionnaire de files d'attente IBM MQ .

Avant de commencer

Depuis IBM MQ 9.4.0, les applications peuvent fournir un jeton d'authentification lorsqu'elles se connectent à un gestionnaire de files d'attente.

L'application doit répondre aux exigences suivantes:

- Il doit être écrit en C ou Java (à l'aide de IBM MQ classes for JMS/ Jakarta Messaging)
- Il doit se connecter au gestionnaire de files d'attente en tant que IBM MQ client. C'est-à-dire que l'application doit se connecter au gestionnaire de files d'attente via un réseau, au lieu d'utiliser des liaisons locales.
- Il doit se connecter à un gestionnaire de files d'attente qui s'exécute sur AIX ou Linux.

Si l'application ne répond pas à ces exigences, la connexion échoue et le code anomalie MQRC_FUNCTION_NOT_SUPPORTED (2298) est renvoyé à l'application.

L'application qui fournit le jeton d'authentification peut s'exécuter sur n'importe quelle plateforme prenant en charge IBM MQ MQI clients.

Les clients qui utilisent la reconnexion client automatique ne peuvent pas fournir de jeton d'authentification lorsqu'ils se connectent. Si une application fournit un jeton d'authentification et spécifie l'option `MQCNO_RECONNECT` ou `MQCNO_RECONNECT_Q_MGR` dans la structure MQCNO, la connexion échoue et le code anomalie MQRC_RECONNECT_INCOMPATIBLE (2547) est renvoyé à l'application. Pour plus d'informations sur la reconnexion automatique du client, voir Reconnexion automatique du client.

Si vous ne pouvez pas écrire l'application pour fournir un jeton d'authentification en raison de ces exigences, vous pouvez également migrer votre application pour utiliser des jetons d'authentification à l'aide d'un exit de sécurité client. L'exit de sécurité client peut être écrit pour définir le jeton d'authentification dans la structure MQCSP. Pour plus d'informations sur les exits de sécurité, voir Exits de sécurité sur une connexion client.

Depuis IBM MQ 9.4.0, les applications client JMS peuvent fournir directement un jeton lors de la connexion (voir «Obtention d'un jeton d'authentification auprès de l'émetteur de jeton que vous avez choisi», à la page 346). Avant IBM MQ 9.4.0, les applications Java peuvent fournir indirectement un jeton par le biais d'un programme d'exit. Pour plus d'informations, voir Java class MQCSP.

Pourquoi et quand exécuter cette tâche

Remarque : Un jeton d'authentification conforme à la norme JWS (JSON Web Signature) est signé pour permettre la validation de l'authenticité du jeton, mais il n'est pas chiffré. Par conséquent, il peut être lu, et éventuellement réutilisé, par toute personne ayant accès au jeton. Configurez la connexion au gestionnaire de files d'attente pour vous assurer que le jeton d'authentification est protégé à l'aide du chiffrement lorsqu'il est envoyé sur le réseau, par exemple à l'aide de TLS. Pour plus d'informations sur les options de protection des données d'identification fournies par une application, voir «Protection par mot de passe MQCSP», à la page 32.

Avant de modifier les applications pour qu'elles se connectent à l'aide d'un jeton, vérifiez que:

- Le gestionnaire de files d'attente a été configuré pour accepter les jetons d'authentification en suivant les étapes de la rubrique «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un magasin de clés local», à la page 343
- Votre application peut obtenir un jeton valide requis à partir de votre serveur d'authentification. Voir «Obtention d'un jeton d'authentification auprès de l'émetteur de jeton que vous avez choisi», à la page 346.

Pour fournir un jeton d'authentification lorsque l'application se connecte à un gestionnaire de files d'attente IBM MQ , incluez le processus suivant.

Procédure

- Pour fournir un jeton d'authentification à partir d'une application C (MQI):

L'application doit se connecter à l'aide de MQCONNX (au lieu de MQCONN) et fournir une structure MQCSP :

- La zone **AuthenticationType** doit être définie sur MQCSP_AUTH_ID_TOKEN.
- La version de la structure doit être définie sur MQCSP_VERSION_3.
- La zone **TokenPtr** ou **TokenOffset** doit faire référence à votre jeton d'authentification.
- La zone **TokenLength** doit être définie sur la longueur du jeton d'authentification.

Exemple de code C pour la connexion à un gestionnaire de files d'attente à l'aide de MQCSP version 3 et d'un jeton d'authentification:

```
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         */
```

- Pour fournir un jeton d'authentification à partir d'une application Java :

  Les applications qui utilisent IBM MQ classes for JMS/Jakarta Messaging peuvent fournir un jeton via n'importe quelle méthode `createContext`ou `createConnection` , qui prend un nom d'utilisateur et un mot de passe.

  Pour fournir un jeton d'authentification, procédez comme suit:

  - **UserID** doit être défini sur null ou sur une chaîne vide, c'est-à-dire sans espaces, " "
  - Le jeton est fourni sous la forme de la chaîne **Password** .

  Cela s'applique à toutes les implémentations IBM MQ de l'interface `ConnectionFactory` .

  Les formes de paramètre explicites, par exemple, `createContext`(String **userID**, String **password**) peuvent être utilisées, ou les versions de paramètre implicites, par exemple, `createContext`().

  Dans ce dernier cas, le **userID** et le jeton **Password** vides doivent d'abord avoir été fournis en tant que propriétés sur la fabrique de connexions.

  Exemple de code Java pour la connexion à un gestionnaire de files d'attente à l'aide d'un jeton d'authentification:

```
// 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
```

Si la connexion échoue avec le code anomalie MQRC_NOT_AUTHORIZED (2035) ou MQRC_SECURITY_ERROR (2063), recherchez dans le journal des erreurs du gestionnaire de files d'attente un message d'erreur contenant plus d'informations sur la cause de l'échec. Pour plus d'informations sur le diagnostic des problèmes liés aux jetons d'authentification, voir Traitement des incidents liés aux jetons d'authentification.

Résultats

L'application est maintenant connectée au gestionnaire de files d'attente. Il reste connecté jusqu'à ce qu'il se déconnecte, même si le jeton utilisé pour l'authentification expire. Si l'application se déconnecte du gestionnaire de files d'attente et doit se reconnecter, elle peut avoir besoin d'obtenir un nouveau jeton d'authentification avec une heure d'expiration ultérieure avant de pouvoir se reconnecter.

Tâches associées

Configuration d'un gestionnaire de files d'attente pour qu'il accepte **AuthTokens**

Référence associée

Section AuthToken du fichier `qm.ini`

MQCSP-Paramètres de sécurité

V 9.4.0 Linux AIX Création d'un référentiel de clés à utiliser comme magasin de clés de confiance TLS

Lors de la création de connexions TLS sortantes, vous devez créer un'magasin de clés de confiance'simple qui peut valider les certificats signés par un ensemble commun d'autorités de certification. Les exemples de connexion TLS sont un canal client IBM MQ ou une connexion HTTPS, tels qu'ils sont utilisés lors de la configuration de certains composants d' IBM MQ.

Pourquoi et quand exécuter cette tâche

Avertissement : La détermination des certificats et des autorités de certification à faire confiance à votre environnement est une étape importante qui a des implications sur la sécurité de votre configuration de bout en bout. Cette rubrique illustre les étapes communes qui permettent aux composants IBM MQ de faire confiance au même ensemble de certificats déjà configurés pour votre système d'exploitation. En cas de doute, toutefois, vous devez discuter de ce processus avec votre administrateur de sécurité.

La plupart des systèmes d'exploitation UNIX et Linux disposent d'un emplacement de système de fichiers contenant un ensemble d'autorités de certification'dignes de confiance'. Ce système de fichiers peut avoir été configuré avec l'installation du système d'exploitation ou personnalisé par votre administrateur système (par exemple pour inclure des autorités de certification internes appartenant à votre organisation). Les emplacements de ces fichiers varient, mais certaines valeurs couramment utilisées pour les systèmes d'exploitation populaires sont les suivantes:

- 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`

Lorsque vous créez et configurez un magasin de clés IBM MQ , vous pouvez facilement ajouter tous les fichiers de certificat d'un répertoire, par exemple, `/etc/ssl/certs`, à une base de données de clés IBM MQ en une seule commande.

Procédure

1. Utilisez la commande suivante pour ajouter les fichiers de certificat à partir du répertoire `/etc/ssl/certs` :

```
runmqakm -cert -add -file /etc/ssl/certs/*.pem -db mykdb.p12 -stashed
```

2. Facultatif : Dans certains cas, il peut être utile de générer un ensemble de certificats'par défaut'pour votre magasin de clés de confiance.

Les composants de sécurité IBM MQ fournis avec le produit fournissent un ensemble de certificats d'autorité de certification'par défaut'.

Remarque : Ces certificats peuvent ne pas être fréquemment mis à jour et / ou avoir une durée de vie relativement courte.

Si vous souhaitez quand même utiliser les certificats de l'autorité de certification préconfigurés, vous pouvez générer un magasin de clés de confiance à l'aide des paramètres **populate** et **ibmcloudtrust** de la commande **runmqakm** :

```
runmqakm -keydb -create -db mqauto.p12 -genpw -stash -type pkcs12 -populate -ibmcloudtrust
```

Concepts associés

Traitement des incidents liés aux jetons d'authentification

Tâches associées

Utilisation de jetons d'authentification dans une application

Référence associée

Section AuthToken du fichier qm.ini

Utilisation des certificats révoqués

Les certificats numériques peuvent être révoqués par les autorités de certification. Vous pouvez vérifier le statut de révocation des certificats à l'aide d'OCSP ou de CRL sur les serveurs LDAP, en fonction de la plateforme.

Lors de l'établissement de liaison TLS, les partenaires communicants s'authentifient mutuellement avec des certificats numériques. L'authentification peut inclure une vérification du certificat reçu. Les autorités de certification révoquent les certificats pour diverses raisons, notamment:

- Le propriétaire a été déplacé vers une autre organisation
- La clé privée n'est plus un secret

Les autorités de certification publient les certificats personnels révoqués dans une liste de révocation de certificat (CRL). Les certificats d'autorités de certification qui ont été révoqués sont publiés dans une liste de révocation des droits d'accès (ARL).

ALW Sur les plateformes AIX, Linux, and Windows , la prise en charge SSL de IBM MQ recherche les certificats révoqués à l'aide du protocole OCSP (Online Certificate Status Protocol) ou des listes CRL et ARL sur les serveurs LDAP (Lightweight Directory Access Protocol). OCSP est la méthode préférée.

IBM MQ classes for Java et IBM MQ classes for JMS ne peuvent pas utiliser les informations OCSP dans un fichier de table de définition de canal du client. Toutefois, vous pouvez configurer OCSP comme indiqué dans Using Online Certificate Protocol.

IBM i Sous IBM i, le support SSL de IBM MQ recherche les certificats révoqués à l'aide des listes de révocation de certificat et des listes de révocation de certificat sur les serveurs LDAP uniquement.

z/OS Sous z/OS, le support SSL de IBM MQ recherche les certificats révoqués à l'aide des listes de révocation de certificat et des listes de révocation de certificat sur les serveurs LDAP uniquement.

Pour plus d'informations sur les autorités de certification, voir «Certificats numériques», à la page 13.

Vérification d'OCSP/CRL

La vérification du protocole OCSP (Online Certificate Status Protocol) /de la liste de révocation de certificat (CRL) est effectuée par rapport aux certificats entrants distants. Le processus vérifie l'ensemble de la chaîne impliquée depuis le certificat personnel du système distant jusqu'à son certificat racine.

Utilisation d' openSSL pour vérifier la validation OCSP

Si votre entreprise utilise openSSL pour valider OCSP, puis que vous tentez d'utiliser une connexion TLS IBM Global Security Kit (GSKit) , vous recevez un avertissement de statut UNKNOWN.

En effet, tous les certificats de la chaîne, à l'exception de la racine, sont vérifiés par GSKit pour leur statut de révocation. L'opération GSKit est conforme à RFC 5280 et est décrite dans la stratégie de confiance GSKit . L'algorithme GSKit essaie toutes les sources disponibles pour les informations de révocation, comme décrit dans RFC 5280 et dans la stratégie de confiance GSKit .

Comment la vérification OCSP/CRL fonctionne-t-elle dans IBM MQ?

IBM MQ prend en charge deux mécanismes permettant de contrôler le comportement lors de la vérification des certificats par rapport aux noeuds finaux OCSP ou CRL nommés, soit dans l'extension de certificat, soit, comme défini dans les objets AUTHINFO:

- Les attributs **OCSPCheckExtensions**, **CDPCheckExtensions**et **OCSPAuthentication** de la strophe SSL du fichier qm.ini, et
- Utilisation du paramètre SSLCRLNL du gestionnaire de files d'attente et des configurations AUTHINFO OCSP et CRLLDAP. Pour plus d'informations, voir ALTER AUTHINFO et ALTER QMGR .

Avertissement :

La commande ALTER AUTHINFO avec **AUTHTYPE(OCSP)** ne s'applique pas aux gestionnaires de files d'attente IBM i ou z/OS . Toutefois, il peut être spécifié sur ces plateformes pour être copié dans la table de définition de canal du client (CCDT) à des fins d'utilisation par le client.

Les attributs de strophe SSL **OCSPCheckExtensions** et **CDPCheckExtensions** contrôlent si IBM MQ doit vérifier un certificat par rapport au serveur OCSP ou CRL détaillé dans l'extension AIA du certificat.

Si cette option n'est pas activée, le serveur OCSP ou CRL de l'extension de certificat n'est pas contacté.

Si des serveurs OCSP ou CRL sont détaillés via des objets AUTHINFO et référencés à l'aide de l'attribut SSLCRLNL **QMGR** , lors du traitement de la révocation de certificat, IBM MQ tente de contacter ces serveurs.

Important : Un seul objet OCSP AUTHINFO peut être défini dans la liste de noms SSLCRLNL.

If :

OCSPCheckExtensions= NO et **CDPCheckExtensions**=*NO* sont définis, et
Aucun serveur OCSP ou CRL n'est défini dans les objets AUTHINFO

aucune vérification de révocation de certificat n'est effectuée.

Lors de la vérification d'un certificat pour son statut de révocation, IBM MQ contacte les serveurs OCSP ou CRL nommés dans l'ordre suivant, s'ils sont activés:

1. Le serveur OCSP est détaillé dans un objet **AUTHTYPE(OCSP)** et référencé dans l'attribut SSLCRLNL **QMGR** .
2. Serveurs OCSP détaillés dans l'extension AIA des certificats, si **OCSPCheckExtensions**=*YES*.
3. Serveurs CRL détaillés dans l'extension **CRLDistributionPoints** des certificats, si **CDPCheckExtensions** =*YES*.
4. Tous les serveurs CRL détaillés dans les objets **AUTHINFO(CRLLDAP)** et référencés dans l'attribut SSLCRLNL **QMGR** .

Lors de la vérification d'un certificat, si une étape aboutit à ce que le serveur OCSP ou le serveur CRL renvoie une réponse REVOKED ou VALID définitive à une requête pour le certificat, aucune autre vérification n'est effectuée et le statut du certificat tel que présenté est utilisé pour déterminer s'il est digne de confiance ou non.

Si un serveur OCSP ou CRL renvoie un résultat UNKNOWN, le traitement se poursuit jusqu'à ce qu'un serveur OCSP ou CRL renvoie un résultat définitif ou que toutes les options soient épuisées.

Le comportement selon lequel un certificat est considéré comme révoqué, si son statut ne peut pas être déterminé, est différent pour les serveurs OCSP et CRL:

- Pour les serveurs CRL, si aucune CRL ne peut être obtenue, le certificat est considéré comme `NOT_REVOKED`
- Pour les serveurs OCSP, si aucun statut de révocation ne peut être obtenu à partir d'un serveur OCSP nommé, le comportement est contrôlé via l'attribut **OCSPAuthentication** dans la strophe SSL du fichier qm.ini .

  Vous pouvez configurer cet attribut pour bloquer une connexion, autoriser une connexion ou autoriser une connexion avec un message d'avertissement.

Vous pouvez utiliser l'attribut **SSLHTTPProxyName**=*string* dans la strophe SSL des fichiers qm.ini et mqclient.ini pour les vérifications OCSP, si nécessaire. La chaîne correspond au nom d'hôte ou à l'adresse réseau du serveur proxy HTTP qui doit être utilisé par GSKit pour les vérifications OCSP.

Vous pouvez définir la valeur **OCSPTimeout** dans la strophe SSL des fichiers `qm.ini` ou `mqclient.ini` qui définit le nombre de secondes d'attente d'un répondeur OCSP lors de l'exécution d'une vérification de révocation.

ALW Certificats révoqués et OCSP

IBM MQ détermine le répondeur OCSP (Online Certificate Status Protocol) à utiliser et traite la réponse reçue. Vous pouvez être amené à réaliser certaines étapes pour pouvoir accéder au répondeur OCSP.

Remarque : Ces informations s'appliquent uniquement à IBM MQ sur les systèmes AIX, Linux, and Windows .

Pour vérifier le statut de révocation d'un certificat numérique à l'aide d'OCSP, IBM MQ peut utiliser deux méthodes pour déterminer le répondeur OCSP à contacter:

- A l'aide de l'extension de certificat AuthorityInfoAccess (AIA) dans le certificat à vérifier.
- A l'aide d'une adresse URL spécifiée dans un objet d'informations d'authentification ou spécifiée par une application client.

Une URL spécifiée dans un objet d'informations d'authentification ou par une application client est prioritaire par rapport à une URL d'une extension de certificat AIA.

Si l'adresse URL du répondeur OCSP se trouve derrière le pare-feu, reconfigurez le pare-feu pour que le répondeur OCSP soit accessible ou configurez un serveur proxy OCSP. Indiquez le nom du serveur proxy en utilisant la variable SSLHTTPProxyName dans la strophe SSL. Sur les systèmes client, vous pouvez également indiquer le nom du serveur proxy à l'aide de la variable d'environnement MQSSLPROXY. Pour plus de détails, consultez les informations connexes.

Si vous n'êtes pas concerné par la révocation des certificats TLS, peut-être parce que vous exécutez un environnement de test, vous pouvez définir OCSPCheckExtensions sur NO dans la strophe SSL. Si vous configurez cette variable, toute extension de certificat AIA est ignorée. Cette solution sera probablement refusée dans un environnement de production, dans lequel vous ne souhaitez sûrement pas autoriser les utilisateurs à accéder aux certificats révoqués.

L'appel d'accès au répondeur OCSP peut entraîner l'un des trois résultats suivants :

Bon
Le certificat est valide.

Révoqué
Le certificat est révoqué.

Inconnu
Ce résultat peut survenir à cause de l'une des trois raisons suivantes :

- IBM MQ ne peut pas accéder au répondeur OCSP.
- Le répondeur OCSP a envoyé une réponse, mais IBM MQ ne peut pas vérifier la signature numérique de la réponse.

- Le répondeur OCSP a envoyé une réponse qui indique qu'il n'existe pas de données de révocation pour le certificat.

Si IBM MQ reçoit un résultat OCSP `Inconnu`, son comportement dépend de la valeur de l'attribut OCSPAuthentication. Pour les gestionnaires de files d'attente, cet attribut est conservé dans l'un des emplacements suivants:

- Linux AIX Dans la section SSL du fichier `qm.ini` sous AIX and Linux.
- Windows Dans le registre Windows .

Cet attribut peut être défini à l'aide de IBM MQ Explorer. Pour les clients, l'attribut est conservé dans la section SSL du fichier de configuration du client.

Si `Inconnu` est reçu et que l'attribut OCSPAuthentication a la valeur REQUIRED (valeur par défaut), IBM MQ rejette la connexion et envoie un message d'erreur de type AMQ9716.
Si les messages d'événements SSL de gestionnaire de files d'attente sont activés, un message d'événement SSL de type MQRC_CHANNEL_SSL_ERROR, avec ReasonQualifier défini sur MQRQ_SSL_HANDSHAKE_ERROR, est généré.

Si `Inconnu` est reçu et que l'attribut OCSPAuthentication a la valeur OPTIONAL, IBM MQ permet au canal SSL de démarrer, et aucun avertissement ou message d'événement SSL n'est généré.

Si `Inconnu` est reçu et que l'attribut OCSPAuthentication a la valeur WARN, le canal SSL démarre, mais IBM MQ génère un message d'avertissement de type AMQ9717 dans le journal des erreurs. Si les messages d'événements SSL de gestionnaire de files d'attente sont activés, un message d'événement SSL de type MQRC_CHANNEL_SSL_WARNING, avec ReasonQualifier défini sur MQRQ_SSL_UNKNOWN_REVOCATION, est généré.

Signature numérique de réponses OCSP

Un répondeur OCSP peut signer ses réponses de trois manières. Votre répondeur vous informe de la méthode à utiliser.

- La réponse OCSP peut être signée numériquement à l'aide du même certificat CA qui a émis le certificat en cours de vérification. Dans ce cas, vous n'avez pas besoin de configurer de certificat supplémentaire ; les étapes que vous avez déjà effectuées pour établir la connectivité TLS sont suffisantes pour vérifier la réponse OCSP.
- La réponse OCSP peut être signée numériquement à l'aide d'un autre certificat signé par la même autorité de certification que celle ayant émis le certificat en cours de vérification. Le certificat signataire est envoyé avec la réponse OCSP dans ce cas. Le certificat transmis à partir du répondeur OCSP doit avoir une extension d'utilisation clé étendue définie sur `id-kp-OCSPSigning` pour pouvoir être digne de confiance. Etant donné que la réponse OCSP est envoyée avec le certificat qui l'a signé (et que ce certificat est signé par une autorité de certification déjà digne de confiance pour la connectivité TLS), aucune configuration de certificat supplémentaire n'est requise.
- La réponse OCSP peut être signée numériquement à l'aide d'un autre certificat qui n'est pas directement lié au certificat en cours de vérification. Dans ce cas, la réponse OCSP est signée par un certificat émis par le répondeur OCSP. Vous devez ajouter une copie du certificat de répondeur OCSP à la base de données de clés du client ou du gestionnaire de files d'attente qui effectue la vérification OCSP. Voir «Ajout d'un certificat de l'autorité de certification ou de la partie publique d'un certificat digne de confiance dans un référentiel de clés sous AIX, Linux, and Windows», à la page 567. Lors de l'ajout d'un certificat CA, il est ajouté par défaut en racine de confiance, ce qui représente le paramètre requis dans ce contexte. Si ce certificat n'est pas ajouté, IBM MQ ne peut pas vérifier la signature numérique sur la réponse OCSP et la vérification OCSP génère un résultat Inconnu, ce qui peut entraîner la fermeture du canal par IBM MQ , en fonction de la valeur d'OCSPAuthentication.

Protocole OCSP (Online Certificate Status Protocol) dans les applications client Java et JMS

En raison d'une limitation de l'API Java , IBM MQ peut utiliser la vérification de révocation de certificat OCSP (Online Certificate Status Protocol) pour les sockets TLS sécurisés uniquement lorsque OCSP est

activé pour l'ensemble du processus de la machine virtuelle Java (JVM). OCSP peut être activé pour toutes les connexions sécurisées de la machine virtuelle Java de deux manières :

- En modifiant le fichier JRE java.security pour y inclure les paramètres de configuration OCSP affichés dans le tableau 1 et en redémarrant l'application.
- Utilisez java.security.Security.setProperty() API, soumise à toute stratégie Java Security Manager en vigueur.

Vous devez au moins spécifier l'une des valeurs ocsp.enable et ocsp.responderURL.

Nom de la propriété	**Description**
ocsp.enable	Cette propriété a la valeur `true` ou `false`. Si la valeur est `true`, la vérification OCSP est activée lors de la vérification de révocation de certificat. Si la valeur est `false` (ou non définie), la vérification OCSP est désactivée.
ocsp.responderURL	La valeur de cette propriété est une adresse URL identifiant l'emplacement du canal répondeur OCSP. Par exemple, `ocsp.responderURL=http://ocsp.example.net:80`. Par défaut, l'emplacement du canal répondeur OCSP est déterminé de manière implicite à partir du certificat en cours de validation. La propriété est utilisée lorsque l'extension Authority Information Access (définie dans RFC 3280) n'est pas indiquée dans le certificat ou lorsqu'elle doit être remplacée.
ocsp.responderCertSubjectName	La valeur de cette propriété est le nom du sujet du certificat du canal répondeur OCSP. Par exemple, `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`. Par défaut, le certificat du canal répondeur OCSP est celui de l'émetteur du certificat en cours de validation. Cette propriété identifie le certificat du canal répondeur OCSP lorsque la valeur par défaut ne s'applique pas. Sa valeur est un nom distinctif de chaîne (défini dans RFC 2253) qui identifie un certificat dans l'ensemble des certificats fournis lors de la validation du chemin aux certificats. Lorsque le seul nom du sujet ne suffit pas à identifier le certificat, alors les propriétés ocsp.responderCertIssuerName et ocsp.responderCertSerialNumber doivent toutes deux être utilisées. Lorsque cette propriété est définie, les propriétés ocsp.responderCertIssuerName et ocsp.responderCertSerialNumber sont ignorées.
ocsp.responderCertIssuerName	La valeur de cette propriété est le nom de l'émetteur du certificat du canal répondeur OCSP. Par exemple, `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`. Par défaut, le certificat du canal répondeur OCSP est celui de l'émetteur du certificat en cours de validation. Cette propriété identifie le certificat du canal répondeur OCSP lorsque la valeur par défaut ne s'applique pas. Sa valeur est un nom distinctif de chaîne (défini dans RFC 2253) qui identifie un certificat dans l'ensemble des certificats fournis lors de la validation du chemin aux certificats. Lorsque cette propriété est définie, la propriété ocsp.responderCertSerialNumber doit également être définie. Cette propriété est ignorée lorsque la propriété ocsp.responderCertSubjectName est définie.
ocsp.responderCertSerialNumber	La valeur de cette propriété est le numéro de série du certificat du canal répondeur OCSP. Par exemple, `ocsp.responderCertSerialNumber=2A:FF:00`. Par défaut, le certificat du canal répondeur OCSP est celui de l'émetteur du

Nom de la propriété	Description
	certificat en cours de validation. Cette propriété identifie le certificat du canal répondeur OCSP lorsque la valeur par défaut ne s'applique pas. Cette valeur est une chaîne de chiffres hexadécimaux (séparés par des signes deux-points ou des espaces) qui identifie un certificat dans l'ensemble des certificats fournis lors de la validation du chemin d'accès aux certificats. Lorsque cette propriété est définie, la propriété ocsp.responderCertIssuerName doit également être définie. Cette propriété est ignorée lorsque la propriété ocsp.responderCertSubjectName est définie.

Avant d'activer OCSP de cette manière, tenez compte des remarques suivantes :

- La définition de la configuration OCSP affecte toutes les connexions sécurisées du processus de la machine virtuelle Java. Dans certains cas, cette configuration peut avoir des effets secondaires indésirables lorsque la machine virtuelle Java est partagée avec un autre code d'application qui utilise des sockets TLS sécurisés. Assurez-vous que la configuration OCSP choisie est appropriée à toutes les applications s'exécutant sur la même machine virtuelle Java.
- L'application de la maintenance à votre environnement d'exécution Java écrase le fichier java.security. Soyez prudent lorsque vous appliquez des correctifs temporaires Java et la maintenance du produit pour éviter d'écraser le fichier java.security . Il peut s'avérer nécessaire d'appliquer à nouveau les modifications de votre fichier java.security après la maintenance. Pour cette raison, vous pouvez envisager de définir la configuration OCSP à l'aide de l'interface de programme d'application java.security.Security.setProperty().
- L'activation de la vérification OCSP n'est effective que si la vérification de la révocation est également activée. La vérification de la révocation est activée via la méthode `PKIXParameters.setRevocationEnabled()`.
- Si vous utilisez l'intercepteur AMS Java décrit dans Activation de la vérification OCSP dans les intercepteurs natifs, prenez soin d'éviter d'utiliser une configuration OCSP java.security qui entre en conflit avec la configuration AMS OCSP dans le fichier de configuration du magasin de clés.

Utilisation des listes de révocation de certificat et des listes de révocation d'autorité

La prise en charge de IBM MQ pour les CRL et les ARL varie en fonction de la plateforme.

Le support CRL et ARL sur chaque plateforme est le suivant:

- Multi Sur Multiplatforms, la prise en charge de CRL et ARL est conforme aux recommandations de profil de CRL PKIX X.509 V2 .
- z/OS Sous z/OS, System SSL prend en charge les listes CRL et ARL stockées sur les serveurs LDAP par le produit Tivoli Public Key Infrastructure.

IBM MQ gère un cache des listes de révocation de certificat et des listes de révocation de certificat qui ont été consultées au cours des 12 dernières heures.

Lorsqu'un gestionnaire de files d'attente ou IBM MQ MQI client reçoit un certificat, il vérifie la liste de révocation de certificat pour confirmer que le certificat est toujours valide. IBM MQ vérifie d'abord dans le cache s'il existe un cache. Si la liste de révocation de certificat n'est pas dans le cache, IBM MQ interroge les emplacements du serveur de listes de révocation de certificat LDAP dans l'ordre dans lequel ils apparaissent dans la liste de noms des objets d'informations d'authentification spécifiée par l'attribut *SSLCRLNL* , jusqu'à ce que IBM MQ trouve une liste de révocation de certificat disponible. Si la liste de noms n'est pas spécifiée ou qu'elle est spécifiée avec une valeur vide, les listes de révocation de nom ne sont pas vérifiées.

Configuration des serveurs LDAP

Configurez la structure de l'arborescence d'informations de l'annuaire LDAP pour refléter la hiérarchie des noms distinctifs des autorités de certification. Pour ce faire, utilisez les fichiers LDAP Data Interchange Format.

Configurez la structure DIT (Directory Information Tree) LDAP pour utiliser la hiérarchie correspondant aux noms distinctifs des autorités de certification qui émettent les certificats et les CRL. Vous pouvez configurer la structure DIT avec un fichier qui utilise le format LDIF (LDAP Data Interchange Format). Vous pouvez également utiliser des fichiers LDIF pour mettre à jour un répertoire.

Les fichiers LDIF sont des fichiers texte ASCII qui contiennent les informations requises pour définir des objets dans un annuaire LDAP. Les fichiers LDIF contiennent une ou plusieurs entrées, dont chacune comprend un nom distinctif, au moins une définition de classe d'objet et, éventuellement, plusieurs définitions d'attribut.

L'attribut `certificateRevocationList;binary` contient une liste, au format binaire, des certificats d'utilisateur révoqués. L'attribut `authorityRevocationList;binary` contient une liste binaire des certificats de l'autorité de certification qui ont été révoqués. Pour une utilisation avec IBM MQ TLS, les données binaires de ces attributs doivent être conformes au format DER (Défini Encoding Rules). Pour plus d'informations sur les fichiers LDIF, reportez-vous à la documentation fournie avec votre serveur LDAP.

La Figure 20, à la page 356 présente un exemple de fichier LDIF que vous pouvez créer en entrée de votre serveur LDAP pour charger les CRL et les ARL émis par CA1, qui est une autorité de certification imaginaire avec le nom distinctif "CN=CA1, OU=Test, O=IBM, C=GB", configuré par l'organisation de test dans IBM.

```
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)
```

Figure 20. Exemple de fichier LDIF pour une autorité de certification. Cela peut varier d'une implémentation à l'autre.

La Figure 21, à la page 357 montre la structure DIT créée par votre serveur LDAP lorsque vous chargez l'exemple de fichier LDIF présenté dans Figure 20, à la page 356 avec un fichier similaire pour CA2, une autorité de certification imaginaire configurée par l'organisation PKI, également dans IBM.

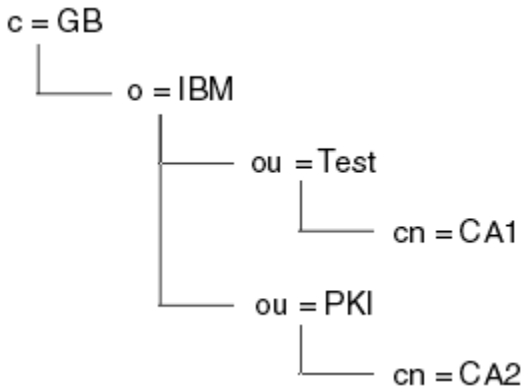


Figure 21. Exemple de structure d'arborescence d'informations d'annuaire LDAP

IBM MQ vérifie les listes de révocation de certificat et les listes de révocation de certificat.

Remarque : Assurez-vous que la liste de contrôle d'accès de votre serveur LDAP permet aux utilisateurs autorisés de lire, de rechercher et de comparer les entrées qui contiennent les CRL et les ARL. IBM MQ accède au serveur LDAP à l'aide des propriétés LDAPUSER et LDAPPWD de l'objet AUTHINFO.

Configuration et mise à jour des serveurs LDAP
Utilisez cette procédure pour configurer ou mettre à jour votre serveur LDAP.

1. Procurez-vous les listes de révocation de certificat et les listes de révocation de certificat au format DER auprès de votre ou de vos autorités de certification.
2. A l'aide d'un éditeur de texte ou de l'outil fourni avec votre serveur LDAP, créez un ou plusieurs fichiers LDIF contenant le nom distinctif de l'autorité de certification et les définitions de classe d'objets requises. Copiez les données au format DER dans le fichier LDIF en tant que valeurs de l'attribut `certificateRevocationList;binary` pour les CRL, de l'attribut `authorityRevocationList;binary` pour les ARL, ou les deux.
3. Démarrez votre serveur LDAP.
4. Ajoutez les entrées du ou des fichiers LDIF que vous avez créés à l'étape «2», à la page 357.

Après avoir configuré votre serveur CRL LDAP, vérifiez qu'il est correctement configuré. Tout d'abord, essayez d'utiliser un certificat qui n'est pas révoqué sur le canal et vérifiez que le canal démarre correctement. Utilisez ensuite un certificat révoqué et vérifiez que le canal ne démarre pas.

Obtenir fréquemment des listes de révocation de certificats mises à jour auprès des autorités de certification. Envisagez de le faire sur vos serveurs LDAP toutes les 12 heures.

Accès aux CRL et aux ARL à l'aide d'un gestionnaire de files d'attente

Un gestionnaire de files d'attente est associé à un ou plusieurs objets d'informations d'authentification, qui contiennent l'adresse d'un serveur CRL LDAP. IBM i IBM MQ on IBM i se comporte différemment des autres plateformes.

Notez que dans cette section, les informations sur les listes de révocation de certificat (CRL) s'appliquent également aux listes de révocation d'autorité (ARL).

Vous indiquez au gestionnaire de files d'attente comment accéder aux listes de révocation de certificat en lui fournissant des objets d'informations d'authentification, chacun contenant l'adresse d'un serveur de listes de révocation de certificat LDAP. Les objets d'informations d'authentification sont conservés dans une liste de noms, qui est spécifiée dans l'attribut de gestionnaire de files d'attente *SSLCRLNL* .

Dans l'exemple suivant, MQSC est utilisé pour spécifier les paramètres:

1. Définissez les objets d'informations d'authentification à l'aide de la commande DEFINE AUTHINFO MQSC, avec le paramètre AUTHTYPE défini sur CRLLDAP. IBM i Sous IBM i, vous pouvez également utiliser la commande CL CRTMQMAUTI.

La valeur CRLLDAP pour le paramètre AUTHTYPE indique que les CRL sont accessibles sur les serveurs LDAP. Chaque objet d'informations d'authentification de type CRLLDAP que vous créez contient l'adresse d'un serveur LDAP. Lorsque vous disposez de plusieurs objets d'informations d'authentification, les serveurs LDAP vers lesquels ils pointent doivent contenir des informations identiques. Cela assure la continuité du service en cas d'échec d'un ou de plusieurs serveurs LDAP.

z/OS De plus, sous z/OS uniquement, tous les serveurs LDAP doivent être accessibles à l'aide des mêmes ID utilisateur et mot de passe. L'ID utilisateur et le mot de passe utilisés sont ceux indiqués dans le premier objet AUTHINFO de la liste de noms.

Sur toutes les plateformes, l'ID utilisateur et le mot de passe sont envoyés au serveur LDAP en clair.

2. A l'aide de la commande DEFINE NAMELIST MQSC, définissez une liste de noms pour les noms de vos objets d'informations d'authentification. z/OS Sous z/OS, vérifiez que l'attribut de liste de noms NLTYPE est défini sur AUTHINFO.
3. A l'aide de la commande ALTER QMGR MQSC, fournissez la liste de noms au gestionnaire de files d'attente. Exemple :

```
ALTER QMGR SSLCRLNL(sslcrlnlname)
```

où `sslcrlnlname` est votre liste de noms d'objets d'informations d'authentification.

Cette commande définit un attribut de gestionnaire de files d'attente appelé *SSLCRLNL*. La valeur initiale du gestionnaire de files d'attente pour cet attribut est vide.

IBM i Sous IBM i, vous pouvez spécifier des objets d'informations d'authentification, mais le gestionnaire de files d'attente n'utilise ni des objets d'informations d'authentification ni une liste de noms d'objets d'informations d'authentification. Seuls les clients IBM MQ qui utilisent une table de connexion client générée par un gestionnaire de files d'attente IBM i utilisent les informations d'authentification spécifiées pour ce gestionnaire de files d'attente IBM i . L'attribut de gestionnaire de files d'attente *SSLCRLNL* dans IBM i détermine les informations d'authentification utilisées par ces clients. Pour savoir comment indiquer à un gestionnaire de files d'attente IBM i comment accéder aux listes de révocation de certificat, voir «Accès aux CRL et aux ARL sous IBM i», à la page 358 .

Vous pouvez ajouter jusqu'à 10 connexions à des serveurs LDAP de remplacement à la liste de noms afin d'assurer la continuité du service en cas d'échec d'un ou de plusieurs serveurs LDAP. Notez que les serveurs LDAP doivent contenir des informations identiques.

IBM i *Accès aux CRL et aux ARL sous IBM i*

Utilisez cette procédure pour accéder aux CRL ou aux ARL sous IBM i.

Notez que dans cette section, les informations sur les listes de révocation de certificat (CRL) s'appliquent également aux listes de révocation d'autorité (ARL).

Pour configurer un emplacement de liste de révocation de certificat pour un certificat spécifique sur IBM i, procédez comme suit:

1. Accédez à l'interface DCM, comme décrit dans «Accès à DCM», à la page 283.
2. Dans la catégorie de tâche **Gérer les emplacements de liste de révocation de certificat** du panneau de navigation, cliquez sur **Ajouter un emplacement de liste de révocation de certificat**. La page Gérer les emplacements de liste de révocation de certificat s'affiche dans le cadre de la tâche.
3. Dans la zone **Nom d'emplacement de liste de révocation de certificat** , entrez un nom d'emplacement de liste de révocation de certificat, par exemple LDAP `Server #1`
4. Dans la zone **Serveur LDAP** , entrez le nom du serveur LDAP.
5. Dans la zone **Utiliser SSL (Secure Sockets Layer)** , sélectionnez **Oui** si vous souhaitez vous connecter au serveur LDAP à l'aide de TLS. Sinon, sélectionnez **Non**.
6. Dans la zone **Numéro de port** , entrez un numéro de port pour le serveur LDAP, par exemple 389.

7. Si votre serveur LDAP n'autorise pas les utilisateurs anonymes à interroger l'annuaire, entrez un nom distinctif de connexion pour le serveur dans la zone **Nom distinctif de connexion** .
8. Cliquez sur **OK**. DCM vous informe qu'il a créé l'emplacement de la liste de révocation de certificat.
9. Dans le panneau de navigation, cliquez sur **Sélectionner un magasin de certificats**. La page Sélectionner un magasin de certificats s'affiche dans le cadre de la tâche.
10. Cochez la case **Autre magasin de certificats système** et cliquez sur **Continuer**. La page Magasin de certificats et mot de passe s'affiche.
11. Dans la zone **Chemin d'accès au magasin de certificats et nom de fichier** , entrez le chemin d'accès au système de fichiers intégré et le nom de fichier que vous définissez lorsque «Création d'un magasin de certificats sous IBM i», à la page 286.
12. Entrez un mot de passe dans la zone **Certificate Store Password** . Cliquez sur **Continuer**. La page Magasin de certificats en cours s'affiche dans le cadre de la tâche.
13. Dans la catégorie de tâche **Gérer les certificats** du panneau de navigation, cliquez sur **Mettre à jour l'affectation d'emplacement de liste de révocation de certificat**. La page Affectation d'emplacement de liste de révocation de certificat s'affiche dans le cadre de la tâche.
14. Sélectionnez le bouton d'option du certificat de l'autorité de certification auquel vous souhaitez affecter l'emplacement de la liste de révocation de certificat. Cliquez sur **Mettre à jour l'affectation d'emplacement de liste de révocation de certificat**. La page Mettre à jour l'affectation d'emplacement de liste de révocation de certificat s'affiche dans le cadre de la tâche.
15. Sélectionnez le bouton d'option correspondant à l'emplacement de la liste de révocation de certificat que vous souhaitez affecter au certificat. Cliquez sur **Mettre à jour l'affectation**. DCM vous informe qu'il a mis à jour l'affectation.

Notez que DCM vous permet d'affecter un serveur LDAP différent par l'autorité de certification.

Accès aux CRL et aux ARL à l'aide de IBM MQ Explorer
Vous pouvez utiliser IBM MQ Explorer pour indiquer à un gestionnaire de files d'attente comment accéder aux CRL.

Notez que dans cette section, les informations sur les listes de révocation de certificat (CRL) s'appliquent également aux listes de révocation d'autorité (ARL).

Utilisez la procédure suivante pour configurer une connexion LDAP à une CRL:

1. Vérifiez que vous avez démarré votre gestionnaire de files d'attente.
2. Cliquez avec le bouton droit de la souris sur le dossier **Informations d'authentification** , puis cliquez sur **Nouveau-> Informations d'authentification**. Dans la feuille de propriétés qui s'ouvre:
   a. Sur la première page **Créer des informations d'authentification**, entrez un nom pour l'objet CRL (LDAP).
   b. Dans la page **Général** de **Modifier les propriétés**, sélectionnez le type de connexion. Vous pouvez éventuellement entrer une description.
   c. Sélectionnez la page **CRL (LDAP)** de **Modifier les propriétés**.
   d. Entrez le nom du serveur LDAP en tant que nom de réseau ou adresse IP.
   e. Si le serveur requiert des détails de connexion, indiquez un ID utilisateur et, si nécessaire, un mot de passe.
   f. Cliquez sur **OK**.
3. Cliquez avec le bouton droit de la souris sur le dossier Listes de noms , puis cliquez sur **Nouveau-> Liste de noms**. Dans la feuille de propriétés qui s'ouvre:
   a. Entrez un nom pour la liste de noms.
   b. Ajoutez le nom de l'objet CRL (LDAP) (à l'étape «2.a», à la page 359) à la liste.
   c. Cliquez sur **OK**.
4. Cliquez avec le bouton droit de la souris sur le gestionnaire de files d'attente, sélectionnez **Propriétés**, puis sélectionnez la page **SSL** :

a. Cochez la case **Vérifier les certificats reçus par ce gestionnaire de files d'attente par rapport aux listes de révocation de certification** .

b. Entrez le nom de la liste de noms (à l'étape «3.a», à la page 359) dans la zone **Liste de noms CRL** .

Accès aux CRL et aux ARL à l'aide d'un IBM MQ MQI client

Vous disposez de trois options pour spécifier les serveurs LDAP qui contiennent des listes CRL à vérifier par un IBM MQ MQI client.

Notez que dans cette section, les informations sur les listes de révocation de certificat (CRL) s'appliquent également aux listes de révocation d'autorité (ARL).

Les trois méthodes de spécification des serveurs LDAP sont les suivantes:

- Utilisation d'une table de définition de canal
- Utilisation de la structure des options de configuration SSL, MQSCO, sur un appel MQCONNX
- Utilisation de Active Directory (sur les systèmes Windows avec prise en charge d' Active Directory)

Pour plus de détails, reportez-vous aux informations associées.

Vous pouvez inclure jusqu'à 10 connexions à d'autres serveurs LDAP pour assurer la continuité du service en cas d'échec d'un ou de plusieurs serveurs LDAP. Notez que les serveurs LDAP doivent contenir des informations identiques.

Vous ne pouvez pas accéder aux CRL LDAP à partir d'un canal IBM MQ MQI client s'exécutant sur Linux (plateforme zSeries).

Emplacement d'un répondeur OCSP et des serveurs LDAP qui contiennent des listes CRL
Sur un système IBM MQ MQI client , vous pouvez spécifier l'emplacement d'un répondeur OCSP et des serveurs LDAP (Lightweight Directory Access Protocol) qui contiennent des listes de révocation de certificat (CRL).

Vous pouvez spécifier ces emplacements de trois manières, décrites ici par ordre de priorité décroissante.

IBM i Pour IBM i, voir Accès aux CRL et aux ARL sur IBM i.

Lorsqu'une application IBM MQ MQI client émet un appel MQCONNX

Vous pouvez spécifier un répondeur OCSP ou un serveur LDAP contenant des CRL sur un appel **MQCONNX** .

Sur un appel **MQCONNX** , la structure d'options de connexion, MQCNO, peut faire référence à une structure d'options de configuration SSL, MQSCO. A son tour, la structure MQSCO peut référencer une ou plusieurs structures d'enregistrement d'informations d'authentification, MQAIR. Chaque structure MQAIR contient toutes les informations dont IBM MQ MQI client a besoin pour accéder à un répondeur OCSP ou à un serveur LDAP contenant des CRL. Par exemple, l'une des zones d'une structure MQAIR est l'URL à laquelle un répondeur peut être contacté. Pour plus d'informations sur la structure MQAIR, voir MQAIR-Enregistrement des informations d'authentification.

Utilisation d'une table de définition de canal du client (ccdt) pour accéder à un répondeur OCSP ou à des serveurs LDAP

Pour qu'un IBM MQ MQI client puisse accéder à un répondeur OCSP ou à des serveurs LDAP qui contiennent des CRL, incluez les attributs d'un ou de plusieurs objets d'informations d'authentification dans une table de définition de canal du client.

Sur un gestionnaire de files d'attente de serveur, vous pouvez définir un ou plusieurs objets d'informations d'authentification. Les attributs d'un objet d'authentification contiennent toutes les informations requises pour accéder à un répondeur OCSP (sur les plateformes où OCSP est pris en charge) ou à un serveur LDAP qui contient des CRL. L'un des attributs spécifie l'URL du répondeur OCSP, l'autre l'adresse de l'hôte ou l'adresse IP d'un système sur lequel s'exécute un serveur LDAP.

z/OS IBM i Un objet d'informations d'authentification avec AUTHTYPE (OCSP) ne s'applique pas aux gestionnaires de files d'attente IBM i ou z/OS , mais il peut être spécifié sur ces plateformes pour être copié dans la table de définition de canal du client (CCDT) à des fins d'utilisation par le client.

Pour permettre à un IBM MQ MQI client d'accéder à un répondeur OCSP ou à des serveurs LDAP qui contiennent des CRL, les attributs d'un ou de plusieurs objets d'informations d'authentification peuvent être inclus dans une table de définition de canal du client. Vous pouvez inclure ces attributs de l'une des manières suivantes:

Multi

Sur les plateformes serveur AIX, Linux, IBM iet Windows

Vous pouvez définir une liste de noms contenant les noms d'un ou de plusieurs objets d'informations d'authentification. Vous pouvez ensuite définir l'attribut de gestionnaire de files d'attente, **SSLCRLNL**, sur le nom de cette liste de noms.

Si vous utilisez des listes de révocation de certificat, plusieurs serveurs LDAP peuvent être configurés pour fournir une disponibilité plus élevée. L'objectif est que chaque serveur LDAP dispose des mêmes CRL. Si un serveur LDAP n'est pas disponible lorsqu'il est requis, un IBM MQ MQI client peut tenter d'accéder à un autre serveur.

Les attributs des objets d'informations d'authentification identifiés par la liste de noms sont appelés collectivement ici *emplacement de révocation de certificat*. Lorsque vous définissez l'attribut de gestionnaire de files d'attente, **SSLCRLNL**, sur le nom de la liste de noms, l'emplacement de révocation de certificat est copié dans la table de définition de canal du client associée au gestionnaire de files d'attente. Si la table de définition de canal du client est accessible à partir d'un système client en tant que fichier partagé, ou si la table de définition de canal du client est ensuite copiée sur un système client, le IBM MQ MQI client sur ce système peut utiliser l'emplacement de révocation de certificat dans la table de définition de canal du client pour accéder à un répondeur OCSP ou à des serveurs LDAP contenant des listes de révocation de certificat.

Si l'emplacement de révocation de certificat du gestionnaire de files d'attente est modifié ultérieurement, la modification est reflétée dans la table de définition de canal du client associée au gestionnaire de files d'attente. Si l'attribut de gestionnaire de files d'attente, **SSLCRLNL**, est mis à blanc, l'emplacement de révocation de certificat est supprimé de la table de définition de canal du client. Ces modifications ne sont reflétées dans aucune copie de la table sur un système client.

Si vous souhaitez que l'emplacement de révocation de certificat aux extrémités client et serveur d'un canal MQI soit différent et que le gestionnaire de files d'attente du serveur est celui qui est utilisé pour créer l'emplacement de révocation de certificat, procédez comme suit:

1. Sur le gestionnaire de files d'attente du serveur, créez l'emplacement de révocation de certificat à utiliser sur le système client.
2. Copiez la table de définition de canal du client contenant l'emplacement de révocation de certificat sur le système client.
3. Sur le gestionnaire de files d'attente du serveur, remplacez l'emplacement de révocation de certificat par l'emplacement requis à l'extrémité serveur du canal MQI.
4. Sur la machine client, vous pouvez utiliser la commande **runmqsc** avec le paramètre **-n** .

Multi

Sur les plateformes client AIX, Linux, IBM iet Windows

Vous pouvez générer une table de définition de canal du client sur la machine client à l'aide de la commande runmqsc avec le paramètre **-n** et les objets **DEFINE AUTHINFO** du fichier CCDT. L'ordre dans lequel les objets sont définis est l'ordre dans lequel ils sont utilisés dans le fichier. Tout nom que vous pouvez utiliser dans un objet **DEFINE AUTHINFO** n'est pas conservé dans le fichier. Seuls les nombres à position fixe sont utilisés lorsque vous **DISPLAY** les objets **AUTHINFO** dans un fichier CCDT.

Remarque : Si vous spécifiez le paramètre **-n** , vous ne devez spécifier aucun autre paramètre.

Utilisation d' Active Directory sous Windows

Windows

Sur les systèmes Windows , vous pouvez utiliser la commande de contrôle **setmqcrl** pour publier les informations CRL en cours dans Active Directory.

La commande **setmqcrl** ne publie pas les informations OCSP.

Pour plus d'informations sur cette commande et sa syntaxe, voir setmqcrl.

Accès aux CRL et aux ARL avec IBM MQ classes for Java et IBM MQ classes for JMS

IBM MQ classes for Java et IBM MQ classes for JMS accèdent aux CRL différemment des autres plateformes.

Pour plus d'informations sur l'utilisation des CRL et des ARL avec IBM MQ classes for Java, voir Utilisation des listes de révocation de certificat

Pour plus d'informations sur l'utilisation des CRL et des ARL avec IBM MQ classes for JMS, voir Propriété d'objet SSLCERTSTORES

Manipulation des objets d'informations d'authentification

Vous pouvez manipuler des objets d'informations d'authentification à l'aide de commandes MQSC ou PCF ou du IBM MQ Explorer.

Les commandes MQSC suivantes agissent sur les objets d'informations d'authentification:

- DEFINE AUTHINFO
- ALTER AUTHINFO
- DELETE AUTHINFO
- INFORMATIONS D'AUTHENTIFICATION D'AFFICHAGE

Pour une description complète de ces commandes, voir Commandes MQSC.

Les commandes PCF (Programmable Command Format) suivantes agissent sur les objets d'informations d'authentification:

- Créer des informations d'authentification
- Copier des informations d'authentification
- Modifier des informations d'authentification
- Supprimer des informations d'authentification
- Consulter des informations d'authentification
- Consulter des noms d'informations d'authentification

Pour une description complète de ces commandes, voir Définitions des formats de commande programmables.

Sur les plateformes où il est disponible, vous pouvez également utiliser le IBM MQ Explorer.

Linux AIX Utilisation de la méthode PAM (Pluggable Authentication Method)

Vous pouvez utiliser PAM uniquement sur les plateformes AIX and Linux . Un système AIX ou Linux standard comporte des modules PAM qui implémentent le mécanisme d'authentification traditionnel ; toutefois, il peut y en avoir d'autres. En plus de la tâche de base de validation des mots de passe, les modules PAM peuvent également être appelés pour exécuter des règles supplémentaires.

Les fichiers de configuration définissent la méthode d'authentification à utiliser pour chaque application. Les exemples d'application incluent la connexion de terminal standard, ftp et telnet.

L'avantage de PAM est que l'application n'a pas besoin de connaître ou de se soucier de la façon dont l'ID utilisateur est authentifié. Tant que l'application peut fournir une forme correcte de données d'authentification à PAM, le mécanisme qui la sous-tend est transparent.

La forme des données d'authentification dépend du système utilisé. Par exemple, IBM MQ obtient un mot de passe via des paramètres, tels que la structure MQCSP utilisée dans l'appel API MQCONNX .

Important : Vous ne pouvez pas définir l'attribut **AUTHENMD** tant que vous n'avez pas installé IBM MQ 8.0.0 Fix Pack 3, puis redémarré le gestionnaire de files d'attente à l'aide d'un niveau **-e CMDLEVEL=** de *802* (dans la commande strmqm) pour définir le niveau de commande requis.

Configuration de votre système pour l'utilisation de PAM

Le nom de service utilisé par IBM MQlors de l'appel de PAM est *ibmmq*.

Notez qu'une installation IBM MQ tente de gérer une configuration PAM par défaut, qui autorise les connexions des utilisateurs du système d'exploitation, en fonction des valeurs par défaut connues pour les différents systèmes d'exploitation.

Toutefois, votre administrateur système doit vérifier que les règles définies dans les fichiers `/etc/pam.conf`ou `/etc/pam.d/ibmmq`sont toujours appropriées.

Autorisation de l'accès aux objets

Cette section contient des informations sur l'utilisation du gestionnaire de droits d'accès aux objets et des programmes d'exit de canal pour contrôler l'accès aux objets.

ALW Sur les systèmes AIX, Linux, and Windows . vous contrôlez l'accès aux objets à l'aide du gestionnaire des droits d'accès aux objets (OAM). Cette collection de rubriques contient des informations sur l'utilisation de l'interface de commande de la méthode d'accès aux objets (OAM).

Cette section contient également une liste de contrôle que vous pouvez utiliser pour déterminer les tâches à effectuer pour appliquer la sécurité à votre système sur toutes les plateformes, ainsi que les considérations à prendre en compte pour accorder aux utilisateurs le droit d'administrer IBM MQ et d'utiliser les objets IBM MQ .

Si les mécanismes de sécurité fournis ne répondent pas à vos besoins, vous pouvez développer vos propres programmes d'exit de canal.

Identification de l'utilisateur utilisé pour l'autorisation

Les droits d'accès aux ressources sont accordés aux groupes dont l'utilisateur est membre ou, dans certains modes, directement à l'utilisateur associé à la connexion. Lors du processus de connexion, et en particulier pour les connexions distantes (client), cette identité peut être modifiée par la configuration du gestionnaire de files d'attente. Cette page répertorie les différentes fonctions d' IBM MQ et leurs options de configuration qui peuvent avoir un impact sur l'identité d'une application de connexion, ainsi que l'ordre de priorité dans lequel ces fonctions sont appliquées.

Fonctions pouvant modifier l'utilisateur adopté

Les différentes fonctions qui peuvent définir l'utilisateur qui doit être autorisé sont les suivantes:

Utilisateur vérifié par l'application
Lorsqu'une connexion distante est démarrée par IBM MQ, l'utilisateur du système d'exploitation sous lequel le processus s'exécute est envoyé au gestionnaire de files d'attente de réception. Cet utilisateur est envoyé pour s'assurer que s'il n'existe aucune configuration supplémentaire qui modifie l'utilisateur, il existe un utilisateur qui peut être utilisé pour la vérification des autorisations.

Il n'est pas recommandé d'utiliser cet utilisateur comme base d'autorisation car il permet aux connexions d'affirmer leur identité sans validation côté serveur. Cela peut même inclure l'administrateur ('mqm').

Paramètre MCAUSER du canal
Les applications qui se connectent via des liaisons réseau le font à l'aide d'une définition de canal IBM MQ . Les définitions de canal prennent en charge l'attribut **MCAUSER** , qui peut être utilisé pour spécifier un utilisateur différent à utiliser pour l'autorisation au lieu de l'utilisateur vérifié par les applications de connexion.

Authentification de connexion ADOPTCTX
Les applications peuvent spécifier un utilisateur et un mot de passe à envoyer à un gestionnaire de files d'attente à des fins d'authentification. Ces données d'identification sont authentifiées à l'aide de la configuration spécifiée pour la fonction d'authentification de connexion. L'option **ADOPTCTX** pour l'authentification de connexion contrôle si un utilisateur doit être utilisé pour l'autorisation après sa validation. Si la valeur est YES, l'utilisateur fourni pour l'authentification est adopté pour les vérifications d'autorisation.

V 9.4.0 Depuis IBM MQ 9.3.4, un jeton peut être fourni pour l'authentification. Si **ADOPTCTX** est défini sur YES, un utilisateur est adopté à partir des revendications contenues dans le jeton.

Enregistrement d'authentification de canal MCAUSER
Lors du traitement de la connexion, le gestionnaire de files d'attente tente de trouver un enregistrement d'authentification de canal correspondant à la connexion. Si un enregistrement d'authentification de canal est mis en correspondance et que sa valeur d'attribut **USERSRC** est définie sur MAP, IBM MQ remplace l'utilisateur utilisé pour les autorisations par la valeur de l'attribut **MCAUSER** .

Exits de sécurité
Les exits de sécurité sont des fonctions personnalisées qui peuvent être écrites et appelées lors du traitement de la sécurité IBM MQ . Lorsque la fonction est appelée, elle est fournie avec une copie de la structure MQCD qui inclut plusieurs zones relatives à l'utilisateur des connexions qui seront utilisées pour les vérifications d'autorisation. Les exits de sécurité peuvent modifier ces zones pour modifier l'utilisateur qui sera autorisé.

ordre de précédence

Le tableau suivant présente l'ordre de priorité de chaque fonction de sécurité décrite dans «Fonctions pouvant modifier l'utilisateur adopté», à la page 363 lorsque IBM MQ sélectionne un utilisateur à autoriser. L'ordre est du plus bas au plus élevé, c'est-à-dire qu'une fonction de sécurité définissant un utilisateur à la première ligne est remplacée par l'une des autres lignes.

Tableau 71. Ordre de priorité des fonctions de sécurité

Commande	**Fonction**
1 (moins important)	ID vérifié par l'application
2	Définition de canal **MCAUSER** , attribut
3	Authentification de la connexion avec **ADOPTCTX(YES)**
4	Enregistrements d'authentification de canal avec **USERSRC(MAP)**
5 (plus élevé)	Exit de sécurité

Implications de l'adoption précoce

Les enregistrements d'authentification de connexion et d'authentification de canal fournissent une option de configuration qui contrôle le moment où l'adoption de l'utilisateur pour l'authentification de connexion est effectuée. Ce paramètre est appelé "adoption précoce". Si l'adoption anticipée est activée,

l'adoption de l'identité d'authentification de connexion a lieu avant le traitement des enregistrements d'authentification de canal (ce qui signifie que les enregistrements d'authentification de canal remplacent toute adoption **CONNAUTH**).

Si cette option est désactivée, l'ordre est inversé, c'est-à-dire que les enregistrements d'authentification de canal sont traités avant l'adoption de **CONNAUTH** . Dans cette situation, l'adoption de l'authentification de connexion a une priorité effective plus élevée que les enregistrements d'authentification de canal.

Le paramètre par défaut pour l'adoption anticipée est `activé`.

ALW Contrôle de l'accès aux objets à l'aide de la méthode d'accès aux objets (OAM) sous AIX, Linux, and Windows

Le gestionnaire des droits d'accès aux objets (OAM) fournit une interface de commande pour l'octroi et la révocation des droits d'accès aux objets IBM MQ .

Vous devez disposer des droits appropriés pour utiliser ces commandes, comme décrit dans «Droit d'administration de IBM MQ sur AIX, Linux, and Windows», à la page 414. Les ID utilisateur autorisés à administrer IBM MQ disposent des droits *superutilisateur* sur le gestionnaire de files d'attente, ce qui signifie que vous n'avez pas besoin de leur accorder d'autres droits pour émettre des demandes ou des commandes MQI.

Linux AIX Droits utilisateur OAM sur AIX and Linux

Sur les systèmes UNIX and Linux, le gestionnaire des droits d'accès aux objets (OAM) peut utiliser l'autorisation utilisateur et l'autorisation de groupe.

Avant la IBM MQ 8.0, les listes de contrôle d'accès sous UNIX and Linux reposent uniquement sur des groupes. DepuisIBM MQ 8.0 , les ACL sont basées à la fois sur les ID utilisateur et sur les groupes, et vous pouvez utiliser soit le modèle basé sur l'utilisateur, soit le modèle basé sur le groupe pour l'autorisation en définissant l'option**SecurityPolicy** attribuer à la valeur appropriée comme décrit dans Strophe de service duqm.ini déposer .

Changements de comportement pour IBM MQ 8.0 et versions ultérieures

Depuis IBM MQ 8.0, lors de l'exécution avec la règle basée sur l'utilisateur, certaines commandes renvoient des informations différentes de celles des versions antérieures du produit:

- Les commandes **dmpmqaut** et **dmpmqcfg** affichent les enregistrements utilisateur, comme le opérations PCF équivalentes.
- Le plug-in OAM pour IBM MQ Explorer indique les enregistrements basés sur l'utilisateur et autorise les modifications basées sur l'utilisateur.
- La fonction **Inquire** d'OAM renvoie des résultats indiquant qu'elle est peut fonctionner en étant basée sur l'utilisateur.

L'utilisation de l'attribut **-p** dans la commande **setmqaut** n'accorde pas l'accès à tous les utilisateurs du même groupe principal, lorsque les autorisations basées sur les utilisateurs sont activées dans le fichier `qm.ini` , comme décrit dans la section Service du fichier qm.ini.

Si vous commencez à utiliser l'autorisation basée sur l'utilisateur et que vous disposez de nombreux utilisateurs, il y aura probablement davantage d'enregistrements stockés sur la file d'attente AUTH qu'avec le modèle basé sur le groupe et le processus d'autorisation risque de durer un peu plus longtemps qu'auparavant car les enregistrements à vérifier sont plus nombreux. Cette augmentation ne devrait pas être significative. Si nécessaire, vous pouvez utiliser une combinaison de droits basés sur l'utilisateur et le groupe.

Migration

Si vous modifiez le modèle du groupe à l'utilisateur pour un gestionnaire de files d'attente existant, l'effet n'est pas immédiat. Les autorisations ayant déjà été accordées continuent de s'appliquer. N'importe

quel utilisateur se connectant au gestionnaire de files d'attente reçoit les mêmes privilèges qu'avant : la combinaison de tous les groupes auxquels leur ID appartient. Lorsque de nouvelles commandes **setmqaut** sont émises pour les ID utilisateur, elles sont immédiatement effectives.

Si vous créez un gestionnaire de files d'attente avec la règle utilisateur, ce gestionnaire de files d'attente dispose des droits uniquement pour l'utilisateur qui l'a créé (qui est normalement, mais pas nécessairement, l'ID utilisateur `mqm`). Il existe également des droits qui sont automatiquement accordés au groupe `mqm` . Toutefois, si vous ne disposez pas de `mqm` comme groupe principal, le groupe `mqm` n'est pas inclus dans l'ensemble d'autorisations initial.

Si vous passez d'une règle d'utilisateur à une règle de groupe, les droits basés sur l'utilisateur ne sont pas automatiquement supprimés. Cependant, ils ne sont plus utilisés lors de la vérification des droits. Avant de modifier les règles, enregistrez la configuration en cours, modifiez les règles, redémarrez le gestionnaire de files d'attente, puis réexécutez le script. Etant donné qu'il s'agit maintenant d'un gestionnaire de files d'attente basé sur le groupe, les règles de l'ID utilisateur sont stockées en fonction du groupe principal.

Concepts associés

Gestionnaire des droits d'accès aux objets (OAM)

«Principaux et groupes sous AIX, Linux, and Windows», à la page 419
Les principaux peuvent appartenir à des groupes. En accordant l'accès aux ressources à des groupes plutôt qu'à des individus, vous pouvez réduire la quantité d'administration requise. Les listes de contrôle d'accès (ACL) sont basées à la fois sur les groupes et les ID utilisateur.

Référence associée

Section de service du fichier qm.ini

Commande **crtmqm** (créer le gestionnaire de files d'attente)

ALW Octroi de l'accès à un objet IBM MQ sur AIX, Linux, and Windows

Utilisez la commande de contrôle **setmqaut** , la commande **SET AUTHREC** MQSC ou la commande PCF **MQCMD_SET_AUTH_REC** pour accorder aux utilisateurs et aux groupes d'utilisateurs l'accès aux objets IBM MQ . Notez que sur IBM MQ Appliance , vous ne pouvez utiliser que la commande **SET AUTHREC** .

Pour une définition complète de la commande de contrôle **setmqaut** et de sa syntaxe, voir setmqaut.

Pour une définition complète de la commande **SET AUTHREC** MQSC et de sa syntaxe, voir SET AUTHREC.

Pour une définition complète de la commande PCF **MQCMD_SET_AUTH_REC** et de sa syntaxe, voir Définition de l'enregistrement des droits d'accès.

Le gestionnaire de files d'attente doit être en cours d'exécution pour pouvoir utiliser cette commande. Lorsque vous avez modifié l'accès à un principal, les modifications sont reflétées immédiatement par la méthode d'accès aux objets (OAM).

Pour accorder aux utilisateurs l'accès à un objet, vous devez spécifier:

- Nom du gestionnaire de files d'attente qui possède les objets que vous utilisez ; si vous ne spécifiez pas le nom d'un gestionnaire de files d'attente, le gestionnaire de files d'attente par défaut est utilisé.
- Nom et type de l'objet (pour identifier l'objet de manière unique). Vous spécifiez le nom en tant que *profil* ; Il s'agit soit du nom explicite de l'objet, soit d'un nom générique, incluant des caractères génériques. Pour une description détaillée des profils génériques et de l'utilisation des caractères génériques qu'ils contiennent, voir «Utilisation des profils génériques OAM sous AIX, Linux, and Windows», à la page 368.
- Un ou plusieurs principaux et noms de groupe auxquels les droits s'appliquent.

  Si un ID utilisateur contient des espaces, placez-le entre guillemets lorsque vous utilisez cette commande. Sur les systèmes Windows , vous pouvez qualifier un ID utilisateur avec un nom de domaine. Si l'ID utilisateur réel contient un symbole arobase (@), remplacez-le par @ @ pour indiquer qu'il fait partie de l'ID utilisateur et non du délimiteur entre l'ID utilisateur et le nom de domaine.

- Liste des autorisations. Chaque élément de la liste indique un type d'accès qui doit être accordé à cet objet (ou révoqué). Chaque autorisation de la liste est indiquée en tant que mot clé, précédé d'un signe plus (+) ou d'un signe moins (-). Utilisez un signe plus pour ajouter l'autorisation spécifiée et un signe moins pour supprimer l'autorisation. Il ne doit pas y avoir d'espaces entre le signe + ou-et le mot clé.

  Vous pouvez spécifier n'importe quel nombre d'autorisations dans une seule commande. Par exemple, la liste des autorisations permettant à un utilisateur ou à un groupe de placer des messages dans une file d'attente et de les parcourir, mais de révoquer l'accès pour obtenir des messages est la suivante:

```
+browse -get +put
```

Exemples d'utilisation de la commande `setmqaut`

Les exemples suivants montrent comment utiliser la commande `setmqaut` pour accorder et révoquer des droits d'utilisation d'un objet:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

Dans cet exemple :

- `saturn.queue.manager` est le nom du gestionnaire de files d'attente
- `queue` est le type d'objet
- `RED.LOCAL.QUEUE` est le nom de l'objet
- `groupa` est l'identificateur du groupe avec les autorisations à modifier
- `+browse -get +put` est la liste d'autorisation pour la file d'attente spécifiée
  - `+browse` ajoute l'autorisation de parcourir les messages dans la file d'attente (pour émettre **MQGET** avec l'option de navigation)
  - `-get` supprime l'autorisation d'obtenir (**MQGET**) des messages de la file d'attente
  - `+put` ajoute l'autorisation d'insertion de messages (**MQPUT**) dans la file d'attente

La commande suivante révoque le droit d'insertion sur la file d'attente MyQueue du principal fvuser et des groupes groupa et groupb. Sur les systèmes AIX and Linux , cette commande révoque également le droit d'insertion pour tous les principaux du même groupe principal que fvuser.

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

Utilisation de la commande `setmqaut` avec un service d'autorisation différent

Si vous utilisez votre propre service d'autorisation à la place de la méthode d'accès aux objets (OAM), vous pouvez spécifier le nom de ce service dans la commande **setmqaut** pour diriger la commande vers ce service. Vous devez spécifier ce paramètre si plusieurs composants installables sont en cours d'exécution en même temps ; si ce n'est pas le cas, la mise à jour est effectuée sur le premier composant installable pour le service d'autorisation. Par défaut, il s'agit de la méthode d'accès aux objets (OAM) fournie.

Remarques sur l'utilisation de la commande SET AUTHREC

La liste des autorisations à ajouter et la liste des autorisations à supprimer ne doivent pas se chevaucher. Par exemple, vous ne pouvez pas ajouter et supprimer des droits d'affichage avec la même commande. Cette règle s'applique même si les droits sont spécifiés à l'aide d'options différentes. Par exemple, la commande suivante échoue car le droit DSP et le droit ALLADM se chevauchent :

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

Il existe une exception à ce comportement de chevauchement dans le cas du droit ALL. La commande suivante ajoute d'abord tous les droits ALL, puis supprime le droit SETID :

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

La commande suivante supprime d'abord tous les droits ALL, puis ajoute le droit DSP :

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

Quel que soit l'ordre dans lequel les droits sont indiqués dans la commande, les droits ALL sont traités en premier.

ALW Utilisation des profils génériques OAM sous AIX, Linux, and Windows

Utilisez les profils génériques OAM pour définir, en une seule opération, les privilèges d'un utilisateur pour de nombreux objets, plutôt que d'avoir à émettre des commandes **setmqaut** ou **SET AUTHREC** distinctes pour chaque objet individuel lors de sa création. Notez que sur IBM MQ Appliance , vous ne pouvez utiliser que la commande **SET AUTHREC** .

L'utilisation de profils génériques dans les commandes setmqaut ou SET AUTHREC vous permet de définir des droits génériques pour tous les objets qui correspondent à ce profil.

Cette collection de rubriques décrit plus en détail l'utilisation des profils génériques.

Utilisation de caractères génériques dans les profils OAM

Ce qui rend un profil générique, c'est l'utilisation de caractères spéciaux (caractères génériques) dans le nom du profil. Par exemple, le caractère générique point d'interrogation (?) correspond à n'importe quel caractère unique dans un nom. Par conséquent, si vous spécifiez `ABC.?EF`, l'autorisation que vous accordez à ce profil s'applique à tous les objets portant les noms `ABC.DEF`, `ABC.CEF`, `ABC.BEF`, etc.

Les caractères génériques disponibles sont les suivants:

?

Utilisez le point d'interrogation (?) au lieu de n'importe quel caractère. Par exemple, `AB.?D` s'applique aux objets `AB.CD`, `AB.EDet AB.FD`.

*****

Utilisez l'astérisque (*) comme suit:

- Un *qualificateur* dans un nom de profil pour correspondre à un qualificateur dans un nom d'objet. Le qualificatif est une partie de nom d'objet délimitée par un point. Par exemple, dans `ABC.DEF.GHI`, les qualificatifs sont `ABC`, `DEF` et `GHI`.

  Par exemple, `ABC.*.JKL` s'applique aux objets `ABC.DEF.JKLet ABC.GHI.JKL`. (Notez qu'il ne s'applique **pas** à `ABC.JKL` ; * utilisé dans ce contexte indique toujours un qualificateur.)
- Caractère dans un qualificatif d'un nom de profil qui doit correspondre à zéro ou plusieurs caractères dans le qualificatif d'un nom d'objet.

  Par exemple, `ABC.DE*.JKL` s'applique aux objets `ABC.DE.JKL`, `ABC.DEF.JKLet ABC.DEGH.JKL`.

******

Utilisez le double astérisque (**) **une fois** dans un nom de profil comme suit:

- Nom de profil complet correspondant à tous les noms d'objet. Par exemple, si vous utilisez `-t prcs` pour identifier les processus, puis utilisez ** comme nom de profil, vous modifiez les autorisations pour tous les processus.
- Comme qualificatif de début, de milieu ou de fin dans un nom de profil pour correspondre à zéro ou plusieurs qualificatifs dans un nom d'objet. Par exemple, `**.ABC` identifie tous les objets avec le qualificateur final ABC.

Vous ne pouvez utiliser que le double astérisque ** comme qualificateur complet:

```
**.DEF
  ABC.**
  A*.**
```

mais pas en tant que

```
A**
```

sinon, vous recevez le message AMQ7226E: `Le nom de profil n'est pas valide.`

Remarque : Lorsque vous utilisez des caractères génériques sur des systèmes AIX and Linux , vous **devez** placer le nom de profil entre apostrophes.

Priorités de profil

Un point important à comprendre lors de l'utilisation de profils génériques est la priorité donnée aux profils lors de la détermination des droits à appliquer à un objet en cours de création. Par exemple, supposons que vous ayez émis les commandes suivantes:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

Le premier accorde le droit d'insertion à toutes les files d'attente pour le principal fred dont les noms correspondent au profil AB. * ; La seconde permet d'obtenir des droits sur les mêmes types de file d'attente qui correspondent au profil AB.C*.

Supposons que vous créiez maintenant une file d'attente appelée AB.CD. Selon les règles de correspondance des caractères génériques, setmqaut peut s'appliquer à cette file d'attente. Alors, a-t-elle mis ou obtenu l'autorité?

Pour trouver la réponse, vous appliquez la règle selon laquelle, chaque fois que plusieurs profils peuvent s'appliquer à un objet, **seuls les plus spécifiques s'appliquent**. La façon dont vous appliquez cette règle consiste à comparer les noms de profil de gauche à droite. Lorsqu'ils diffèrent, un caractère non générique est plus spécifique qu'un caractère générique. Ainsi, dans cet exemple, la file d'attente AB.CD dispose du droit **get** (AB.C* est plus spécifique que AB. *).

Lorsque vous comparez des caractères génériques, l'ordre de la *spécificité* est le suivant:

1. ?
2. *
3. **

Vidage des paramètres de profil

Pour une définition complète de la commande de contrôle **dmpmqaut** et de sa syntaxe, voir dmpmqaut.

Pour une définition complète de la commande **DISPLAY AUTHREC** MQSC et de sa syntaxe, voir DISPLAY AUTHREC.

Pour une définition complète de la commande PCF **MQCMD_INQUIRE_AUTH_RECS** et de sa syntaxe, voir Inquire Authority Records.

Les exemples suivants illustrent l'utilisation de la commande de contrôle **dmpmqaut** pour vider des enregistrements de droits d'accès pour des profils génériques:

1. Cet exemple vide tous les enregistrements de droits d'accès dont le profil correspond à la file d'attente a.b.c pour le principal user1.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   Le vidage résultant se présente comme suit:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

Remarque : Bien que les utilisateurs sous AIX and Linux puissent utiliser l'option `-p` pour la commande **dmpmqaut** , ils doivent utiliser `-g groupname` à la place lors de la définition des autorisations.

2. Cet exemple vide tous les enregistrements de droits d'accès dont le profil correspond à la file d'attente a.b.c.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

Le vidage résultant se présente comme suit:

```
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. Cet exemple vide tous les enregistrements de droits d'accès pour le profil a.b. *, de type file d'attente.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

Le vidage résultant se présente comme suit:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Cet exemple vide tous les enregistrements de droits d'accès pour le gestionnaire de files d'attente qmX.

```
dmpmqaut -m qmX
```

Le vidage résultant se présente comme suit:

```
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. Cet exemple vide tous les noms de profil et tous les types d'objet pour le gestionnaire de files d'attente qmX.

```
dmpmqaut -m qmX -l
```

Le vidage résultant se présente comme suit:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Remarque : Pour IBM MQ for Windows uniquement, tous les principaux affichés incluent des informations de domaine, par exemple:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW *Utilisation de caractères génériques dans les profils OAM sous AIX, Linux, and Windows*

Utilisez des caractères génériques dans un nom de profil de gestionnaire des droits d'accès aux objets (OAM) pour rendre ce profil applicable à plusieurs objets.

Ce qui rend un profil générique, c'est l'utilisation de caractères spéciaux (caractères génériques) dans le nom du profil. Par exemple, le caractère générique point d'interrogation (?) correspond à n'importe quel caractère unique dans un nom. Par conséquent, si vous spécifiez ABC.?EF, l'autorisation que vous accordez à ce profil s'applique à tous les objets portant les noms ABC.DEF, ABC.CEF, ABC.BEF, etc.

Les caractères génériques disponibles sont les suivants:

?

Utilisez le point d'interrogation (?) au lieu de n'importe quel caractère. Par exemple, AB.?D s'applique aux objets AB.CD, AB.EDet AB.FD.

*****

Utilisez l'astérisque (*) comme suit:

- Un *qualificateur* dans un nom de profil pour correspondre à un qualificateur dans un nom d'objet. Le qualificatif est une partie de nom d'objet délimitée par un point. Par exemple, dans ABC.DEF.GHI, les qualificatifs sont ABC, DEF et GHI.

  Par exemple, ABC.*.JKL s'applique aux objets ABC.DEF.JKLet ABC.GHI.JKL. (Notez qu'il ne s'applique **pas** à ABC.JKL ; * utilisé dans ce contexte indique toujours un qualificateur.)
- Caractère dans un qualificatif d'un nom de profil qui doit correspondre à zéro ou plusieurs caractères dans le qualificatif d'un nom d'objet.

  Par exemple, ABC.DE*.JKL s'applique aux objets ABC.DE.JKL, ABC.DEF.JKLet ABC.DEGH.JKL.

******

Utilisez le double astérisque (**) **une fois** dans un nom de profil comme suit:

- Nom de profil complet correspondant à tous les noms d'objet. Par exemple, si vous utilisez `-t prcs` pour identifier les processus, puis utilisez ** comme nom de profil, vous modifiez les autorisations pour tous les processus.
- Comme qualificatif de début, de milieu ou de fin dans un nom de profil pour correspondre à zéro ou plusieurs qualificatifs dans un nom d'objet. Par exemple, `**.ABC` identifie tous les objets avec le qualificateur final ABC.

Remarque : Lorsque vous utilisez des caractères génériques sur des systèmes AIX and Linux , vous **devez** placer le nom de profil entre apostrophes.

ALW *Priorités de profil sous AIX, Linux, and Windows*

Plusieurs profils génériques peuvent s'appliquer à un seul objet. Lorsque c'est le cas, la règle la plus spécifique s'applique.

Un point important à comprendre lors de l'utilisation de profils génériques est la priorité donnée aux profils lors de la détermination des droits à appliquer à un objet en cours de création. Par exemple, supposons que vous ayez émis les commandes suivantes:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

Le premier accorde le droit d'insertion à toutes les files d'attente pour le principal fred dont les noms correspondent au profil AB. * ; La seconde permet d'obtenir des droits sur les mêmes types de file d'attente qui correspondent au profil AB.C*.

Supposons que vous créiez maintenant une file d'attente appelée AB.CD. Selon les règles de correspondance des caractères génériques, setmqaut peut s'appliquer à cette file d'attente. Alors, a-t-elle mis ou obtenu l'autorité?

Pour trouver la réponse, vous appliquez la règle selon laquelle, chaque fois que plusieurs profils peuvent s'appliquer à un objet, **seuls les plus spécifiques s'appliquent**. La façon dont vous appliquez cette règle consiste à comparer les noms de profil de gauche à droite. Lorsqu'ils diffèrent, un caractère non générique est plus spécifique qu'un caractère générique. Ainsi, dans cet exemple, la file d'attente AB.CD dispose du droit **get** (AB.C* est plus spécifique que AB. *).

Lorsque vous comparez des caractères génériques, l'ordre de la *spécificité* est le suivant:

1. ?
2. *
3. **

Pour obtenir des informations équivalentes lors de l'utilisation de cette commande MQSC, voir SET AUTHREC .

ALW *Vidage des paramètres de profil sous AIX, Linux, and Windows*

Utilisez la commande de contrôle **dmpmqaut** , la commande **DISPLAY AUTHREC** MQSC ou la commande **MQCMD_INQUIRE_AUTH_RECS** PCF pour vider les autorisations en cours associées à un profil spécifié. Notez que sur IBM MQ Appliance , vous ne pouvez utiliser que la commande **DISPLAY AUTHREC** .

Pour une définition complète de la commande de contrôle **dmpmqaut** et de sa syntaxe, voir dmpmqaut.

Pour une définition complète de la commande **DISPLAY AUTHREC** MQSC et de sa syntaxe, voir DISPLAY AUTHREC.

Pour une définition complète de la commande PCF **MQCMD_INQUIRE_AUTH_RECS** et de sa syntaxe, voir Inquire Authority Records.

Les exemples suivants illustrent l'utilisation de la commande de contrôle **dmpmqaut** pour vider des enregistrements de droits d'accès pour des profils génériques:

1. Cet exemple vide tous les enregistrements de droits d'accès dont le profil correspond à la file d'attente a.b.c pour le principal user1.

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

Le vidage résultant ressemble à l'exemple suivant:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

Remarque : Les utilisateurs AIX and Linux ne peuvent pas utiliser l'option `-p` ; ils doivent utiliser `-g groupname` à la place.

2. Cet exemple vide tous les enregistrements de droits d'accès dont le profil correspond à la file d'attente a.b.c.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

Le vidage résultant ressemble à l'exemple suivant:

```
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. Cet exemple vide tous les enregistrements de droits d'accès pour le profil a.b. *, de type file d'attente.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

Le vidage résultant ressemble à l'exemple suivant:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Cet exemple vide tous les enregistrements de droits d'accès pour le gestionnaire de files d'attente qmX.

```
dmpmqaut -m qmX
```

Le vidage résultant ressemble à l'exemple suivant:

```
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. Cet exemple vide tous les noms de profil et tous les types d'objet pour le gestionnaire de files d'attente qmX.

```
dmpmqaut -m qmX -l
```

   Le vidage résultant ressemble à l'exemple suivant:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Remarque : Pour IBM MQ for Windows uniquement, tous les principaux affichés incluent des informations de domaine, par exemple:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW Affichage des paramètres d'accès sur AIX, Linux, and Windows

Utilisez la commande de contrôle **dspmqaut** , la commande **DISPLAY AUTHREC** MQSC ou la commande **MQCMD_INQUIRE_ENTITY_AUTH** PCF pour afficher les autorisations dont dispose un principal ou un groupe spécifique pour un objet particulier. Notez que sur IBM MQ Appliance , vous ne pouvez utiliser que la commande **DISPLAY AUTHREC** .

Le gestionnaire de files d'attente doit être en cours d'exécution pour pouvoir utiliser cette commande. Lorsque vous modifiez l'accès à un principal, les modifications sont répercutées immédiatement par la méthode d'accès aux objets (OAM). L'autorisation ne peut être affichée que pour un seul groupe ou principal à la fois.

Pour une définition complète de la commande de contrôle **dmpmqaut** et de sa syntaxe, voir dmpmqaut.

Pour une définition complète de la commande **DISPLAY AUTHREC** MQSC et de sa syntaxe, voir DISPLAY AUTHREC.

Pour une définition complète de la commande PCF **MQCMD_INQUIRE_AUTH_RECS** et de sa syntaxe, voir Inquire Authority Records.

L'exemple suivant illustre l'utilisation de la commande de contrôle **dspmqaut** pour afficher les autorisations dont dispose le groupe `GpAdmin` pour une définition de processus nommée `Annuities` qui se trouve sur le gestionnaire de files d'attente `QueueMan1`.

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW Modification et révocation de l'accès à un objet IBM MQ sous AIX, Linux, and Windows

Pour modifier le niveau d'accès d'un utilisateur ou d'un groupe à un objet, utilisez la commande de contrôle **setmqaut** , la commande **DELETE AUTHREC** MQSC ou la commande PCF **MQCMD_DELETE_AUTH_REC** . MQ Appliance Notez que sur IBM MQ Appliance , vous ne pouvez utiliser que la commande **DELETE AUTHREC** .

Le processus de suppression de l'utilisateur d'un groupe est décrit dans:

- Windows «Création et gestion de groupes sur Windows», à la page 156
- AIX «Création et gestion de groupes sur AIX», à la page 154
- Linux «Création et gestion de groupes sur Linux», à la page 155

.

L'ID utilisateur qui crée un objet IBM MQ dispose de droits de contrôle complets sur cet objet. Si vous supprimez cet ID utilisateur du groupe mqm local (ou du groupe Administrateurs sur les systèmes Windows), ces droits ne sont pas révoqués. Utilisez la commande de contrôle **setmqaut** ou la commande PCF **MQCMD_DELETE_AUTH_REC** pour révoquer l'accès à un objet pour l'ID utilisateur qui l'a créé, après l'avoir supprimé du groupe mqm ou Administrateurs.

Pour une définition complète de la commande de contrôle setmqaut et de sa syntaxe, voir setmqaut.

Pour une définition complète de la commande **DELETE AUTHREC** MQSC et de sa syntaxe, voir DELETE AUTHREC.

Pour une définition complète de la commande PCF **MQCMD_DELETE_AUTH_REC** et de sa syntaxe, voir Delete Authority Record.

Windows Sous Windows, depuis IBM MQ 8.0, vous pouvez supprimer à tout moment les entrées OAM correspondant à un compte utilisateur Windows particulier à l'aide du paramètre **-u** *SID* de **setmqaut**.

Avant IBM MQ 8.0, vous deviez supprimer les entrées OAM correspondant à un compte utilisateur Windows particulier avant de supprimer le profil utilisateur. Il était impossible de supprimer les entrées OAM après la suppression du compte utilisateur.

ALW Prévention des contrôles d'accès de sécurité sur les systèmes AIX, Linux, and Windows

Remarque: Cette rubrique décrit les fonctionnalités qui ne doivent pas être activées. Pour désactiver la vérification de la sécurité, vous pouvez désactiver le gestionnaire des droits d'accès aux objets (OAM). Cela peut convenir à un environnement de test. Lorsque cette option est désactivée, le gestionnaire de files d'attente ne peut plus effectuer de vérifications d'autorisation ou d'authentification de connexion. TLS, les enregistrements d'authentification de canal et les exits de sécurité peuvent toujours être utilisés. Après avoir désactivé ou supprimé la méthode d'accès aux objets (OAM), vous ne pouvez pas ajouter une méthode d'accès aux objets (OAM) à un gestionnaire de files d'attente existant.

Si vous décidez de ne pas effectuer de contrôles de sécurité (par exemple, dans un environnement de test), vous pouvez désactiver la méthode d'accès aux objets (OAM) de l'une des deux manières suivantes:

- Avant de créer un gestionnaire de files d'attente, définissez la variable d'environnement du système d'exploitation **MQSNOAUT**.

  Pour plus d'informations sur les implications de la définition de la variable d'environnement **MQSNOAUT** et sur la manière de définir **MQSNOAUT** sur AIX, Linux, and Windows, voir Description des variables d'environnement.
- Editez le fichier de configuration du gestionnaire de files d'attente pour supprimer le service.

Avertissement : Lorsqu'une méthode d'accès aux objets (OAM) est supprimée, elle ne peut pas être remise sur un gestionnaire de files d'attente existant. En effet, la méthode d'accès aux objets (OAM) doit être en place au moment de la création de l'objet. Pour utiliser à nouveau la méthode d'accès aux objets (OAM) IBM MQ une fois qu'elle a été supprimée, régénérez le gestionnaire de files d'attente.

Si vous utilisez la commande **setmqaut**ou **dspmqaut** alors que la méthode d'accès aux objets (OAM) est désactivée, notez les points suivants:

- La méthode d'accès aux objets (OAM) ne valide pas le principal ou le groupe spécifié, ce qui signifie que la commande peut accepter des valeurs non valides.
- La méthode d'accès aux objets (OAM) n'effectue pas de contrôles de sécurité et indique que tous les principaux et groupes sont autorisés à effectuer toutes les opérations d'objet applicables.
- Les données d'identification transmises à la méthode d'accès aux objets (OAM) pour les vérifications d'authentification ne sont pas validées.

Concepts associés

Services et composants installables pour AIX, Linux, and Windows

Tâches associées

Configuration des services installables

Référence associée

Informations de référence sur les services installables

Octroi de l'accès requis aux ressources

Cette rubrique permet de déterminer les tâches à effectuer pour appliquer la sécurité à votre système IBM MQ .

Pourquoi et quand exécuter cette tâche

Au cours de cette tâche, vous décidez des actions nécessaires pour appliquer le niveau de sécurité approprié aux éléments de votre installation IBM MQ . Chaque tâche individuelle à qui vous vous référez fournit des instructions étape par étape pour toutes les plateformes.

Procédure

1. Avez-vous besoin de limiter l'accès à votre gestionnaire de files d'attente à certains utilisateurs?
   a) Non: ne prenez aucune autre mesure.
   b) Oui: Passez à la question suivante.
2. Ces utilisateurs ont-ils besoin d'un accès administratif partiel sur un sous-ensemble de ressources de gestionnaire de files d'attente?
   a) Non: Passez à la question suivante.
   b) Oui: voir «Octroi d'un accès administrateur partiel sur un sous-ensemble de ressources de gestionnaire de files d'attente», à la page 377.
3. Ces utilisateurs ont-ils besoin d'un accès administrateur complet sur un sous-ensemble de ressources de gestionnaire de files d'attente?
   a) Non: Passez à la question suivante.
   b) Oui: voir «Octroi d'un accès administrateur complet sur un sous-ensemble de ressources de gestionnaire de files d'attente», à la page 386.
4. Ces utilisateurs ont-ils besoin d'un accès en lecture seule à toutes les ressources du gestionnaire de files d'attente?
   a) Non: Passez à la question suivante.
   b) Oui: voir «Octroi d'un accès en lecture seule à toutes les ressources d'un gestionnaire de files d'attente», à la page 392.

5. Ces utilisateurs ont-ils besoin d'un accès administrateur complet sur toutes les ressources du gestionnaire de files d'attente?
   a) Non: Passez à la question suivante.
   b) Oui: voir «Octroi d'un accès administrateur complet à toutes les ressources d'un gestionnaire de files d'attente», à la page 393.
6. Avez-vous besoin d'applications utilisateur pour vous connecter à votre gestionnaire de files d'attente?
   a) Non: Désactiver la connectivité, comme décrit dans «Suppression de la connectivité au gestionnaire de files d'attente», à la page 394
   b) Oui: voir «Autorisation des applications utilisateur à se connecter à votre gestionnaire de files d'attente», à la page 395.

Multi z/OS Octroi d'un accès administrateur partiel sur un sous-ensemble de ressources de gestionnaire de files d'attente

Vous devez accorder à certains utilisateurs un accès administrateur partiel à certaines ressources de gestionnaire de files d'attente, mais pas à toutes. Utilisez ce tableau pour déterminer les actions que vous devez effectuer.

Tableau 72. Octroi d'un accès administrateur partiel à un sous-ensemble de ressources de gestionnaire de files d'attente

Les utilisateurs doivent administrer les objets de ce type	**Effectuer cette action**
Files d'attente	Accordez un accès administrateur partiel aux files d'attente requises, comme décrit dans «Octroi d'un accès administrateur limité à certaines files d'attente», à la page 378
Rubriques	Accordez un accès administrateur partiel aux rubriques requises, comme décrit dans «Octroi d'un accès administrateur limité à certaines rubriques», à la page 379
Canaux	Accordez un accès administrateur partiel aux canaux requis, comme décrit dans «Octroi d'un accès administratif limité à certains canaux», à la page 380
Gestionnaire de files d'attente	Accordez un accès administrateur partiel au gestionnaire de files d'attente, comme décrit dans «Octroi d'un accès administrateur limité à un gestionnaire de files d'attente», à la page 381
Processus	Accordez un accès administratif partiel aux processus requis, comme décrit dans «Octroi d'un accès administrateur limité à certains processus», à la page 382
Listes de noms	Accordez un accès administrateur partiel aux listes de noms requises, comme décrit dans «Octroi d'un accès administrateur limité à certaines listes de noms», à la page 383
Services	Accordez un accès administrateur partiel aux services requis, comme décrit dans «Octroi d'un accès administratif limité à certains services», à la page 385

Octroi d'un accès administrateur limité à certaines files d'attente

Accordez un accès administrateur partiel à certaines files d'attente d'un gestionnaire de files d'attente, à chaque groupe d'utilisateurs qui en ont besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité à certaines files d'attente pour certaines actions, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Remarque : MQ Appliance Sous IBM MQ Appliance , vous ne pouvez utiliser que la commande **SET AUTHREC** .

Procédure

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
  ```

- IBM i

  Pour IBM i, exécutez la commande suivante:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Pour z/OS, exécutez les commandes suivantes pour accorder l'accès à une file d'attente spécifiée:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Pour spécifier les commandes MQSC que l'utilisateur peut exécuter sur la file d'attente, émettez les commandes suivantes pour chaque commande MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
  PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Pour permettre à l'utilisateur d'utiliser la commande DISPLAY QUEUE, exécutez les commandes suivantes:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
  PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

ReqdAction
L'action que vous autorisez le groupe à effectuer:

- ALW Sur les systèmes AIX, Linux, and Windows , toute combinaison des autorisations suivantes: + chg, + clr, + dlt, + dsp. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.
- IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMDLT, *ADMDSP. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.
- z/OS Sous z/OS, l'une des valeurs ALTER, CLEAR, DELETE ou MOVE.

Remarque : L'octroi + crt pour les files d'attente fait indirectement de l'utilisateur ou du groupe un administrateur. N'utilisez pas le droit + crt pour accorder un accès administrateur limité à certaines files d'attente.

QType

Pour la commande DISPLAY, l'une des valeurs QUEUE, QLOCAL, QALIAS, QMODEL, QREMOTE ou QCLUSTER.

Pour les autres valeurs de *ReqdAction*, l'une des valeurs QLOCAL, QALIAS, QMODEL ou QREMOTE.

Octroi d'un accès administrateur limité à certaines rubriques

Accordez un accès administratif partiel à certaines rubriques d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité à certaines rubriques pour certaines actions, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- IBM i
  Pour IBM i, exécutez la commande suivante:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Pour z/OS, exécutez les commandes suivantes:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Ces commandes permettent d'accéder à la rubrique spécifiée. Pour déterminer les commandes MQSC que l'utilisateur peut exécuter sur la rubrique, exécutez les commandes suivantes pour chaque commande MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Pour permettre à l'utilisateur d'utiliser la commande DISPLAY TOPIC, exécutez les commandes suivantes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

ReqdAction
L'action que vous autorisez le groupe à effectuer:

- ALW Sur les systèmes AIX, Linux, and Windows , toute combinaison des autorisations suivantes: + chg, + clr, + crt, + dlt, + dsp. + ctrl. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.
- IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.
- z/OS Sous z/OS, l'une des valeurs ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Octroi d'un accès administratif limité à certains canaux

Accordez un accès administratif partiel à certains canaux d'un gestionnaire de files d'attente, à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité à certains canaux pour certaines actions, utilisez les commandes appropriées à votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
  ```

- IBM i
  Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Sous z/OS :

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Ces commandes permettent d'accéder au canal spécifié. Pour déterminer les commandes MQSC que l'utilisateur peut exécuter sur le canal, exécutez les commandes suivantes pour chaque commande MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Pour permettre à l'utilisateur d'utiliser la commande DISPLAY CHANNEL, exécutez les commandes suivantes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

ReqdAction
L'action que vous autorisez le groupe à effectuer:

- ALW Sous AIX, Linux, and Windows, toute combinaison des autorisations suivantes: + chg, + clr, + crt, + dlt, + dsp. + ctrl, + ctrlx. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.
- IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLx. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.
- z/OS Sous z/OS, l'une des valeurs ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Octroi d'un accès administrateur limité à un gestionnaire de files d'attente

Accordez un accès administrateur partiel à un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité afin d'effectuer certaines actions sur le gestionnaire de files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Sous AIX, Linux, and Windows :

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- IBM i
  Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS
  Sous z/OS :

  Pour déterminer les commandes MQSC que vous pouvez exécuter sur le gestionnaire de files d'attente, exécutez les commandes suivantes pour chaque commande MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
  PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Pour permettre à l'utilisateur d'utiliser la commande DISPLAY QMGR, exécutez les commandes suivantes:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
  PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

  ReqdAction
  L'action que vous autorisez le groupe à effectuer:

  – ALW Sous AIX, Linux, and Windows, toute combinaison des autorisations suivantes: + chg, + clr, + crt, + dlt, + dsp. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.

    Bien que + set soit une autorisation MQI et qu'elle ne soit normalement pas considérée comme administrative, l'octroi de + set sur le gestionnaire de files d'attente peut indirectement conduire à des droits d'administration complets. N'accordez pas + défini aux utilisateurs et aux applications ordinaires.

  – IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.

Octroi d'un accès administrateur limité à certains processus

Accordez un accès administratif partiel à certains processus d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité à certains processus pour certaines actions, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

Sous AIX, Linux, and Windows :

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
```

- IBM i
  Sous IBM i :

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Sous z/OS :

```
RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Ces commandes permettent d'accéder au canal spécifié. Pour déterminer les commandes MQSC que l'utilisateur peut exécuter sur le canal, exécutez les commandes suivantes pour chaque commande MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Pour permettre à l'utilisateur d'utiliser la commande DISPLAY PROCESS, exécutez les commandes suivantes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

ReqdAction
L'action que vous autorisez le groupe à effectuer:

- ALW Sous AIX, Linux, and Windows, toute combinaison des autorisations suivantes: + chg, + clr, + crt, + dlt, + dsp. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.
- IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.
- z/OS Sous z/OS, l'une des valeurs ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Octroi d'un accès administrateur limité à certaines listes de noms

Accordez un accès administrateur partiel à certaines listes de noms sur un gestionnaire de files d'attente, à chaque groupe d'utilisateurs qui en ont besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité à certaines listes de noms pour certaines actions, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
  ```

- IBM i

  Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Sous z/OS :

  ```
  RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
  PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Ces commandes permettent d'accéder à la liste de noms spécifiée. Pour déterminer les commandes MQSC que l'utilisateur peut exécuter sur la liste de noms, exécutez les commandes suivantes pour chaque commande MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
  PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Pour permettre à l'utilisateur d'utiliser la commande DISPLAY NAMELIST, exécutez les commandes suivantes:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
  PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

  ReqdAction
  L'action que vous autorisez le groupe à effectuer:

  – ALW Sous AIX, Linux, and Windows, toute combinaison des autorisations suivantes: + chg, + clr, + crt, + dlt, + ctrl, + ctrlx, + dsp. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.

– IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.

– z/OS Sous z/OS, l'une des valeurs ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Octroi d'un accès administratif limité à certains services

Accordez un accès administratif partiel à certains services d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur limité à certains services pour certaines actions, utilisez les commandes appropriées pour votre système d'exploitation. z/OS Notez que les objets de service n'existent pas sur z/OS.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
  ```

- Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Sous z/OS :

  Ces commandes permettent d'accéder au service spécifié. Pour déterminer les commandes MQSC que l'utilisateur peut exécuter sur le service, exécutez les commandes suivantes pour chaque commande MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
  PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Pour permettre à l'utilisateur d'utiliser la commande DISPLAY SERVICE, exécutez les commandes suivantes:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
  PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

  ReqdAction
  L'action que vous autorisez le groupe à effectuer:

– ALW Sur les systèmes AIX, Linux, and Windows , toute combinaison des autorisations suivantes: + chg, + clr, + crt, + dlt, + ctrl, + ctrlx, + dsp. L'autorisation + alladm est équivalente à + chg + clr + dlt + dsp.

– IBM i Sous IBM i, toute combinaison des droits suivants: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. L'autorisation *ALLADM est équivalente à toutes ces autorisations individuelles.

Octroi d'un accès administrateur complet sur un sous-ensemble de ressources de gestionnaire de files d'attente

Vous devez accorder à certains utilisateurs un accès administrateur complet à certaines ressources de gestionnaire de files d'attente, mais pas à toutes. Utilisez ces tableaux pour déterminer les actions que vous devez effectuer.

Tableau 73. Octroi d'un accès administrateur complet à un sous-ensemble de ressources de gestionnaire de files d'attente

Les utilisateurs doivent administrer les objets de ce type	**Effectuer cette action**
Files d'attente	Accordez un accès administrateur complet aux files d'attente requises, comme décrit dans «Octroi d'un accès administrateur complet à certaines files d'attente», à la page 386
Rubriques	Accordez un accès administrateur complet aux rubriques requises, comme décrit dans «Octroi d'un accès administrateur complet à certaines rubriques», à la page 387
Canaux	Accordez un accès administrateur complet aux canaux requis, comme décrit dans «Octroi d'un accès administrateur complet à certains canaux», à la page 388
Gestionnaire de files d'attente	Accordez un accès administrateur complet au gestionnaire de files d'attente, comme décrit dans «Octroi d'un accès administrateur complet à un gestionnaire de files d'attente», à la page 389
Processus	Accordez un accès administrateur complet aux processus requis, comme décrit dans «Octroi d'un accès administrateur complet à certains processus», à la page 389
Listes de noms	Accordez un accès administrateur complet aux listes de noms requises, comme décrit dans «Octroi d'un accès administrateur complet à certaines listes de noms», à la page 390
Services	Accordez un accès administrateur complet aux services requis, comme décrit dans «Octroi d'un accès administrateur complet à certains services», à la page 391

Octroi d'un accès administrateur complet à certaines files d'attente

Accordez un accès administrateur complet à certaines files d'attente d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet à certaines files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
  ```

- IBM i

  Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  Sous z/OS :

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à certaines rubriques

Accordez un accès administrateur complet à certaines rubriques d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet à certaines rubriques pour certaines actions, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
  ```

- IBM i

Sous IBM i :

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

Sous z/OS :

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à certains canaux

Accordez un accès administrateur complet à certains canaux d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet à certains canaux, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

Sous AIX, Linux, and Windows :

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- IBM i

Sous IBM i :

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

Sous z/OS :

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à un gestionnaire de files d'attente

Accordez un accès administrateur complet à un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet au gestionnaire de files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +alladm
  ```

- IBM i
  Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  Sous z/OS :

  ```
  RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
  PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à certains processus

Accordez un accès administrateur complet à certains processus d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet à certains processus, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
  ```

- IBM i

  Sous IBM i :

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  Sous z/OS :

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à certaines listes de noms

Accordez un accès administrateur complet à certaines listes de noms d'un gestionnaire de files d'attente, à chaque groupe d'utilisateurs qui en ont besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet à certaines listes de noms, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Sous AIX, Linux, and Windows :

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
  ```

- IBM i

Sous IBM i :

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

Sous z/OS :

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à certains services

Accordez un accès administrateur complet à certains services d'un gestionnaire de files d'attente à chaque groupe d'utilisateurs qui en ont besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder un accès administrateur complet à certains services, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

Sous AIX, Linux, and Windows :

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- IBM i

Sous IBM i :

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

Sous z/OS :

```
RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente.

z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès en lecture seule à toutes les ressources d'un gestionnaire de files d'attente

Accordez un accès en lecture seule à toutes les ressources d'un gestionnaire de files d'attente, à chaque utilisateur ou groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Utilisez l'assistant d'ajout de droits basés sur les rôles ou les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Après avoir modifié les détails d'autorisation, effectuez une actualisation de la sécurité à l'aide de la commande REFRESH SECURITY .

Procédure

- A l'aide de l'assistant:
  a) Dans le panneau IBM MQ Explorer Navigator , cliquez avec le bouton droit de la souris sur le gestionnaire de files d'attente, puis cliquez sur **Droits sur les objets** > **Ajouter des droits basés sur les rôles**

  L'assistant Ajout de droits basés sur des rôles s'ouvre.
- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez les commandes suivantes:

```
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
```

  Droits d'accès spécifiques à SYSTEM.ADMIN.COMMAND.QUEUE et SYSTEM.MQEXPLORER.REPLY.MODEL n'est nécessaire que si vous souhaitez utiliser le IBM MQ Explorer.
- IBM i
  Pour IBM i, exécutez les commandes suivantes:

```
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

  Pour z/OS, exécutez les commandes suivantes:

```
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)
```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi d'un accès administrateur complet à toutes les ressources d'un gestionnaire de files d'attente

Accordez un accès administrateur complet à toutes les ressources d'un gestionnaire de files d'attente, à chaque utilisateur ou groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'assistant Ajouter des droits basés sur les rôles ou les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Remarques : ALW

1. Si vous utilisez **runmqsc** pour administrer le gestionnaire de files d'attente à la place de IBM MQ Explorer, vous devez accorder les droits permettant d'interroger, d'obtenir et de parcourir SYSTEM.MQSC.REPLY.QUEUE, et vous n'avez pas besoin d'accorder des droits sur SYSTEM.MQEXPLORER.REPLY.MODEL .
2. Lorsque vous octroyez à un utilisateur l'accès à toutes les ressources d'un gestionnaire de files d'attente, il existe des commandes que l'utilisateur ne peut pas exécuter, à moins qu'il ne dispose d'un accès en lecture au fichier `qm.ini` . Cela est dû aux restrictions qui s'appliquent aux utilisateurs non `mqm` qui peuvent lire le fichier `qm.ini` .

   L'utilisateur ne peut pas exécuter les commandes suivantes sauf si vous lui avez accordé un accès en lecture au fichier qm.ini :

   - Définition d'un canal configuré pour utiliser TLS

- Définition d'un canal à l'aide de variables d'insertion de configuration automatique définies dans `qm.ini`

Procédure

- Si vous utilisez l'assistant, dans le panneau IBM MQ Explorer Navigator , cliquez avec le bouton droit de la souris sur le gestionnaire de files d'attente, puis cliquez sur **Droits sur les objets** > **Ajouter des droits basés sur les rôles**.

  L'assistant Ajout de droits basés sur des rôles s'ouvre.

- Linux AIX

  Pour les systèmes AIX and Linux , exécutez les commandes suivantes:

```
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
```

  Voir **setmqaut** pour plus d'informations sur `@class`

- Windows

  Pour les systèmes Windows , exécutez les mêmes commandes que pour les systèmes AIX and Linux , mais en utilisant le nom de profil @CLASS au lieu de @class.

- IBM i

  Pour IBM i, exécutez la commande suivante:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
```

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Suppression de la connectivité au gestionnaire de files d'attente

Si vous ne souhaitez pas que les applications utilisateur se connectent à votre gestionnaire de files d'attente, supprimez leurs droits de connexion.

Pourquoi et quand exécuter cette tâche

Révoquez le droit de tous les utilisateurs de se connecter au gestionnaire de files d'attente à l'aide de la commande appropriée pour votre système d'exploitation.

Sous Multiplateformes, vous pouvez également utiliser la commande DELETE AUTHREC .

Remarque : Sur IBM MQ Appliance , vous ne pouvez utiliser que la commande **DELETE AUTHREC** .

Procédure

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName -connect
  ```

- IBM i

  Pour IBM i, exécutez la commande suivante:

  ```
  RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
  ```

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  ```

  N'émettez aucune commande PERMIT.

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente.

  z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  GroupName
  Nom du groupe auquel l'accès doit être refusé.

Autorisation des applications utilisateur à se connecter à votre gestionnaire de files d'attente

Vous souhaitez autoriser l'application utilisateur à se connecter à votre gestionnaire de files d'attente. Utilisez les tableaux de cette rubrique pour déterminer les actions à effectuer.

Tout d'abord, déterminez si les applications client se connecteront à votre gestionnaire de files d'attente.

Si aucune des applications qui se connecteront à votre gestionnaire de files d'attente ne sont des applications client, désactivez l'accès distant comme décrit dans «Désactivation de l'accès distant au gestionnaire de files d'attente», à la page 403.

Si une ou plusieurs des applications qui se connecteront à votre gestionnaire de files d'attente sont des applications client, sécurisez la connectivité à distance comme décrit dans «Sécurisation de la connectivité distante au gestionnaire de files d'attente», à la page 396.

Dans les deux cas, configurez la sécurité de connexion comme décrit dans «Configuration de la sécurité de connexion», à la page 403

Si vous souhaitez contrôler l'accès aux ressources pour chaque utilisateur se connectant au gestionnaire de files d'attente, voir le tableau suivant. Si l'instruction de la première colonne est vraie, effectuez l'action indiquée dans la deuxième colonne.

Instruction	Effectuez cette action
Vous disposez d'applications qui utilisent des files d'attente	Voir le «Contrôle de l'accès utilisateur aux files d'attente», à la page 404
Vous disposez d'applications qui utilisent des rubriques	Voir «Contrôle de l'accès des utilisateurs aux rubriques», à la page 410.
Vous disposez d'applications qui s'interrogent sur l'objet gestionnaire de files d'attente	Voir «Octroi de droits d'interrogation sur un gestionnaire de files d'attente», à la page 412.
Vous disposez d'applications qui utilisent des objets de processus	Voir le «Octroi de droits d'accès aux processus», à la page 413
Vous disposez d'applications qui utilisent des listes de noms	Voir le «Octroi de droits d'accès aux listes de noms», à la page 414

Sécurisation de la connectivité distante au gestionnaire de files d'attente

Vous pouvez sécuriser la connectivité distante au gestionnaire de files d'attente à l'aide de TLS, d'un exit de sécurité, d'enregistrements d'authentification de canal ou d'une combinaison de ces méthodes.

Pourquoi et quand exécuter cette tâche

Vous connectez un client au gestionnaire de files d'attente à l'aide d'un canal de connexion client sur le poste de travail client et d'un canal de connexion serveur sur le serveur. Sécurisez ces connexions de l'une des manières suivantes.

Procédure

1. Utilisation de TLS avec des enregistrements d'authentification de canal:
   a) Empêchez tout nom distinctif (DN) d'ouvrir un canal en utilisant un enregistrement d'authentification de canal SSLPEERMAP pour mapper tous les noms distinctifs à USERSRC (NOACCESS).
   b) Autoriser des noms distinctifs ou des ensembles de noms distinctifs spécifiques à ouvrir un canal à l'aide d'un enregistrement d'authentification de canal SSLPEERMAP pour les mapper à USERSRC (CHANNEL).
2. Utilisation de TLS avec un exit de sécurité:
   a) Définissez MCAUSER sur le canal de connexion serveur sur un identificateur utilisateur sans privilèges.
   b) Ecrivez un exit de sécurité pour affecter une valeur MCAUSER en fonction de la valeur du nom distinctif TLS qu'il reçoit dans les zones SSLPeerNamePtr et SSLPeerNameLength transmises à l'exit dans la structure MQCD.
3. Utilisation de TLS avec des valeurs de définition de canal fixes:
   a) Définissez SSLPEER sur le canal de connexion serveur sur une valeur spécifique ou une plage de valeurs étroite.
   b) Définissez MCAUSER sur le canal de connexion serveur sur l'ID utilisateur avec lequel le canal doit s'exécuter.
4. Utilisation des enregistrements d'authentification de canal sur les canaux qui n'utilisent pas TLS:
   a) Empêchez toute adresse IP d'ouvrir des canaux en utilisant un enregistrement d'authentification de canal de mappage d'adresse avec ADDRESS (*) et USERSRC (NOACCESS).
   b) Autorisez des adresses IP spécifiques à ouvrir des canaux, en utilisant des enregistrements d'authentification de canal de mappage d'adresse pour ces adresses avec USERSRC (CHANNEL).

5. A l'aide d'un exit de sécurité:
   a) Ecrivez un exit de sécurité pour autoriser les connexions en fonction de la propriété que vous choisissez, par exemple, l'adresse IP d'origine.
6. Il est également possible d'utiliser des enregistrements d'authentification de canal avec un exit de sécurité ou d'utiliser les trois méthodes, si vos circonstances particulières l'exigent.

Blocage d'adresses IP spécifiques
Vous pouvez empêcher un canal spécifique d'accepter une connexion entrante à partir d'une adresse IP ou empêcher l'ensemble du gestionnaire de files d'attente d'autoriser l'accès à partir d'une adresse IP, à l'aide d'un enregistrement d'authentification de canal.

Avant de commencer

Activez les enregistrements d'authentification de canal en exécutant la commande suivante:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Pourquoi et quand exécuter cette tâche

Pour empêcher des canaux spécifiques d'accepter une connexion entrante et s'assurer que les connexions sont uniquement acceptées lors de l'utilisation du nom de canal correct, un type de règle peut être utilisé pour bloquer les adresses IP. Pour interdire l'accès d'une adresse IP à l'ensemble du gestionnaire de files d'attente, vous devez normalement utiliser un pare-feu pour le bloquer définitivement. Toutefois, un autre type de règle peut être utilisé pour vous permettre de bloquer temporairement quelques adresses, par exemple pendant que vous attendez que le pare-feu soit mis à jour.

Procédure

- Pour empêcher les adresses IP d'utiliser un canal spécifique, définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**.

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

La commande comporte trois parties:

SET CHLAUTH (*nom-canal-générique*)
Cette partie de la commande permet de contrôler si vous souhaitez bloquer une connexion pour l'ensemble du gestionnaire de files d'attente, un canal unique ou une plage de canaux. Ce que vous mettez ici détermine les zones qui sont couvertes.

Exemple :

- SET CHLAUTH('*') -bloque tous les canaux d'un gestionnaire de files d'attente, c'est-à-dire l'intégralité du gestionnaire de files d'attente
- SET CHLAUTH ('SYSTEM. *')-bloque chaque canal commençant par SYSTEM.
- SET CHLAUTH ('SYSTEM.DEF.SVRCONN')-bloque le canal SYSTEM.DEF.SVRCONN

Type de règle CHLAUTH
Utilisez cette partie de la commande pour spécifier le type de commande et déterminer si vous souhaitez fournir une adresse unique ou une liste d'adresses.

Exemple :

- TYPE(ADDRESSMAP) -Utilisez ADDRESSMAP si vous souhaitez fournir une adresse unique ou une adresse générique. Par exemple, ADDRESS('192.168.*') bloque toutes les connexions provenant d'une adresse IP à partir de 192.168.

Pour plus d'informations sur le filtrage des adresses IP à l'aide de modèles, voir Adresses IP génériques.

- TYPE(BLOCKADDR) -Utilisez BLOCKADDR si vous souhaitez fournir une liste d'adresses à bloquer.

Paramètres supplémentaires

Ces paramètres dépendent du type de règle que vous avez utilisé dans la deuxième partie de la commande:

- Pour TYPE(ADDRESSMAP) , vous utilisez ADDRESS
- Pour TYPE(BLOCKADDR) , vous utilisez ADDRLIST

Référence associée

SET CHLAUTH

Blocage temporaire d'adresses IP spécifiques si le gestionnaire de files d'attente n'est pas en cours d'exécution

Vous pouvez bloquer des adresses IP particulières ou des plages d'adresses lorsque le gestionnaire de files d'attente n'est pas en cours d'exécution et que vous ne pouvez donc pas émettre de commandes MQSC. Vous pouvez temporairement bloquer des adresses IP de manière exceptionnelle en modifiant le fichier `blockaddr.ini` .

Pourquoi et quand exécuter cette tâche

Le fichier `blockaddr.ini` contient une copie des définitions BLOCKADDR utilisées par le gestionnaire de files d'attente. Ce fichier est lu par le programme d'écoute si ce dernier est démarré avant le gestionnaire de files d'attente. Dans ces circonstances, le programme d'écoute utilise toutes les valeurs que vous avez ajoutées manuellement au fichier `blockaddr.ini` .

Toutefois, sachez que lorsque le gestionnaire de files d'attente est démarré, il écrit l'ensemble des définitions BLOCKADDR dans le fichier `blockaddr.ini` , en écrase les modifications manuelles que vous avez éventuellement effectuées. De même, chaque fois que vous ajoutez ou supprimez une définition BLOCKADDR à l'aide de la commande **SET CHLAUTH** , le fichier `blockaddr.ini` est mis à jour. Vous pouvez donc apporter des modifications permanentes aux définitions BLOCKADDR uniquement à l'aide de la commande **SET CHLAUTH** lorsque le gestionnaire de files d'attente est en cours d'exécution.

Procédure

1. Ouvrez le fichier `blockaddr.ini` dans n'importe quel éditeur de texte.

   Le fichier se trouve dans le répertoire de données du gestionnaire de files d'attente.
2. Ajoutez des adresses IP sous forme de paires mot clé-valeur simples, où le mot clé est `Addr`.

   Pour plus d'informations sur le filtrage des adresses IP à l'aide de modèles, voir Adresses IP génériques.

   Exemple :

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

Tâches associées

«Blocage d'adresses IP spécifiques», à la page 397

Vous pouvez empêcher un canal spécifique d'accepter une connexion entrante à partir d'une adresse IP ou empêcher l'ensemble du gestionnaire de files d'attente d'autoriser l'accès à partir d'une adresse IP, à l'aide d'un enregistrement d'authentification de canal.

Référence associée

SET CHLAUTH

Blocage d'ID utilisateur spécifiques
Vous pouvez empêcher des utilisateurs spécifiques d'utiliser un canal en spécifiant des ID utilisateur qui, s'ils sont vérifiés, provoquent l'arrêt du canal. Pour cela, définissez un enregistrement d'authentification de canal.

Avant de commencer

Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procédure

Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
La liste d'utilisateurs fournie sur un TYPE(BLOCKUSER) s'applique uniquement aux canaux SVRCONN et non aux canaux de gestionnaire de files d'attente.
userID1 et *userID2* sont chacun l'ID d'un utilisateur qui doit être empêché d'utiliser le canal. Vous pouvez également spécifier la valeur spéciale *MQADMIN pour faire référence aux administrateurs privilégiés. Pour plus d'informations sur les utilisateurs privilégiés, voir «Utilisateurs privilégiés», à la page 329. Pour plus d'informations sur *MQADMIN, voir SET CHLAUTH.

Référence associée
SET CHLAUTH

Mappage d'un gestionnaire de files d'attente éloignées à un ID utilisateur MCAUSER
Vous pouvez utiliser un enregistrement d'authentification de canal pour définir l'attribut MCAUSER d'un canal, en fonction du gestionnaire de files d'attente à partir duquel le canal se connecte.

Avant de commencer

Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Pourquoi et quand exécuter cette tâche

Vous pouvez éventuellement restreindre les adresses IP auxquelles la règle s'applique.

Notez que cette technique ne s'applique pas aux canaux de connexion serveur. Si vous spécifiez le nom d'un canal de connexion serveur dans les commandes suivantes, cela n'a aucun effet.

Procédure

- Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

  ```
  SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
  ) USERSRC(MAP) MCAUSER(user)
  ```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
generic-partner-qmgr-name est soit le nom du gestionnaire de files d'attente, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du gestionnaire de files d'attente.
user est l'ID utilisateur à utiliser pour toutes les connexions à partir du gestionnaire de files d'attente spécifié.

- Pour limiter cette commande à certaines adresses IP, incluez le paramètre **ADDRESS** comme suit:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user) ADDRESS(
generic-ip-address)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
generic-ip-address est soit une adresse unique, soit un modèle incluant l'astérisque (*) comme caractère générique ou le trait d'union (-) pour indiquer une plage, qui correspond à l'adresse. Pour plus d'informations sur les adresses IP génériques, voir Adresses IP génériques.

Référence associée
SET CHLAUTH

Mappage d'un ID utilisateur client à un ID utilisateur MCAUSER
Vous pouvez utiliser un enregistrement d'authentification de canal pour modifier l'attribut MCAUSER d'un canal de connexion serveur, en fonction de l'ID utilisateur reçu d'un client.

Avant de commencer
Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Pourquoi et quand exécuter cette tâche
Notez que cette technique s'applique uniquement aux canaux de connexion serveur. Il n'a aucun effet sur les autres types de canal.

Procédure
Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record** . Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
client-user-name est l'ID utilisateur associé à la connexion client. La valeur peut être vérifiée par l'application client, modifiée par l'authentification de connexion à l'aide de l'adoption anticipée ou définie via un exit de canal.
user est l'ID utilisateur à utiliser à la place du nom d'utilisateur du client.

Référence associée
SET CHLAUTH
Attributs de la strophe channels (ChlauthEarlyAdopt)

Mappage d'un nom distinctif SSL ou TLS à un ID utilisateur MCAUSER
Vous pouvez utiliser un enregistrement d'authentification de canal pour définir l'attribut MCAUSER d'un canal, en fonction du nom distinctif (DN) reçu.

Avant de commencer

Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procédure

Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
generic-ssl-peer-name est une chaîne qui suit les règles IBM MQ standard pour les valeurs SSLPEER. Voir Règles IBM MQ pour les valeurs SSLPEER.
user est l'ID utilisateur à utiliser pour toutes les connexions utilisant le nom distinctif spécifié.
generic-issuer-name fait référence au nom distinctif de l'émetteur du certificat à mettre en correspondance. Ce paramètre est facultatif, mais vous devez l'utiliser afin d'éviter toute correspondance avec le mauvais certificat, si plusieurs autorités de certification sont utilisées.

Référence associée
SET CHLAUTH

Blocage de l'accès à partir d'un gestionnaire de files d'attente éloignées
Vous pouvez utiliser un enregistrement d'authentification de canal pour empêcher un gestionnaire de files d'attente éloignées de démarrer des canaux.

Avant de commencer

Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Pourquoi et quand exécuter cette tâche

Notez que cette technique ne s'applique pas aux canaux de connexion serveur. Si vous indiquez le nom d'un canal de connexion serveur dans la commande suivante, cela n'a aucun effet.

Procédure

Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.

generic-partner-qmgr-name est soit le nom du gestionnaire de files d'attente, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du gestionnaire de files d'attente.

Référence associée

SET CHLAUTH

Blocage de l'accès pour un ID utilisateur client

Vous pouvez utiliser un enregistrement d'authentification de canal pour empêcher un ID utilisateur client d'établir une connexion de canal.

Avant de commencer

Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Pourquoi et quand exécuter cette tâche

Notez que cette technique s'applique uniquement aux canaux de connexion serveur. Il n'a aucun effet sur les autres types de canal.

Procédure

Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
client-user-name est l'ID utilisateur associé à la connexion client. La valeur peut être vérifiée par l'application client, modifiée par l'authentification de connexion à l'aide de l'adoption anticipée ou définie via un exit de canal.

Référence associée

SET CHLAUTH

Blocage de l'accès pour un nom distinctif SSL ou TLS

Vous pouvez utiliser un enregistrement d'authentification de canal pour empêcher un nom distinctif TLS de démarrer des canaux.

Avant de commencer

Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procédure

Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
generic-ssl-peer-name est une chaîne qui suit les règles IBM MQ standard pour les valeurs SSLPEER. Voir Règles IBM MQ pour les valeurs SSLPEER.
generic-issuer-name fait référence au nom distinctif de l'émetteur du certificat à mettre en correspondance. Ce paramètre est facultatif, mais vous devez l'utiliser afin d'éviter toute correspondance avec le mauvais certificat, si plusieurs autorités de certification sont utilisées.

Référence associée
SET CHLAUTH

Mappage d'une adresse IP à un ID utilisateur MCAUSER
Vous pouvez utiliser un enregistrement d'authentification de canal pour définir l'attribut MCAUSER d'un canal, en fonction de l'adresse IP à partir de laquelle la connexion est reçue.

Avant de commencer
Vérifiez que les enregistrements d'authentification de canal sont activés comme suit:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procédure
Définissez un enregistrement d'authentification de canal à l'aide de la commande MQSC **SET CHLAUTH**ou de la commande PCF **Set Channel Authentication Record**. Par exemple, vous pouvez émettre la commande MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

nom-canal-générique est soit le nom d'un canal auquel vous souhaitez contrôler l'accès, soit un modèle incluant l'astérisque (*) comme caractère générique qui correspond au nom du canal.
user est l'ID utilisateur à utiliser pour toutes les connexions utilisant le nom distinctif spécifié.
generic-ip-address est soit l'adresse à partir de laquelle la connexion est établie, soit un modèle incluant l'astérisque (*) comme caractère générique ou le trait d'union (-) pour indiquer une plage qui correspond à l'adresse.

Référence associée
SET CHLAUTH

Désactivation de l'accès distant au gestionnaire de files d'attente
Si vous ne souhaitez pas que les applications client se connectent à votre gestionnaire de files d'attente, désactivez l'accès à distance à ce dernier.

Pourquoi et quand exécuter cette tâche
Empêchez les applications client de se connecter au gestionnaire de files d'attente de l'une des manières suivantes:

Procédure
- Supprimez tous les canaux de connexion serveur à l'aide de la commande MQSC **DELETE CHANNEL**.
- Définissez l'ID utilisateur de l'agent de canal de transmission de messages (MCAUSER) du canal sur un ID utilisateur sans droits d'accès, à l'aide de la commande MQSC **ALTER CHANNEL**.

Configuration de la sécurité de connexion
Accordez le droit de se connecter au gestionnaire de files d'attente à chaque utilisateur ou groupe d'utilisateurs dont l'entreprise a besoin pour le faire.

Pourquoi et quand exécuter cette tâche

Pour configurer la sécurité de la connexion, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Sous AIX, Linux, and Windows :

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

- IBM i

  Sous IBM i :

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```

- z/OS

  Sous 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)
  ```

  Ces commandes donnent le droit de se connecter pour le traitement par lots, CICS, IMS et l'initiateur de canal (CHIN). Si vous n'utilisez pas un type particulier de connexion, omettez les commandes appropriées.

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Concepts associés

«Connection security profiles for the channel initiator», à la page 209
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.

Contrôle de l'accès utilisateur aux files d'attente

Vous souhaitez contrôler l'accès des applications aux files d'attente. Cette rubrique permet de déterminer les actions à effectuer.

Pour chaque instruction true de la première colonne, effectuez l'action indiquée dans la deuxième colonne.

Instruction	Action
L'application extrait des messages d'une file d'attente	Voir le «Octroi de droits pour l'obtention de messages à partir de files d'attente», à la page 405
L'application définit le contexte	Voir le «Octroi de droits d'accès pour définir le contexte», à la page 406
L'application transmet le contexte	Voir le «Octroi de l'autorisation de transmettre le contexte», à la page 407
L'application insère des messages dans une file d'attente en cluster	Voir le «Autorisation d'insertion de messages dans des files d'attente de cluster éloignées», à la page 493
L'application insère des messages dans une file d'attente locale	Voir le «Octroi du droit d'insertion de messages dans une file d'attente locale», à la page 408
L'application insère des messages dans une file d'attente modèle	Voir le «Octroi du droit d'insertion de messages dans une file d'attente modèle», à la page 408
L'application insère des messages dans une file d'attente éloignée	Voir le «Octroi du droit d'insertion de messages dans une file d'attente de cluster éloignée», à la page 409

Octroi de droits pour l'obtention de messages à partir de files d'attente
Accordez le droit d'extraire des messages d'une file d'attente ou d'un ensemble de files d'attente à chaque groupe d'utilisateurs qui en a besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit d'extraire des messages de certaines files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- Windows
  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM i
  Pour IBM i, exécutez la commande suivante:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
```

- z/OS
  Pour z/OS, exécutez les commandes suivantes:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi de droits d'accès pour définir le contexte
Accordez le droit de définir le contexte d'un message en cours d'insertion à chaque groupe d'utilisateurs qui en ont besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit de définir le contexte sur certaines files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez l'une des commandes suivantes:
  - Pour définir le contexte d'identité uniquement:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
    ```

  - Pour définir tous les contextes:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
    ```

  Remarque : Pour utiliser les droits `setid` ou `setall` , les autorisations doivent être accordées à la fois sur l'objet de file d'attente approprié et sur l'objet de gestionnaire de files d'attente.
- IBM i
  Pour IBM i, exécutez l'une des commandes suivantes:
  - Pour définir le contexte d'identité uniquement:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
    QMgrName ')
    ```

  - Pour définir tous les contextes:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
    QMgrName ')
    ```

- z/OS
  Pour z/OS, exécutez l'un des ensembles de commandes suivants:
  - Pour définir le contexte d'identité uniquement:

    ```
    RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
    PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
    ```

  - Pour définir tous les contextes:

    ```
    RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
    PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
    ```

  Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi de l'autorisation de transmettre le contexte
Accordez le droit de transmettre le contexte d'un message extrait à un message en cours d'insertion, à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit de transmettre le contexte sur certaines files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez l'une des commandes suivantes:
  - Pour transmettre uniquement le contexte d'identité:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
    ```

  - Pour transmettre tous les contextes:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
    ```

- IBM i
  Pour IBM i, exécutez l'une des commandes suivantes:
  - Pour transmettre uniquement le contexte d'identité:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
    QMgrName ')
    ```

  - Pour transmettre tous les contextes:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
    QMgrName ')
    ```

- z/OS
  Pour z/OS, exécutez les commandes suivantes pour transmettre le contexte d'identité ou tout le contexte:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi du droit d'insertion de messages dans une file d'attente locale
Accordez le droit d'insérer des messages dans une file d'attente locale ou dans un ensemble de files d'attente, à chaque groupe d'utilisateurs qui en a besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit d'insertion de messages dans certaines files d'attente locales, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
  ```

- IBM i
  Pour IBM i, exécutez la commande suivante:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  ```

- z/OS
  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi du droit d'insertion de messages dans une file d'attente modèle
Accordez le droit d'insérer des messages dans une file d'attente modèle ou dans un ensemble de files d'attente modèle, à chaque groupe d'utilisateurs qui en a besoin.

Pourquoi et quand exécuter cette tâche

Les files d'attente modèles sont utilisées pour créer des files d'attente dynamiques. Vous devez donc accorder des droits d'accès au modèle et aux files d'attente dynamiques. Pour accorder ces droits, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez les commandes suivantes:

```
setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i

  Pour IBM i, exécutez les commandes suivantes:

```
GRTMQMAUT OBJ(' ModelQueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

```
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)
```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  Nom ModelQueue
  Nom de la file d'attente modèle sur laquelle sont basées les files d'attente dynamiques.

  ObjectProfile
  Nom de la file d'attente dynamique ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi du droit d'insertion de messages dans une file d'attente de cluster éloignée
Accordez le droit d'insérer des messages dans une file d'attente de cluster éloignée ou dans un ensemble de files d'attente, à chaque groupe d'utilisateurs qui en a besoin.

Pourquoi et quand exécuter cette tâche

Pour placer un message dans une file d'attente de cluster éloignée, vous pouvez le placer dans une définition locale d'une file d'attente éloignée ou dans une file d'attente éloignée qualifiée complète. Si vous utilisez une définition locale d'une file d'attente éloignée, vous devez disposer des droits d'accès à l'objet local: voir «Octroi du droit d'insertion de messages dans une file d'attente locale», à la page 408. Si vous utilisez une file d'attente éloignée qualifiée complète, vous devez disposer des droits nécessaires pour la placer dans la file d'attente éloignée. Accordez ces droits à l'aide des commandes appropriées pour votre système d'exploitation.

Le comportement par défaut consiste à effectuer un contrôle d'accès sur le SYSTEM.CLUSTER.TRANSMIT.QUEUE. Notez que ce comportement s'applique, même si vous utilisez plusieurs files d'attente de transmission.

Le comportement spécifique décrit dans cette rubrique s'applique uniquement lorsque vous avez configuré l'attribut **ClusterQueueAccessControl** dans le fichier qm.ini comme étant *RQMName*, comme décrit dans la rubrique Strophe de sécurité , puis redémarré le gestionnaire de files d'attente.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -t rqmname -n
  ObjectProfile -g GroupName +put
  ```

  Notez que vous pouvez utiliser l'objet *rqmname* uniquement pour les files d'attente de cluster éloignées.

- IBM i

  Pour IBM i, exécutez la commande suivante:

  ```
  GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
  ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
  QMgrName')
  ```

  Notez que vous pouvez utiliser l'objet RMTMQMNAME uniquement pour les files d'attente de cluster éloignées.

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Notez que vous pouvez utiliser le nom du gestionnaire de files d'attente éloignées (ou du groupe de partage de files d'attente) uniquement pour les files d'attente de cluster éloignées.

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom du gestionnaire de files d'attente éloignées ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Contrôle de l'accès des utilisateurs aux rubriques

Vous devez contrôler l'accès des applications aux rubriques. Cette rubrique permet de déterminer les actions à effectuer.

Pour chaque instruction true de la première colonne, effectuez l'action indiquée dans la deuxième colonne.

Tableau 74. Contrôle de l'accès des utilisateurs aux rubriques

Instruction	Action
L'application publie des messages dans un sujet	Voir le «Octroi du droit de publier des messages dans une rubrique», à la page 411
L'application s'abonne à une rubrique	Voir le «Octroi de droits d'abonnement à des rubriques», à la page 411

Octroi du droit de publier des messages dans une rubrique
Accordez le droit de publier des messages dans une rubrique ou un ensemble de rubriques, à chaque groupe d'utilisateurs qui en a besoin.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit de publier des messages dans certaines rubriques, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
  ```

- IBM i
  Pour IBM i, exécutez la commande suivante:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('
  QMgrName ')
  ```

- z/OS
  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi de droits d'abonnement à des rubriques
Accordez le droit de s'abonner à une rubrique ou à un ensemble de rubriques, à chaque groupe d'utilisateurs ayant besoin d'une rubrique ou d'un ensemble de rubriques.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit d'abonnement à certaines rubriques, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- IBM i

  Pour IBM i, exécutez la commande suivante:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('
QMgrName ')
```

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi de droits d'interrogation sur un gestionnaire de files d'attente

Accordez les droits d'interrogation sur un gestionnaire de files d'attente à chaque groupe d'utilisateurs ayant besoin d'un gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit d'interrogation sur un gestionnaire de files d'attente, utilisez les commandes appropriées pour votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- IBM i

  Pour IBM i, exécutez la commande suivante:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
')
```

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Ces commandes permettent d'accéder au gestionnaire de files d'attente spécifié. Pour permettre à l'utilisateur d'utiliser la commande MQINQ, exécutez les commandes suivantes:

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Les noms de variable ont les significations suivantes:

QMgrName
Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

ObjectProfile
Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

GroupName
Nom du groupe auquel l'accès doit être accordé.

Octroi de droits d'accès aux processus

Accordez le droit d'accès à un processus ou à un ensemble de processus à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder le droit d'accès à certains processus, utilisez les commandes appropriées à votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW
  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
  ```

- IBM i
  Pour IBM i, exécutez la commande suivante:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
  ')
  ```

- z/OS
  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

Octroi de droits d'accès aux listes de noms

Accordez le droit d'accéder à une liste de noms ou à un ensemble de listes de noms, à chaque groupe d'utilisateurs ayant un besoin métier.

Pourquoi et quand exécuter cette tâche

Pour accorder les droits d'accès à certaines listes de noms, utilisez les commandes appropriées à votre système d'exploitation.

Multi Sur Multiplatforms, vous pouvez également utiliser la commande SET AUTHREC .

Procédure

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez la commande suivante:

  ```
  setmqaut -m QMgrName -n
  ObjectProfile -t namelist -g GroupName
  +all
  ```

- IBM i

  Pour IBM i, exécutez la commande suivante:

  ```
  GRTMQMAUT OBJ('ObjectProfile
  ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
  QMgrName')
  ```

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQNLIST
  QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile
  CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
  ```

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  ObjectProfile
  Nom de l'objet ou du profil générique pour lequel modifier les autorisations.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

ALW Droit d'administration de IBM MQ sur AIX, Linux, and Windows

Les administrateurs IBM MQ peuvent utiliser toutes les commandes IBM MQ et accorder des droits à d'autres utilisateurs. Lorsque les administrateurs émettent des commandes pour les gestionnaires de files d'attente éloignées, ils doivent disposer des droits requis sur le gestionnaire de files d'attente éloignées. D'autres considérations s'appliquent aux systèmes Windows .

Les administrateurs IBM MQ ont le droit d'utiliser toutes les commandes IBM MQ (y compris les commandes permettant d'accorder des droits IBM MQ à d'autres utilisateurs).

Pour être un administrateur IBM MQ , vous devez être membre d'un groupe spécial appelé groupe **mqm** .

Windows Sinon, sous Windows uniquement, les comptes locaux peuvent administrer IBM MQ s'ils sont membres du groupe Administrateurs sur les systèmes Windows .

Avertissement : Vous pouvez ajouter votre utilisateur Azure AD au groupe mqm à l'aide d'une commande d'administrateur. Par exemple, utilisez la commande `net localgroup mqm AzureAD\<your userID> /add`. Exécutez ensuite les commandes d'administration IBM MQ ou utilisez IBM MQ Explorer.

Le groupe **mqm** est créé automatiquement lorsque IBM MQ est installé. Vous pouvez ajouter d'autres utilisateurs au groupe pour leur permettre d'effectuer des tâches d'administration. Tous les membres de ce groupe peuvent accéder à toutes les ressources. Cet accès peut être révoqué uniquement en supprimant un utilisateur du groupe **mqm** et en exécutant la commande **REFRESH SECURITY** .

Les administrateurs peuvent utiliser des commandes de contrôle pour administrer IBM MQ. L'une de ces commandes de contrôle est **setmqaut**, qui est utilisée pour accorder des droits à d'autres utilisateurs afin de leur permettre d'accéder aux ressources IBM MQ ou de les contrôler. Les commandes PCF pour les enregistrements d'autorité de gestion sont disponibles pour les non-administrateurs auxquels sont accordés des droits dsp et chg sur le gestionnaire de files d'attente. Pour plus d'informations sur les droits de gestion à l'aide des commandes PCF, voir Programmable Command Formats.

Les administrateurs doivent disposer des droits requis pour que les commandes MQSC soient traitées par le gestionnaire de files d'attente éloignées. IBM MQ Explorer émet des commandes PCF pour effectuer des tâches d'administration. Les administrateurs n'ont pas besoin de droits supplémentaires pour utiliser IBM MQ Explorer afin d'administrer un gestionnaire de files d'attente sur le système local. Lorsque IBM MQ Explorer est utilisé pour administrer un gestionnaire de files d'attente sur un autre système, les administrateurs doivent disposer des droits requis pour que les commandes PCF soient traitées par le gestionnaire de files d'attente éloignées.

Avertissement : Vous n'avez pas besoin d'être un administrateur pour utiliser la commande de contrôle **runmqsc**, qui émet des commandes IBM MQ Script (MQSC).

Lorsque **runmqsc** est utilisé en mode indirect pour envoyer des commandes MQSC à un gestionnaire de files d'attente éloignées, chaque commande MQSC est encapsulée dans une commande Escape PCF.

Pour plus d'informations sur les vérifications des droits d'accès lors du traitement des commandes PCF et MQSC, voir les rubriques suivantes:

- Pour les commandes PCF qui fonctionnent sur les gestionnaires de files d'attente, les files d'attente, les processus, les listes de noms et les objets d'informations d'authentification, voir Droits d'utilisation des objets IBM MQ. Reportez-vous à cette section pour connaître les commandes MQSC équivalentes encapsulées dans les commandes Escape PCF.
- Pour les commandes PCF qui fonctionnent sur des canaux, des initiateurs de canal, des programmes d'écoute et des clusters, voir Sécurité des canaux.
- Pour les commandes PCF qui fonctionnent sur des enregistrements de droits d'accès, voir Contrôle des droits d'accès pour les commandes PCF
- z/OS Pour les commandes MQSC traitées par le serveur de commandes sous IBM MQ for z/OS, voir Command security and command resource security on z/OS .

En outre, sur les systèmes Windows , le compte SYSTEM dispose d'un accès complet aux ressources IBM MQ .

Sur les plateformes AIX and Linux , un ID utilisateur spécial **mqm** est également créé pour être utilisé par le produit uniquement. Il ne doit jamais être disponible pour les utilisateurs non privilégiés. Tous les objets IBM MQ appartiennent à l'ID utilisateur **mqm**.

Sur les systèmes Windows , les membres du groupe Administrateurs peuvent également administrer n'importe quel gestionnaire de files d'attente, tout comme le compte SYSTEM. Vous pouvez également créer un groupe **mqm** de domaine sur le contrôleur de domaine qui contient tous les ID utilisateur privilégiés actifs dans le domaine et l'ajouter au groupe **mqm** local. Certaines commandes, par exemple **crtmqm**, manipulent les droits sur les objets IBM MQ et ont donc besoin de droits pour utiliser ces objets (comme décrit dans les sections suivantes). Les membres du groupe **mqm** ont le droit d'utiliser tous les objets, mais sur les systèmes Windows , il se peut que les droits d'accès soient refusés si vous disposez

d'un utilisateur local et d'un utilisateur authentifié par le domaine portant le même nom. Ceci est décrit dans «Principaux et groupes sous AIX, Linux, and Windows», à la page 419.

Les versions de Windows comportant une fonction de contrôle de compte utilisateur restreignent les actions que les utilisateurs peuvent exécuter sur certaines fonctions du système d'exploitation, même s'ils sont membres du groupe des administrateurs. Si votre ID utilisateur se trouve dans le groupe Administrateurs mais pas dans le groupe **mqm** , vous devez utiliser une invite de commande avec des droits élevés pour émettre des commandes d'administration IBM MQ telles que **crtmqm**. Sinon, l'erreur `AMQ7077: Vous n'êtes pas autorisé à effectuer l'opération demandée` est générée. Pour ouvrir une invite de commande avec des droits élevés, cliquez à l'aide du bouton droit de la souris sur l'option de menu Démarrer ou sur l'icône de l'invite de commande, puis sélectionnez **Exécuter en tant qu'administrateur**.

Vous n'avez pas besoin d'être membre du groupe **mqm** pour effectuer les actions suivantes:

- Emettez des commandes à partir d'un programme d'application qui émet des commandes PCF ou des commandes MQSC dans une commande PCF d'échappement, sauf si les commandes manipulent des initiateurs de canal. (Ces commandes sont décrites dans «Protection des définitions d'initialisateur de canal», à la page 123).
- Emettez des appels MQI à partir d'un programme d'application (sauf si vous souhaitez utiliser les liaisons Fast Path dans l'appel `MQCONNX`).
- Utilisez la commande `crtmqcvx` pour créer un fragment de code qui effectue la conversion de données sur les structures de type de données.
- Utilisez la commande `dspmq` pour afficher les gestionnaires de files d'attente.
- Utilisez la commande `dspmqtrc` pour afficher la sortie de trace formatée IBM MQ .

Une limitation de 12 caractères s'applique aux ID groupe et utilisateur.

Les plateformes UNIX and Linux limitent généralement la longueur d'un ID utilisateur à 12 caractères. AIX 5.3 a augmenté cette limite, mais IBM MQ continue d'observer une restriction de 12 caractères sur toutes les plateformes UNIX and Linux . Si vous utilisez un ID utilisateur de plus de 12 caractères, IBM MQ le remplace par la valeur `UNKNOWN`. Ne définissez pas d'ID utilisateur avec la valeur `UNKNOWN`.

ALW Gestion du groupe mqm sur AIX, Linux, and Windows

Les utilisateurs du groupe mqm bénéficient de privilèges d'administration complets sur IBM MQ. Pour cette raison, vous ne devez pas inscrire d'applications et d'utilisateurs ordinaires dans le groupe mqm. Le groupe mqm doit contenir uniquement les comptes des administrateurs IBM MQ .

Ces tâches sont décrites dans:

- Windows Création et gestion de groupes sur Windows
- AIX Création et gestion de groupes sur AIX
- Linux Création et gestion de groupes sur Linux

Windows Si votre contrôleur de domaine s'exécute sous Windows 2000 ou Windows 2003 ou version ultérieure, votre administrateur de domaine devra peut-être configurer un compte spécial à utiliser par IBM MQ . Pour plus d'informations, voir Configuration de IBM MQ avec Prepare IBM MQ Wizard et Création et configuration de comptes de domaine Windows pour IBM MQ.

ALW Droits d'utiliser les objets IBM MQ sur AIX, Linux, and Windows

Tous les objets sont protégés par IBM MQet les principaux doivent disposer des droits appropriés pour y accéder. Les différents principaux ont besoin de droits d'accès différents à des objets différents.

Les gestionnaires de files d'attente, les files d'attente, les définitions de processus, les listes de noms, les canaux, les canaux de connexion client, les programmes d'écoute, les services et les objets d'informations d'authentification sont tous accessibles à partir d'applications qui utilisent des appels MQI

ou des commandes PCF. Ces ressources sont toutes protégées par IBM MQet les applications doivent être autorisées à y accéder. L'entité qui effectue la demande peut être un utilisateur, un programme d'application qui émet un appel MQI ou un programme d'administration qui émet une commande PCF. L'identificateur du demandeur est appelé *principal*.

Différents groupes de principaux peuvent être accordés à différents types de droits d'accès sur le même objet. Par exemple, pour une file d'attente spécifique, un groupe peut être autorisé à effectuer des opérations d'insertion et d'extraction ; un autre groupe peut être autorisé uniquement à parcourir la file d'attente (MQGET avec l'option de navigation). De même, certains groupes peuvent avoir des droits d'insertion et d'obtention sur une file d'attente, mais ils ne sont pas autorisés à modifier les attributs de la file d'attente ou à la supprimer.

Certaines opérations sont particulièrement sensibles et doivent être limitées aux utilisateurs privilégiés. Exemple :

- Accès à certaines files d'attente spéciales, telles que les files d'attente de transmission ou la file d'attente de commandes SYSTEM.ADMIN.COMMAND.QUEUE
- Exécution de programmes qui utilisent des options de contexte MQI complètes
- Création et suppression de files d'attente d'application

Le droit d'accès complet à un objet est automatiquement accordé à l'ID utilisateur qui a créé l'objet et à tous les membres du groupe mqm (ainsi qu'aux membres du groupe Administrateurs locaux sur les systèmes Windows).

Concepts associés

«Droit d'administration de IBM MQ sur AIX, Linux, and Windows», à la page 414
Les administrateurs IBM MQ peuvent utiliser toutes les commandes IBM MQ et accorder des droits à d'autres utilisateurs. Lorsque les administrateurs émettent des commandes pour les gestionnaires de files d'attente éloignées, ils doivent disposer des droits requis sur le gestionnaire de files d'attente éloignées. D'autres considérations s'appliquent aux systèmes Windows .

ALW Lorsque des contrôles de sécurité sont effectués sur AIX, Linux, and Windows

Les contrôles de sécurité sont généralement effectués lors de la connexion à un gestionnaire de files d'attente, de l'ouverture ou de la fermeture d'objets et de l'insertion ou de l'obtention de messages.

Les contrôles de sécurité effectués pour une application standard sont les suivants:

Connexion au gestionnaire de files d'attente (appels MQCONN ou MQCONNX)
C'est la première fois que l'application est associée à un gestionnaire de files d'attente particulier. Le gestionnaire de files d'attente interroge l'environnement d'exploitation pour découvrir l'ID utilisateur associé à l'application. IBM MQ vérifie ensuite que l'ID utilisateur est autorisé à se connecter au gestionnaire de files d'attente et conserve l'ID utilisateur pour des vérifications ultérieures.

Les utilisateurs n'ont pas besoin de se connecter à IBM MQ; IBM MQ suppose que les utilisateurs se sont connectés au système d'exploitation sous-jacent et qu'ils ont été authentifiés par ce système.

Ouverture de l'objet (appels MQOPEN ou MQPUT1)
Les objets IBM MQ sont accessibles en ouvrant l'objet et en exécutant des commandes sur celui-ci. Toutes les vérifications de ressources sont effectuées lorsque l'objet est ouvert, plutôt que lorsqu'il est réellement consulté. Cela signifie que la demande **MQOPEN** doit spécifier le type d'accès requis (par exemple, si l'utilisateur souhaite uniquement parcourir l'objet ou effectuer une mise à jour comme placer des messages dans une file d'attente).

IBM MQ vérifie la ressource nommée dans la demande **MQOPEN** . Pour un alias ou un objet de file d'attente éloignée, l'autorisation utilisée est celle de l'objet lui-même, et non celle de la file d'attente dans laquelle l'alias ou la file d'attente éloignée est résolu. Cela signifie que l'utilisateur n'a pas besoin de droits pour y accéder. Limitez les droits de création de files d'attente aux utilisateurs privilégiés. Si vous ne le faites pas, les utilisateurs peuvent ignorer le contrôle d'accès normal simplement en créant un alias. Si une file d'attente éloignée est référencée explicitement avec les noms de file d'attente et

de gestionnaire de files d'attente, la file d'attente de transmission associée au gestionnaire de files d'attente éloignées est vérifiée.

Les droits d'accès à une file d'attente dynamique sont basés sur ceux de la file d'attente modèle dont elle est dérivée, mais ne sont pas nécessairement les mêmes. Ceci est décrit dans la remarque «1», à la page 142.

L'ID utilisateur utilisé par le gestionnaire de files d'attente pour les contrôles d'accès est l'ID utilisateur obtenu à partir de l'environnement d'exploitation de l'application connectée au gestionnaire de files d'attente. Une application dûment autorisée peut émettre un appel **MQOPEN** en spécifiant un autre ID utilisateur ; des vérifications de contrôle d'accès sont ensuite effectuées sur l'autre ID utilisateur. Cela ne modifie pas l'ID utilisateur associé à l'application, mais uniquement celui utilisé pour les vérifications de contrôle d'accès.

Insertion et obtention de messages (appels MQPUT ou MQGET)
Aucune vérification de contrôle d'accès n'est effectuée.

Fermeture de l'objet (MQCLOSE)
Aucune vérification de contrôle d'accès n'est effectuée, sauf si **MQCLOSE** entraîne la suppression d'une file d'attente dynamique. Dans ce cas, il est vérifié que l'ID utilisateur est autorisé à supprimer la file d'attente.

Abonnement à une rubrique (MQSUB)
Lorsqu'une application s'abonne à une rubrique, elle spécifie le type d'opération qu'elle doit effectuer. Il s'agit soit de créer un nouvel abonnement, soit de modifier un abonnement existant, soit de reprendre un abonnement existant sans le modifier. Pour chaque type d'opération, le gestionnaire de files d'attente vérifie que l'ID utilisateur associé à l'application est autorisé à effectuer l'opération.

Lorsqu'une application s'abonne à une rubrique, les vérifications des droits d'accès sont effectuées sur les objets de rubrique qui se trouvent dans l'arborescence de rubriques au niveau ou au-dessus du point de l'arborescence de rubriques auquel l'application s'est abonnée. Les vérifications des droits d'accès peuvent impliquer des vérifications sur plusieurs objets de rubrique.

L'ID utilisateur utilisé par le gestionnaire de files d'attente pour les vérifications des droits d'accès est l'ID utilisateur obtenu auprès du système d'exploitation lorsque l'application se connecte au gestionnaire de files d'attente.

Le gestionnaire de files d'attente effectue des vérifications des droits d'accès sur les files d'attente d'abonné, mais pas sur les files d'attente gérées.

ALW Comment le contrôle d'accès est implémenté par IBM MQ sur AIX, Linux, and Windows

IBM MQ utilise les services de sécurité fournis par le système d'exploitation sous-jacent, à l'aide du gestionnaire des droits d'accès aux objets. IBM MQ fournit des commandes pour créer et gérer des listes de contrôle d'accès.

Une interface de contrôle d'accès appelée Interface de service d'autorisation fait partie de IBM MQ. IBM MQ fournit une implémentation d'un gestionnaire de contrôle d'accès (conforme à l'interface de service d'autorisation) appelé *gestionnaire des droits d'accès aux objets (OAM)*. Elle est automatiquement installée et activée pour chaque gestionnaire de files d'attente que vous créez, sauf indication contraire (comme décrit dans «Prévention des contrôles d'accès de sécurité sur les systèmes AIX, Linux, and Windows», à la page 375). La méthode d'accès aux objets (OAM) peut être remplacée par tout composant écrit par un utilisateur ou un fournisseur conforme à l'interface de service d'autorisation.

La méthode d'accès aux objets (OAM) exploite les fonctions de sécurité du système d'exploitation sous-jacent, à l'aide des ID utilisateur et de groupe du système d'exploitation. Les utilisateurs ne peuvent accéder aux objets IBM MQ que s'ils disposent des droits appropriés. «Contrôle de l'accès aux objets à l'aide de la méthode d'accès aux objets (OAM) sous AIX, Linux, and Windows», à la page 365 décrit comment accorder et révoquer ces droits.

La méthode d'accès aux objets (OAM) gère une liste de contrôle d'accès (ACL) pour chaque ressource qu'elle contrôle. Les données d'autorisation sont stockées dans une file d'attente locale appelée SYSTEM.AUTH.DATA.QUEUE. L'accès à cette file d'attente est limité aux utilisateurs du groupe mqm, ainsi

qu'à Windows, aux utilisateurs du groupe Administrateurs et aux utilisateurs connectés avec l'ID SYSTEM. L'accès utilisateur à la file d'attente ne peut pas être modifié.

IBM MQ fournit des commandes pour créer et gérer des listes de contrôle d'accès. Pour plus d'informations sur ces commandes, voir «Contrôle de l'accès aux objets à l'aide de la méthode d'accès aux objets (OAM) sous AIX, Linux, and Windows», à la page 365.

IBM MQ transmet à la méthode d'accès aux objets (OAM) une demande contenant un principal, un nom de ressource et un type d'accès. La méthode d'accès aux objets (OAM) accorde ou rejette l'accès en fonction de la liste de contrôle d'accès qu'elle gère. IBM MQ suit la décision de la méthode d'accès aux objets (OAM) ; si la méthode d'accès aux objets (OAM) ne peut pas prendre de décision, IBM MQ n'autorise pas l'accès.

ALW Identification de l'ID utilisateur sous AIX, Linux, and Windows

Le gestionnaire des droits d'accès aux objets identifie le principal qui demande l'accès à une ressource. L'ID utilisateur utilisé comme principal varie en fonction du contexte.

Le gestionnaire des droits d'accès aux objets (OAM) doit pouvoir identifier qui demande l'accès à une ressource particulière. IBM MQ utilise le terme *principal* pour désigner cet identificateur. Le principal est établi lorsque l'application se connecte pour la première fois au gestionnaire de files d'attente ; il est déterminé par le gestionnaire de files d'attente à partir de l'ID utilisateur associé à l'application de connexion. (Si l'application émet des appels XA sans se connecter au gestionnaire de files d'attente, l'ID utilisateur associé à l'application qui émet l'appel xa_open est utilisé pour les vérifications des droits d'accès par le gestionnaire de files d'attente.)

Sur les systèmes AIX and Linux , les routines d'autorisation vérifient l'ID utilisateur réel (loggedin) ou l'ID utilisateur effectif associé à l'application. L'ID utilisateur vérifié peut dépendre du type de liaison. Pour plus de détails, voir Services optionnels.

IBM MQ propage l'ID utilisateur reçu du système dans l'en-tête de message (structure MQMD) de chaque message en tant qu'identification de l'utilisateur. Cet identificateur fait partie des informations de contexte de message et est décrit dans «Droits de contexte sur AIX, Linux, and Windows», à la page 422. Les applications ne peuvent pas modifier ces informations sauf si elles ont été autorisées à modifier les informations de contexte.

ALW *Principaux et groupes sous AIX, Linux, and Windows*

Les principaux peuvent appartenir à des groupes. En accordant l'accès aux ressources à des groupes plutôt qu'à des individus, vous pouvez réduire la quantité d'administration requise. Les listes de contrôle d'accès (ACL) sont basées à la fois sur les groupes et les ID utilisateur.

Par exemple, vous pouvez définir un groupe composé d'utilisateurs qui souhaitent exécuter une application particulière. Les autres utilisateurs peuvent avoir accès à toutes les ressources dont ils ont besoin en ajoutant leur ID utilisateur au groupe approprié.

Ce processus de définition et de gestion de groupes est décrit pour des plateformes particulières:

- AIX Création et gestion de groupes sur AIX
- Linux Création et gestion de groupes sur Linux
- Windows Création et gestion de groupes sur Windows

Un principal peut appartenir à plusieurs groupes (son ensemble de groupes). Il dispose de l'ensemble des droits accordés à chaque groupe de son groupe. Ces droits étant mis en cache, les modifications apportées à l'appartenance au groupe du principal ne sont pas reconnues tant que le gestionnaire de files d'attente n'est pas redémarré, sauf si vous émettez la commande MQSC **REFRESH SECURITY** (ou son équivalent PCF).

Systèmes AIX and Linux

Les listes de contrôle d'accès (ACL) sont basées sur les ID utilisateur et les groupes et vous pouvez utiliser l'un ou l'autre pour l'autorisation en définissant l'attribut **SecurityPolicy** sur la valeur appropriée, comme décrit dans la strophe de service du fichier qm.ini.

Vous pouvez utiliser le *modèle basé sur l'utilisateur* pour l'autorisation, ce qui vous permet d'utiliser à la fois des utilisateurs et des groupes. Toutefois, lorsque vous spécifiez un utilisateur dans la commande setmqaut , les nouveaux droits s'appliquent à cet utilisateur seul et non aux groupes auxquels cet utilisateur appartient. Pour plus d'informations, voir «Droits utilisateur OAM sur AIX and Linux», à la page 365.

Lorsque vous utilisez le *modèle basé sur un groupe* pour l'autorisation, le groupe principal auquel appartient l'ID utilisateur est inclus dans la liste de contrôle d'accès. L'ID utilisateur individuel n'est pas inclus et des droits sont accordés à tous les membres de ce groupe. Pour cette raison, sachez que vous pouvez modifier par inadvertance les droits d'un principal en modifiant les droits d'un autre principal du même groupe.

Tous les utilisateurs sont nominalement affectés au groupe d'utilisateurs par défaut `personne` et, par défaut, aucune autorisation n'est accordée à ce groupe. Vous pouvez modifier l'autorisation dans le groupe `personne` pour accorder l'accès aux ressources IBM MQ à des utilisateurs sans autorisation spécifique.

A partir de IBM MQ 9.3.0, vous pouvez utiliser l'option `UserExternal` de l'attribut **SecurityPolicy** pour créer un nom d'utilisateur du système d'exploitation. Si vous créez un nom d'utilisateur de système d'exploitation, cet utilisateur est considéré comme n'appartenant à aucun groupe, à l'exception du groupe `nobody` . Pour plus d'informations sur cette option, voir crtmqm et la section de service du fichier qm.ini.

Ne définissez pas d'ID utilisateur avec la valeur `UNKNOWN`. La valeur `UNKNOWN` est utilisée lorsqu'un ID utilisateur est trop long. Par conséquent, les ID utilisateur arbitraires utilisent les droits d'accès de `UNKNOWN`.

Pour plus d'informations sur l'utilisation de LDAP, voir «Définition des autorisations», à la page 428 .

Les ID utilisateur peuvent contenir jusqu'à 12 caractères et les noms de groupe jusqu'à 12 caractères.

Systèmes Windows

Les listes de contrôle d'accès sont basées sur les ID utilisateur et les groupes. Les vérifications sont les mêmes que pour AIX and Linux. Vous pouvez avoir différents utilisateurs sur différents domaines avec le même ID utilisateur. IBM MQ permet aux ID utilisateur d'être qualifiés par un nom de domaine afin que ces utilisateurs puissent disposer de différents niveaux d'accès.

Le nom de groupe peut éventuellement inclure un nom de domaine, spécifié dans les formats suivants:

```
GroupName@domain domain_name\group_name
```

Les groupes globaux sont vérifiés par la méthode d'accès aux objets (OAM) dans deux cas uniquement:

1. La section de sécurité du gestionnaire de files d'attente inclut le paramètre: `GroupModel=GlobalGroups`. Voir Sécurisation.
2. Le gestionnaire de files d'attente utilise un autre groupe d'accès de sécurité. Voir **crtmqm** .

Les ID utilisateur peuvent contenir jusqu'à 20 caractères, les noms de domaine jusqu'à 15 caractères et les noms de groupe jusqu'à 64 caractères.

La méthode d'accès aux objets (OAM) vérifie d'abord la base de données de sécurité locale, puis la base de données du domaine principal, et enfin la base de données des domaines de confiance. Le premier ID utilisateur rencontré est utilisé par la méthode d'accès aux objets (OAM) pour la vérification. Chacun de ces ID utilisateur peut avoir des appartenances de groupe différentes sur un ordinateur particulier.

Certaines commandes de contrôle (par exemple, **crtmqm**) modifient les droits sur les objets IBM MQ à l'aide du gestionnaire des droits d'accès aux objets (OAM). La méthode d'accès aux objets (OAM) recherche les bases de données de sécurité dans l'ordre indiqué dans le paragraphe précédent afin de déterminer les droits d'accès pour un ID utilisateur particulier. Par conséquent, les droits d'accès déterminés par la méthode d'accès aux objets (OAM) peuvent remplacer le fait qu'un ID utilisateur soit membre du groupe mqm local. Par exemple, si vous émettez la commande **crtmqm** à partir d'un ID utilisateur authentifié par un contrôleur de domaine qui est membre du groupe mqm local via un groupe global, la commande échoue si le système possède un utilisateur local du même nom qui ne fait pas partie du groupe mqm local.

Pour plus d'informations sur la définition de l'attribut **SecurityPolicy** sur Windows, voir Strophe de service du fichier qm.ini.

Windows *Identificateurs de sécurité (SID) Windows*

IBM MQ on Windows utilise le SID où il est disponible. Si un SID Windows n'est pas fourni avec une demande d'autorisation, IBM MQ identifie l'utilisateur en fonction du nom d'utilisateur uniquement, mais cela peut entraîner l'octroi de droits d'accès incorrects.

Sur les systèmes Windows , l'identificateur de sécurité (SID) est utilisé pour compléter l'ID utilisateur. Le SID contient des informations qui identifient les détails complets du compte utilisateur sur la base de données du gestionnaire de compte de sécurité Windows (SAM) dans laquelle l'utilisateur est défini. Lorsqu'un message est créé sur IBM MQ for Windows, IBM MQ stocke le SID dans le descripteur de message. Lorsque IBM MQ on Windows effectue des vérifications d'autorisation, il utilise le SID pour interroger les informations complètes de la base de données SAM. (La base de données SAM dans laquelle l'utilisateur est défini doit être accessible pour que cette requête aboutisse.)

Par défaut, si un SID Windows n'est pas fourni avec une demande d'autorisation, IBM MQ identifie l'utilisateur en se basant uniquement sur le nom d'utilisateur. Pour ce faire, il effectue des recherches dans les bases de données de sécurité dans l'ordre suivant:

1. Base de données de sécurité locale
2. Base de données de sécurité du domaine principal
3. Base de données de sécurité des domaines de confiance

Si le nom d'utilisateur n'est pas unique, des droits IBM MQ incorrects peuvent être accordés. Pour éviter ce problème, incluez un SID dans chaque demande d'autorisation ; le SID est utilisé par IBM MQ pour établir les données d'identification de l'utilisateur.

Pour indiquer que toutes les demandes d'autorisation doivent inclure un SID, utilisez **regedit**. Définissez SecurityPolicy sur NTSIDsRequired.

ALW Droits d'utilisateur de remplacement sous AIX, Linux, and Windows

Vous pouvez indiquer qu'un ID utilisateur peut utiliser les droits d'un autre utilisateur lors de l'accès à un objet IBM MQ . Il s'agit des *droits d'accès utilisateur de remplacement*, que vous pouvez utiliser sur n'importe quel objet IBM MQ .

Les droits d'utilisateur de remplacement sont essentiels lorsqu'un serveur reçoit des demandes d'un programme et souhaite s'assurer que le programme dispose des droits requis pour la demande. Le serveur peut disposer des droits requis, mais il doit savoir si le programme dispose des droits requis pour les actions qu'il a demandées.

Par exemple, supposons qu'un programme serveur s'exécutant sous l'ID utilisateur PAYSERV extrait un message de demande d'une file d'attente qui a été placée dans la file d'attente par l'ID utilisateur USER1. Lorsque le programme serveur obtient le message de demande, il traite la demande et insère la réponse dans la file d'attente de réponse spécifiée dans le message de demande. Au lieu d'utiliser son propre ID utilisateur (PAYSERV) pour autoriser l'ouverture de la file d'attente de réponse, le serveur peut spécifier un ID utilisateur différent, dans ce cas, USER1. Dans cet exemple, vous pouvez utiliser les droits d'utilisateur de remplacement pour contrôler si PAYSERV est autorisé à spécifier USER1 comme ID utilisateur de remplacement lorsqu'il ouvre la file d'attente de réponse.

L'ID utilisateur de remplacement est indiqué dans la zone **AlternateUserId** du descripteur d'objet.

Linux Résolution de certains problèmes d'appartenance à un groupe sur Linux

Certains systèmes sont lents à renvoyer des informations de groupe via la série normale d'appels API de système d'exploitation **getgrent** et si votre entreprise dispose de milliers de groupes à rechercher, en recherchant les groupes dans lesquels se trouve l'utilisateur `mqm` , la réponse lente peut entraîner un dépassement du délai d'attente du gestionnaire de files d'attente interne. Pour contourner ce problème, il existe une autre API de système d'exploitation.

Pour utiliser l'API alternative la plus rapide et renvoyer tous les groupes à partir d'un seul appel, définissez la variable d'environnement MQS_GETGROUPLIST_API.

Il se peut que vous ayez reçu une erreur RC2035 lors de l'octroi de l'accès de connexion au groupe secondaire de l'utilisateur et que l'activation de la variable MQS_GETGROUPLIST_API atténue le problème.

IBM MQ utilise ensuite l'API **getgrouplist** à la place de l'API **getgrent** .

Pour activer **getgrouplist**:

1. Arrêter le gestionnaire de files d'attente
2. Emettez la commande d'exportation MQS_GETGROUPLIST_API=1
3. Redémarrez le gestionnaire de files d'attente

Réessayez le scénario qui a échoué et si votre problème a été résolu, vous pouvez envisager de modifier le fichier `.bashrc / .profile` pour l'utilisateur `mqm` afin d'ajouter cette variable d'environnement ou d'ajouter la variable d'environnement dans le script que vous utilisez pour démarrer le gestionnaire de files d'attente.

Si votre système fusionne des informations d'utilisateur ou de groupe pour le système d'exploitation à partir de plusieurs référentiels tels que NIS ou LDAP, assurez-vous que le groupe ou l'ID utilisateur est cohérent dans tous les référentiels, y compris le référentiel local, car ils sont utilisés pour installer et définir les droits d'accès au niveau du système d'exploitation.

ALW Droits de contexte sur AIX, Linux, and Windows

Le contexte est une information qui s'applique à un message particulier et qui est contenue dans le descripteur de message, MQMD, qui fait partie du message. Les applications peuvent spécifier les données contextuelles lorsqu'un appel `MQOPEN` ou `MQPUT` est effectué.

Les informations contextuelles comportent deux sections :

La section d'identité
D'où vient le message. Il se compose des zones `UserIdentifier`, `AccountingTokenet` `ApplIdentityData` .

Section d'origine
D'où provient le message et quand il a été placé dans la file d'attente. Il se compose des zones `PutApplType`, `PutApplName`, `PutDate`, `PutTimeet` `ApplOriginData` .

Les applications peuvent spécifier les données contextuelles lorsqu'un appel `MQOPEN` ou `MQPUT` est effectué. Ces données peuvent être générées par l'application, transmises à partir d'un autre message ou générées par le gestionnaire de files d'attente par défaut. Par exemple, les données contextuelles peuvent être utilisées par les programmes serveur pour vérifier l'identité du demandeur, en vérifiant si le message provient d'une application s'exécutant sous un ID utilisateur autorisé.

Un programme serveur peut utiliser le `UserIdentifier` pour déterminer l'ID utilisateur d'un autre utilisateur. Vous utilisez l'autorisation de contexte pour contrôler si l'utilisateur peut spécifier l'une des options de contexte dans un appel `MQOPEN` ou `MQPUT1` .

Voir Contrôle des informations de contexte pour plus d'informations sur les options de contexte et MQMD-Descripteur de message pour obtenir des descriptions des zones de descripteur de message relatives au contexte.

Implémentation du contrôle d'accès dans les exits de sécurité

Vous pouvez implémenter le contrôle d'accès dans un exit de sécurité à l'aide de MCAUserIdentifier ou du gestionnaire des droits d'accès aux objets.

MCAUserIdentifier

Chaque instance d'un canal en cours est associée à une structure de définition de canal, MQCD. Les valeurs initiales des zones de MQCD sont déterminées par la définition de canal créée par un administrateur IBM MQ . En particulier, la valeur initiale de l'une des zones, *MCAUserIdentifier*, est déterminée par la valeur du paramètre MCAUSER dans la commande DEFINE CHANNEL ou par l'équivalent de MCAUSER si la définition de canal est créée d'une autre manière.

La structure MQCD est transmise à un programme d'exit de canal lorsqu'elle est appelée par un agent MCA. Lorsqu'un exit de sécurité est appelé par un agent MCA, l'exit de sécurité peut modifier la valeur de *MCAUserIdentifier*, en remplaçant toute valeur spécifiée dans la définition de canal.

Multi Sous Multiplateformes, sauf si la valeur de *MCAUserIdentifier* est vide, le gestionnaire de files d'attente utilise la valeur de *MCAUserIdentifier* comme ID utilisateur pour les vérifications des droits d'accès lorsqu'un agent MCA tente d'accéder aux ressources du gestionnaire de files d'attente après s'être connecté au gestionnaire de files d'attente. Si la valeur de *MCAUserIdentifier* est vide, le gestionnaire de files d'attente utilise à la place l'ID utilisateur par défaut de l'agent MCA. Cela s'applique aux canaux RCVR, RQSTR, CLUSRCVR et SVRCONN. Pour l'envoi d'agents MCA, l'ID utilisateur par défaut est toujours utilisé pour les vérifications des droits d'accès, même si la valeur de *MCAUserIdentifier* n'est pas vide.

z/OS Sous z/OS, le gestionnaire de files d'attente peut utiliser la valeur de *MCAUserIdentifier* pour les vérifications des droits d'accès, à condition qu'elle ne soit pas vide. Pour la réception des MCM et des MCM de connexion au serveur, le fait que le gestionnaire de files d'attente utilise ou non la valeur de *MCAUserIdentifier* pour les vérifications des droits d'accès dépend des éléments suivants:

- Valeur du paramètre PUTAUT dans la définition de canal
- Profil RACF utilisé pour les vérifications
- Niveau d'accès de l'ID utilisateur de l'espace adresse de l'initiateur de canal au profil RESLEVEL

Pour l'envoi des MCM, il dépend des éléments suivants:

- Indique si l'agent MCA émetteur est un appelant ou un répondeur
- Niveau d'accès de l'ID utilisateur de l'espace adresse de l'initiateur de canal au profil RESLEVEL

L'ID utilisateur stocké par un exit de sécurité dans *MCAUserIdentifier* peut être acquis de différentes manières. Voici quelques exemples :

- A condition qu'il n'y ait pas d'exit de sécurité à l'extrémité client d'un canal MQI, un ID utilisateur associé à l'application client IBM MQ est transmis de l'agent MCA de connexion client à l'agent MCA de connexion serveur lorsque l'application client émet un appel MQCONN.L'agent MCA de connexion serveur stocke cet ID utilisateur dans la zone *RemoteUserIdentifier* de la structure de définition de canal, MQCD. Si la valeur de *MCAUserIdentifier* est vide à ce stade, l'agent MCA stocke le même ID utilisateur dans *MCAUserIdentifier*. Si l'agent MCA ne stocke pas l'ID utilisateur dans *MCAUserIdentifier*, un exit de sécurité peut le faire ultérieurement en affectant à *MCAUserIdentifier* la valeur *RemoteUserIdentifier*.

  Si l'ID utilisateur qui provient du système client entre dans un nouveau domaine de sécurité et n'est pas valide sur le système serveur, l'exit de sécurité peut remplacer l'ID utilisateur par un ID utilisateur valide et stocker l'ID utilisateur remplacé dans *MCAUserIdentifier*.
- L'ID utilisateur peut être envoyé par l'exit de sécurité partenaire dans un message de sécurité.

Sur un canal de transmission de messages, un exit de sécurité appelé par l'agent MCA émetteur peut envoyer l'ID utilisateur sous lequel l'agent MCA émetteur s'exécute. Un exit de sécurité appelé par l'agent MCA récepteur peut ensuite stocker l'ID utilisateur dans *MCAUserIdentifier*. De même, sur un canal MQI, un exit de sécurité à l'extrémité client du canal peut envoyer l'ID utilisateur associé à l'application IBM MQ MQI client . Un exit de sécurité à l'extrémité serveur du canal peut ensuite stocker l'ID utilisateur dans *MCAUserIdentifier*. Comme dans l'exemple précédent, si l'ID utilisateur n'est pas valide sur le système cible, l'exit de sécurité peut remplacer l'ID utilisateur par un ID utilisateur valide et stocker l'ID utilisateur remplacé dans *MCAUserIdentifier*.

Si un certificat numérique est reçu dans le cadre du service d'identification et d'authentification, un exit de sécurité peut mapper le nom distinctif du certificat à un ID utilisateur valide sur le système cible. Il peut ensuite stocker l'ID utilisateur dans *MCAUserIdentifier*.

- Si TLS est utilisé sur le canal, le nom distinctif (DN) du partenaire est transmis à l'exit dans la zone Ptr SSLPeerNamede MQCD, et le nom distinctif de l'émetteur de ce certificat est transmis à l'exit dans la zone Ptr SSLRemCertIssNamede MQCXP.

Pour plus d'informations sur la zone *MCAUserIdentifier* , la structure de définition de canal, MQCD et la structure de paramètre d'exit de canal, MQCXP, voir Appels d'exit de canal et structures de données. Pour plus d'informations sur l'ID utilisateur qui provient d'un système client sur un canal MQI, voir Contrôle d'accès.

Remarque : Les applications d'exit de sécurité construites avant l'édition de IBM WebSphere MQ 7.1 peuvent nécessiter une mise à jour. Pour plus d'informations, voir Programmes d'exit de sécurité de canal.

Authentification d'utilisateur du gestionnaire des droits d'accès aux objets IBM MQ

Sur les connexions IBM MQ MQI client , les exits de sécurité peuvent être utilisés pour modifier ou créer la structure MQCSP utilisée dans l'authentification d'utilisateur du gestionnaire des droits d'accès aux objets (OAM). Ceci est décrit dans Programmes d'exit de canal pour les canaux de messagerie

Implémentation du contrôle d'accès dans les exits de message

Vous devrez peut-être utiliser un exit de message pour remplacer un ID utilisateur par un autre.

Prenons l'exemple d'une application client qui envoie un message à une application serveur. L'application serveur peut extraire l'ID utilisateur de la zone *UserIdentifier* du descripteur de message et, à condition qu'elle dispose de droits d'utilisateur de remplacement, demander au gestionnaire de files d'attente d'utiliser cet ID utilisateur pour les vérifications des droits d'accès lorsqu'il accède aux ressources IBM MQ pour le compte du client.

Si le paramètre PUTAUT est défini sur CTX (ou ALTMCA on z/OS) dans la définition de canal, l'ID utilisateur dans la zone *UserIdentifier* de chaque message entrant est utilisé pour les vérifications des droits d'accès lorsque l'agent MCA ouvre la file d'attente de destination.

Dans certains cas, lorsqu'un message de rapport est généré, il est placé à l'aide des droits de l'ID utilisateur dans la zone *UserIdentifier* du message à l'origine du rapport. En particulier, les rapports de confirmation à la livraison (COD) et les rapports d'expiration sont toujours soumis à cette autorité.

En raison de ces situations, il peut être nécessaire de remplacer un ID utilisateur par un autre dans la zone *UserIdentifier* lorsqu'un message entre dans un nouveau domaine de sécurité. Ceci peut être réalisé par un exit de message à l'extrémité réceptrice du canal. Vous pouvez également vous assurer que l'ID utilisateur dans la zone *UserIdentifier* d'un message entrant est défini dans le nouveau domaine de sécurité.

Si un message entrant contient un certificat numérique pour l'utilisateur de l'application qui a envoyé le message, un exit de message peut valider le certificat et mapper le nom distinctif du certificat à un ID utilisateur valide sur le système de réception. Il peut ensuite définir la zone *UserIdentifier* du descripteur de message sur cet ID utilisateur.

S'il est nécessaire qu'un exit de message modifie la valeur de la zone *UserIdentifier* dans un message entrant, il peut être approprié que l'exit de message authentifie l'expéditeur du message en même temps. Pour plus de détails, voir «Mappage d'identité dans les exits de message», à la page 333.

Implémentation du contrôle d'accès dans l'exit d'API et l'exit de croisement d'API

Une API ou un exit de croisement d'API peut fournir des contrôles d'accès pour compléter ceux fournis par IBM MQ. En particulier, l'exit peut fournir un contrôle d'accès au niveau du message. L'exit peut s'assurer qu'une application insère dans une file d'attente, ou extrait d'une file d'attente, uniquement les messages qui répondent à certains critères.

Prenons les exemples suivants :

- Un message contient des informations sur une commande. Lorsqu'une application tente d'insérer un message dans une file d'attente, une API ou un exit de croisement d'API peut vérifier que la valeur totale de la commande est inférieure à une limite prescrite.
- Les messages arrivent dans une file d'attente de destination à partir de gestionnaires de files d'attente éloignées. Lorsqu'une application tente d'extraire un message de la file d'attente, une API ou un exit de croisement d'API peut vérifier que l'expéditeur du message est autorisé à envoyer un message à la file d'attente.

Multi Sécurité des files d'attente de flux

La fonction de files d'attente de diffusion en flux permet à un administrateur de configurer une file d'attente locale (ou modèle) avec une file d'attente secondaire, où sont placés les messages en double, chaque fois qu'un message est inséré dans la file d'attente d'origine. Il existe deux aspects à prendre en compte en ce qui concerne les droits de diffusion en file d'attente.

Droit de configuration d'une file d'attente pour la diffusion en flux de messages en double

Si vous souhaitez activer la diffusion en flux des messages en double d'une file d'attente vers une file d'attente secondaire, vous devez disposer des droits nécessaires. Pour pouvoir configurer l'attribut **STREAMQ** d'une file d'attente, vous devez disposer des droits suivants:

1. Les droits CHG de la file d'attente pour laquelle ils modifient l'attribut **STREAMQ** pour
2. Droits CHG de la file d'attente dans laquelle vous souhaitez placer les messages de duplication

La combinaison de ces deux vérifications de droits lors de la configuration garantit qu'un utilisateur, qui ne dispose que des droits CHG sur la file d'attente d'origine, ne peut pas placer des messages dans une autre file d'attente sur laquelle il ne dispose pas de droits.

Droit d'ouverture de la ou des files d'attente et d'insertion de messages

Lorsqu'une application ouvre une file d'attente qui a été configurée avec une file d'attente secondaire, via son attribut **STREAMQ** , il est vérifié que l'utilisateur de l'application dispose des droits d'accès PUT sur la file d'attente d'origine.

Remarque : Aucun contrôle de droits supplémentaire n'est effectué pour l'utilisateur de l'application dans la file d'attente secondaire, qui est similaire au modèle de droits utilisé pour les files d'attente alias.

Les applications qui consomment des messages à partir de la file d'attente d'origine ou de la file d'attente secondaire requièrent des droits d'accès GET ou BROWSE, uniquement sur la file d'attente à partir de laquelle elles consomment des messages.

Aucune vérification supplémentaire des droits d'accès n'est effectuée au moment de la mise en oeuvre ou de l'obtention.

Exemple

L'exemple suivant montre les droits corrects définis pour permettre à l'utilisateur `admin` de configurer une file d'attente d'origine, INQUIRIES.QUEUE, pour transmettre ses messages en double à la

file d'attente locale ANALYTICS.QUEUE, mais empêchant admin de dupliquer des messages dans PURCHASES.QUEUE:

```
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)
```

L'utilisateur admin peut alors exécuter la commande suivante:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

mais si le même utilisateur émet la commande suivante:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

pour configurer INQUIRIES.QUEUE pour insérer des messages en double dans PURCHASES.QUEUE, ils reçoivent l'erreur suivante:

```
AMQ8135E Non autorisé
```

Avec INQUIRIES.QUEUE configuré pour dupliquer des messages dans ANALYTICS.QUEUE, les enregistrements de droits d'accès suivants sont utilisés pour permettre à une application s'exécutant en tant qu'utilisateur appuser d'insérer des messages dans INQUIRIES.QUEUEet messages en double dans ANALYTICS.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

Remarque : appuser ne requiert pas d'enregistrement de droits d'accès sur ANALYTICS.QUEUE. Les messages en double sont insérés dans la file d'attente par le gestionnaire de files d'attente.

Concepts associés

Files d'attente de flux

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 Autorisation LDAP

Vous pouvez utiliser l'autorisation LDAP pour supprimer la nécessité d'un ID utilisateur local.

Disponibilité de l'autorisation LDAP sur les plateformes prises en charge

L'autorisation LDAP est disponible sur Multiplatforms:

Avertissement :

A partir de la disponibilité générale d' IBM MQ 9.0 , cette fonctionnalité est disponible sur tous les gestionnaires de files d'attente, qu'ils soient nouveaux ou migrés à partir d'une édition antérieure.

Présentation de l'autorisation LDAP

Avec l'autorisation LDAP, les commandes qui gèrent la configuration d'autorisation, telles que **setmqaut** et **DISPLAY AUTHREC**, peuvent traiter les noms distinctifs. Auparavant, les utilisateurs étaient authentifiés en comparant leurs données d'identification avec le nombre maximal de caractères disponibles pour les utilisateurs et les groupes sur le système d'exploitation local.

Avertissement : Si vous avez exécuté la commande **DEFINE AUTHINFO** , vous devez redémarrer le gestionnaire de files d'attente. Si vous ne redémarrez pas le gestionnaire de files d'attente, la commande setmqaut ne renvoie pas le résultat correct.

Si un utilisateur fournit un ID utilisateur plutôt qu'un nom distinctif, l'ID utilisateur est traité. Par exemple, lorsqu'il existe un message entrant sur un canal avec PUTAUT (CTX), les caractères de l'ID utilisateur sont mappés à un nom distinctif LDAP et les vérifications d'autorisation appropriées sont effectuées.

D'autres commandes, telles que **DISPLAY CONN**, continuent d'utiliser et d'afficher la valeur réelle de l'ID utilisateur, même si cet ID utilisateur n'existe pas sur le système d'exploitation local.

Linux AIX Lorsque l'autorisation LDAP est en place, le gestionnaire de files d'attente utilise toujours le modèle utilisateur de sécurité sur les plateformes AIX and Linux , quel que soit l'attribut **SecurityPolicy** dans le fichier qm.ini . Par conséquent, la définition des droits d'accès d'un utilisateur individuel n'affecte que cet utilisateur, et personne d'autre n'appartient à aucun des groupes de cet utilisateur.

Comme pour le modèle de système d'exploitation, un utilisateur dispose toujours des droits combinés qui ont été affectés à la fois à l'individu et à tous les groupes (le cas échéant) auxquels l'utilisateur appartient.

Par exemple, supposons que les enregistrements suivants ont été définis dans un référentiel LDAP.

- Dans la classe **inetOrgPerson** :

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- Dans la classe **groupOfNames** :

```
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"
```

A des fins d'authentification, un gestionnaire de files d'attente utilisant ce serveur LDAP doit avoir été défini de sorte que sa valeur **CONNAUTH** pointe vers un objet **AUTHINFO** de type IDPWLDAP, et dont les attributs de résolution de nom appropriés sont probablement définis comme suit:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

Etant donné cette configuration pour l'authentification, une application peut renseigner la zone CSPUserID , utilisée dans l'appel MQCNO, avec l'un des ensembles de valeurs suivants:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

ou

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

Dans les deux cas, le système peut utiliser les valeurs fournies pour authentifier le contexte de système d'exploitation de " jodoe".

Multi Définition des autorisations

Comment utiliser le nom abrégé ou **USRFIELD** pour définir les autorisations.

L'approche de l'utilisation de plusieurs formats, décrite dans «Autorisation LDAP», à la page 427, se poursuit avec les commandes d'autorisation, avec une extension supplémentaire que shortname ou USRFIELD peut être utilisé de manière non décorée.

La chaîne de caractères spécifie un attribut particulier dans l'enregistrement LDAP lors de la désignation des utilisateurs (principaux) pour l'autorisation.

Important : La chaîne de caractères ne doit pas contenir le caractère = , car ce caractère ne peut pas être utilisé dans un ID utilisateur du système d'exploitation.

Si vous transmettez un nom de principal à la méthode d'accès aux objets (OAM) pour une autorisation qui peut être `shortname`, la chaîne de caractères doit contenir 12 caractères. L'algorithme de mappage tente d'abord de le résoudre en nom distinctif à l'aide de l'attribut SHORTUSR dans sa requête LDAP.

Si cela échoue avec une erreur UNKNOWN_ENTITY ou si la chaîne donnée ne peut pas être un `shortname`, une nouvelle tentative est effectuée à l'aide de l'attribut USRFIELD pour construire la requête LDAP.

Avertissement : Si vous avez exécuté la commande DEFINE AUTHINFO, vous devez redémarrer le gestionnaire de files d'attente. Si vous ne redémarrez pas le gestionnaire de files d'attente, la commande setmqaut ne renvoie pas le résultat correct.

Pour le traitement des autorisations utilisateur, les paramètres de commande setmqaut suivants sont tous équivalents.

Tableau 75. Paramètres d'autorisation de l'utilisateur

Commande	Remarque
`setmqaut -m QM -t qmgr -p jodoe +connect`	Il s'agit d'un nom non qualifié non hiérarchique, résolu via SHORTUSR.
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	Il s'agit également d'un nom non qualifié non hiérarchique, qui se résout via USRFIELD en une même entité.
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	Utilisation d'un attribut nommé.
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	Utilisation d'un autre attribut nommé qui ne doit pas nécessairement être l'un de ceux configurés sur l'objet AUTHINFO.

Vous pouvez utiliser la commande MQSC SET AUTHREC comme alternative à la commande **setmqaut** :

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

ou la commande PCF Set Authority Record (MQCMD_SET_AUTH_REC) avec l'élément MQCACF_PRINCIPAL_ENTITY_NAMES contenant la chaîne:

```
"cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

Lors du traitement des groupes, il n'y a pas d'ambiguïté sur le traitement de `shortname` , car il n'est pas nécessaire d'insérer une forme de nom de groupe en 12 caractères. Par conséquent, il n'existe pas d'équivalent de l'attribut SHORTUSR pour les groupes.

Cela signifie que les exemples de syntaxe décrits dans Tableau 76, à la page 429 sont valides, en supposant que vous avez configuré l'objet AUTHINFO avec les attributs étendus et défini sur:

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

Tableau 76. Paramètres d'autorisation de groupe

Commande	Remarque
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	Utilisation de GRPFIELD pour la résolution
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	Attribution d'un nom à un attribut unique

Tableau 76. Paramètres d'autorisation de groupe (suite)	
Commande	**Remarque**
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	Utilisation du nom distinctif complet

Vous pouvez utiliser la commande MQSC SET AUTHREC comme alternative à la commande **setmqaut** précédente:

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

ou la commande PCF Set Authority Record (MQCMD_SET_AUTH_REC) avec l'élément MQCACF_GROUP_ENTITY_NAMES contenant la chaîne:

```
 "ApplicationGroupA"
```

Important :

Quel que soit le format utilisé pour faire référence à un nom, que ce soit pour un utilisateur ou un groupe, il doit être possible de dériver un nom distinctif unique.

Par exemple, vous ne devez pas avoir deux enregistrements distincts ayant tous deux "`shortu=jodoe`".

Si un seul nom distinctif unique ne peut pas être déterminé, la méthode d'accès aux objets (OAM) renvoie MQRC_UNKNOWN_ENTITY.

Multi Affichage des autorisations

Diverses méthodes d'affichage de l'autorisation des utilisateurs ou des groupes.

Commande dspmqaut

La méthode la plus simple pour afficher les autorisations disponibles pour un utilisateur ou un groupe consiste à utiliser la commande dspmqaut .

Vous pouvez utiliser une requête sur l'une des variantes de syntaxe pour identifier un utilisateur ou un groupe. Notez que la sortie de la commande répète l'identité dans le format indiqué sur la ligne de commande. La sortie ne signale pas le nom distinctif résolu complet.

Exemple :

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

ou

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

commandes dmpmqaut et dmpmqcfg

La commande dmpmqaut et ses équivalents MQSC ou PCF peuvent spécifier le principal ou le groupe dans n'importe lequel des formats pris en charge, comme les tables **setmqaut** décrites dans «Définition

des autorisations», à la page 428. Cependant, contrairement à **dspmqaut**, la commande **dmpmqaut** signale toujours le nom distinctif complet.

```
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
```

De même, la commande dmpmqcfg , qui ne comporte aucun filtrage sur les enregistrements sélectionnés, affiche toujours le nom distinctif complet dans un format qui peut être réexécuté ultérieurement.

```
dmpmqcfg -m QM -x authrec
-----------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi

Autres considérations lors de l'utilisation de l'autorisation LDAP

Brève description des modifications apportées à l'interface MQI (Message Queue Interface) et aux autres commandes MQSC et PCF dont vous devez tenir compte lors de l'utilisation de l'autorisation LDAP de IBM MQ 9.0.0.

ADOPTCTX

Il n'est pas nécessaire que les applications fournissent des informations d'authentification ou que l'attribut ADOPTCTX soit défini sur YES.

Si une application ne s'authentifie pas explicitement ou si **ADOPTCTX** est défini sur NO pour l'objet CONNAUTH actif, le contexte d'identité associé à l'application est extrait de l'ID utilisateur du système d'exploitation.

Lorsque des autorisations doivent être appliquées, ce contexte est mappé à une identité LDAP à l'aide des mêmes règles que pour les commandes setmqaut .

Paramètres d'entrée des appels MQI

MQOPEN, MQPUT1et MQSUB possèdent des structures qui permettent d'indiquer un autre ID utilisateur.

Si ces zones sont utilisées, l'ID utilisateur de 12 caractères est mappé à un nom distinctif en utilisant les mêmes règles que sur les commandes **setmqaut**, **dmpmqaut**et **dspmqaut** .

MQPUT et MQPUT1 permettent également aux programmes disposant des droits appropriés de définir la zone MQMD UserIdentifier . La valeur de cette zone n'est pas contrôlée lors du processus PUT et peut être définie sur n'importe quelle valeur.

Toutefois, comme d'habitude, la valeur **UserIdentifier** peut être utilisée pour l'autorisation à des étapes ultérieures du traitement des messages, par exemple lorsque PUTAUT (CTX) est défini sur un canal récepteur.

A ce stade, l'autorisation de l'identificateur sera vérifiée à l'aide de la configuration du gestionnaire de files d'attente de réception-qui peut être LDAP ou basé sur le système d'exploitation.

Paramètres de sortie des appels MQI

Chaque fois qu'un ID utilisateur est fourni à un programme dans une structure MQI, il s'agit de la version de nom abrégé à 12 caractères associée à la connexion.

Par exemple, la valeur **MQAXC.UserId** pour les exits API est le nom abrégé renvoyé par le mappage LDAP.

Autres commandes MQSC et PCF d'administration

Les commandes qui affichent des informations utilisateur dans le statut de l'objet, telles que DISPLAY CONN USERID , renvoient le nom abrégé de 12 caractères associé au contexte. Le nom distinctif complet n'est pas affiché.

Les commandes qui permettent l'assertion d'identités, telles que les règles de mappage CHLAUTH ou les valeurs MCAUSER pour les canaux, peuvent prendre des valeurs jusqu'à la longueur maximale définie pour ces attributs (actuellement 64 caractères).

La syntaxe n'est pas modifiée. Lorsque l'autorisation est requise pour cette identité, elle est mappée en interne à un nom distinctif à l'aide des mêmes règles que pour les commandes **setmqaut**, **dmpmqaut**et **dspmqaut** .

Cela signifie que la valeur MCAUSER sur une définition de canal peut ne pas s'afficher comme la même chaîne que DISPLAY CHSTATUS mais qu'elle fait référence à la même identité.

Exemple :

```
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 affiche la valeur SHORTUSR, *MCAUSER (jodoe)* pour toutes les connexions.

Multi Basculement entre les modèles d'autorisation du système d'exploitation et LDAP

Comment basculer entre les différentes méthodes d'autorisation sur différentes plateformes.

L'attribut CONNAUTH du gestionnaire de files d'attente pointe vers un objet AUTHINFO. Lorsque l'objet est de type IDPWLDAP, un référentiel LDAP est utilisé pour l'authentification.

Vous pouvez maintenant appliquer une méthode d'autorisation à ce même objet, ce qui vous permet de continuer avec l'autorisation basée sur le système d'exploitation ou d'utiliser l'autorisation LDAP

IBM i, AIX and Linux

Linux IBM i AIX

Le gestionnaire de files d'attente peut être permuté à tout moment entre les modèles OS et LDAP. Vous pouvez modifier la configuration et la rendre active à l'aide de la commande REFRESH SECURITY TYPE (CONNAUTH) .

Par exemple, si cet objet a déjà été configuré avec les informations de connexion pour l'authentification:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

Si une modification de la configuration des droits d'accès implique le basculement entre les modèles OS et LDAP, le gestionnaire de files d'attente doit être redémarré pour que la modification soit prise en

compte. Sinon, vous pouvez activer la modification à l'aide de la commande REFRESH SECURITY TYPE (CONNAUTH) .

Règles de traitement

Lorsque vous passez de l'autorisation du système d'exploitation à l'autorisation LDAP, toutes les règles d'autorité du système d'exploitation existantes qui ont été définies deviennent inactives et invisibles.

Les commandes telles que **dmpmqaut** n'affichent pas ces règles de système d'exploitation. De même, lorsque vous revenez de LDAP au système d'exploitation, toutes les autorisations LDAP définies deviennent inactives et invisibles, ce qui restaure les règles du système d'exploitation d'origine.

Si vous souhaitez sauvegarder les définitions d'un gestionnaire de files d'attente pour une raison quelconque, à l'aide de la commande **dmpmqcfg** , cette sauvegarde contiendra uniquement les règles définies pour la méthode d'autorisation en vigueur au moment de la sauvegarde.

Multi Administration LDAP

Présentation de la façon dont chaque plateforme administre LDAP.

Lors de l'utilisation de l'autorisation LDAP, l'appartenance au groupe `mqm` (ou équivalent) dans le système d'exploitation n'est pas si importante. Le fait d'être membre de ce groupe contrôle uniquement si certaines commandes de ligne de commande peuvent être traitées.

En particulier, vous devez faire partie de ce groupe pour exécuter les commandes strmqm et endmqm .

Une fois que le gestionnaire de files d'attente est en cours d'exécution, le compte privilégié est désormais limité. Outre l'ID utilisateur de la personne qui émet la commande **strmqm** , les autres utilisateurs appartenant au groupe `mqm` du système d'exploitation (ou équivalent) n'obtiennent pas de privilèges spéciaux.

Les autorisations des autres utilisateurs sont basées sur les groupes LDAP auxquels ils appartiennent. Une utilisation non qualifiée du nom de groupe `mqm` dans des commandes telles que **setmqaut** n'est pas autorisée à être mappée à un groupe LDAP.

AIX and Linux

Linux AIX

Une fois que le gestionnaire de files d'attente est en cours d'exécution, le seul compte automatiquement privilégié est l'utilisateur réel qui a démarré le gestionnaire de files d'attente.

L'ID `mqm` existe toujours et est utilisé en tant que propriétaire des ressources du système d'exploitation, telles que les fichiers, car `mqm` est l'ID effectif sous lequel le gestionnaire de files d'attente s'exécute. Toutefois, l'utilisateur `mqm` ne pourra pas effectuer automatiquement les tâches d'administration contrôlées par la méthode d'accès aux objets (OAM).

Windows

Windows

Sous Windows, les comptes avec privilèges complets automatiques sont l'utilisateur du système d'exploitation qui a démarré le gestionnaire de files d'attente, ainsi que l'utilisateur exécutant les processus du gestionnaire de files d'attente principaux, tels que MUSR_MQADMIN si le gestionnaire de files d'attente a été démarré en tant que service Windows .

Lors de l'exécution en mode d'autorisation LDAP, Windows se comporte de manière très similaire aux plateformes AIX and Linux . Il traite des noms abrégés de 12 caractères et des noms distinctifs complets.

IBM i

IBM i

Sous IBM i, les comptes avec privilèges automatiques sont ceux qui démarrent le gestionnaire de files d'attente et l'ID QMQM.

Vous avez besoin des deux ID, car l'ID utilisateur qui démarre le gestionnaire de files d'attente n'est requis que pour démarrer le système. Une fois en cours d'exécution, les processus du gestionnaire de files d'attente disposent uniquement des droits QMQM.

Exemple de script permettant de fournir des privilèges MQADMIN

Linux AIX

Comme il est utile qu'un groupe puisse effectuer une administration complète sur un gestionnaire de files d'attente, un exemple de script est fourni sur les plateformes AIX and Linux comme suit:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

Cet exemple utilise deux paramètres:

- Un nom de gestionnaire de files d'attente
- Nom de groupe LDAP

L'exemple traite les commandes setmqaut , en accordant des droits complets sur tous les objets. Il s'agit du même script que celui généré par l'assistant OAM IBM MQ Explorer pour les rôles d'administration. Par exemple, le code démarre:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

Confidentialité des messages

Le chiffrement des messages garantit que le contenu des messages reste confidentiel. Il existe différentes méthodes de chiffrement des messages dans IBM MQ en fonction de vos besoins.

Si vous avez besoin d'une protection des données de bout en bout au niveau de l'application pour votre infrastructure de messagerie point à point, vous pouvez utiliser Advanced Message Security pour chiffrer les messages ou écrire votre propre exit d'API ou exit de croisement d'API.

La solution la plus sécurisée consiste à fournir un chiffrement de bout en bout, en chiffrant un message à partir du point où il est placé par une application, jusqu'au point où il est obtenu par l'application consommatrice. Cette opération peut être effectuée à l'aide de «Planification de Advanced Message Security», à la page 116 (AMS) ou en écrivant votre propre exit d'API ou exit de croisement d'API ; voir «Implémentation de la confidentialité dans les programmes d'exit utilisateur», à la page 482 pour plus d'informations.

Si vous devez chiffrer les messages uniquement lorsqu'ils sont transportés sur un réseau, vous pouvez utiliser TLS ; voir «Protocoles de sécurité TLS dans IBM MQ», à la page 25 pour plus d'informations, ou vous pouvez écrire votre propre exit de sécurité, exit de message ou programmes d'exit d'envoi et de réception pour effectuer le chiffrement.

z/OS Si vous devez chiffrer des messages au repos sur un gestionnaire de files d'attente, vous pouvez utiliser le chiffrement de fichier z/OS sur ce gestionnaire de files d'attente ; voir «Confidentiality for data at rest on IBM MQ for z/OS with data set encryption», à la page 484 pour plus d'informations.

Tâches associées

Connexion de deux gestionnaires de files d'attente via le protocole TLS

Connexion sécurisée d'un client à un gestionnaire de files d'attente

Activation des CipherSpecs

Activez un CipherSpec à l'aide du paramètre **SSLCIPH** dans la commande **DEFINE CHANNEL** ou **ALTER CHANNEL** MQSC.

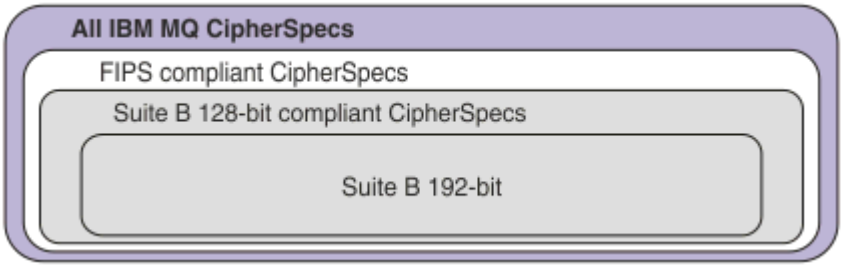
Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

Certains CipherSpecs que vous pouvez utiliser avec IBM MQ sont conformes à la norme FIPS. Certains des CipherSpecs conformes à la norme FIPS sont également conformes à la norme Suite B, alors que d'autres, tels que TLS_RSA_WITH_AES_256_CBC_SHA, ne le sont pas.

Tous les CipherSpecs conformes à la norme Suite B sont également conformes à la norme FIPS. Tous les CipherSpecs conformes à Suite B appartiennent à deux groupes: 128 bits (par exemple, ECDHE_ECDSA_AES_128_GCM_SHA256) et 192 bits (par exemple, ECDHE_ECDSA_AES_256_GCM_SHA384),

Le diagramme suivant illustre la relation entre ces sous-ensembles:



Le produit prend en charge le protocole de sécurité TLS 1.3 sur toutes les plateformes.

Les CipherSpecs que vous pouvez utiliser pour chacune de ces plateformes sont répertoriés dans Tableau 77, à la page 436. Pour plus d'informations sur l'utilisation de ces CipherSpecs, voir «Utilisation de TLS 1.3 dans IBM MQ», à la page 439 et «IBM MQ MQI client et TLS 1.3», à la page 439.

Pour faciliter la configuration et la migration ultérieure, IBM MQ fournit également un ensemble d'alias CipherSpecs. La migration des configurations de sécurité existantes en vue de l'utilisation d'un alias CipherSpec signifie que vous pouvez vous adapter aux ajouts et aux dépréciations de chiffrement sans avoir à effectuer d'autres modifications de configuration invasives à l'avenir. Ces alias CipherSpecs sont répertoriés dans la section Alias CipherSpecs de Tableau 77, à la page 436. Pour plus d'informations sur la migration pour utiliser un alias CipherSpec, voir Migration des configurations de sécurité existantes pour utiliser un alias CipherSpec.

Vous pouvez configurer les CipherSpecs par défaut comme décrit dans «Valeurs CipherSpec par défaut activées dans IBM MQ», à la page 440. Vous pouvez également fournir un autre ensemble de CipherSpecs qui sont activés pour être utilisés avec les canaux sur:

- Multi IBM MQ for Multiplatforms, comme décrit dans «Fourniture d'une liste personnalisée de CipherSpecs commandés et activés sur IBM MQ for Multiplatforms», à la page 448.
- z/OS IBM MQ for z/OS, comme décrit dans «Fourniture d'une liste personnalisée de CipherSpecs commandés et activés sur IBM MQ for z/OS», à la page 449.

Les CipherSpecs obsolètes que vous pouvez réactiver pour les utiliser avec IBM MQ si nécessaire sont répertoriés dans «CipherSpecs obsolètes», à la page 450.

CipherSpecs que vous pouvez utiliser avec la prise en charge de TLS dans IBM MQ

CipherSpecs que vous pouvez utiliser automatiquement avec le gestionnaire de files d'attente IBM MQ sont répertoriés dans le tableau suivant. Lorsque vous demandez un certificat personnel, vous définissez une taille de clé pour la paire de clé publique et de clé privée. La taille de clé utilisée lors de l'établissement de liaison TLS est la taille stockée dans le certificat, sauf si elle est déterminée par le CipherSpec, comme indiqué dans le tableau.

Tableau 77. CipherSpecs que vous pouvez utiliser avec la prise en charge du protocole TLS dans IBM MQ

Prise en charge des plateformes «1», à la page 438	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Algorithme MAC	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 438	Suite B
CipherSpecs alias							
Tous	ANY_TLS13_OR_HIGHER «3», à la page 438 «4», à la page 438	Non disponible	Négocié	Négocié	Négocié	Négocié	Négocié
Tous	ANY_TLS13 «4», à la page 438 «5», à la page 438	Non disponible	TLS 1.3	Négocié	Négocié	Négocié	Négocié
Tous	ANY_TLS12_OR_HIGHER «4», à la page 438 «6», à la page 438	Non disponible	Négocié	Négocié	Négocié	Négocié	Négocié
Tous	ANY_TLS12 «7», à la page 438	Non disponible	TLS 1.2	Négocié	Négocié	Négocié	Négocié
Tous	ANY «8», à la page 438	Non disponible	Négocié	Négocié	Négocié	Négocié	Négocié
CipherSpecs pour TLS 1.3							
Tous	TLS_AES_128_GCM_SHA256	1301	TLS 1.3	GCM	AES-128 avec GCM (128)	Oui	Non
Tous	TLS_AES_256_GCM_SHA384	1302	TLS 1.3	GCM	AES-256 avec GCM (256)	Oui	Non
Tous	TLS_CHACHA20_POLY1305_SHA256	1303	TLS 1.3	POLY1305	CHACHA20 (256)	Non	Non
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 avec CTR (128)	Oui	Non
ALW	TLS_AES_128_CCM_8_SHA256 «10», à la page 438	1305	TLS 1.3	CBC-MAC	AES-128 avec CTR (128)	Oui	Non
CipherSpecs pour TLS 1.2							

Tableau 77. CipherSpecs que vous pouvez utiliser avec la prise en charge du protocole TLS dans IBM MQ (suite)

Prise en charge des plateformes «1», à la page 438	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Algorithme MAC	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 438	Suite B
Tous	TLS_RSA_WITH_AES_128_CBC_SHA256 «9», à la page 438	003C	TLS 1.2	SHA-256	AES (128)	Oui	Non
Tous	TLS_RSA_WITH_AES_256_CBC_SHA256 «9», à la page 438 «11», à la page 438	003D	TLS 1.2	SHA-256	AES (256)	Oui	Non
Tous	TLS_RSA_WITH_AES_128_GCM_SHA256 «9», à la page 438 «12», à la page 438	009C	TLS 1.2	SHA-256 et AEAD GCM	AES (128)	Oui	Non
Tous	TLS_RSA_WITH_AES_256_GCM_SHA384 «9», à la page 438 «11», à la page 438 «12», à la page 438	009D	TLS 1.2	SHA-384 et AEAD GCM	AES (256)	Oui	Non
Tous	ECDHE_ECDSA_AES_128_CBC_SHA256 «9», à la page 438	C023	TLS 1.2	SHA-256	AES (128)	Oui	Non
Tous	ECDHE_ECDSA_AES_256_CBC_SHA384 «9», à la page 438 «11», à la page 438	C024	TLS 1.2	SHA-384	AES (256)	Oui	Non
Tous	ECDHE_RSA_AES_128_CBC_SHA256 «9», à la page 438	C027	TLS 1.2	SHA-256	AES (128)	Oui	Non
Tous	ECDHE_RSA_AES_256_CBC_SHA384 «9», à la page 438 «11», à la page 438	C028	TLS 1.2	SHA-384	AES (256)	Oui	Non
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 «11», à la page 438 «12», à la page 438	C02B	TLS 1.2	SHA-256 et AEAD GCM	AES (SHA384)	Oui	128 bits
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 «11», à la page 438 «12», à la page 438	C02C	TLS 1.2	SHA-384 et AEAD GCM	AES (SHA384)	Oui	192 bits
Tous	ECDHE_RSA_AES_128_GCM_SHA256 «12», à la page 438	C02F	TLS 1.2	SHA-256 et AEAD GCM	AES (128)	Oui	Non
Tous	ECDHE_RSA_AES_256_GCM_SHA384 «11», à la page 438 «12», à la page 438	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	Oui	Non

Tableau 77. CipherSpecs que vous pouvez utiliser avec la prise en charge du protocole TLS dans IBM MQ (suite)

Prise en charge des plateformes «1», à la page 438	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Algorithme MAC	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 438	Suite B

Remarques :

1. Pour obtenir la liste des plateformes couvertes par chaque icône de plateforme, voir Icônes utilisées dans la documentation du produit.
2. Indique si le CipherSpec est certifié FIPS sur une plateforme certifiée FIPS. Voir la rubrique sur la norme FIPS (Federal Information Processing Standards) pour une explication de la norme FIPS.
3. ALW Le CipherSpec alias ANY_TLS13_OR_HIGHER négocie le niveau supérieur de sécurité que l'extrémité distante autorise mais ne se connecte qu'avec le protocole TLS 1.3 ou une version ultérieure.
4. IBM i Pour que vous puissiez utiliser TLS 1.3 ou le CipherSpec ANY sous IBM i, la version du système d'exploitation sous-jacent doit prendre en charge TLS 1.3. Voir System TLS support for TLSv1.3 pour plus d'informations.
5. ALW Le CipherSpec alias ANY_TLS13 représente un sous-ensemble des CipherSpecs acceptables qui utilisent le protocole TLS 1.3, conformément à la liste dans ce tableau pour chaque plateforme.
6. ALW Le CipherSpec alias ANY_TLS12_OR_HIGHER négocie le niveau supérieur de sécurité que l'extrémité distante autorise mais ne se connecte qu'avec le protocole TLS 1.2 ou une version ultérieure.
7. Le CipherSpec ANY_TLS12 représente un sous-ensemble des CipherSpecs acceptables qui utilisent le protocole TLS 1.2, conformément à la liste dans ce tableau pour chaque plateforme.
8. ALW Le CipherSpec alias ANY négocie le niveau supérieur de sécurité que l'extrémité distante autorise.
9. IBM i Ces CipherSpecs ne sont pas activés sur les systèmes IBM i 7.4 dont la valeur système QSSLCSLCTL a pour valeur *OPSSYS.
10. ALW Ces CipherSpecs utilisent une valeur de contrôle d'intégrité de 8 octets au lieu de 16 octets.
11. Ce CipherSpec ne peut pas être utilisé pour sécuriser une connexion d'IBM MQ Explorer à un gestionnaire de files d'attente, sauf si les fichiers de règles sans restriction appropriés sont appliqués à l'environnement d'exécution Java utilisé par l'explorateur.
12. ALW Suite à une recommandation de GSKit, TLS 1.2 GCM CipherSpecs a une restriction qui signifie qu'après l'envoi d'enregistrements TLS24.5 à l'aide de la même clé de session, la connexion se termine par le message AMQ9288E. Cette restriction GCM est active, quel que soit le mode FIPS utilisé.

    Pour éviter cette erreur, évitez d'utiliser les chiffrements TLS 1.2 GCM , activez la réinitialisation de clé confidentielle ou démarrez votre gestionnaire de files d'attente ou client IBM MQ avec la variable d'environnement GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE définie. Pour les bibliothèques GSKit , vous devez définir cette variable d'environnement des deux côtés de la connexion et l'appliquer aux connexions client à gestionnaire de files d'attente et gestionnaire de files d'attente à gestionnaire de files d'attente. Notez que ce paramètre affecte les clients .NET non gérés, mais pas les clients Java ou .NET gérés. Pour plus d'informations, voir AES-GCM cipher restriction.

    z/OS Cette restriction ne s'applique pas à IBM MQ for z/OS.

Utilisation de TLS 1.3 dans IBM MQ

Le produit prend en charge TLS 1.3 sur toutes les plateformes.

Les gestionnaires de files d'attente créés dans IBM MQ 9.2.0 ou version ultérieure prennent en charge TLS 1.3 par défaut. TLS 1.3 doit être activé pour les gestionnaires de files d'attente migrés à partir de versions antérieures d' IBM MQ . Vous pouvez activer TLS 1.3 sur les gestionnaires de files d'attente migrés en définissant la propriété **AllowTLSV13**=TRUE :

- Multi Pour les gestionnaires de files d'attente IBM MQ for Multiplatforms , éditez le fichier qm.ini et ajoutez la propriété **AllowTLSV13**=TRUE sous la strophe SSL (lien vers

  ```
  SSL:
    AllowTLSV13=TRUE
  ```

- z/OS Pour les gestionnaires de files d'attente IBM MQ for z/OS , éditez le fichier QMINI spécifié dans le JCL de démarrage du gestionnaire de files d'attente et ajoutez la propriété **AllowTLSV13**=TRUE sous la strophe TransportSecurity .

  ```
  TransportSecurity:
      AllowTLSV13=TRUE
  ```

Lorsque TLS 1.3 est activé, et conformément à la spécification TLS 1.3, toute tentative de communication avec un CipherSpecfaible, qu'ils soient activés dans IBM MQ ou non, est rejetée. Les CipherSpecs que TLS 1.3 considère comme faibles sont les CipherSpecs qui répondent à un ou plusieurs des critères suivants:

- Utilise le protocole SSL 3.0 .
- Utilise RC4 ou RC2 comme algorithme de chiffrement.
- A une taille de clé de chiffrement (bit) égale ou inférieure à 112.

Ces restrictions sont signalées par la remarque [3] dans le tableau 1 des CipherSpecsobsolètes.

Si vous devez continuer à utiliser ces CipherSpecs, vous devez désactiver le mode TLS 1.3 :

- ALW Editez le fichier qm.ini du gestionnaire de files d'attente et remplacez la valeur de la propriété **AllowTLSV13** par:

  ```
  SSL:
    AllowTLSV13=FALSE
  ```

- z/OS Editez le fichier QMINI du gestionnaire de files d'attente et modifiez le paramètre de la propriété **AllowTLSV13** comme suit:

  ```
  TransportSecurity:
      AllowTLSV13=FALSE
  ```

IBM MQ MQI client et TLS 1.3

ALW

Lors de l'utilisation de IBM MQ MQI client, la valeur de **AllowTLSV13** est déduite sauf si elle est explicitement spécifiée dans la strophe SSL du fichier mqclient.ini utilisé par l'application.

- Si des CipherSpecs faibles sont activés, **AllowTLSV13** est défini sur FALSE et aucun 1.3 CipherSpecs TLS ne peut être utilisé.
- Sinon, **AllowTLSV13** est défini sur TRUE et les nouveaux CipherSpecs TLS 1.3 CipherSpecs et alias CipherSpecs peuvent être utilisés.

Valeurs CipherSpec par défaut activées dans IBM MQ

Dans la configuration par défaut d'un nouveau gestionnaire de files d'attente IBM MQ , IBM MQ prend en charge les protocoles TLS 1.2 et TLS 1.3 et divers algorithmes de cryptographie à l'aide de CipherSpecs. A des fins de compatibilité, IBM MQ peut également être configuré pour utiliser les protocoles SSL 3.0 et TLS 1.0 et un certain nombre d'algorithmes de cryptographie connus pour être faibles ou vulnérables aux vulnérabilités en matière de sécurité. La liste des CipherSpecs qui sont activés dans la configuration par défaut peut changer en appliquant la maintenance.

Il est possible de configurer IBM MQ pour restreindre ou autoriser l'utilisation de CipherSpecs à l'aide des contrôles suivants:

- Autorisez uniquement les CipherSpecs conformes à la norme FIPS 140-2 à l'aide de SSLFIPS.
- ALW Autorisez uniquement les CipherSpecs conformes à NSA Suite B à l'aide de SUITEB.
- Multi Autorisez une liste personnalisée de CipherSpecs à l'aide de **AllowedCipherSpecs**.
- ALW Autorisez une liste personnalisée de CipherSpecs à l'aide de la variable d'environnement **AMQ_ALLOWED_CIPHERS** .
- ALW Autorisez l'utilisation de CipherSpecs obsolètes à l'aide de **AllowWeakCipher** ou de la variable d'environnement **AMQ_SSL_WEAK_CIPHER_ENABLE** .
- z/OS Autorisez l'utilisation de CipherSpecs obsolètes à l'aide d'instructions de définition de données dans le JCL CHINIT.

Remarque : Si vous spécifiez une liste personnalisée de CipherSpecs à l'aide de **AllowedCipherSpecs** ou **AMQ_ALLOWED_CIPHERS** , cela remplace l'activation des CipherSpecsobsolètes. Notez que lorsque vous utilisez des restrictions NSA Suite B ou FIPS 140-2 en combinaison avec une liste CipherSpec personnalisée, vous devez vous assurer que la liste personnalisée contient uniquement des CipherSpecs autorisés par les paramètres Suite B ou FIPS 140-2.

Concepts associés

«Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49
Cette rubrique fournit des informations sur la façon de choisir les CipherSpecs et les certificats numériques appropriés pour votre règle de sécurité, en soulignant la relation entre les CipherSpecs et les certificats numériques dans IBM MQ.

«CipherSpecs et CipherSuites», à la page 22
Les protocoles de sécurité cryptographique doivent convenir des algorithmes utilisés par une connexion sécurisée. CipherSpecs et CipherSuites définissent des combinaisons spécifiques d'algorithmes.

«Configuration de IBM MQ pour Suite B», à la page 46
IBM MQ peut être configuré pour fonctionner conformément à la norme NSA Suite B sur les plateformes AIX, Linux, and Windows .

«FIPS (Federal Information Processing Standards)», à la page 35
Cette rubrique présente le programme FIPS (Federal Information Processing Standards) Cryptomodule Validation Program du US National Institute of Standards and Technology et les fonctions cryptographiques qui peuvent être utilisées sur les canaux TLS.

Tâches associées

Migration des configurations de sécurité existantes pour utiliser un alias CipherSpe

Référence associée

De la définition d'un canal

ALTER CHANNEL

Modifier, copier et créer un canal

ALW AES-Restriction de chiffrementGCM

Guide des restrictions imposées sur les chiffrements AES-GCM lorsqu'ils sont utilisés pour la cryptographie TLS. Ces restrictions sont imposées par les organisations IETF et NIST et nécessitent que la même clé de session ne soit pas utilisée pour transférer de manière sécurisée plus de 2 enregistrements TLS$^{24.5}$ lors de l'utilisation des chiffrements AES-GCM .

Pour plus d'informations sur ces restrictions, voir RFC 9325 Section 4.4 Limites d'utilisation des clés et RFC 8446 section 5.5.

IBM MQ n'implémente pas directement la fonctionnalité cryptographique. A la place, plusieurs bibliothèques cryptographiques différentes sont utilisées pour fournir les fonctionnalités TLS et Advanced Message Security . Sur les systèmes d'exploitation Windows, Linuxet AIX , la bibliothèque cryptographique utilisée par IBM MQ est IBM Global Security Kit (GSKit). Pour les applications, les bibliothèques C et .NET non gérées utilisent GSKit pour la fonctionnalité cryptographique. L'implémentation des algorithmes de chiffrement AES-GCM par GSKit inclut les restrictions spécifiées par le groupe de normes. En outre, ces restrictions sont activées par défaut. Ainsi, la communication TLS IBM MQ , lors de l'utilisation des chiffrements AES-GCM , s'arrête si plus de 2 enregistrements TLS$^{24.5}$ sont transmis à l'aide de la même clé de session.

Remarque : Cette restriction n'est pas présente sur les plateformes IBM i, IBM Z ou IBM MQ for HPE NonStop ou Java/JMS, les applications .NET gérées car des bibliothèques cryptographiques différentes sont utilisées et ces bibliothèques n'ont pas implémenté la même restriction.

Si un canal IBM MQ reste en cours d'exécution pendant une durée suffisante pour que plus de 2 enregistrements TLS$^{24.5}$ soient transmis à l'aide de la même clé de session, la bibliothèque cryptographique sous-jacente met fin à la connexion. Cela provoque l'arrêt du canal et un message d'erreur AMQ9288E est généré. Les applications dont la communication est interrompue de cette manière reçoivent un code retour `MQRC_CONNECTION_BROKEN` de l'opération IBM MQ exécutée.

L'arrêt de la connexion peut être effectué à chaque extrémité de la communication, mais uniquement sur les extrémités qui utilisent GSKit pour la fonctionnalité cryptographique.

Conseils pour l'atténuation de la restriction

Voici quelques options permettant d'empêcher ou de gérer les communications arrêtées en raison de cette restriction:

Utiliser des clients reconnectables

Les applications peuvent être configurées pour tenter automatiquement une reconnexion, en cas d'échec d'une connexion. Cela inclut les connexions qui sont arrêtées en raison de la restriction GCM . Lorsqu'elle est configurée pour la reconnexion, l'application client est restaurée automatiquement à tout point de défaillance et tous les descripteurs permettant d'ouvrir les objets sont restaurés. Cette opération est effectuée sans revenir au code de l'application.

Pour plus d'informations, voir Reconnexion automatique du client.

Définir une valeur de réinitialisation de clé secrète

IBM MQ peut être configuré pour demander une réinitialisation de clé de session après qu'un nombre d'octets configurable a été transféré sur un canal. Une fois cette limite atteinte, IBM MQ demande à la couche cryptographique d'effectuer une réinitialisation de la clé de session, ce qui génère une nouvelle clé de session.

Il est important de noter que la valeur spécifiée est le nombre d'octets transférés, qui est lié à la taille des messages envoyés par IBM MQ. La restriction concerne le nombre d'enregistrements TLS envoyés. Il n'existe pas de mappage direct entre les octets de message et les enregistrements TLS car un enregistrement TLS peut envoyer un nombre maximal d'octets dépendant de l'unité de transmission maximale (MTU) du réseau. Tous les messages envoyés dont la taille est supérieure à cette valeur sont transmis sous la forme de plusieurs enregistrements TLS. La valeur MTU varie d'un réseau à l'autre. En outre, il existe d'autres raisons pour lesquelles un enregistrement TLS peut avoir besoin d'être envoyé en dehors de la transmission de données de message IBM MQ , par exemple des vérifications de pulsation IBM MQ , des alertes TLS, d'autres messages de protocole IBM MQ . Ces

enregistrements TLS supplémentaires sont comptabilisés dans le nombre maximal d'enregistrements TLS, mais ne sont pas comptabilisés dans la valeur de réinitialisation de la clé confidentielle IBM MQ .

La réinitialisation régulière d'une clé de session à l'aide de la réinitialisation de clé secrète peut empêcher l'arrêt du canal en raison de la restriction AES-GCM .

Pour plus d'informations, voir Réinitialisation des clés secrètes SSL et TLS.

Utiliser les spécifications de chiffrement TLS 1.3

Alors que la restriction AES-GCM est toujours présente lors de l'utilisation du protocole TLS 1.3 , le protocole TLS 1.3 prend en charge l'exécution automatique d'une réinitialisation de clé de session sans qu'il soit nécessaire d'interrompre les communications TLS. Cela permet à GSKit de gérer la réinitialisation de la clé de session lorsqu'elle est nécessaire sans qu' IBM MQ n'ait besoin de demander la réinitialisation d'une clé secrète.

Pour plus d'informations, voir Utilisation de TLS 1.3 dans IBM MQ dans «Activation des CipherSpecs», à la page 435.

Désactivation de la restriction AES-GCM

Si nécessaire, la restriction peut être désactivée en définissant la variable d'environnement **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** pour désactiver la restriction AES-GCM . Cela permet d'envoyer n'importe quel nombre d'enregistrements TLS à l'aide de la même clé de session. Si vous choisissez cette atténuation, la variable d'environnement doit être définie à chaque extrémité de la communication qui utilise GSKit pour les communications sécurisées.

Avertissement : Cette option n'est pas recommandée car, après l'envoi de plus de 2 enregistrements TLS[24.5] , les agresseurs peuvent effectuer une analyse sur les enregistrements envoyés afin de déterminer la clé de session utilisée. Une fois la clé de session déterminée, toutes les communications existantes et futures utilisant cette clé de session sont compromises.

CipherSpec dans l'ordre d'établissement de liaison TLS

L'ordre des CipherSpecs est utilisé lors du choix entre plusieurs CipherSpecs possibles, par exemple lors de l'utilisation de l'un des CipherSpecs ANY*.

Lors d'un établissement de liaison TLS, un client et un serveur échangent les CipherSpecs et les protocoles qu'ils prennent en charge dans l'ordre de leurs préférences. Un CipherSpec commun que les deux parties hiérarchisent est choisi et utilisé pour la communication TLS. Lors du choix d'un protocole CipherSpec , la version est également prise en compte, par exemple si un serveur répertorie TLS 1.2 CipherSpecs avant TLS 1.3 CipherSpecs , il donne toujours la priorité à TLS 1.3 tant que le client peut le prendre en charge et qu'il dispose d'un TLS 1.3 CipherSpec commun qui peut être utilisé.

Lorsque IBM MQ est configuré pour TLS, il définit les CipherSpecs dans l'ordre indiqué dans le tableau suivant, de la plus préférée à la moins préférée.

Remarque : Si un CipherSpec n'est pas activé via l'attribut **AllowedCipherSpecs** , il ne sera pas configuré pour être utilisé lors de l'établissement d'une liaison TLS.

Si l'attribut **AllowedCipherSpecs** n'est pas spécifié, une liste par défaut des chiffrements activés, indiquée par le tableau suivant, est utilisée.

Tableau 78. CipherSpecs à partir de IBM MQ 9.2.0

Plateforme	CipherSpec	Protocole	Code hexadécimal	Activé par défaut
Tous	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Oui
Tous	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Oui
Tous	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Oui

Tableau 78. CipherSpecs à partir de IBM MQ 9.2.0 (suite)

Plateforme	CipherSpec	Protocole	Code hexadécimal	Activé par défaut
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	Oui
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	Oui
Tous	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Oui
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	Oui
Tous	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Oui
Tous	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Oui
Tous	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Oui
Tous	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Oui
Tous	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Oui
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	Oui
Tous	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Oui
Tous	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Oui
Tous	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Oui
Tous	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Oui
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	Non
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	Non
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	Non

Tableau 78. CipherSpecs à partir de IBM MQ 9.2.0 (suite)				
Plateforme	**CipherSpec**	**Protocole**	**Code hexadécimal**	**Activé par défaut**
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	Non
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	Non
Tous	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Non
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	Non
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	Non
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	0000	Non
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Non
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Non
IBM i	AES_SHA_US	TLS 1.0	002E	Non
Tous	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Non
Tous	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Non
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	Non
Tous	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Non
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	Non
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	Non
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	Non
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	Non
Tous	TRIPLE_DES_SHA_US	SSL v3	000A	Non
Tous	RC4_SHA_US	SSL v3	0005	Non

Tableau 78. CipherSpecs à partir de IBM MQ 9.2.0 (suite)				
Plateforme	**CipherSpec**	**Protocole**	**Code hexadécimal**	**Activé par défaut**
Tous	RC4_MD5_US	SSL v3	0004	Non
Tous	DES_SHA_EXPORT	SSL v3	0009	Non
Tous	RC4_MD5_EXPORT	SSL v3	0003	Non
Tous	RC2_MD5_EXPORT	SSL v3	0006	Non
Tous	NULL_SHA	SSL v3	0002	Non
Tous	NULL_MD5	SSL v3	0001	Non
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	FEFF	Non
ALW	RC4_56_SHA_EXPORT1024	SSL v3	0064	Non
ALW	DES_SHA_EXPORT1024	SSL v3	0062	Non
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	FEFE	Non

Cette liste a été créée en classant les protocoles avec la liste par défaut fournie par la bibliothèque cryptographique utilisée par IBM MQ sur z/OS et est cohérente sur les plateformes z/OS et réparties.

changer l'ordre

Si un ordre différent est souhaité, un nouvel ordre de CipherSpecs peut être fourni à l'aide de l'attribut **AllowedCipherSpecs** de la strophe SSL sur IBM MQ for Multiplatforms z/OS ou de la strophe TransportSecurity sur IBM MQ for z/OS, avec les règles suivantes:

- Les versions de protocole supérieures sont toujours utilisées, quelle que soit leur position dans la liste.
- Tous les CipherSpecs désactivés sont réactivés s'ils sont fournis dans la liste.
- L'ordre de liste du serveur TLS a une priorité plus élevée que celle du client TLS.
- Lorsque TLS 1.3 est activé, certains CipherSpecs ne sont pas pris en charge.

Par exemple, sous IBM MQ for Multiplatforms, si les éléments suivants sont configurés sur le gestionnaire de files d'attente:

```
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 et sous IBM MQ for z/OS, si les éléments suivants sont configurés sur le gestionnaire de files d'attente:

```
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
```

puis :

- Un client se connectant à ANY_TLS12 utilisera probablement le protocole TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256.
- Un client se connectant à ANY_TLS12_OR_HIGHER utilisera probablement le protocole TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256 (en supposant que le client prend en charge TLS 1.3).

- Un client se connectant avec le protocole TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA utilisera ce CipherSpec.

Versions précédentes de IBM MQ

Avant IBM MQ 9.2.0, l'ordre suivant des CipherSpecs était utilisé:

Tableau 79. CipherSpecs ordre avant IBM MQ 9.2.0

Plateforme	CipherSpec	Protocole	Activé par défaut
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	Non
IBM i	AES_SHA_US	TLS 1.0	Non
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	Non
Tous	RC4_SHA_US	SSL v3	Non
Tous	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	Non
Tous	RC4_MD5_US	SSL v3	Non
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	Non
Tous	TRIPLE_DES_SHA_US	SSL v3	Non
Tous	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	Non
ALW	DES_SHA_EXPORT1024	SSL v3	Non
Tous	RC4_56_SHA_EXPORT1024	SSL v3	Non
Tous	RC4_MD5_EXPORT	SSL v3	Non
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	Non
Tous	RC2_MD5_EXPORT	SSL v3	Non
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	Non
Tous	DES_SHA_EXPORT	SSL v3	Non
Tous	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	Non
Tous	NULL_SHA	SSL v3	Non
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	Non
Tous	NULL_MD5	SSL v3	Non

Tableau 79. CipherSpecs ordre avant IBM MQ 9.2.0 (suite)			
Plateforme	**CipherSpec**	**Protocole**	**Activé par défaut**
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	Non
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	Non
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	Non
Tous	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	Oui
Tous	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	Oui
Tous	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	Non
Tous	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	Oui
Tous	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	Oui
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	Non
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	Non
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	Non
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	Non
Tous	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	Oui
Tous	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	Oui
Tous	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	Oui
Tous	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	Oui
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	Oui
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	Oui
Tous	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	Oui
Tous	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	Oui
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	Non

Tableau 79. CipherSpecs ordre avant IBM MQ 9.2.0 (suite)			
Plateforme	**CipherSpec**	**Protocole**	**Activé par défaut**
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	Non
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	Non
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	Non
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	Oui
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	Oui
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	Oui
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	Oui
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	Oui

Important : Depuis le 23rd juillet 2020, l'attribut AllowedCipherSpecs suivant active uniquement les CipherSpecs actuellement activés par défaut. Toutefois, vous devez vérifier les CipherSpecs activés par l'attribut AllowedCipherSpecs avec les données en cours afin de vous assurer que les CipherSpecs qui sont obsolètes depuis cette date ne sont pas réactivés par inadvertance.

Si vous devez revenir à cet ordre de CipherSpecs, vous pouvez le faire en utilisant la valeur d'attribut de section **AllowedCipherSpecs** SSL/TransportSecurity suivante:

```
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
```

Fourniture d'une liste personnalisée de CipherSpecs commandés et activés sur IBM MQ for Multiplatforms

Multi

Vous pouvez fournir un autre ensemble de CipherSpecs qui sont activés, et dans l'ordre de vos préférences, pour une utilisation avec les canaux IBM MQ , à l'aide de la variable d'environnement ALW **AMQ_ALLOWED_CIPHERS** ou de l'attribut de strophe SSL **AllowedCipherSpecs** du fichier `.ini` . Vous pouvez utiliser ce paramètre pour l'une des raisons suivantes:

- Pour empêcher les programmes d'écoute IBM MQ d'accepter les demandes de démarrage de canal entrantes, sauf s'ils utilisent l'une des CipherSpecsnommées.
- Permet de modifier l'ordre de priorité des CipherSpecs utilisés dans un établissement de liaison TLS.

Cette fonctionnalité peut être utilisée pour contrôler les CipherSpecs qui sont inclus dans les CipherSpecsANY*.

La variable d'environnement **AMQ_ALLOWED_CIPHERS** ou l'attribut de strophe SSL **AllowedCipherSpecs** accepte:

- Nom CipherSpec unique.
- Liste de noms CipherSpec séparés par des virgules à réactiver.

- Valeur spéciale de ALL, représentant tous les CipherSpecs.

Remarque : Vous ne devez pas activer **ALL** CipherSpecs, car cela activera les protocoles SSL 3.0 et TLS 1.0 et un grand nombre d'algorithmes de cryptographie faibles.

Si ce paramètre est configuré, il remplace la liste CipherSpec par défaut et oblige IBM MQ à ignorer les paramètres de dépréciation de chiffrement faible (voir ci-dessous):

- Les programmes d'écoute IBM MQ acceptent uniquement les propositions SSL/TLS qui utilisent l'un des CipherSpecsnommés.
- Les canaux IBM MQ n'autorisent qu'une valeur SSLCIPH vide ou l'un des CipherSpecsnommés.
- La saisie semi-automatique **runmqsc** des valeurs SSLCIPH limite les valeurs de saisie semi-automatique à l'un des noms CipherSpecs.

Par exemple, si vous souhaitez uniquement autoriser les canaux à être définis / modifiés et les programmes d'écoute à accepter ECDHE_RSA_AES_128_GCM_SHA256 ou ECDHE_ECDSA_AES_256_GCM_SHA384 , vous pouvez définir ce qui suit dans le fichier qm.ini :

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

De plus, les CipherSpecs de cette liste seront utilisés pour déterminer la priorité des CipherSpecs utilisés lors de l'établissement d'une liaison TLS. Par exemple, si vous spécifiez une liste de TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 , il est probable que, lors de l'établissement de liaison, TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec sera choisi sur TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec si un client se connecte en spécifiant les deux CipherSpecs, c'est-à-dire un client se connectant à ANY_TLS12.

Notez que les chiffrements utilisés par les canaux AMQP ou MQTT peuvent être restreints à l'aide des paramètres de fichier java.security .

Fourniture d'une liste personnalisée de CipherSpecs commandés et activés sur IBM MQ for z/OS

z/OS

Vous pouvez fournir un autre ensemble de CipherSpecs activés, et dans l'ordre de vos préférences, à utiliser avec les canaux IBM MQ , à l'aide de l'attribut de strophe **AllowedCipherSpecs** TransportSecurity de l' ensemble de données QMINI. Vous pouvez effectuer cette opération pour l'une des raisons suivantes:

- Pour empêcher les programmes d'écoute IBM MQ d'accepter les demandes de démarrage de canal entrantes, sauf s'ils utilisent l'une des CipherSpecsnommées.
- Permet de modifier l'ordre de priorité des CipherSpecs utilisés dans un établissement de liaison TLS.

Vous pouvez utiliser cette fonctionnalité pour contrôler les CipherSpecs qui sont inclus dans les CipherSpecsANY*. L'attribut **AllowedCipherSpecs** accepte:

- Nom CipherSpec unique.
- Liste de noms CipherSpec séparés par des virgules à réactiver.
- Valeur spéciale de ALL, représentant tous les CipherSpecs.

  Remarque : Vous ne devez pas activer **ALL** CipherSpecs, car cela activera les protocoles SSL 3.0 et TLS 1.0 et un grand nombre d'algorithmes de cryptographie faibles. Si vous configurez ce paramètre, il remplace la liste CipherSpec par défaut et oblige IBM MQ à ignorer les paramètres de dépréciation de chiffrement faible ; voir «Activation des CipherSpecs obsolètes sous z/OS», à la page 454.

Les programmes d'écoute IBM MQ acceptent uniquement les propositions SSL/TLS qui utilisent l'un des CipherSpecs et les canaux IBM MQ n'autorisent qu'une valeur SSLCIPH vide ou l'un des CipherSpecsnommés.

Par exemple, si vous souhaitez uniquement autoriser la définition / modification de canaux et que les programmes d'écoute acceptent ECDHE_RSA_AES_128_GCM_SHA256 ou ECDHE_RSA_AES_256_GCM_SHA384 , vous pouvez définir ce qui suit:

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

De plus, les CipherSpecs de cette liste sont utilisés pour déterminer la priorité des CipherSpecs utilisés lors de l'établissement d'une liaison TLS. Par exemple, si vous spécifiez une liste de TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 il est probable que, lors de l'établissement de liaison, le TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec sera choisi sur le TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec si un client se connecte en spécifiant les deux CipherSpecs, c'est-à-dire un client se connectant à ANY_TLS12.

Deprecated CipherSpecs obsolètes

Liste des CipherSpecs obsolètes que vous pouvez utiliser avec IBM MQ si nécessaire.

Les CipherSpecs obsolètes que vous pouvez utiliser avec la prise en charge de TLS dans IBM MQ sont répertoriés dans le tableau suivant.

Tableau 80. CipherSpecs dépréciés que vous pouvez réactiver afin de les utiliser avec IBM MQ

Prise en charge des plateformes «1», à la page 453	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Intégrité des données	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 453	Suite B	Mettre à jour si déprécié
CipherSpecs pour SSL 3.0								
IBM i	AES_SHA_US «3», à la page 453	002F	SSL 3.0	SHA-1	AES (128)	Non	Non	9.0.0.0
Tous	DES_SHA_EXPORT «3», à la page 453 «4», à la page 453 «5», à la page 453	0009	SSL 3.0	SHA-1	DES (56)	Non	Non	9.0.0.0
ALW	DES_SHA_EXPORT1024 «3», à la page 453 «6», à la page 453	0062	SSL 3.0	SHA-1	DES (56)	Non	Non	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA «3», à la page 453	FEFE	SSL 3.0	SHA-1	DES (56)	Non «7», à la page 453	Non	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA «3», à la page 453	FEFF	SSL 3.0	SHA-1	3DES (168)	Non «8», à la page 453	Non	9.0.0.1 et 9.0.1
Tous	NULL_MD5 «3», à la page 453	0001	SSL 3.0	MD5	Aucun	Non	Non	9.0.0.1
Tous	NULL_SHA «3», à la page 453	0002	SSL 3.0	SHA-1	Aucun	Non	Non	9.0.0.1
Tous	RC2_MD5_EXPORT «3», à la page 453 «4», à la page 453 «5», à la page 453	0006	SSL 3.0	MD5	RC2 (40)	Non	Non	9.0.0.0

Tableau 80. CipherSpecs dépréciés que vous pouvez réactiver afin de les utiliser avec IBM MQ (suite)

Prise en charge des plateformes «1», à la page 453	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Intégrité des données	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 453	Suite B	Mettre à jour si déprécié
Tous	RC4_MD5_EXPORT «4», à la page 453 «3», à la page 453	0003	SSL 3.0	MD5	RC4 (40)	Non	Non	9.0.0.0
Tous	RC4_MD5_US «3», à la page 453	0004	SSL 3.0	MD5	RC4 (128)	Non	Non	9.0.0.0
Tous	RC4_SHA_US «3», à la page 453 «5», à la page 453	0005	SSL 3.0	SHA-1	RC4 (128)	Non	Non	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 «3», à la page 453 «6», à la page 453	0064	SSL 3.0	SHA-1	RC4 (56)	Non	Non	9.0.0.0
Tous	TRIPLE_DES_SHA_US «3», à la page 453 «5», à la page 453	000A	SSL 3.0	SHA-1	3DES (168)	Non	Non	9.0.0.1 et 9.0.1
CipherSpecs pour TLS 1.0								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 «3», à la page 453	0006	TLS 1.0	MD5	RC2 (40)	Non	Non	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5«3», à la page 453 «4», à la page 453	0003	TLS 1.0	MD5	RC4 (40)	Non	Non	9.0.0.0
Tous	TLS_RSA_WITH_DES_CBC_SHA «3», à la page 453	0009	TLS 1.0	SHA-1	DES (56)	Non« 9», à la page 453	Non	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 «3», à la page 453	0001	TLS 1.0	MD5	Aucun	Non	Non	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA «3», à la page 453	0002	TLS 1.0	SHA-1	Aucun	Non	Non	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 «3», à la page 453	0004	TLS 1.0	MD5	RC4 (128)	Non	Non	9.0.0.0
z/OS ALW	TLS_RSA_WITH_AES_128_CBC_SHA «10», à la page 453	002F	TLS 1.0	SHA-1	AES (128)	Oui	Non	9.0.5
z/OS ALW	TLS_RSA_WITH_AES_256_CBC_SHA «6», à la page 453 «10», à la page 453	0035	TLS 1.0	SHA-1	AES (256)	Oui	Non	9.0.5
Tous	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	Oui	Non	9.0.0.1 et 9.0.1
CipherSpecs pour TLS 1.2								

Tableau 80. CipherSpecs dépréciés que vous pouvez réactiver afin de les utiliser avec IBM MQ (suite)

Prise en charge des plateformes «1», à la page 453	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Intégrité des données	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 453	Suite B	Mettre à jour si déprécié
ALW	ECDHE_ECDSA_NULL_SHA256 «3», à la page 453	C006	TLS 1.2	SHA-1	Aucun	Non	Non	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 «3», à la page 453	C007	TLS 1.2	SHA-1	RC4 (128)	Non	Non	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 «3», à la page 453	C010	TLS 1.2	SHA-1	Aucun	Non	Non	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 «3», à la page 453	C011	TLS 1.2	SHA-1	RC4 (128)	Non	Non	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL «3», à la page 453	0000	TLS 1.2	Aucun	Aucun	Non	Non	9.0.0.1
Tous	TLS_RSA_WITH_NULL_SHA256 «3», à la page 453	003B	TLS 1.2	SHA-256	Aucun	Non	Non	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 «3», à la page 453	0005	TLS 1.2	SHA-1	RC4 (128)	Non	Non	9.0.0.0
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	Oui	Non	9.0.0.1 et 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	Oui	Non	9.0.0.1 et 9.0.1

Tableau 80. CipherSpecs dépréciés que vous pouvez réactiver afin de les utiliser avec IBM MQ (suite)

Prise en charge des plateformes «1», à la page 453	Nom du CipherSpec	Code hexadécimal	Protocole utilisé	Intégrité des données	Algorithme de chiffrement (bits de chiffrement)	FIPS «2», à la page 453	Suite B	Mettre à jour si déprécié

Remarques :

1. Pour obtenir la liste des plateformes couvertes par chaque icône de plateforme, voir Icônes utilisées dans la documentation du produit.
2. Indique si le CipherSpec est certifié FIPS sur une plateforme certifiée FIPS. Voir la rubrique sur la norme FIPS (Federal Information Processing Standards) pour une explication de la norme FIPS.
3. ALW Ces CipherSpecs sont désactivés lorsque TLS 1.3 est activé (via la propriété AllowTLSV13 dans `qm.ini`).

   z/OS Les gestionnaires de files d'attente créés dans IBM MQ for z/OS 9.2.0 ou version ultérieure activent TLS 1.3 par défaut, ce qui désactive ces CipherSpecs. Si nécessaire, vous pouvez activer ces CipherSpecs en désactivant TLS V1.3. Pour ce faire, ajoutez **AllowTLSV13**=*FALSE* à la strophe TransportSecurity du fichier QMINI dans le JCL du gestionnaire de files d'attente. TLS 1.3 n'est pas activé par défaut dans les gestionnaires de files d'attente migrés vers IBM MQ for z/OS 9.2.0 depuis une version précédente et par conséquent, ces CipherSpecs sont activés.
4. La taille de clé d'établissement de liaison maximale est de 512 bits. Si l'un ou l'autre des certificats échangés lors de l'établissement de liaison SSL a une taille de clé supérieure à 512 bit, une clé temporaire de 512 bits est générée pour l'établissement de liaison.
5. Ces CipherSpec ne sont plus pris en charge par IBM MQ classes for Java et IBM MQ classes for JMS. Pour plus d'informations, voir CipherSpecs et suites de chiffrement SSL/TLS dans IBM MQ classes for Java ou CipherSpecs et suites de chiffrement SSL/TLS dans IBM MQ classes for JMS.
6. La taille de clé d'établissement de liaison maximale est de 1024 bits.
7. Deprecated Ce CipherSpec a été certifié FIPS 140-2 avant le 19 mai 2007. Le nom `FIPS_WITH_DES_CBC_SHA` est historique et reflète le fait que CipherSpec était précédemment (mais n'est plus) conforme à la norme FIPS. Ce CipherSpec est déprécié et son utilisation est déconseillée.
8. Deprecated Le nom `FIPS_WITH_3DES_EDE_CBC_SHA` est historique et reflète le fait que CipherSpec était précédemment (mais n'est plus) conforme à la norme FIPS. L'utilisation de ce CipherSpec a été dépréciée.
9. Ce CipherSpec a été certifié FIPS 140-2 avant le 19 mai 2007.
10. La réactivation de ces CipherSpec ne nécessite pas l'utilisation de l'instruction de définition de données (DD) CSQXWEAK.

Activation des CipherSpecs obsolètes sous IBM MQ for Multiplatforms

Multi

Par défaut, vous n'êtes pas autorisé à spécifier un CipherSpec obsolète dans une définition de canal. Si vous tentez de spécifier un CipherSpec obsolète sur IBM MQ for Multiplatforms, vous recevez le message AMQ8242: la définition SSLCIPH est incorrecte et PCF renvoie `MQRCCF_SSL_CIPHER_SPEC_ERROR`.

Vous ne pouvez pas démarrer un canal avec un CipherSpecobsolète. Si vous tentez de le faire avec un CipherSpecobsolète, le système renvoie `MQCC_FAILED (2)`, ainsi qu'un **Reason** de `MQRC_SSL_INITIALIZATION_ERROR (2393)` au client.

Vous pouvez réactiver une ou plusieurs des CipherSpecs obsolètes pour définir des canaux, lors de l'exécution sur le serveur, en définissant la variable d'environnement **AMQ_SSL_WEAK_CIPHER_ENABLE**.

La variable d'environnement **AMQ_SSL_WEAK_CIPHER_ENABLE** accepte:

- Un nom CipherSpec unique, ou
- Liste de noms CipherSpec séparés par des virgules à réactiver, ou
- Valeur spéciale de ALL, représentant tous les CipherSpecs.

Avertissement : Bien que ALL soit une option valide, vous devez l'utiliser **uniquement** dans une situation spécifique requise par votre entreprise, car la réactivation de ALL CipherSpecs active les protocoles SSL 3.0 et TLS 1.0 , ainsi qu'un grand nombre d'algorithmes de cryptographie faibles.

Par exemple, si vous souhaitez réactiver ECDHE_RSA_RC4_128_SHA256, définissez la variable d'environnement suivante:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

ou modifiez la strophe SSL dans le fichier `qm.ini` en définissant:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

Activation des CipherSpecs obsolètes sous z/OS

Par défaut, vous n'êtes pas autorisé à spécifier un CipherSpec obsolète dans une définition de canal. Si vous tentez de spécifier un CipherSpec obsolète sur z/OS, vous recevez le message CSQM102E, le message CSQX616Eou CSQX674E.

Suivez les instructions répertoriées dans cette section si vous recevez l'un de ces messages et que votre entreprise doit réactiver l'utilisation de CipherSpecsfaibles.

Avertissement : Dans les instructions suivantes, pour que les instructions de définition de données prennent effet, SSLTASKS doit être une valeur différente de zéro. Si cela nécessite une modification de SSLTASKS, vous devez recycler l'initiateur de canal.

Sous IBM MQ for z/OS, la méthode actuelle de contrôle des CipherSpecs faibles ou rompus est la suivante:

- Si vous souhaitez réactiver l'utilisation de CipherSpecsfaibles, vous devez ajouter une instruction de définition de données (DD) factice nommée CSQXWEAK au JCL de l'initiateur de canal. S'il est spécifié seul, cela active uniquement les CipherSpecs faibles associées au protocole TLS 1.2 ; par exemple:

  ```
  //CSQXWEAK DD DUMMY
  ```

  Remarque : Tous les CipherSpecs obsolètes ne nécessitent pas l'utilisation de cette instruction de définition de données (voir la remarque 10 dans le tableau précédent).
- Si vous souhaitez réactiver l'utilisation de CipherSpecs SSLv3 , vous devez également ajouter une instruction de définition de données factice nommée CSQXSSL3 au JCL de l'initiateur de canal. Tous les CipherSpecs SSLv3 CipherSpecs sont considérés comme **faibles**. Vous devez donc également spécifier CSQXWEAK:

  ```
  //CSQXSSL3 DD DUMMY
  ```
- Si vous souhaitez réactiver les CipherSpecsTLS V1 obsolètes, ajoutez une instruction de définition de données factice nommée TLS10ON (activez TLS V1.0) au JCL de l'initiateur de canal. S'il est spécifié seul, cela active les CipherSpecs forts associés au protocole TLS 1.0 :

  ```
  //TLS10ON DD DUMMY
  ```

S'il est spécifié avec CSQXWEAK , cela active également les CipherSpecs **faibles** associées à TLS 1.0.

- Si vous souhaitez désactiver explicitement les CipherSpecsTLS V1 obsolètes, vous devez ajouter une instruction de définition de données factice nommée TLS10OFF (désactivez TLS V1.0) au JCL de l'initiateur de canal. Par exemple:

```
//TLS10OFF DD DUMMY
```

Si vous souhaitez négocier uniquement avec le programme d'écoute à l'aide des spécifications de chiffrement répertoriées dans la liste des spécifications de chiffrement par défaut **System SSL** , vous devez définir l'instruction de définition de données suivante dans le JCL CHINIT:

```
JCL: //GSKDCIPS DD DUMMY
```

Important : Pour IBM MQ for z/OS 9.2.0 et les versions ultérieures, les cartes DD répertoriées précédemment et la valeur de **AllowTLSV13** sont prises en compte lors de l'affichage des messages lors du démarrage de l'initiateur de canal pour indiquer les protocoles qui sont activés et ceux qui ne le sont pas. Ainsi, même si l'une des cartes DD précédemment listées est spécifiée, cela peut signifier que, en raison d'une combinaison de ces paramètres, un certain protocole ne peut pas être activé avec un autre protocole. Par exemple, le protocole SSL 3.0 n'est pas autorisé si TLS 1.3 est activé.

Des mécanismes alternatifs peuvent être utilisés pour réactiver de force les CipherSpecsfaibles et la prise en charge de SSLv3 , si la modification de la définition de données ne convient pas. Pour plus d'informations, contactez le service de maintenance IBM .

Concepts associés

«Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49
Cette rubrique fournit des informations sur la façon de choisir les CipherSpecs et les certificats numériques appropriés pour votre règle de sécurité, en soulignant la relation entre les CipherSpecs et les certificats numériques dans IBM MQ.

Référence associée

De la définition d'un canal
ALTER CHANNEL

Relation entre les paramètres d'alias CipherSpec

Ces informations décrivent le comportement attendu avec différentes combinaisons d'alias CipherSpecs dans les configurations client et serveur. Ici, un client fait référence à l'entité initiant la communication, par example une application client ou un canal émetteur de gestionnaire de files d'attente, et un serveur fait référence à l'entité recevant la communication du client, par example un canal de connexion serveur ou un canal récepteur.

Protocole minimum et protocole fixe CipherSpecs

IBM MQ prend en charge deux types différents de CipherSpecs:

Protocole minimal
Les CipherSpecs de protocole minimum sont ceux qui ne définissent pas de limite supérieure, par exemple ANY, ANY_TLS12_OR_HIGHER ou ANY_TLS13_OR_HIGHER.

Protocole fixe
Les CipherSpecs de protocole fixe sont ceux qui identifient un protocole spécifique, par exemple ANY_TLS12 et ANY_TLS13, ou un algorithme spécifique, tel que ECDHE_ECDSA_3DES_EDE_CBC_SHA256.

Les protocoles minimum et fixe CipherSpecs sont pris en charge sur toutes les plateformes.

Pour optimiser la simplicité de la configuration tout en conservant la sécurité, l'utilisation du **protocole minimal** CipherSpecs est recommandée des deux côtés du canal. Cela permet à vos communications de prendre automatiquement en charge et d'utiliser une version de protocole TLS supérieure lorsque les deux côtés prennent en charge une nouvelle version sans qu'il soit nécessaire de modifier la configuration de l'un ou l'autre côté.

L'utilisation d'un **protocole minimum** CipherSpec du côté du lancement, mais un **protocole fixe** CipherSpec du côté de la réception peut entraîner le rejet de la connexion, et

- Multi Messages AMQ9631 et AMQ9641 émis.
- z/OS Messages CSQX631E et CSQX641E émis.

Les tableaux suivants présentent la relation entre les différents paramètres d'alias CipherSpec et le résultat attendu. Tableau 81, à la page 456 montre le comportement attendu lorsque TLS 1.3 n'est pas activé sur le client, le serveur ou les deux. Tableau 82, à la page 456 montre le comportement attendu lorsque TLS 1.3 est activé sur le client et le serveur. Dans les deux cas, les CipherSpecs du client sont affichés sur l'axe Y du tableau et les CipherSpecs du serveur sont affichés sur l'axe X du tableau.

Remarque : Dans les tableaux suivants, les cellules marquées *Probablement d'échec* indiquent le risque de conflit lorsque vous spécifiez un **protocole minimum** CipherSpec pour une partie d'une connexion et un CipherSpec spécifique (**protocole fixe**) pour une autre partie.

Par exemple, supposons que le client et le serveur soient définis pour utiliser ANY CipherSpec, et que le canal serveur soit défini pour utiliser un CipherSpecspécifique:

- Si le CipherSpec pris en charge le plus fort pour le client et le serveur correspond au CipherSpec spécifique configuré sur le canal, l'établissement de liaison TLS est résolu avec succès.
- Si, toutefois, il existe un CipherSpec plus fort pris en charge par le client et le serveur, l'établissement de liaison TLS se résout à l'utiliser, même s'il ne correspond pas au CipherSpec spécifié sur le canal, et l'établissement de liaison TLS échoue.

Tableau 81. Comportement attendu lorsque TLS 1.3 n'est pas activé sur le client, le serveur ou les deux

	serveur			
Environnement	**TLS spécifique 1.2 CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ OR_HIGHER**
TLS spécifique 1.2 CipherSpec	Connecte	Connecte	Connecte	Connecte
Tout	*Susceptible d'échouer*	Connecte	Connecte	Connecte
ANY_ TLS12	*Susceptible d'échouer*	Connecte	Connecte	Connecte
ANY_TLS12_ OR_HIGHER	*Susceptible d'échouer*	Connecte	Connecte	Connecte

Tableau 82. Comportement attendu lorsque TLS 1.3 est activé sur le client et le serveur

	serveur						
Environnement	**TLS spécifique 1.2 CipherSpec**	**TLS spécifique 1.3 CipherSpec**	**ANY**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_SUPERIEUR**	**ANY_TLS13_ OR_SUPERIEUR**
TLS spécifique 1.2 CipherSpec	Connecte	**Echoue**	Connecte	Connecte	**Echoue**	Connecte	**Echoue**
TLS spécifique 1.3 CipherSpec	**Echoue**	Connecte	Connecte	**Echoue**	Connecte	Connecte	Connecte

Tableau 82. Comportement attendu lorsque TLS 1.3 est activé sur le client et le serveur (suite)							
	serveur						
Environnement	**TLS spécifique 1.2 CipherSpec**	**TLS spécifique 1.3 CipherSpec**	**ANY**	**ANY_TLS12**	**ANY_TLS13**	**ANY_TLS12_OR_SUPERIEUR**	**ANY_TLS13_OR_SUPERIEUR**
Tout	**Echoue**	*Susceptible d'échouer*	Connecte	**Echoue**	Connecte	Connecte	Connecte
ANY_TLS12	*Susceptible d'échouer*	**Echoue**	Connecte	Connecte	**Echoue**	Connecte	**Echoue**
ANY_TLS13	**Echoue**	*Susceptible d'échouer*	Connecte	**Echoue**	Connecte	Connecte	Connecte
ANY_TLS12_OR_HIGHER	**Echoue**	*Susceptible d'échouer*	Connecte	**Echoue**	Connecte	Connecte	Connecte
ANY_TLS13_OR_HIGHER	**Echoue**	*Susceptible d'échouer*	Connecte	**Echoue**	Connecte	Connecte	Connecte

Concepts associés

«Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49
Cette rubrique fournit des informations sur la façon de choisir les CipherSpecs et les certificats numériques appropriés pour votre règle de sécurité, en soulignant la relation entre les CipherSpecs et les certificats numériques dans IBM MQ.

«CipherSpecs et CipherSuites», à la page 22
Les protocoles de sécurité cryptographique doivent convenir des algorithmes utilisés par une connexion sécurisée. CipherSpecs et CipherSuites définissent des combinaisons spécifiques d'algorithmes.

«Activation des CipherSpecs», à la page 435
Activez un CipherSpec à l'aide du paramètre **SSLCIPH** dans la commande **DEFINE CHANNEL** ou **ALTER CHANNEL** MQSC.

Tâches associées

Migration des configurations de sécurité existantes en vue de l'utilisation de l'élément CipherSpec ANY_TLS12_OR_HIGHER

Obtention d'informations sur les CipherSpecs à l'aide de IBM MQ Explorer

Vous pouvez utiliser IBM MQ Explorer pour afficher les descriptions des CipherSpecs.

Utilisez la procédure suivante pour obtenir des informations sur les CipherSpecs dans «Activation des CipherSpecs», à la page 435:

1. Ouvrez IBM MQ Explorer et développez le dossier des **gestionnaires de files d'attente**.
2. Vérifiez que vous avez démarré votre gestionnaire de files d'attente.
3. Sélectionnez le gestionnaire de files d'attente à utiliser et cliquez sur **Canaux**.
4. Cliquez avec le bouton droit de la souris sur le canal que vous souhaitez utiliser et sélectionnez **Propriétés**.
5. Sélectionnez la page de propriétés **SSL** .
6. Dans la liste, sélectionnez le CipherSpec que vous souhaitez utiliser. Une description s'affiche dans la fenêtre située sous la liste.

z/OS IBM i Alternatives pour la spécification de CipherSpecs

Pour les plateformes sur lesquelles le système d'exploitation fournit la prise en charge TLS, votre système peut prendre en charge de nouveaux CipherSpecs qui ne sont pas inclus dans «Activation des CipherSpecs», à la page 435.

Vous pouvez spécifier un nouveau CipherSpec avec le paramètre SSLCIPH, mais la valeur que vous fournissez dépend de votre plateforme. Dans tous les cas, la spécification doit correspondre à un CipherSpec TLS valide et pris en charge par la version de TLS exécutée par votre système.

Remarque : Cette section ne s'applique pas aux systèmes AIX, Linux, and Windows , car les CipherSpecs sont fournis avec le produit IBM MQ , de sorte que les nouveaux CipherSpecs ne deviennent pas disponibles après l'expédition.

IBM i IBM i

Chaîne de deux caractères représentant une valeur hexadécimale.

Pour plus d'informations sur les valeurs autorisées, voir le point 3 de la section Remarques sur l'utilisation de la rubrique Définition des informations sur les caractères pour une session sécurisée.

Avertissement : Vous ne devez pas spécifier de valeurs de chiffrement hexadécimal dans **SSLCIPH**, car il n'est pas clair à partir de la valeur qui sera utilisée, et le choix du protocole à utiliser est indéterminé. L'utilisation de valeurs de chiffrement hexadécimales peut entraîner des erreurs de non-concordance de CipherSpec .

Vous pouvez utiliser **CHGMQMCHL** ou la commande **CRTMQMCHL** pour spécifier la valeur, par exemple:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

Vous pouvez également utiliser la commande **ALTER QMGR** MQSC pour définir le paramètre **SSLCIPH** .

z/OS z/OS

Chaîne de quatre caractères représentant une valeur hexadécimale. Les codes hexadécimaux correspondent aux valeurs définies dans le protocole TLS.

Pour plus d'informations, voir Cipher Suite Definitions où figure la liste de toutes les spécifications de chiffrement TLS 1.0, TLS 1.2et TLS 1.3 prises en charge, sous la forme de codes hexadécimaux à 4 chiffres.

Remarque : Deprecated Pour utiliser un CipherSpecfaible ou un CipherSpec appartenant à un protocole obsolète, tel que SSL V3.0 ou TLS 1.0, vous devez spécifier la carte DD appropriée dans le JCL de démarrage de l'initiateur de canal. Pour plus d'informations, voir «CipherSpecs obsolètes», à la page 450.

Remarques relatives aux clusters IBM MQ

Avec les clusters IBM MQ , il est plus sûr d'utiliser les noms CipherSpec dans «Activation des CipherSpecs», à la page 435. Si vous utilisez une autre spécification, sachez que la spécification peut ne pas être valide sur d'autres plateformes. Pour plus d'informations, voir «SSL/TLS et clusters», à la page 497.

Spécification d'un CipherSpec pour un IBM MQ MQI client

Vous disposez de trois options pour spécifier un CipherSpec pour un IBM MQ MQI client.

Ces options sont les suivantes :

- Utilisation d'une table de définition de canal
- Utilisation de la zone SSLCipherSpec dans la structure MQCD, à l'adresse MQCD_VERSION_7 ou supérieure, sur un appel MQCONNX.

- Utilisation de Active Directory (sur les systèmes Windows avec prise en charge d' Active Directory)

Spécification d'une CipherSuite avec IBM MQ classes for Java et IBM MQ classes for JMS

IBM MQ classes for Java et IBM MQ classes for JMS spécifient les CipherSuites différemment des autres plateformes.

Pour plus d'informations sur la spécification d'une CipherSuite avec IBM MQ classes for Java, voir Transport Layer Security (TLS) support for Java

Pour plus d'informations sur la spécification d'une CipherSuite avec IBM MQ classes for JMS, voir Utilisation de TLS (Transport Layer Security) avec IBM MQ classes for JMS

Spécification d'un CipherSpec pour IBM MQ.NET

Pour IBM MQ.NET , vous pouvez spécifier le CipherSpec à l'aide de la classe MQEnvironment ou à l'aide de la propriété MQC.SSL_CIPHER_SPEC_PROPERTY dans la table de hachage des propriétés de connexion.

Pour plus d'informations sur la spécification d'un CipherSpec pour le client non géré .NET , voir Activation de TLS pour le client .NET non géré

Pour plus d'informations sur la spécification d'un CipherSpec pour le client géré .NET , voir Prise en charge deCipherSpec pour le client .NET géré .

z/OS Utilisation d'AT-TLS avec IBM MQ for z/OS

Application Transparent Transport Layer Security (AT-TLS) fournit une prise en charge TLS pour les applications z/OS sans que ces applications aient à implémenter la prise en charge TLS, ou même à savoir que TLS est utilisé. AT-TLS est disponible uniquement sur z/OS.

AT-TLS peut être utilisé avec toutes les versions de IBM MQ for z/OS.

Avant d'utiliser AT-TLS avec IBM MQ for z/OS, assurez-vous de bien comprendre le «Restrictions», à la page 462 impliqué.

Pour utiliser Application Transparent Transport Layer Security , vous devez définir des instructions de stratégie contenant un ensemble de règles utilisées par z/OS Communications Server pour déterminer les connexions TCP/IP pour lesquelles TLS est activé de manière transparente.

IBM MQ for z/OS possède sa propre implémentation TLS, qui requiert que les canaux aient le paramètre SSLCIPH configuré avec un CipherSpecpris en charge.

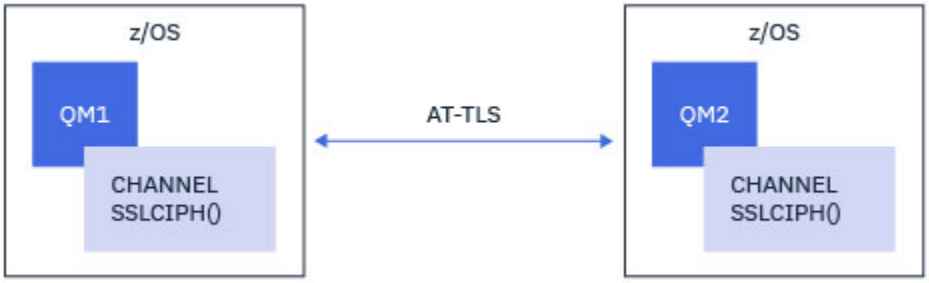
Lorsqu'il décide d'activer TLS sur un canal, l'administrateur IBM MQ peut décider d'utiliser AT-TLS ou IBM MQ TLS. La décision est souvent prise en fonction de l'utilisation d'AT-TLS pour d'autres middlewares ou en raison d'implications en termes de performances. Pour une comparaison de base des performances d'AT-TLS et de IBM MQ TLS, voir MP16: Capacity Planning and Tuning for IBM MQ for z/OS.

Scénarios

L'utilisation d'AT-TLS avec IBM MQ est prise en charge dans les scénarios suivants:

Scénario 1

Entre deux gestionnaires de files d'attente IBM MQ for z/OS où les deux côtés du canal utilisent AT-TLS. Autrement dit, aucun des deux canaux ne spécifie l'attribut SSLCIPH. Cette approche peut être utilisée avec n'importe quel canal de message.



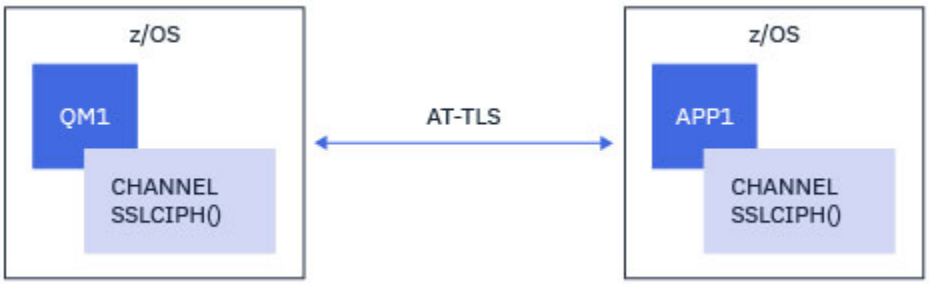
L'implémentation de ce scénario consiste à définir deux règles AT-TLS, une pour chaque côté du canal. Ces règles sont identiques à celles utilisées avec le scénario 3 ou le scénario 4.

Par exemple, si le canal est passé de l'utilisation d'un seul CipherSpec CipherSpec à l'utilisation d'AT-TLS, le canal sortant utilise la règle de «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 463 et le canal entrant utilise la règle de «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 472.

Si le canal est passé de l'utilisation d'un alias CipherSpec à l'utilisation d'AT-TLS, le canal de communications sortantes utilise la règle de «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide de l'alias CipherSpecs», à la page 468 et le canal de communications entrantes utilise la règle de «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec», à la page 476.

Scénario 2

Entre un gestionnaire de files d'attente IBM MQ for z/OS et une application client IBM MQ Java s'exécutant sur z/OS où les deux côtés du canal utilisent AT-TLS. Autrement dit, ni le canal de connexion serveur, ni le canal de connexion client ne spécifient l'attribut SSLCIPH.



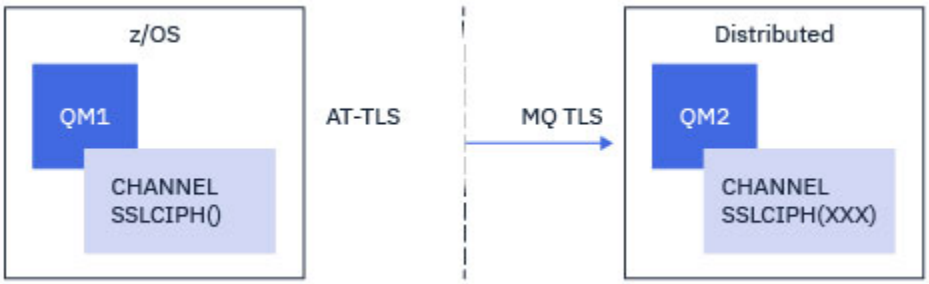
L'implémentation de ce scénario consiste à définir deux règles AT-TLS, une pour chaque côté du canal. Ces règles sont identiques à celles utilisées avec le scénario 3 ou le scénario 4.

Par exemple, si le canal est passé d'un CipherSpec CipherSpec unique à un CipherSpec AT-TLS, le canal de connexion client utilise la règle de «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 463 et le canal de connexion serveur utilise la règle de «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 472.

Si le canal est modifié de l'alias CipherSpec à l'utilisation d'AT-TLS, le canal de connexion client utilise la règle de «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide de l'alias CipherSpecs», à la page 468 et le canal de connexion serveur utilise la règle de «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec», à la page 476.

Scénario 3

Entre un gestionnaire de files d'attente IBM MQ for z/OS et un gestionnaire de files d'attente s'exécutant sous IBM MQ for Multiplatforms, où le gestionnaire de files d'attente IBM MQ for z/OS utilise AT-TLS et le gestionnaire de files d'attente IBM MQ for Multiplatforms utilise IBM MQ TLS, en spécifiant l'attribut SSLCIPH avec un seul CipherSpecnommé. Cela s'applique à tous les types de canaux de transmission de messages autres que les types émetteur et récepteur de cluster.

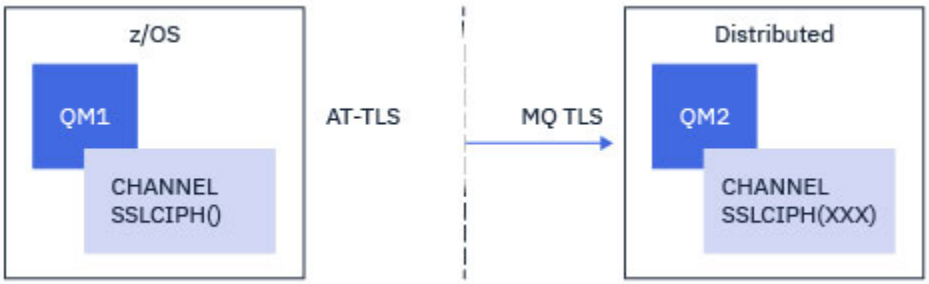


Voir «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 463 pour un exemple de configuration AT-TLS pour les canaux sortants du gestionnaire de files d'attente IBM MQ for z/OS vers le gestionnaire de files d'attente IBM MQ for Multiplatforms et «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 472 pour un exemple de configuration AT-TLS pour les canaux entrants du gestionnaire de files d'attente IBM MQ for Multiplatforms vers le gestionnaire de files d'attente IBM MQ for z/OS .

La même configuration AT-TLS peut être utilisée lorsque les deux gestionnaires de files d'attente sont sous z/OS, mais le gestionnaire de files d'attente à droite n'a pas été configuré pour utiliser AT-TLS.

Scénario 4

Entre un gestionnaire de files d'attente IBM MQ for z/OS et un gestionnaire de files d'attente s'exécutant sous IBM MQ for Multiplatforms, où le gestionnaire de files d'attente IBM MQ for z/OS utilise AT-TLS et le gestionnaire de files d'attente IBM MQ for Multiplatforms utilise IBM MQ TLS, en spécifiant l'attribut SSLCIPH avec un alias CipherSpec. Cela s'applique à tous les types de canaux de transmission de messages autres que les types émetteur et récepteur de cluster.

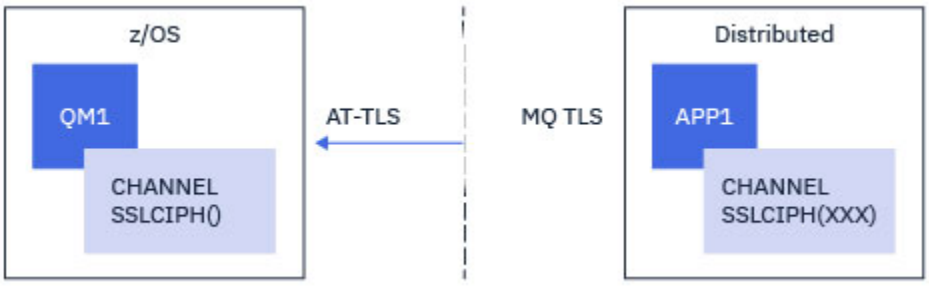


Voir «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide de l'alias CipherSpecs», à la page 468 pour un exemple de configuration AT-TLS pour les canaux sortants du gestionnaire de files d'attente IBM MQ for z/OS vers le gestionnaire de files d'attente IBM MQ for Multiplatforms et «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec», à la page 476, et «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec», à la page 476 pour un exemple de configuration AT-TLS pour les canaux entrants du gestionnaire de files d'attente IBM MQ for Multiplatforms vers le gestionnaire de files d'attente IBM MQ for z/OS .

La même configuration AT-TLS peut être utilisée lorsque les deux gestionnaires de files d'attente sont sous z/OS, mais le gestionnaire de files d'attente à droite n'a pas été configuré pour utiliser AT-TLS.

Scénario 5

Entre un gestionnaire de files d'attente IBM MQ for z/OS et une application client s'exécutant sous IBM MQ for Multiplatforms, où le gestionnaire de files d'attente IBM MQ for z/OS utilise AT-TLS et l'application client utilise IBM MQ TLS en spécifiant l'attribut SSLCIPH avec un seul CipherSpec CipherSpec.

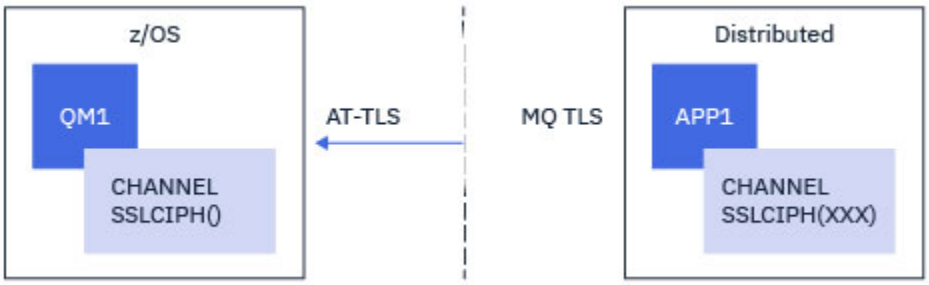


Ce scénario requiert une règle AT-TLS unique qui répond aux mêmes exigences que celles utilisées par un canal de message entrant ; voir «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec», à la page 472.

La même configuration AT-TLS peut être utilisée lorsque l'application client est une application Java et qu'elle s'exécute également sur z/OS, mais elle n'a pas été configurée pour utiliser AT-TLS.

Scénario 6

Entre un gestionnaire de files d'attente IBM MQ for z/OS et une application client s'exécutant sous IBM MQ for Multiplatforms, où le gestionnaire de files d'attente IBM MQ for z/OS utilise AT-TLS et l'application client utilise IBM MQ TLS en spécifiant l'attribut SSLCIPH avec un alias CipherSpec.



Ce scénario requiert une règle AT-TLS unique qui répond aux mêmes exigences que celles utilisées par un canal de message entrant ; voir «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec», à la page 476.

La même configuration AT-TLS peut être utilisée lorsque l'application client est une application Java et qu'elle s'exécute également sur z/OS, mais elle n'a pas été configurée pour utiliser AT-TLS.

Restrictions

IBM MQ for z/OS n'est pas sensible à AT-TLS, il existe donc plusieurs restrictions qui s'appliquent aux scénarios précédents:

- AT-TLS combiné à IBM MQ TLS ne fonctionne pas avec les canaux émetteur de cluster et récepteur de cluster.

- Les gestionnaires de files d'attente IBM MQ for z/OS ne savent pas qu'ils utilisent AT-TLS et ne reçoivent pas d'informations de certificat de leur gestionnaire de files d'attente ou client partenaire. Par conséquent, les attributs suivants n'ont aucun effet sur le côté z/OS d'un canal utilisant AT-TLS:
  - Attributs de canal SSLCAUTH et SSLPEER
  - Attribut de gestionnaire de files d'attente SSLRKEYC
  - Attributs SSLPEERMAP des règles CHLAUTH
- L'utilisation de la renégociation de clé secrète TLS requiert que les deux côtés du canal utilisent le protocole TLS IBM MQ . Par conséquent, la renégociation des clés secrètes TLS ne doit pas être activée pour un gestionnaire de files d'attente ou un client IBM MQ for Multiplatforms si vous vous connectez à un gestionnaire de files d'attente IBM MQ for z/OS à l'aide d'AT-TLS.

  Pour désactiver la renégociation de clé secrète TLS pour un gestionnaire de files d'attente, définissez le paramètre SSLRKEYC du gestionnaire de files d'attente sur `0`. Pour un client, définissez le paramètre approprié sur `0` en fonction du type de client. Pour plus de détails sur la procédure à suivre, voir «Réinitialisation des clés secrètes SSL et TLS», à la page 481.

Instructions de configuration AT-TLS

AT-TLS est configuré à l'aide d'un ensemble d'instructions. Les scénarios utilisés dans les scénarios décrits dans cette rubrique sont les suivants:

Règle TTLSRule
Indique un ensemble de critères pour la mise en correspondance d'une connexion TCP/IP à une configuration TLS. Fait référence aux autres types d'instruction.

TTLSGroupAction
Indique si le `TTLSRule` de référencement est activé ou non.

TTLSEnvironmentAction
Indique la configuration détaillée du `TTLSRule` de référence et fait référence à un certain nombre d'autres instructions.

TTLSKeyringParms
Fait référence au fichier de clés qui doit être utilisé par AT-TLS.

TTLSCipherParms
Définit les suites de chiffrement à utiliser.

TTLSEnvironmentAdvancedParms
Définit les protocoles TLS ou SSL qui sont activés.

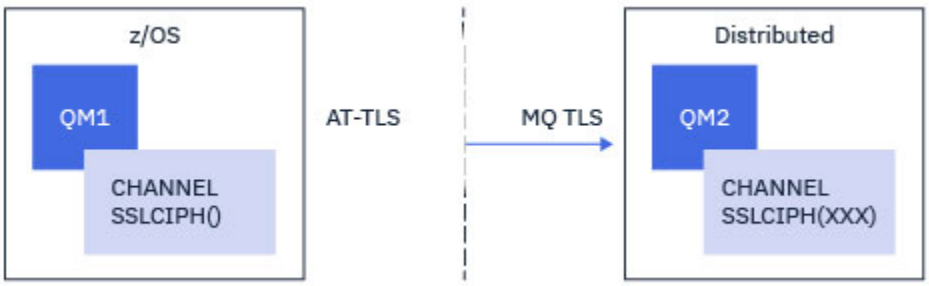
Avertissement : Il existe d'autres instructions de règle AT-TLS avec AT-TLS qui ne sont pas documentées ici et qui peuvent être utilisées avec IBM MQ en fonction des besoins. Toutefois, IBM MQ n'a été testé qu'avec les règles décrites dans cette rubrique.

z/OS *Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec*

Comment configurer AT-TLS sur un canal sortant d'un gestionnaire de files d'attente IBM MQ for z/OS vers un gestionnaire de files d'attente IBM MQ for Multiplatforms . Dans ce cas, le canal du gestionnaire de files d'attente z/OS est un canal émetteur pour lequel l'attribut SSLCIPH n'est pas défini et le canal du gestionnaire de files d'attente nonz/OS est un canal récepteur avec l'attribut SSLCIPH défini sur un seul, nommé CipherSpec.

Voir «Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide de l'alias CipherSpecs», à la page 468 pour un exemple d'utilisation d'un alias CipherSpec.

Dans cet exemple, une paire de canaux émetteur-récepteur existante qui utilise le protocole TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec va être ajustée pour que le canal émetteur utilise AT-TLS au lieu de IBM MQ TLS.



D'autres protocoles TLS et CipherSpecs peuvent être utilisés en apportant des ajustements mineurs à la configuration. D'autres types de canaux de transmission de messages, à l'exception des canaux émetteur de cluster et récepteur de cluster, pourraient être utilisés sans modification de la configuration AT-TLS.

Procédure

Etape 1: Arrêter le canal

Etape 2: Création et application d'une règle AT-TLS

Vous devez créer les instructions AT-TLS suivantes pour ce scénario:

1. Une instruction TTLSRule pour faire correspondre les connexions sortantes de l'espace adresse de l'initiateur de canal à l'adresse IP et au numéro de port du canal récepteur cible. Ces valeurs doivent correspondre aux informations utilisées dans CONNAME du canal émetteur. Ici, un filtrage supplémentaire a été inclus pour correspondre à un nom de travail d'initiateur de canal spécifique.

```
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
}
```

   La règle précédente correspond aux connexions à l'adresse IP 123.456.78.9 sur le port 1414 à partir du travail CSQ1CHIN .

   Des options de filtrage plus avancées sont décrites dans TTLSRule.

2. Une instruction TTLSGroupAction activant la règle. `TTLSRule` fait référence à `TTLSGroupAction` à l'aide de la propriété **`TTLSGroupActionRef`** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Instruction TTLSEnvironmentAction associée à `TTLSRule` par la propriété **`TTLSEnvironmentActionRef`** . Un `TTLSEnvironmentAction` configure l'environnement TLS et spécifie le fichier de clés à utiliser.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Instruction TTLSKeyringParms associée à TTLSEnvironmentAction par la propriété **TTLSKeyringParmsRef** et qui définit le fichier de clés utilisé par AT-TLS.

   Le fichier de clés doit contenir des certificats sécurisés par le gestionnaire de files d'attente nonz/OS distant. Ce fichier de clés peut être défini de la même manière qu'un fichier de clés utilisé par l'initiateur de canal ; voir «Configuring your z/OS system to use TLS», à la page 262.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Une instruction TTLSCipherParms associée à TTLSEnvironmentAction par la propriété **TTLSCipherParmsRef** .

   Cette instruction doit contenir un nom de suite de chiffrement unique qui doit être l'équivalent du nom IBM MQ CipherSpec utilisé sur le canal récepteur cible.

   Remarque : Les noms de suite de chiffrement AT-TLS ne correspondent pas nécessairement aux noms IBM MQ CipherSpec . Toutefois, il est possible de trouver le nom de la suite de chiffrement AT-TLS qui correspond à un nom IBM MQ CipherSpec en recherchant le nom IBM MQ CipherSpec dans le tableau suivant et en croisant la colonne de code hexadécimal avec la colonne de caractères développée du tableau 2 de la rubrique d'instruction TTLSCipherParms .

Tableau 83. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0

CipherSpec	Protocole	Code hexadécimal	Activé par défaut
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Oui
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Oui
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Oui
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Oui
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Oui
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Oui
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Oui
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Oui
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Oui
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Oui

Tableau 83. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0 (suite)			
CipherSpec	**Protocole**	**Code hexadécimal**	**Activé par défaut**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Oui
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Oui
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Oui
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Non
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Non
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Non
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Non
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Non
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Non
TRIPLE_DES_SHA_US	SSL v3	000A	Non
RC4_SHA_US	SSL v3	0005	Non
RC4_MD5_US	SSL v3	0004	Non
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Non
RC2_MD5_EXPORT	SSL v3	0006	Non
NULL_SHA	SSL v3	0002	Non
NULL_MD5	SSL v3	0001	Non

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. Une instruction TTLSEnvironmentAdvancedParms est associée à TTLSEnvironmentAction par la propriété **TTLSEnvironmentAdvancedParmsRef** .

   Cette instruction peut être utilisée pour spécifier les protocoles SSL et TLS qui sont activés. Avec IBM MQ , vous devez activer uniquement le protocole unique qui correspond au nom de l'algorithme de cryptographie utilisé dans l'instruction TTLSCipherParms .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

L'ensemble complet des instructions est le suivant et doit être appliqué à l'agent de règles:

```
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
}
```

Etape 3: Suppression de SSLCIPH du canal z/OS

Supprimez le CipherSpec du canal z/OS à l'aide de la commande suivante:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Etape 4: Démarrez le canal

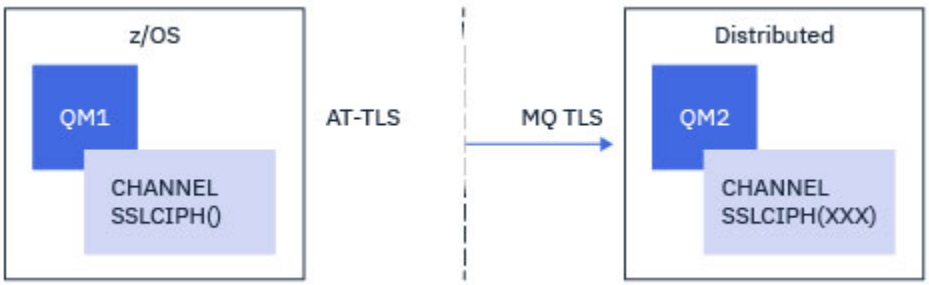
Une fois le canal démarré, il utilise une combinaison d'AT-TLS et IBM MQ TLS.

Avertissement : Les instructions AT-TLS précédentes ne sont qu'une configuration minimale. Il existe d'autres instructions de règle AT-TLS avec AT-TLS qui ne sont pas documentées ici et qui peuvent être utilisées avec IBM MQ en fonction des besoins. Toutefois, IBM MQ n'a été testé qu'avec les règles décrites.

z/OS *Configuration d'AT-TLS sur un canal sortant vers un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide de l'alias CipherSpecs*

Comment configurer AT-TLS sur un canal sortant d'un gestionnaire de files d'attente IBM MQ for z/OS vers un gestionnaire de files d'attente IBM MQ for Multiplatforms . Dans ce cas, le canal du gestionnaire de files d'attente z/OS est un canal émetteur pour lequel l'attribut SSLCIPH n'est pas défini et le canal du gestionnaire de files d'attente nonz/OS est un canal récepteur avec l'attribut SSLCIPH défini sur un alias CipherSpec

Dans cet exemple, une paire de canaux émetteur-récepteur existante, qui utilise l'alias ANY_TLS13 CipherSpec , va être ajustée pour que le canal émetteur utilise AT-TLS au lieu de TLS IBM MQ .



D'autres protocoles TLS et CiperSpecs peuvent être utilisés en apportant des ajustements mineurs à la configuration. D'autres types de canaux de transmission de messages, à l'exception des canaux émetteur de cluster et récepteur de cluster, pourraient être utilisés sans modification de la configuration AT-TLS.

Procédure

Etape 1: Arrêter le canal

Etape 2: Création et application d'une règle AT-TLS

Vous devez créer les instructions AT-TLS suivantes pour ce scénario:

1. Une instruction TTLSRule pour faire correspondre les connexions sortantes de l'espace adresse de l'initiateur de canal à l'adresse IP et au numéro de port du canal récepteur cible. Ces valeurs doivent correspondre aux informations utilisées dans CONNAME du canal émetteur. Ici, un filtrage supplémentaire a été inclus pour correspondre à un nom de travail d'initiateur de canal spécifique.

```
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
}
```

   La règle précédente correspond aux connexions à l'adresse IP 123.456.78.9 sur le port 1414 à partir du travail CSQ1CHIN .

   Des options de filtrage plus avancées sont décrites dans TTLSRule.

2. Une instruction TTLSGroupAction activant la règle. TTLSRule fait référence à TTLSGroupAction à l'aide de la propriété **TTLSGroupActionRef** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Instruction TTLSEnvironmentAction associée à TTLSRule par la propriété **TTLSEnvironmentActionRef** . Un TTLSEnvironmentAction configure l'environnement TLS et spécifie le fichier de clés à utiliser.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Instruction TTLSKeyringParms associée à TTLSEnvironmentAction par la propriété **TTLSKeyringParmsRef** et qui définit le fichier de clés utilisé par AT-TLS.

   Le fichier de clés doit contenir des certificats sécurisés par le gestionnaire de files d'attente nonz/OS distant. Ce fichier de clés peut être défini de la même manière qu'un fichier de clés utilisé par l'initiateur de canal ; voir «Configuring your z/OS system to use TLS», à la page 262.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Une instruction TTLSCipherParms associée à TTLSEnvironmentAction par la propriété **TTLSCipherParmsRef** .

   Cette instruction doit contenir un ou plusieurs noms de suite de chiffrement, dont au moins un doit être compatible avec l'ensemble de CipherSpecs impliqué par l'alias CipherSpec utilisé sur le canal récepteur cible.

   Remarque : Les noms de suite de chiffrement AT-TLS ne correspondent pas nécessairement aux noms IBM MQ CipherSpec . Toutefois, il est possible de trouver le nom de la suite de chiffrement AT-TLS qui correspond à un nom IBM MQ CipherSpec en recherchant le nom IBM MQ CipherSpec dans le tableau suivant et en faisant référence à la colonne de code hexadécimal avec la colonne de caractères développée du tableau 2 dans la rubrique TTLSCipherParms .

Tableau 84. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0

CipherSpec	Protocole	Code hexadécimal	Activé par défaut
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Oui
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Oui
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Oui
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Oui
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Oui
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Oui

Tableau 84. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0 (suite)

CipherSpec	Protocole	Code hexadécimal	Activé par défaut
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Oui
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Oui
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Oui
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Oui
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Oui
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Oui
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Oui
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Non
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Non
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Non
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Non
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Non
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Non
TRIPLE_DES_SHA_US	SSL v3	000A	Non
RC4_SHA_US	SSL v3	0005	Non
RC4_MD5_US	SSL v3	0004	Non
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Non
RC2_MD5_EXPORT	SSL v3	0006	Non
NULL_SHA	SSL v3	0002	Non
NULL_MD5	SSL v3	0001	Non

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Avertissement : Si le gestionnaire de files d'attente et la règle AT-TLS prennent en charge TLS 1.3, seuls les alias CipherSpecs qui contiennent au moins une spécification TLS 1.3 CipherSpec permettent au canal de démarrer. Par exemple, l'utilisation de ANY_TLS12 entraîne l'échec du démarrage du canal, même si TTLSCipherParms contient TLS 1.2 CipherSpecs, mais que l'utilisation de ANY_TLS12_OR_HIGHER ou ANY_TLS13 permet le démarrage du canal. Pour plus d'informations, voir «Relation entre les paramètres d'alias CipherSpec», à la page 455 .

6. Une instruction TTLSEnvironmentAdvancedParms est associée à TTLSEnvironmentAction par la propriété **TTLSEnvironmentAdvancedParmsRef** .

   Cette instruction peut être utilisée pour spécifier les protocoles SSL et TLS qui sont activés et doit être cohérente avec les suites de chiffrement de l'instruction TTLSCipherParms .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

L'ensemble complet des instructions est le suivant et doit être appliqué à l'agent de règles:

```
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
}
```

Etape 3: Suppression de SSLCIPH du canal z/OS

Supprimez le CipherSpec du canal z/OS à l'aide de la commande suivante:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Etape 4: Démarrez le canal

Une fois le canal démarré, il utilise une combinaison d'AT-TLS et IBM MQ TLS.

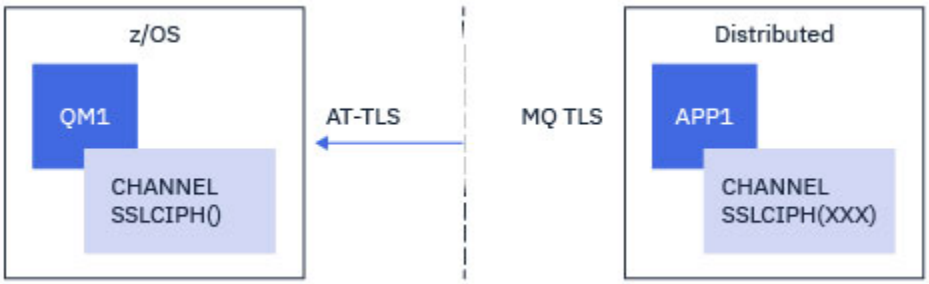
Avertissement : Les instructions AT-TLS précédentes ne sont qu'une configuration minimale. Il existe d'autres instructions de règle AT-TLS avec AT-TLS qui ne sont pas documentées ici et qui peuvent être utilisées avec IBM MQ en fonction des besoins. Toutefois, IBM MQ n'a été testé qu'avec les règles décrites.

z/OS *Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un seul CipherSpec CipherSpec*

Comment configurer AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms vers un gestionnaire de files d'attente IBM MQ for z/OS . Dans ce cas, le canal du gestionnaire de files d'attente z/OS est un canal récepteur dont l'attribut SSLCIPH n'est pas défini et le canal du gestionnaire de files d'attente nonz/OS est un canal émetteur dont l'attribut SSLCIPH est défini sur un seul, nommé CipherSpec.

Voir «Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec», à la page 476 pour un exemple d'utilisation d'un alias CipherSpec.

Dans cet exemple, une paire de canaux émetteur-récepteur existante qui utilise le protocole TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec va être ajustée pour que le canal récepteur utilise AT-TLS au lieu de IBM MQ TLS.



D'autres protocoles TLS et CipherSpecs peuvent être utilisés en apportant des ajustements mineurs à la configuration. D'autres types de canaux de transmission de messages, à l'exception des canaux émetteur de cluster et récepteur de cluster, pourraient être utilisés sans modification de la configuration AT-TLS.

Procédure

Etape 1: Arrêter le canal

Etape 2: Création et application d'une règle AT-TLS

Vous devez créer les instructions AT-TLS suivantes pour ce scénario:

1. Une instruction TTLSRule permettant de faire correspondre les connexions entrantes à l'espace adresse de l'initiateur de canal à partir de l'adresse IP du canal émetteur. Ici, un filtrage supplémentaire a été inclus pour correspondre à un nom de travail d'initiateur de canal spécifique.

```
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
}
```

La règle précédente correspond aux connexions entrant dans le travail CSQ1CHIN sur le port local 1414 à partir de l'adresse IP distante 123.456.78.9.

Des options de filtrage plus avancées sont décrites dans TTLSRule.

2. Une instruction TTLSGroupAction activant la règle. TTLSRule fait référence à TTLSGroupAction à l'aide de la propriété **TTLSGroupActionRef** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Une instruction TTLSEnvironmentAction est associée à TTLSRule par la propriété **TTLSEnvironmentActionRef** . Un TTLSEnvironmentAction configure l'environnement TLS et spécifie le fichier de clés à utiliser.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS offre la possibilité de fournir une authentification mutuelle, ce qui est l'équivalent de l'utilisation de l'attribut de canal SSLCAUTH. Pour cela, une instruction TTLSEnvironmentAction est associée à la valeur **HandshakeRole** *ServerWithClientAuth* pour l'instruction TTLSEnvironmentAction entrante.

4. Une instruction TTLSKeyringParms est associée à TTLSEnvironmentAction par la propriété **TTLSKeyringParmsRef** et définit le fichier de clés utilisé par AT-TLS.

   Le fichier de clés doit contenir des certificats sécurisés par le gestionnaire de files d'attente nonz/OS distant. Ce fichier de clés peut être défini de la même manière qu'un fichier de clés utilisé par l'initiateur de canal ; voir «Configuring your z/OS system to use TLS», à la page 262.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Une instruction TTLSCipherParms associée à TTLSEnvironmentAction par la propriété **TTLSCipherParmsRef** .

   Cette instruction doit contenir un nom de suite de chiffrement unique qui doit être l'équivalent du nom IBM MQ CipherSpec utilisé sur le canal émetteur distant.

   Remarque : Les noms de suite de chiffrement AT-TLS ne correspondent pas nécessairement aux noms IBM MQ CipherSpec . Toutefois, il est possible de trouver le nom de la suite de chiffrement AT-TLS qui correspond à un nom IBM MQ CipherSpec en recherchant le nom IBM MQ CipherSpec dans le tableau suivant et en croisant la colonne de code hexadécimal avec la colonne de caractères développée du tableau 2 de la rubrique d'instruction TTLSCipherParms .

Tableau 85. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0

CipherSpec	Protocole	Code hexadécimal	Activé par défaut
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Oui
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Oui
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Oui
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Oui
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Oui
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Oui
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Oui
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Oui
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Oui
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Oui
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Oui
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Oui
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Oui
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Non
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Non
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Non
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Non
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Non
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Non
TRIPLE_DES_SHA_US	SSL v3	000A	Non
RC4_SHA_US	SSL v3	0005	Non
RC4_MD5_US	SSL v3	0004	Non

Tableau 85. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0 (suite)			
CipherSpec	**Protocole**	**Code hexadécimal**	**Activé par défaut**
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Non
RC2_MD5_EXPORT	SSL v3	0006	Non
NULL_SHA	SSL v3	0002	Non
NULL_MD5	SSL v3	0001	Non

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. Une instruction TTLSEnvironmentAdvancedParms est associée à TTLSEnvironmentAction par la propriété **TTLSEnvironmentAdvancedParmsRef** .

   Cette instruction peut être utilisée pour spécifier les protocoles SSL et TLS qui sont activés. Avec IBM MQ , vous devez activer uniquement le protocole unique qui correspond au nom de l'algorithme de cryptographie utilisé dans l'instruction TTLSCipherParms .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

L'ensemble complet des instructions est le suivant et doit être appliqué à l'agent de règles:

```
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
}
```

Etape 3: Suppression de SSLCIPH du canal z/OS

Supprimez le CipherSpec du canal z/OS à l'aide de la commande suivante:

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

Etape 4: Démarrez le canal

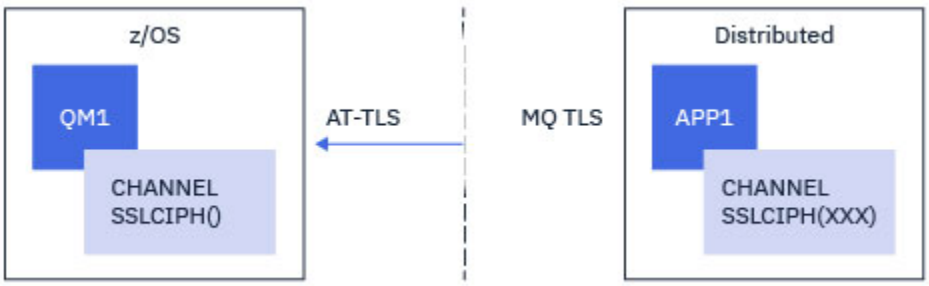
Une fois le canal démarré, il utilise une combinaison d'AT-TLS et IBM MQ TLS.

Avertissement : Les instructions AT-TLS précédentes ne sont qu'une configuration minimale. Il existe d'autres instructions de règle AT-TLS avec AT-TLS qui ne sont pas documentées ici et qui peuvent être utilisées avec IBM MQ en fonction des besoins. Toutefois, IBM MQ n'a été testé qu'avec les règles décrites.

z/OS *Configuration d'AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms à l'aide d'un alias CipherSpec*

Comment configurer AT-TLS sur un canal entrant à partir d'un gestionnaire de files d'attente IBM MQ for Multiplatforms vers un gestionnaire de files d'attente IBM MQ for z/OS . Dans ce cas, le canal du gestionnaire de files d'attente z/OS est un canal récepteur dont l'attribut SSLCIPH n'est pas défini et le canal du gestionnaire de files d'attente nonz/OS est un canal émetteur dont l'attribut SSLCIPH est défini sur un alias CipherSpec.

Dans cet exemple, une paire de canaux émetteur-récepteur existante, qui utilise n'importe quel protocole TLS 1.3 CipherSpec , va être ajustée de sorte que le canal récepteur utilise AT-TLS au lieu de IBM MQ TLS.



D'autres protocoles TLS et CiperSpecs peuvent être utilisés en apportant des ajustements mineurs à la configuration. D'autres types de canaux de transmission de messages, à l'exception des canaux émetteur de cluster et récepteur de cluster, pourraient être utilisés sans modification de la configuration AT-TLS.

Procédure

Etape 1: Arrêter le canal

Etape 2: Création et application d'une règle AT-TLS

Vous devez créer les instructions AT-TLS suivantes pour ce scénario:

1. Une instruction TTLSRule permettant de faire correspondre les connexions entrantes à l'espace adresse de l'initiateur de canal à partir de l'adresse IP du canal émetteur. Ici, un filtrage supplémentaire a été inclus pour correspondre à un nom de travail d'initiateur de canal spécifique.

```
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
}
```

   La règle précédente correspond aux connexions entrant dans le travail CSQ1CHIN sur le port local 1414 à partir de l'adresse IP distante 123.456.78.9.

   Des options de filtrage plus avancées sont décrites dans TTLSRule.

2. Une instruction TTLSGroupAction activant la règle. TTLSRule fait référence à TTLSGroupAction à l'aide de la propriété **TTLSGroupActionRef** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Une instruction TTLSEnvironmentAction est associée à TTLSRule par la propriété **TTLSEnvironmentActionRef** . Un TTLSEnvironmentAction configure l'environnement TLS et spécifie le fichier de clés à utiliser.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS offre la possibilité de fournir une authentification mutuelle, ce qui est l'équivalent de l'utilisation de l'attribut de canal SSLCAUTH. Pour cela, une instruction TTLSEnvironmentAction est associée à la valeur **HandshakeRole** *ServerWithClientAuth* pour l'instruction TTLSEnvironmentAction entrante.

4. Une instruction TTLSKeyringParms est associée à TTLSEnvironmentAction par la propriété **TTLSKeyringParmsRef** et définit le fichier de clés utilisé par AT-TLS.

   Le fichier de clés doit contenir des certificats sécurisés par le gestionnaire de files d'attente nonz/OS distant. Ce fichier de clés peut être défini de la même manière qu'un fichier de clés utilisé par l'initiateur de canal ; voir «Configuring your z/OS system to use TLS», à la page 262.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Une instruction TTLSCipherParms associée à TTLSEnvironmentAction par la propriété **TTLSCipherParmsRef** .

   Cette instruction doit contenir au moins un nom de suite de chiffrement inclus dans l'alias CipherSpec défini sur le canal émetteur distant.

   Remarque : Les noms de suite de chiffrement AT-TLS ne correspondent pas nécessairement aux noms IBM MQ CipherSpec . Toutefois, il est possible de trouver le nom de la suite de chiffrement AT-TLS qui correspond à un nom IBM MQ CipherSpec en recherchant le nom IBM MQ CipherSpec dans le tableau suivant et en croisant la colonne de code hexadécimal avec la colonne de caractères développée du tableau 2 de la rubrique d'instruction TTLSCipherParms .

Tableau 86. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0

CipherSpec	Protocole	Code hexadécimal	Activé par défaut
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Oui
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Oui
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Oui
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Oui
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Oui
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Oui
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Oui
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Oui
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Oui
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Oui
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Oui

Tableau 86. CipherSpecs sur z/OS à partir de IBM MQ for z/OS 9.2.0 (suite)			
CipherSpec	**Protocole**	**Code hexadécimal**	**Activé par défaut**
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Oui
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Oui
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Non
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Non
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Non
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Non
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Non
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Non
TRIPLE_DES_SHA_US	SSL v3	000A	Non
RC4_SHA_US	SSL v3	0005	Non
RC4_MD5_US	SSL v3	0004	Non
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Non
RC2_MD5_EXPORT	SSL v3	0006	Non
NULL_SHA	SSL v3	0002	Non
NULL_MD5	SSL v3	0001	Non

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Avertissement : Si le gestionnaire de files d'attente et la règle AT-TLS prennent en charge TLS 1.3, seuls les alias CipherSpecs qui contiennent au moins une spécification TLS 1.3 CipherSpec permettent au canal de démarrer. Par exemple, l'utilisation de ANY_TLS12 entraîne l'échec du démarrage du canal, même si TTLSCipherParms contient TLS 1.2 CipherSpecs, mais que l'utilisation de ANY_TLS12_OR_HIGHER ou ANY_TLS13 permet le démarrage du canal. Pour plus d'informations, voir «Relation entre les paramètres d'alias CipherSpec», à la page 455 .

6. Une instruction TTLSEnvironmentAdvancedParms est associée à `TTLSEnvironmentAction` par la propriété **`TTLSEnvironmentAdvancedParmsRef`** .

   Cette instruction peut être utilisée pour spécifier les protocoles SSL et TLS qui sont activés et doit être cohérente avec les suites de chiffrement de l'instruction `TTLSCipherParms` .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

L'ensemble complet des instructions est le suivant et doit être appliqué à l'agent de règles:

```
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
}
```

Etape 3: Suppression de SSLCIPH du canal z/OS

Supprimez le CipherSpec du canal z/OS à l'aide de la commande suivante:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Etape 4: Démarrez le canal

Une fois le canal démarré, il utilise une combinaison d'AT-TLS et IBM MQ TLS.

Avertissement : Les instructions AT-TLS précédentes ne sont qu'une configuration minimale. Il existe d'autres instructions de règle AT-TLS avec AT-TLS qui ne sont pas documentées ici et qui

peuvent être utilisées avec IBM MQ en fonction des besoins. Toutefois, IBM MQ n'a été testé qu'avec les règles décrites.

Réinitialisation des clés secrètes SSL et TLS

IBM MQ prend en charge la réinitialisation des clés secrètes sur les gestionnaires de files d'attente et les clients.

Les clés secrètes sont réinitialisées lorsqu'un nombre spécifié d'octets chiffrés de données ont transité sur le canal. Si les pulsations de canal sont activées, la clé secrète est réinitialisée avant que les données ne soient envoyées ou reçues à la suite d'une pulsation de canal.

La valeur de réinitialisation de clé est toujours définie par le côté initiateur du canal IBM MQ .

Gestionnaire de files d'attente

Pour un gestionnaire de files d'attente, utilisez la commande **ALTER QMGR** avec le paramètre **SSLRKEYC** pour définir les valeurs utilisées lors de la renégociation de clé.

Sous IBM i, utilisez **CHGMQM** avec le paramètre **SSLRSTCNT** .

MQI Client

Par défaut, les clients MQI ne renégocient pas la clé secrète. Vous pouvez faire en sorte qu'un client MQI renégocie la clé de trois manières. Dans la liste suivante, les méthodes sont affichées par ordre de priorité. Si vous spécifiez plusieurs valeurs, la valeur de priorité la plus élevée est utilisée.

1. En utilisant la zone KeyResetCount dans la structure MQSCO sur un appel MQCONNX.
2. En utilisant la variable d'environnement **MQSSLRESET**.
3. En définissant l'attribut **SSLKeyResetCount** dans la strophe SSL du fichier de configuration client.

Ces variables peuvent être définies sur un entier compris entre 0 et 999 999 999, représentant le nombre d'octets non chiffrés envoyés et reçus dans une conversation TLS avant que la clé secrète TLS ne soit renégociée. La valeur 0 indique que les clés secrètes TLS ne sont jamais renégociées. Si vous spécifiez un nombre de réinitialisations de clé confidentielle TLS compris entre 1 octet et 32 Ko, les canaux TLS utiliseront un nombre de réinitialisations de clé confidentielle de 32 Ko. Cela permet d'éviter un nombre excessif de réinitialisations de clé qui se produiraient pour les petites valeurs de réinitialisation de clé confidentielle TLS.

Si une valeur supérieure à zéro est spécifiée et que les pulsations de canal sont activées pour le canal, la clé secrète est également renégociée avant que les données de message ne soient envoyées ou reçues à la suite d'une pulsation de canal.

Nombre d'octets jusqu'à ce que la prochaine renégociation de clé secrète soit réinitialisée après chaque renégociation réussie.

Java

Pour IBM MQ classes for Java, une application peut réinitialiser la clé secrète de l'une des manières suivantes:

- En définissant la zone sslResetCount dans la classe MQEnvironment.
- En définissant la propriété d'environnement MQC.SSL_RESET_COUNT_PROPERTY dans un objet Hashtable. L'application affecte ensuite la table de hachage à la zone `properties` de la classe MQEnvironment ou transmet la table de hachage à un objet MQQueueManager sur son constructeur.

Si l'application utilise plusieurs de ces méthodes, les règles de priorité habituelles s'appliquent. Voir Class com.ibm.mq.MQEnvironment pour les règles de priorité.

La valeur de la zone sslResetCount ou de la propriété d'environnement MQC.SSL_RESET_COUNT_PROPERTY représente le nombre total d'octets envoyés et reçus par le code

client IBM MQ classes for Java avant la renégociation de la clé secrète. Le nombre d'octets envoyés est le nombre avant chiffrement et le nombre d'octets reçus est le nombre après déchiffrement. Le nombre d'octets inclut également les informations de contrôle envoyées et reçues par le client IBM MQ classes for Java .

Si le nombre de réinitialisations est égal à zéro, ce qui correspond à la valeur par défaut, la clé secrète n'est jamais renégociée. Le nombre de réinitialisations est ignoré si CipherSuite n'est pas spécifié.

JMS

Pour IBM MQ classes for JMS, la propriété SSLRESETCOUNT représente le nombre total d'octets envoyés et reçus par une connexion avant que la clé secrète utilisée pour le chiffrement ne soit renégociée. Le nombre d'octets envoyés est le nombre avant chiffrement et le nombre d'octets reçus est le nombre après déchiffrement. Le nombre d'octets inclut également les informations de contrôle envoyées et reçues par IBM MQ classes for JMS. Par exemple, pour configurer un objet ConnectionFactory pouvant être utilisé pour créer une connexion via un canal MQI activé par TLS avec une clé secrète renégociée après la transmission de 4 Mo de données, exécutez la commande suivante à JMSAdmin:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Si la valeur de SSLRESETCOUNT est zéro, qui est la valeur par défaut, la clé secrète n'est jamais renégociée. La propriété SSLRESETCOUNT est ignorée si SSLCIPHERSUITE n'est pas défini.

.NET

Pour les clients .NET non gérés, la propriété d'entier **SSLKeyResetCount** indique le nombre d'octets non chiffrés envoyés et reçus dans une conversation TLS avant que la clé secrète ne soit renégociée. Pour plus d'informations sur l'utilisation des propriétés d'objet dans IBM MQ classes for .NET, voir Obtention et définition des valeurs d'attribut.

Pour les clients gérés .NET , la classe SSLStream ne prend pas en charge la réinitialisation / renégociation des clés secrètes. Toutefois, pour être cohérent avec les autres clients IBM MQ , le client IBM MQ géré .NET permet aux applications de définir **SSLKeyResetCount**. Pour plus d'informations, voir Secret key reset or renégociation.

XMS .NET

Pour les clients XMS .NET non gérés, voir Connexions sécurisées à un gestionnaire de files d'attente IBM MQ.

Référence associée
ALTER QMGR
DISPLAYQMGR
Modification du gestionnaire de files d'attente de messages (CHGMQM)
Afficher le gestionnaire de files d'attente de messages (DSPMQM)

Implémentation de la confidentialité dans les programmes d'exit utilisateur

Implémentation de la confidentialité dans les exits de sécurité

Les exits de sécurité peuvent jouer un rôle dans le service de confidentialité en générant et en distribuant la clé symétrique pour le chiffrement et le déchiffrement des données qui circulent sur le canal. Une technique courante pour ce faire utilise la technologie PKI.

Un exit de sécurité génère une valeur de données aléatoire, le chiffre à l'aide de la clé publique du gestionnaire de files d'attente ou de l'utilisateur que l'exit de sécurité partenaire représente et envoie les données chiffrées à son partenaire dans un message de sécurité. L'exit de sécurité partenaire déchiffre la valeur de données aléatoires avec la clé privée du gestionnaire de files d'attente ou de l'utilisateur qu'il

représente. Chaque exit de sécurité peut désormais utiliser la valeur de données aléatoires pour dériver la clé symétrique indépendamment l'une de l'autre en utilisant un algorithme connu des deux. Ils peuvent également utiliser la valeur de données aléatoire comme clé.

Si le premier exit de sécurité n'a pas authentifié son partenaire à ce moment-là, le message de sécurité suivant envoyé par le partenaire peut contenir une valeur attendue chiffrée avec la clé symétrique. Le premier exit de sécurité peut désormais authentifier son partenaire en vérifiant que l'exit de sécurité du partenaire a pu chiffrer correctement la valeur attendue.

Les exits de sécurité peuvent également utiliser cette opportunité pour convenir de l'algorithme de chiffrement et de déchiffrement des données qui circulent sur le canal, si plusieurs algorithmes sont disponibles.

Implémentation de la confidentialité dans les exits de message

Un exit de message à l'extrémité émettrice d'un canal peut chiffrer les données d'application dans un message et un autre exit de message à l'extrémité réceptrice du canal peut déchiffrer les données. Pour des raisons de performances, un algorithme de clé symétrique est normalement utilisé à cette fin. Pour plus d'informations sur la façon dont la clé symétrique peut être générée et distribuée, voir «Implémentation de la confidentialité dans les programmes d'exit utilisateur», à la page 482.

Les en-têtes d'un message, tels que l'en-tête de file d'attente de transmission, MQXQH, qui inclut le descripteur de message imbriqué, ne doivent pas être chiffrés par un exit de message. En effet, la conversion des données des en-têtes de message a lieu soit après l'appel d'un exit de message à l'extrémité émettrice, soit avant l'appel d'un exit de message à l'extrémité réceptrice. Si les en-têtes sont chiffrés, la conversion des données échoue et le canal s'arrête.

Implémentation de la confidentialité dans les exits d'envoi et de réception

Les exits d'envoi et de réception peuvent être utilisés pour chiffrer et déchiffrer les données qui circulent sur un canal. Ils sont plus appropriés que les exits de message pour fournir ce service pour les raisons suivantes:

- Sur un canal de message, les en-têtes de message peuvent être chiffrés ainsi que les données d'application dans les messages.
- Les exits d'envoi et de réception peuvent être utilisés sur les canaux MQI ainsi que sur les canaux de message. Les paramètres des appels MQI peuvent contenir des données d'application sensibles qui doivent être protégées alors qu'elles circulent sur un canal MQI. Vous pouvez donc utiliser les mêmes exits d'émission et de réception sur les deux types de canaux.

Implémentation de la confidentialité dans l'exit d'API et l'exit de croisement d'API

Les données d'application d'un message peuvent être chiffrées par une API ou un exit de croisement d'API lorsque le message est inséré par l'application émettrice et déchiffré par un deuxième exit lorsque le message est extrait par l'application réceptrice. Pour des raisons de performances, un algorithme de clé symétrique est généralement utilisé à cette fin. Toutefois, au niveau de l'application, où de nombreux utilisateurs peuvent s'envoyer des messages les uns aux autres, le problème est de s'assurer que seul le destinataire prévu d'un message est en mesure de déchiffrer le message. Une solution consiste à utiliser une clé symétrique différente pour chaque paire d'utilisateurs qui s'envoient des messages. Mais cette solution peut être difficile et longue à administrer, en particulier si les utilisateurs appartiennent à des organisations différentes. Un moyen standard de résoudre ce problème est appelé *enveloppage numérique* et utilise la technologie PKI.

Lorsqu'une application place un message dans une file d'attente, une API ou un exit de croisement d'API génère une clé symétrique aléatoire et utilise la clé pour chiffrer les données d'application dans le message. L'exit chiffre la clé symétrique avec la clé publique du destinataire prévu. Il remplace ensuite les données d'application du message par les données d'application chiffrées et la clé symétrique chiffrée. De cette manière, seul le récepteur visé peut déchiffrer la clé symétrique et donc les données d'application. Si un message chiffré a plus d'un récepteur prévu possible, l'exit peut chiffrer une copie de la clé symétrique pour chaque récepteur prévu.

Si différents algorithmes de chiffrement et de déchiffrement des données d'application sont disponibles, l'exit peut inclure le nom de l'algorithme qu'il a utilisé.

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 484
- Archive log data sets; see note “2” on page 484
- Page sets; see note “1” on page 484
- BSDS; see note “2” on page 484
- CSQINP* data sets; see note “2” on page 484
- SMDS; see note “1” on page 484

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 485 to "6" on page 485 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 485
2. "Configuring data set encryption for the log data sets" on page 486

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 486.

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 486

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 485

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 486 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 485
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 485.
   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 489.

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 490 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 489.

   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 489 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 489 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).

Intégrité des données de messages

Pour préserver l'intégrité des données, vous pouvez utiliser différents types de programme d'exit utilisateur pour fournir des prétraitements de message ou des signatures numériques pour vos messages.

Intégrité des données

Implémentation de l'intégrité des données dans les messages

Lorsque vous utilisez TLS, votre choix de CipherSpec détermine le niveau d'intégrité des données dans l'entreprise. Si vous utilisez le service AMS (IBM MQ Advanced Message Service), vous pouvez spécifier l'intégrité d'un message unique.

Implémentation de l'intégrité des données dans les exits de message

Un message peut être signé numériquement par un exit de message à l'extrémité émettrice d'un canal. La signature numérique peut alors être vérifiée par une sortie de message à l'extrémité réceptrice d'un canal pour détecter si le message a été volontairement modifié.

Une certaine protection peut être fournie à l'aide d'un résumé de message au lieu d'une signature numérique. Un résumé de message peut être efficace contre les manipulations occasionnelles ou indiscriminées, mais il n'empêche pas l'individu le plus informé de changer ou de remplacer le message, et de générer un résumé complètement nouveau pour lui. Cela est particulièrement vrai si l'algorithme utilisé pour générer le résumé de message est un algorithme bien connu.

Implémentation de l'intégrité des données dans les exits d'envoi et de réception

Sur un canal de message, les exits de message sont plus appropriés pour fournir ce service car un exit de message a accès à l'ensemble d'un message. Sur un canal MQI, les paramètres des appels MQI peuvent contenir des données d'application qui doivent être protégées et seuls les exits d'envoi et de réception peuvent fournir cette protection.

Implémentation de l'intégrité des données dans l'exit API ou l'exit de croisement d'API

Un message peut être signé numériquement par une API ou un exit de croisement d'API lorsque le message est inséré par l'application émettrice. La signature numérique peut alors être vérifiée par une deuxième sortie lorsque le message est récupéré par l'application réceptrice pour détecter si le message a été volontairement modifié.

Une certaine protection peut être fournie à l'aide d'un résumé de message au lieu d'une signature numérique. Un résumé de message peut être efficace contre les manipulations occasionnelles ou indiscriminées, mais il n'empêche pas l'individu le plus informé de changer ou de remplacer le message, et de générer un résumé complètement nouveau pour lui. Ceci est particulièrement vrai si l'algorithme utilisé pour générer le résumé de message est bien connu,

Plus d'informations

Pour plus d'informations sur la garantie de l'intégrité des données, voir la section «Activation des CipherSpecs», à la page 435 .

Tâches associées

Connexion de deux gestionnaires de files d'attente via le protocole TLS

Connexion sécurisée d'un client à un gestionnaire de files d'attente

Audit

Vous pouvez vérifier les intrusions de sécurité ou les tentatives d'intrusion à l'aide de messages d'événement. Vous pouvez également vérifier la sécurité de votre système à l'aide de la IBM MQ Explorer.

Pour détecter les tentatives d'exécution d'actions non autorisées, telles que la connexion à un gestionnaire de files d'attente ou l'insertion d'un message dans une file d'attente, examinez les messages d'événement générés par vos gestionnaires de files d'attente, en particulier les messages d'événement de droits d'accès. Pour plus d'informations sur les messages d'événement du gestionnaire de files d'attente, voir Evénements du gestionnaire de files d'attente, et pour plus d'informations sur la surveillance des événements en général, voir Surveillance des événements.

Maintien de la sécurité des clusters

Autorisez ou empêchez les gestionnaires de files d'attente de rejoindre des clusters ou d'insérer des messages dans des files d'attente de cluster. Forcer un gestionnaire de files d'attente à quitter un cluster. Prenez en compte certaines considérations supplémentaires lors de la configuration de TLS pour les clusters.

Arrêt des gestionnaires de files d'attente non autorisés envoyant des messages

Empêchez les gestionnaires de files d'attente non autorisés d'envoyer des messages à votre gestionnaire de files d'attente à l'aide d'un exit de sécurité de canal.

Avant de commencer

La mise en cluster n'a aucun effet sur le fonctionnement des exits de sécurité. Vous pouvez restreindre l'accès à un gestionnaire de files d'attente de la même manière que dans un environnement de mise en file d'attente répartie.

Pourquoi et quand exécuter cette tâche

Empêchez les gestionnaires de files d'attente sélectionnés d'envoyer des messages à votre gestionnaire de files d'attente:

Procédure

1. Définissez un programme d'exit de sécurité de canal sur la définition de canal CLUSRCVR .
2. Ecrivez un programme qui authentifie les gestionnaires de files d'attente en tentant d'envoyer des messages sur votre canal récepteur de cluster et leur refuse l'accès s'ils ne sont pas autorisés.

Que faire ensuite

Les programmes d'exit de sécurité de canal sont appelés au démarrage et à l'arrêt de l'agent MCA.

Arrêt des gestionnaires de files d'attente non autorisés à insérer des messages dans vos files d'attente

Utilisez l'attribut de droit d'insertion de canal sur le canal récepteur de cluster pour arrêter les gestionnaires de files d'attente non autorisés à placer des messages dans vos files d'attente. Autorisez un gestionnaire de files d'attente éloignées en vérifiant l'ID utilisateur dans le message à l'aide de RACF on z/OSou de la méthode d'accès aux objets (OAM) sur Multiplatforms.

Pourquoi et quand exécuter cette tâche

Utilisez les fonctions de sécurité d'une plateforme et le mécanisme de contrôle d'accès dans IBM MQ pour contrôler l'accès aux files d'attente.

Procédure

1. Pour empêcher certains gestionnaires de files d'attente d'insérer des messages dans une file d'attente, utilisez les fonctions de sécurité disponibles sur votre plateforme.

   Exemple :

   - z/OS RACF ou autres gestionnaires de sécurité externes sous IBM MQ for z/OS
   - Multi Gestionnaire des droits d'accès aux objets (OAM) sur d'autres plateformes Multiplatforms.
2. Utilisez les droits d'insertion, PUTAUT, sur l'attribut de la définition de canal CLUSRCVR .

   L'attribut PUTAUT permet de spécifier les identificateurs utilisateur à utiliser pour établir le droit d'insertion d'un message dans une file d'attente.

   Les options de l'attribut PUTAUT sont les suivantes:

   infrastructure d"évaluation de déploiement
   Utilisez l'ID utilisateur par défaut.

   z/OS Sous z/OS, la vérification peut impliquer l'utilisation à la fois de l'ID utilisateur reçu du réseau et de l'ID utilisateur dérivé de MCAUSER.

   CTX
   Utilisez l'ID utilisateur dans les informations de contexte associées au message.

   z/OS Sous z/OS , la vérification peut impliquer l'utilisation de l'ID utilisateur reçu du réseau, ou de l'ID utilisateur dérivé de MCAUSER, ou des deux. Utilisez cette option si le lien est sécurisé et authentifié.

   z/OS **ONLYMCA (z/OS uniquement)**
   Comme pour DEF, mais tout ID utilisateur reçu du réseau n'est pas utilisé. Utilisez cette option si le lien n'est pas sécurisé. Vous souhaitez autoriser uniquement un ensemble spécifique d'actions sur celui-ci, qui sont définies pour MCAUSER.

   z/OS **ALTMCA (z/OS uniquement)**
   Comme pour CTX, mais aucun ID utilisateur reçu du réseau n'est utilisé.

Autorisation d'insertion de messages dans des files d'attente de cluster éloignées

Sur z/OS , configurez l'autorisation d'insertion dans une file d'attente de cluster à l'aide de RACF. Sur Multiplatforms, autorisez l'accès pour la connexion aux gestionnaires de files d'attente et l'insertion dans les files d'attente de ces gestionnaires de files d'attente.

Pourquoi et quand exécuter cette tâche

Le comportement par défaut consiste à effectuer un contrôle d'accès sur le SYSTEM.CLUSTER.TRANSMIT.QUEUE. Notez que ce comportement s'applique, même si vous utilisez plusieurs files d'attente de transmission.

Le comportement spécifique décrit dans cette rubrique s'applique uniquement lorsque vous avez configuré l'attribut **ClusterQueueAccessControl** dans le fichier qm.ini comme étant *RQMName*, comme décrit dans la rubrique Strophe de sécurité , puis redémarré le gestionnaire de files d'attente.

Procédure

- z/OS

  Pour z/OS, exécutez les commandes suivantes:

  ```
  RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
  PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
  ```

- ALW

  Pour les systèmes AIX, Linux, and Windows , exécutez les commandes suivantes:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
  ```

- IBM i

  Pour IBM i, exécutez les commandes suivantes:

  ```
  GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
  GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  ```

  L'utilisateur peut placer des messages uniquement dans la file d'attente de cluster spécifiée, et aucune autre file d'attente de cluster.

  Les noms de variable ont les significations suivantes:

  QMgrName
  Nom du gestionnaire de files d'attente. Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.

  GroupName
  Nom du groupe auquel l'accès doit être accordé.

  QueueName
  Nom de la file d'attente ou du profil générique pour lequel modifier les autorisations.

Que faire ensuite

Si vous indiquez une file d'attente de réponse lorsque vous placez un message dans une file d'attente de cluster, l'application destinataire doit être autorisée à envoyer la réponse. Définissez ces droits en suivant les instructions de la rubrique «Octroi du droit d'insertion de messages dans une file d'attente de cluster éloignée», à la page 409.

Concepts associés

Section de sécurité dans qm.ini

Empêcher les gestionnaires de files d'attente de rejoindre un cluster

Si un gestionnaire de files d'attente corrompu rejoint un cluster, il est difficile de l'empêcher de recevoir des messages que vous ne souhaitez pas recevoir.

Procédure

Si vous souhaitez vous assurer que seuls certains gestionnaires de files d'attente autorisés rejoignent un cluster, vous avez le choix entre trois techniques:

- A l'aide des enregistrements d'authentification de canal, vous pouvez bloquer la connexion de canal de cluster en fonction de l'adresse IP distante, du nom du gestionnaire de files d'attente éloignées ou du nom distinctif TLS fourni par le système distant.
- Ecrire un programme d'exit pour empêcher les gestionnaires de files d'attente non autorisés d'écrire dans SYSTEM.CLUSTER.COMMAND.QUEUE. Ne limitez pas l'accès à

SYSTEM.CLUSTER.COMMAND.QUEUE de sorte qu'aucun gestionnaire de files d'attente ne puisse y écrire, sinon vous empêcheriez tout gestionnaire de files d'attente de rejoindre le cluster.

- Un programme d'exit de sécurité sur la définition de canal CLUSRCVR .

Exits de sécurité sur les canaux de cluster

Remarques supplémentaires à prendre en compte lors de l'utilisation des exits de sécurité sur les canaux de cluster.

Pourquoi et quand exécuter cette tâche

Lorsqu'un canal émetteur de cluster est démarré pour la première fois, il utilise les attributs définis manuellement par un administrateur système. Lorsque le canal est arrêté et redémarré, il récupère les attributs de la définition de canal récepteur de cluster correspondante. La définition de canal émetteur de cluster d'origine est remplacée par les nouveaux attributs, y compris l'attribut SecurityExit .

Procédure

1. Vous devez définir un exit de sécurité à la fois sur l'extrémité émettrice du cluster et sur l'extrémité réceptrice du cluster d'un canal.

   La connexion initiale doit être établie avec un établissement de liaison d'exit de sécurité, même si le nom de l'exit de sécurité est envoyé à partir de la définition du récepteur de cluster.
2. Validez PartnerName dans la structure MQCXP de l'exit de sécurité.

   L'exit doit autoriser le démarrage du canal uniquement si le gestionnaire de files d'attente partenaire est autorisé
3. Concevez l'exit de sécurité sur la définition de récepteur de cluster à lancer.
4. Si vous le concevez comme étant initié par l'expéditeur, un gestionnaire de files d'attente non autorisé sans exit de sécurité peut rejoindre le cluster car aucun contrôle de sécurité n'est effectué.

   Ce n'est que lorsque le canal est arrêté et redémarré que le nom SCYEXIT peut être envoyé à partir de la définition du récepteur de cluster et que des contrôles de sécurité complets ont été effectués.
5. Pour afficher la définition de canal émetteur de cluster en cours d'utilisation, utilisez la commande suivante:

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   La commande affiche les attributs qui ont été envoyés à partir de la définition du récepteur de cluster.
6. Pour afficher la définition d'origine, utilisez la commande suivante:

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```
7. Vous devrez peut-être définir un exit de définition automatique de canal, CHADEXIT, sur le gestionnaire de files d'attente émetteur de cluster, si les gestionnaires de files d'attente se trouvent sur des plateformes différentes.

   Utilisez l'exit de définition automatique de canal pour définir l'attribut SecurityExit sur un format approprié pour la plateforme cible.
8. Déployez et configurez l'exit de sécurité.

   z/OS

   Le module de chargement de l'exit de sécurité doit se trouver dans le fichier spécifié dans l'instruction CSQXLIB DD de la procédure d'espace adresse de l'initiateur de canal.

   AIX, Linux, and Windows systèmes

   - La bibliothèque de liens dynamiques d'exit de sécurité doit se trouver dans le chemin indiqué dans l'attribut SCYEXIT de la définition de canal.

- La bibliothèque de liaison dynamique de l'exit de définition automatique de canal doit se trouver dans le chemin indiqué dans l'attribut CHADEXIT de la définition de gestionnaire de files d'attente.

Forcer les gestionnaires de files d'attente indésirables à quitter un cluster

Forcez un gestionnaire de files d'attente non souhaité à quitter un cluster en exécutant la commande RESET CLUSTER sur un gestionnaire de files d'attente de référentiel complet.

Pourquoi et quand exécuter cette tâche

Vous pouvez forcer un gestionnaire de files d'attente indésirable à quitter un cluster. Si, par exemple, un gestionnaire de files d'attente est supprimé mais que ses canaux récepteurs de cluster sont toujours définis dans le cluster. Vous voudrez peut-être ranger.

Seuls les gestionnaires de files d'attente de référentiel complet sont autorisés à éjecter un gestionnaire de files d'attente d'un cluster.

Remarque : Bien que l'utilisation de la commande RESET CLUSTER force la suppression d'un gestionnaire de files d'attente d'un cluster, l'utilisation de la commande RESET CLUSTER seule n'empêche pas le gestionnaire de files d'attente de rejoindre le cluster ultérieurement. Pour vous assurer que le gestionnaire de files d'attente ne rejoint pas le cluster, suivez les étapes décrites dans «Empêcher les gestionnaires de files d'attente de rejoindre un cluster», à la page 494.

Procédez comme suit pour éjecter le gestionnaire de files d'attente OSLO du cluster NORWAY:

Procédure

1. Sur un gestionnaire de files d'attente de référentiel complet, exécutez la commande suivante:

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. Vous pouvez également utiliser QMID à la place de QMNAME dans la commande:

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

   Remarque : QMID est une chaîne. Par conséquent, la valeur de qmid doit être placée entre apostrophes, par exemple, QMID('FR01_2019-07-15_14.42.42').

Résultats

Le gestionnaire de files d'attente qui est supprimé de force ne change pas ; ses définitions de cluster local indiquent qu'il se trouve dans le cluster. Les définitions de tous les autres gestionnaires de files d'attente ne l'affichent pas dans le cluster.

Empêcher les gestionnaires de files d'attente de recevoir des messages

Vous pouvez empêcher un gestionnaire de files d'attente de cluster de recevoir des messages qu'il n'est pas autorisé à recevoir à l'aide de programmes d'exit.

Pourquoi et quand exécuter cette tâche

Il est difficile d'empêcher un gestionnaire de files d'attente membre d'un cluster de définir une file d'attente. Il existe un risque qu'un gestionnaire de files d'attente incontrôlable rejoigne un cluster et définisse sa propre instance de l'une des files d'attente du cluster. Il peut désormais recevoir des messages qu'il n'est pas autorisé à recevoir. Pour empêcher un gestionnaire de files d'attente de recevoir des messages, utilisez l'une des options suivantes fournies dans la procédure.

Procédure

- Un programme d'exit de canal sur chaque canal émetteur de cluster. Le programme d'exit utilise le nom de connexion pour déterminer si le gestionnaire de files d'attente de destination est approprié pour l'envoi des messages.
- Un programme d'exit de charge de travail de cluster, qui utilise les enregistrements de destination pour déterminer l'adéquation de la file d'attente de destination et du gestionnaire de files d'attente pour l'envoi des messages.

SSL/TLS et clusters

Lors de la configuration de TLS pour les clusters, sachez qu'une définition de canal CLUSRCVR est propagée à d'autres gestionnaires de files d'attente en tant que canal CLUSSDR défini automatiquement. Si un canal CLUSRCVR utilise TLS, vous devez configurer TLS sur tous les gestionnaires de files d'attente qui communiquent à l'aide du canal.

Pour plus d'informations sur le protocole TLS, voir «Protocoles de sécurité TLS dans IBM MQ», à la page 25. Les conseils qui s'y appliquent sont généralement applicables aux canaux de cluster, mais vous souhaiterez peut-être accorder une attention particulière aux éléments suivants:

Dans un cluster IBM MQ , une définition de canal CLUSRCVR particulière est fréquemment propagée à de nombreux autres gestionnaires de files d'attente où elle est transformée en un CLUSSDRdéfini automatiquement. Par la suite, le CLUSSDR défini automatiquement est utilisé pour démarrer un canal vers CLUSRCVR. Si CLUSRCVR est configuré pour la connectivité TLS, les considérations suivantes s'appliquent:

- Tous les gestionnaires de files d'attente qui souhaitent communiquer avec ce CLUSRCVR doivent avoir accès au support TLS. Cette mise à disposition TLS doit prendre en charge le CipherSpec pour le canal.
- Les différents gestionnaires de files d'attente auxquels les canaux émetteurs de cluster définis automatiquement ont été propagés auront chacun un nom distinctif différent associé. Si la vérification d'homologue de nom distinctif doit être utilisée sur le CLUSRCVR , elle doit être configurée de sorte que tous les noms distinctifs pouvant être reçus soient correctement mis en correspondance.

  Par exemple, supposons que tous les gestionnaires de files d'attente qui hébergeront des canaux émetteurs de cluster qui se connecteront à un CLUSRCVRparticulier soient associés à des certificats. Supposons également que les noms distinctifs de tous ces certificats définissent le pays en tant que Royaume-Uni, l'organisation en tant que IBM, l'unité organisationnelle en tant que IBM MQ Development, et aient tous des noms communs sous la forme DEVT.QMnnn, où nnn est numérique.

  Dans ce cas, la valeur SSLPEER de C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM* sur le CLUSRCVR permet à tous les canaux émetteurs de cluster requis de se connecter correctement, mais empêche les canaux émetteurs de cluster indésirables de se connecter.
- Si des chaînes CipherSpec personnalisées sont utilisées, sachez que les formats de chaîne personnalisés ne sont pas autorisés sur toutes les plateformes. Par exemple, la CipherSpec chaîne RC4_SHA_US a la valeur 05 on IBM i mais n'est pas une spécification valide sur les systèmes AIX, Linux, and Windows . Ainsi, si des paramètres SSLCIPH personnalisés sont utilisés sur un CLUSRCVR, tous les canaux émetteurs de cluster définis automatiquement doivent résider sur des plateformes sur lesquelles le support TLS sous-jacent implémente ce CipherSpec et sur lesquelles il peut être spécifié avec la valeur personnalisée. Si vous ne pouvez pas sélectionner une valeur pour le paramètre SSLCIPH qui sera comprise dans votre cluster, vous aurez besoin d'un exit de définition automatique de canal pour que les plateformes utilisées puissent la comprendre. Utilisez les chaînes textuelles CipherSpec lorsque cela est possible (par exemple, TLS_RSA_WITH_AES_128_CBC_SHA).

Un paramètre SSLCRLNL s'applique à un gestionnaire de files d'attente individuel et n'est pas propagé à d'autres gestionnaires de files d'attente au sein d'un cluster.

Mise à niveau des gestionnaires de files d'attente en cluster et des canaux vers SSL/TLS

Mettez à niveau les canaux de cluster un par un, en modifiant tous les canaux CLUSRCVR avant les canaux CLUSSDR .

Avant de commencer

Tenez compte des considérations suivantes, car elles peuvent affecter votre choix de CipherSpec pour un cluster:

- Certains CipherSpecs ne sont pas disponibles sur toutes les plateformes. Prenez soin de choisir un CipherSpec pris en charge par tous les gestionnaires de files d'attente du cluster.
- Certains CipherSpecs peuvent être nouveaux dans la version actuelle de IBM MQ et ne pas être pris en charge dans les versions plus anciennes. Un cluster contenant des gestionnaires de files d'attente s'exécutant dans différentes éditions de MQ ne peut utiliser que les CipherSpecs prises en charge par chaque édition.

  Pour utiliser un nouveau CipherSpec dans un cluster, vous devez d'abord migrer tous les gestionnaires de files d'attente de cluster vers l'édition en cours.
- Certains CipherSpecs nécessitent l'utilisation d'un type spécifique de certificat numérique, notamment ceux qui utilisent la cryptographie Elliptic Curve.

Avertissement : Il n'est pas possible d'utiliser un mélange de certificats signés par Elliptic Curve et de certificats signés par RSA sur les gestionnaires de files d'attente que vous souhaitez joindre dans le cadre d'un cluster.

Les gestionnaires de files d'attente d'un cluster doivent tous utiliser des certificats signés par RSA ou tous utiliser des certificats signés par EC, et non une combinaison des deux.

Pour plus d'informations, voir «Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49.

Mettez à niveau tous les gestionnaires de files d'attente du cluster vers IBM MQ V8 ou ultérieure, s'ils ne sont pas déjà à ces niveaux. Distribuez les certificats et les clés pour que TLS fonctionne à partir de chacun d'eux.

Avant de pouvoir effectuer une mise à niveau vers ou utiliser l'un des alias CipherSpecs (ANY_TLS13, ANY_TLS13_OR_HIGHER, ANY_TLS12, ANY_TLS12_OR_HIGHER, etc.), vous devez mettre à niveau vos gestionnaires de files d'attente:

- Multi Mettez à niveau tous les gestionnaires de files d'attente IBM MQ for Multiplatforms du cluster vers la IBM MQ 9.1.4 ou une version ultérieure.
- z/OS Mettez à niveau tous les gestionnaires de files d'attente IBM MQ for z/OS du cluster vers la IBM MQ for z/OS 9.2.0 ou une version ultérieure.

Vous devez

Pourquoi et quand exécuter cette tâche

Modifiez les canaux CLUSRCVR avant les canaux CLUSSDR .

Procédure

1. Basculez les canaux CLUSRCVR vers TLS dans l'ordre de votre choix, en modifiant un CLUSRCVR à la fois, et autorisez les modifications à circuler dans le cluster avant de modifier le suivant.

   Important : Veillez à ne pas modifier le chemin inverse tant que les modifications du canal en cours n'ont pas été distribuées dans le cluster.
2. Facultatif : Commuter tous les canaux CLUSSDR manuels vers TLS.

Cela n'a aucun effet sur le fonctionnement du cluster, sauf si vous utilisez la commande REFRESH CLUSTER avec l'option REPOS(YES).

Remarque : Pour les clusters de grande taille, l'utilisation de la commande **REFRESH CLUSTER** peut perturber le cluster pendant qu'il est en cours, et à nouveau à des intervalles de 27 jours par la suite lorsque les objets de cluster envoient automatiquement des mises à jour de statut à tous les gestionnaires de files d'attente intéressés. Voir L'actualisation d'un grand cluster peut affecter les performances et la disponibilité du cluster.

3. Utilisez la commande DISPLAY CLUSQMGR pour vous assurer que la nouvelle configuration de sécurité a été propagée dans le cluster.
4. Redémarrez les canaux pour utiliser TLS et exécutez REFRESH SECURITY (SSL).

Concepts associés

«Activation des CipherSpecs», à la page 435
Activez un CipherSpec à l'aide du paramètre **SSLCIPH** dans la commande **DEFINE CHANNEL** ou **ALTER CHANNEL** MQSC.

«Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49
Cette rubrique fournit des informations sur la façon de choisir les CipherSpecs et les certificats numériques appropriés pour votre règle de sécurité, en soulignant la relation entre les CipherSpecs et les certificats numériques dans IBM MQ.

Information associée

Mise en cluster : meilleures pratiques d'utilisation REFRESH CLUSTER

Désactivation de SSL/TLS sur les gestionnaires de files d'attente et les canaux en cluster

Pour désactiver TLS, définissez le paramètre SSLCIPH sur ' '. Désactivez TLS sur les canaux de cluster individuellement, en modifiant tous les canaux récepteurs de cluster avant les canaux émetteurs de cluster.

Pourquoi et quand exécuter cette tâche

Modifiez un canal récepteur de cluster à la fois et autorisez les modifications à transiter par le cluster avant de modifier le suivant.

Important : Veillez à ne pas modifier le chemin inverse tant que les modifications du canal en cours n'ont pas été distribuées dans le cluster.

Procédure

1. Définissez la valeur du paramètre SSLCIPH sur ' ', une chaîne vide entre apostrophes IBM i ou *NONE sur IBM i.

   Vous pouvez désactiver TLS sur les canaux récepteurs de cluster dans l'ordre de votre choix.

   Notez que les modifications circulent dans la direction opposée sur les canaux sur lesquels vous laissez TLS actif.
2. Vérifiez que la nouvelle valeur est reflétée dans tous les autres gestionnaires de files d'attente à l'aide de la commande **DISPLAY CLUSQMGR(*)** ALL.
3. Désactivez TLS sur tous les canaux émetteurs de cluster manuels.

   Cela n'a aucun effet sur le fonctionnement du cluster, sauf si vous utilisez la commande **REFRESH CLUSTER** avec l'option REPOS (YES).

   Pour les clusters de grande taille, l'utilisation de la commande **REFRESH CLUSTER** peut perturber le cluster pendant qu'il est en cours, puis à intervalles réguliers, lorsque les objets de cluster envoient automatiquement des mises à jour de statut à tous les gestionnaires de files d'attente intéressés. Pour plus d'informations, voir La régénération dans un cluster de grande taille peut affecter les performances et la disponibilité du cluster.

4. Arrêtez et redémarrez les canaux émetteurs de cluster.

Sécurité de publication / abonnement

Les composants et les interactions impliqués dans la publication / l'abonnement sont décrits comme une introduction aux explications et exemples plus détaillés qui suivent.

Un certain nombre de composants sont impliqués dans la publication et l'abonnement à une rubrique. Certaines des relations de sécurité entre eux sont illustrées dans Figure 22, à la page 500 et décrites dans l'exemple suivant.

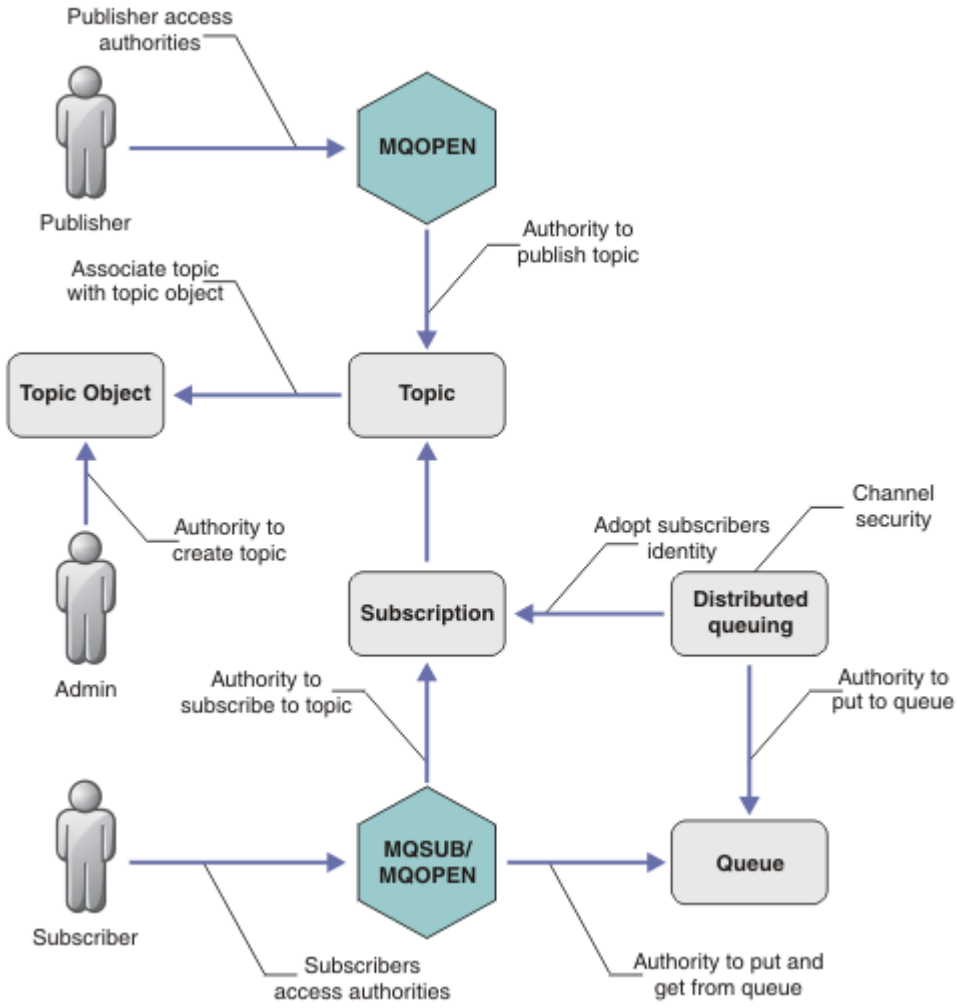


Figure 22. Relations de sécurité de publication / abonnement entre différents composants

Rubriques

Les rubriques sont identifiées par des chaînes de rubrique et sont généralement organisées en arborescences. Voir Arborescences de rubriques. Vous devez associer une rubrique à un objet de rubrique pour contrôler l'accès à la rubrique. «Modèle de sécurité de rubrique», à la page 502 explique comment sécuriser des rubriques à l'aide d'objets de rubrique.

Objets de rubrique d'administration

Vous pouvez contrôler qui a accès à une rubrique et dans quel but, à l'aide de la commande **setmqaut** avec une liste d'objets de rubrique d'administration. Consultez les exemples, «Accorder l'accès à un utilisateur pour s'abonner à une rubrique», à la page 507 et «Accorder l'accès à un utilisateur pour la publication dans une rubrique», à la page 515.

z/OS Pour contrôler l'accès aux objets de rubrique sur z/OS, voir Profils pour la sécurité des rubriques.

Abonnements

Abonnez-vous à une ou plusieurs rubriques en créant un abonnement fournissant une chaîne de rubrique, qui peut inclure des caractères génériques, à mettre en correspondance avec les chaînes de rubrique des publications. Pour plus de détails, voir:

S'abonner à l'aide d'un objet de rubrique
«Abonnement à l'aide du nom d'objet de rubrique», à la page 504

S'abonner à l'aide d'une rubrique
«Abonnement à l'aide d'une chaîne de rubrique dans laquelle le noeud de rubrique n'existe pas», à la page 504

S'abonner à l'aide d'une rubrique avec des caractères génériques
«Abonnement à l'aide d'une chaîne de sujet contenant des caractères génériques», à la page 505

Un abonnement contient des informations sur l'identité de l'abonné et sur l'identité de la file d'attente de destination dans laquelle les publications doivent être placées. Il contient également des informations sur la manière dont la publication doit être placée dans la file d'attente de destination.

En plus de définir les abonnés qui ont le droit de s'abonner à certaines rubriques, vous pouvez limiter les abonnements à l'utilisation par un abonné individuel. Vous pouvez également contrôler les informations sur l'abonné qui sont utilisées par le gestionnaire de files d'attente lorsque des publications sont placées dans la file d'attente de destination. Voir «Sécurité des abonnements», à la page 521.

Files d'attente

La file d'attente de destination est une file d'attente importante à sécuriser. Il est local pour l'abonné et les publications qui correspondent à l'abonnement y sont placées. Vous devez envisager d'accéder à la file d'attente de destination à partir de deux perspectives:

1. Insertion d'une publication dans la file d'attente de destination.
2. Extraction de la publication de la file d'attente de destination.

Le gestionnaire de files d'attente place une publication dans la file d'attente de destination à l'aide d'une identité fournie par l'abonné. L'abonné, ou un programme auquel la tâche d'obtention de publications a été déléguée, enlève les messages de la file d'attente. Voir «Droits d'accès aux files d'attente de destination», à la page 505.

Il n'existe pas d'alias d'objet de rubrique, mais vous pouvez utiliser une file d'attente alias comme alias d'un objet de rubrique. Dans ce cas, en plus de vérifier les droits d'utilisation de la rubrique pour la publication ou l'abonnement, le gestionnaire de files d'attente vérifie les droits d'utilisation de la file d'attente.

«Sécurité de publication / abonnement entre les gestionnaires de files d'attente», à la page 522

Votre droit de publication ou d'abonnement à une rubrique est vérifié sur le gestionnaire de files d'attente local à l'aide des identités et des autorisations locales. L'autorisation ne dépend pas de la définition ou non de la rubrique, ni de l'endroit où elle est définie. Par conséquent, vous devez effectuer une autorisation de rubrique sur chaque gestionnaire de files d'attente d'un cluster lorsque des rubriques en cluster sont utilisées.

Remarque : Le modèle de sécurité des rubriques diffère du modèle de sécurité des files d'attente. Vous pouvez obtenir le même résultat pour les files d'attente en définissant un alias de file d'attente en local pour chaque file d'attente en cluster.

Les gestionnaires de files d'attente échangent des abonnements dans un cluster. Dans la plupart des configurations de cluster IBM MQ , les canaux sont configurés avec PUTAUT=DEF pour placer des messages dans des files d'attente cible en utilisant les droits du processus de canal. Vous pouvez modifier la configuration de canal pour utiliser PUTAUT=CTX afin que l'utilisateur abonné ait le droit de propager un abonnement à un autre gestionnaire de files d'attente dans un cluster.

«Sécurité de publication / abonnement entre les gestionnaires de files d'attente», à la page 522 décrit comment modifier vos définitions de canal pour contrôler qui est autorisé à propager des abonnements sur d'autres serveurs du cluster.

Autorisation

Vous pouvez appliquer une autorisation à des objets de rubrique, tout comme des files d'attente et d'autres objets. Il existe trois opérations d'autorisation, pub, subet `resume` , que vous pouvez appliquer uniquement aux rubriques. Les détails sont décrits dans Spécification des droits pour différents types d'objet.

Appels de fonction

Dans les programmes de publication et d'abonnement, comme dans les programmes en file d'attente, des vérifications d'autorisation sont effectuées lorsque des objets sont ouverts, créés, modifiés ou supprimés. Les vérifications ne sont pas effectuées lorsque des appels `MQPUT` ou `MQGET` `MQI` sont effectués pour placer et obtenir des publications.

Pour publier une rubrique, effectuez un `MQOPEN` sur la rubrique, qui effectue les vérifications d'autorisation. Publiez des messages dans le descripteur de rubrique à l'aide de la commande `MQPUT` , qui n'effectue aucune vérification d'autorisation.

Pour vous abonner à une rubrique, vous exécutez généralement une commande `MQSUB` pour créer ou reprendre l'abonnement, ainsi que pour ouvrir la file d'attente de destination afin de recevoir des publications. Vous pouvez également effectuer un `MQOPEN` distinct pour ouvrir la file d'attente de destination, puis exécuter la commande `MQSUB` pour créer ou reprendre l'abonnement.

Quels que soient les appels que vous utilisez, le gestionnaire de files d'attente vérifie que vous pouvez vous abonner à la rubrique et obtenir les publications résultantes de la file d'attente de destination. Si la file d'attente de destination n'est pas gérée, des vérifications d'autorisation sont également effectuées pour que le gestionnaire de files d'attente puisse placer des publications dans la file d'attente de destination. Il utilise l'identité qu'il a adoptée à partir d'un abonnement correspondant. Il est supposé que le gestionnaire de files d'attente est toujours en mesure de placer des publications dans des files d'attente de destination gérées.

Rôles

Les utilisateurs sont impliqués dans quatre rôles lors de l'exécution d'applications de publication / abonnement:

1. Diffuseur de publications
2. Abonné
3. Administrateur de rubriques
4. IBM MQ Administrateur-membre du groupe `mqm`

Définissez des groupes avec les autorisations appropriées correspondant aux rôles de publication, d'abonnement et d'administration de sujet. Vous pouvez ensuite affecter des principaux à ces groupes en les autorisant à effectuer des tâches de publication et d'abonnement spécifiques.

En outre, vous devez étendre les autorisations d'opérations d'administration à l'administrateur des files d'attente et des canaux en charge du déplacement des publications et des abonnements.

Modèle de sécurité de rubrique

Seuls les objets de rubrique définis peuvent être associés à des attributs de sécurité. Pour obtenir une description des objets de rubrique, voir Objets de rubrique d'administration. Les attributs de sécurité indiquent si un ID utilisateur ou un groupe de sécurité spécifié est autorisé à effectuer une opération d'abonnement ou de publication sur chaque objet de rubrique.

Les attributs de sécurité sont associés au noeud d'administration approprié dans l'arborescence de rubriques. Lorsqu'une vérification des droits est effectuée pour un ID utilisateur particulier lors d'une opération d'abonnement ou de publication, les droits accordés sont basés sur les attributs de sécurité du noeud d'arborescence de rubriques associé.

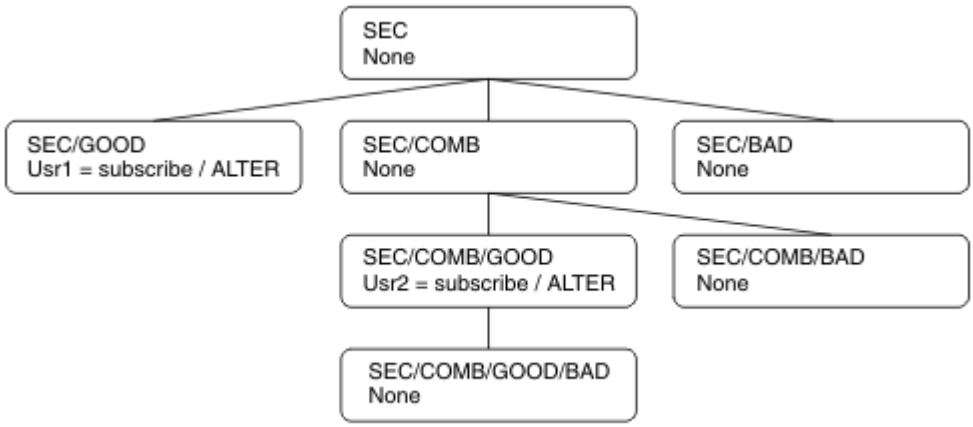
Les attributs de sécurité sont une liste de contrôle d'accès qui indique les droits d'accès d'un ID utilisateur ou d'un groupe de sécurité du système d'exploitation sur l'objet de rubrique.

Prenez l'exemple suivant dans lequel les objets de rubrique ont été définis avec les attributs de sécurité ou les droits affichés:

Tableau 87. Exemples de droits sur les objets de rubrique

Nom de la rubrique	**Chaîne de rubrique**	**Droits d'accès-Multiplatforms**	**z/OS droits**
SECROOT	SEC	Aucun	Aucun
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	Aucun	Aucun HLQ.SUBSCRIBE.SECBAD
SECCOMB	SEC/COMB	Aucun	Aucun HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/ GOOD/BAD	Aucun	Aucun HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	Aucun	Aucun HLQ.SUBSCRIBE.SECCOMBN

L'arborescence de rubriques avec les attributs de sécurité associés sur chaque noeud peut être représentée comme suit:



Les exemples répertoriés donnent les autorisations suivantes:

- Sur le noeud racine de l'arborescence /SEC, aucun utilisateur n'a de droits sur ce noeud.
- usr1 a reçu le droit d'abonnement à l'objet /SEC/GOOD
- usr2 a reçu le droit d'abonnement à l'objet /SEC/COMB/GOOD

Abonnement à l'aide du nom d'objet de rubrique

Lors de l'abonnement à un objet de rubrique en spécifiant le nom MQCHAR48 , le noeud correspondant dans l'arborescence de rubriques est localisé. Si les attributs de sécurité associés au noeud indiquent que l'utilisateur est autorisé à s'abonner, l'accès est accordé.

Si l'accès n'est pas accordé à l'utilisateur, le noeud parent de l'arborescence détermine si l'utilisateur est autorisé à s'abonner au niveau du noeud parent. Si tel est le cas, l'accès est accordé. Si ce n'est pas le cas, le parent de ce noeud est pris en compte. La récursivité se poursuit jusqu'à ce qu'un noeud soit localisé et accorde le droit d'abonnement à l'utilisateur. La récursivité s'arrête lorsque le noeud racine est pris en compte sans que les droits aient été accordés. Dans ce dernier cas, l'accès est refusé.

En résumé, si un noeud du chemin accorde le droit de s'abonner à cet utilisateur ou à cette application, l'abonné est autorisé à s'abonner à ce noeud ou à n'importe quel emplacement situé en dessous de ce noeud dans l'arborescence de rubriques.

Le noeud racine de l'exemple est SEC.

Le droit d'abonnement est accordé à l'utilisateur si la liste de contrôle d'accès indique que l'ID utilisateur lui-même dispose de droits ou qu'un groupe de sécurité du système d'exploitation dont l'ID utilisateur est membre dispose de droits.

Ainsi, par exemple:

- Si `usr1` tente de s'abonner à l'aide d'une chaîne de rubrique `SEC/GOOD`, l'abonnement est autorisé car l'ID utilisateur a accès au noeud associé à cette rubrique. Toutefois, si `usr1` tente de s'abonner à l'aide de la chaîne de rubrique `SEC/COMB/GOOD` , l'abonnement ne sera pas autorisé car l'ID utilisateur n'a pas accès au noeud qui lui est associé.
- Si `usr2` tente de s'abonner, à l'aide d'une chaîne de rubrique `SEC/COMB/GOOD` , l'abonnement est autorisé car l'ID utilisateur a accès au noeud associé à la rubrique. Toutefois, si `usr2` tentait de s'abonner à `SEC/GOOD` , l'abonnement ne serait pas autorisé car l'ID utilisateur n'a pas accès au noeud qui lui est associé.
- Si `usr2` tente de s'abonner à l'aide d'une chaîne de rubrique `SEC/COMB/GOOD/BAD` , l'abonnement est autorisé car l'ID utilisateur a accès au noeud parent `SEC/COMB/GOOD`.
- Si `usr1` ou `usr2` tente de s'abonner à l'aide d'une chaîne de rubrique `/SEC/COMB/BAD`, aucune n'est autorisée car ils n'ont pas accès au noeud de rubrique qui lui est associé, ni aux noeuds parent de cette rubrique.

Une opération d'abonnement spécifiant le nom d'un objet de rubrique qui n'existe pas génère une erreur MQRC_UNKNOWN_OBJECT_NAME.

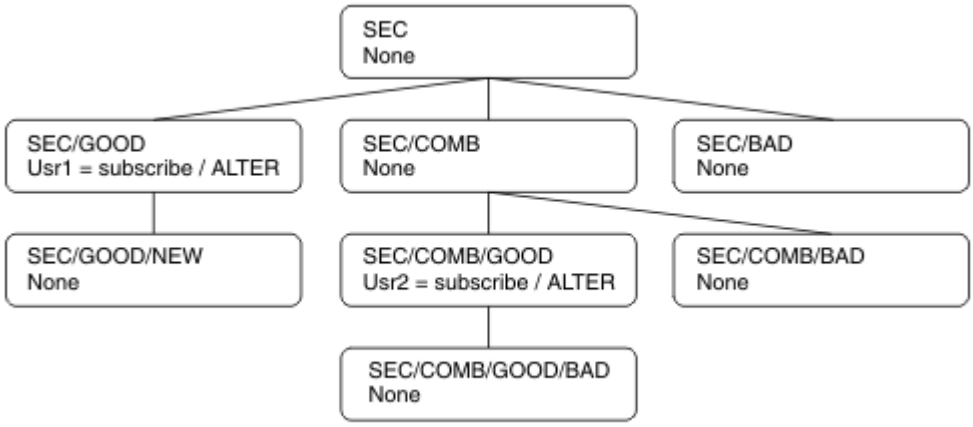
Abonnement à l'aide d'une chaîne de rubrique dans laquelle le noeud de rubrique existe

Le comportement est le même que lors de la spécification de la rubrique par le nom d'objet MQCHAR48 .

Abonnement à l'aide d'une chaîne de rubrique dans laquelle le noeud de rubrique n'existe pas

Prenons le cas d'une application abonnée, en spécifiant une chaîne de rubrique représentant un noeud de rubrique qui n'existe pas actuellement dans l'arborescence de rubriques. La vérification des droits d'accès est effectuée comme indiqué dans la section précédente. La vérification commence par le noeud parent de celui qui est représenté par la chaîne de rubrique. Si les droits sont accordés, un nouveau noeud représentant la chaîne de rubrique est créé dans l'arborescence de rubriques.

Par exemple, `usr1` tente de s'abonner à une rubrique `SEC/GOOD/NEW`. Les droits sont accordés car `usr1` a accès au noeud parent `SEC/GOOD`. Un nouveau noeud de rubrique est créé dans l'arborescence, comme le montre le diagramme suivant. Le nouveau noeud de rubrique n'est pas un objet de rubrique auquel aucun attribut de sécurité n'est directement associé ; les attributs sont hérités de son parent.



Abonnement à l'aide d'une chaîne de sujet contenant des caractères génériques

Prenez en compte le cas de l'abonnement à l'aide d'une chaîne de rubrique contenant un caractère générique. La vérification des droits est réalisée sur le noeud dans l'arborescence de sujets qui correspond à la partie qualifiée complète de la chaîne de sujet.

Par conséquent, si une application s'abonne à SEC/COMB/GOOD/*, une vérification des droits d'accès est effectuée comme indiqué dans les deux sections précédentes sur le noeud SEC/COMB/GOOD dans l'arborescence de rubriques.

De même, si une application doit s'abonner à SEC/COMB/*/GOOD, une vérification des droits d'accès est effectuée sur le noeud SEC/COMB.

Droits d'accès aux files d'attente de destination

Lors de l'abonnement à une rubrique, l'un des paramètres est le descripteur hobj d'une file d'attente qui a été ouverte pour la sortie afin de recevoir les publications.

Si hobj n'est pas spécifié, mais qu'il est vide, une file d'attente gérée est créée si les conditions suivantes s'appliquent:

- L'option MQSO_MANAGED a été spécifiée.
- L'abonnement n'existe pas.
- La création est spécifiée.

Si hobj est vide et que vous modifiez ou reprenez un abonnement existant, la file d'attente de destination précédemment fournie peut être gérée ou non gérée.

L'application ou l'utilisateur qui effectue la demande MQSUB doit avoir le droit d'insérer des messages dans la file d'attente de destination qu'elle a fournie ; en effet, il doit avoir le droit d'insérer des messages publiés dans cette file d'attente. La vérification des droits d'accès suit les règles existantes pour la vérification de la sécurité de la file d'attente.

La vérification de la sécurité inclut un ID utilisateur alternatif et des vérifications de la sécurité du contexte, le cas échéant. Pour pouvoir définir l'une des zones de contexte d'identité, vous devez spécifier l'option MQSO_SET_IDENTITY_CONTEXT ainsi que l'option MQSO_CREATE ou MQSO_ALTER . Vous ne pouvez pas définir de zones de contexte d'identité dans une demande MQSO_RESUME .

Si la destination est une file d'attente gérée, aucun contrôle de sécurité n'est effectué sur la destination gérée. Si vous êtes autorisé à vous abonner à une rubrique, il est supposé que vous pouvez utiliser des destinations gérées.

Publication à l'aide du nom de rubrique ou de la chaîne de rubrique dans laquelle le noeud de rubrique existe

Le modèle de sécurité pour la publication est le même que pour l'abonnement, à l'exception des caractères génériques. Les publications ne contiennent pas de caractères génériques ; il n'y a donc pas de cas d'une chaîne de rubrique contenant des caractères génériques à prendre en compte.

Les droits de publication et d'abonnement sont distincts. Un utilisateur ou un groupe peut avoir le droit d'en effectuer un sans être nécessairement en mesure d'en effectuer un autre.

Lors de la publication dans un objet de rubrique en spécifiant le nom MQCHAR48 ou la chaîne de rubrique, le noeud correspondant dans l'arborescence de rubriques est localisé. Si les attributs de sécurité associés au noeud de rubrique indiquent que l'utilisateur est autorisé à publier, l'accès est accordé.

Si l'accès n'est pas accordé, le noeud parent de l'arborescence détermine si l'utilisateur a le droit de publier à ce niveau. Si tel est le cas, l'accès est accordé. Si ce n'est pas le cas, la récursivité se poursuit jusqu'à ce qu'un noeud soit localisé et accorde le droit de publication à l'utilisateur. La récursivité s'arrête lorsque le noeud racine est pris en compte sans que les droits aient été accordés. Dans ce dernier cas, l'accès est refusé.

En bref, si un noeud du chemin accorde le droit de publier à cet utilisateur ou à cette application, le diffuseur de publications est autorisé à publier sur ce noeud ou n'importe où en dessous de ce noeud dans l'arborescence de rubriques.

Publication à l'aide du nom de rubrique ou de la chaîne de rubrique où le noeud de rubrique n'existe pas

Comme pour l'opération d'abonnement, lorsqu'une application publie, en spécifiant une chaîne de rubrique représentant un noeud de rubrique qui n'existe pas actuellement dans l'arborescence de rubriques, la vérification des droits d'accès est effectuée en commençant par le parent du noeud représenté par la chaîne de rubrique. Si les droits sont accordés, un nouveau noeud représentant la chaîne de rubrique est créé dans l'arborescence de rubriques.

Publication à l'aide d'une file d'attente alias qui se résout en un objet de rubrique

Si vous publiez à l'aide d'une file d'attente alias qui se résout en un objet de rubrique, la vérification de la sécurité est effectuée à la fois sur la file d'attente alias et sur la rubrique sous-jacente à laquelle elle se résout.

Le contrôle de sécurité de la file d'attente alias vérifie que l'utilisateur est autorisé à placer des messages dans cette file d'attente alias et le contrôle de sécurité de la rubrique vérifie que l'utilisateur peut publier des messages dans cette rubrique. Lorsqu'une file d'attente alias est résolue en une autre file d'attente, les vérifications ne sont pas effectuées sur la file d'attente sous-jacente. La vérification des droits d'accès est effectuée différemment pour les rubriques et les files d'attente.

Fermeture d'un abonnement

Un contrôle de sécurité supplémentaire est effectué si vous fermez un abonnement à l'aide de l'option `MQCO_REMOVE_SUB` si vous n'avez pas créé l'abonnement sous ce descripteur.

Un contrôle de sécurité est effectué pour vous assurer que vous disposez des droits appropriés pour effectuer cette opération, car l'action entraîne la suppression de l'abonnement. Si les attributs de sécurité associés au noeud de rubrique indiquent que l'utilisateur dispose de droits d'accès, l'accès est accordé. Si ce n'est pas le cas, le noeud parent de l'arborescence est pris en compte pour déterminer si l'utilisateur a le droit de fermer l'abonnement. La récursivité se poursuit jusqu'à ce que les droits soient accordés ou que le noeud racine soit atteint.

Définition, modification et suppression d'un abonnement

Aucun contrôle de sécurité d'abonnement n'est effectué lorsqu'un abonnement est créé de manière administrative au lieu d'utiliser une demande d'API MQSUB . Ce droit a déjà été accordé à l'administrateur via la commande.

Des contrôles de sécurité sont effectués pour s'assurer que les publications peuvent être placées dans la file d'attente de destination associée à l'abonnement. Les vérifications sont effectuées de la même manière que pour une demande MQSUB .

L'ID utilisateur utilisé pour ces contrôles de sécurité dépend de la commande émise. Si le paramètre **SUBUSER** est spécifié, il affecte la manière dont la vérification est effectuée, comme illustré dans la Tableau 88, à la page 507:

Tableau 88. ID utilisateur utilisés pour les contrôles de sécurité des commandes

Commande	SUBUSER indiqué et vide	SUBUSER indiqué et terminé	SUBUSER non indiqué
	Utiliser l'ID administrateur		Utiliser l'ID utilisateur de l'abonnement LIKE
	Utiliser l'ID administrateur		Utilisez l'ID.DEFAULT.SU utilisateurB de SYSTEMabonnement -si la zone est vide, utilisez l'ID administrateur
	Utiliser l'ID administrateur		Utiliser l'ID utilisateur de l'abonnement existant

Le seul contrôle de sécurité effectué lors de la suppression d'abonnements à l'aide de la commande DELETE SUB est le contrôle de sécurité de la commande.

Exemple de configuration de la sécurité de publication / abonnement

Cette section décrit un scénario dans lequel le contrôle d'accès est configuré sur les rubriques de manière à permettre l'application du contrôle de sécurité selon les besoins.

Accorder l'accès à un utilisateur pour s'abonner à une rubrique

Cette rubrique est la première d'une liste de tâches qui vous indique comment accorder l'accès aux rubriques à plusieurs utilisateurs.

Pourquoi et quand exécuter cette tâche

Cette tâche suppose qu'aucun objet de rubrique d'administration n'existe et qu'aucun profil n'a été défini pour l'abonnement ou la publication. Les applications créent de nouveaux abonnements, plutôt que de reprendre des abonnements existants, et utilisent uniquement la chaîne de rubrique.

Une application peut s'abonner en fournissant un objet de rubrique, une chaîne de rubrique ou une combinaison des deux. Quelle que soit la façon dont l'application sélectionne, l'effet est de créer un

abonnement à un certain point de l'arborescence de rubriques. Si ce point de l'arborescence de rubriques est représenté par un objet de rubrique d'administration, un profil de sécurité est vérifié en fonction du nom de cet objet de rubrique.

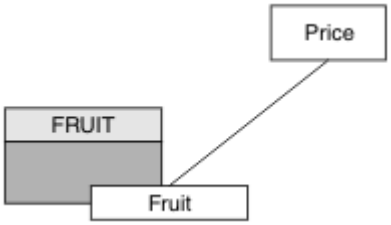


Figure 23. Exemple d'accès à un objet de rubrique

Tableau 89. Exemple d'accès à un objet de rubrique

Topic	Accès à l'abonnement requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun
Prix / Fruits	USER1	fruit

Définissez un nouvel objet de rubrique comme suit:

Procédure

1. Exécutez la commande MQSC `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')`.
2. Accordez l'accès comme suit:

- z/OS **z/OS** :

  Accordez l'accès à USER1 pour vous abonner à la rubrique `"Price/Fruit"` en accordant à l'utilisateur l'accès au profil `hlq.SUBSCRIBE.FRUIT` . Pour ce faire, utilisez les commandes RACF suivantes:

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi **Multiplatforms**:

  Accordez l'accès à USER1 pour vous abonner à la rubrique `"Price/Fruit"` en accordant à l'utilisateur l'accès à l'objet `FRUIT` . Pour ce faire, utilisez la commande d'autorisation pour la plateforme:

  ALW **AIX, Linux, and Windows systèmes**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

Résultats

Lorsque USER1 tente de s'abonner à la rubrique `"Price/Fruit"` , le résultat est un succès.

Lorsque USER2 tente de s'abonner à la rubrique `"Price/Fruit"` , le résultat est un échec avec un message `MQRC_NOT_AUTHORIZED` , ainsi que:

- z/OS Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Sous AIX, Linux, and Windows, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i Sous IBMi, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Notez qu'il s'agit d'une illustration de ce que vous voyez ; pas de tous les champs.

Accorder l'accès à un utilisateur pour s'abonner à une rubrique plus en profondeur dans l'arborescence

Cette rubrique est la deuxième d'une liste de tâches qui vous indique comment accorder l'accès aux rubriques à plusieurs utilisateurs.

Avant de commencer

Cette rubrique utilise la configuration décrite dans «Accorder l'accès à un utilisateur pour s'abonner à une rubrique», à la page 507.

Pourquoi et quand exécuter cette tâche

Si le point dans l'arborescence de rubriques où l'application effectue l'abonnement n'est pas représenté par un objet de rubrique d'administration, déplacez l'arborescence vers le haut jusqu'à ce que l'objet de rubrique d'administration parent le plus proche soit localisé. Le profil de sécurité est vérifié en fonction du nom de cet objet de rubrique.

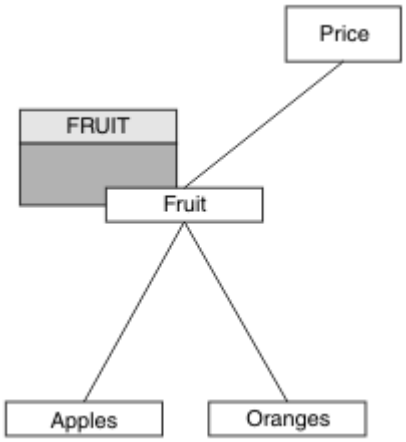


Figure 24. Exemple d'octroi d'accès à une rubrique dans une arborescence de rubriques

Tableau 90. Exigences d'accès pour des exemples de rubriques et d'objets de rubrique

Topic	Accès à l'abonnement requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun
Prix / Fruits	USER1	fruit
Prix / Fruits / Pommes	USER1	
Prix / Fruits / Oranges	USER1	

Dans «Accorder l'accès à un utilisateur pour s'abonner à une rubrique», à la page 507, USER1 a été autorisé à s'abonner à la rubrique `"Price/Fruit"` en lui accordant l'accès au profil `hlq.SUBSCRIBE.FRUIT` sur z/OS et l'accès à l'abonnement au profil FRUIT sur Multiplatforms. Ce profil unique accorde également à USER1 l'accès pour s'abonner à `"Price/Fruit/Apples"`, `"Price/Fruit/Oranges"` et `"Price/Fruit/#"`.

Lorsque USER1 tente de s'abonner à la rubrique `"Price/Fruit/Apples"` , le résultat est un succès.

Lorsque USER2 tente de s'abonner à la rubrique `"Price/Fruit/Apples"` , le résultat est un échec avec un message `MQRC_NOT_AUTHORIZED` , ainsi que:

- z/OS Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Multi Sur Multiplatforms, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

Notez ce qui suit :

- z/OS Les messages que vous recevez sur z/OS sont identiques à ceux reçus lors de la tâche précédente car les mêmes objets de rubrique et les mêmes profils contrôlent l'accès.
- Multi Le message d'événement que vous recevez sur Multiplatforms est similaire à celui reçu dans la tâche précédente, mais la chaîne de rubrique réelle est différente.

Accorder à un autre utilisateur l'accès permettant de s'abonner uniquement à la rubrique située plus en profondeur dans l'arborescence

Cette rubrique est la troisième d'une liste de tâches qui vous indique comment accorder l'accès à l'abonnement à des rubriques par plusieurs utilisateurs.

Avant de commencer

Cette rubrique utilise la configuration décrite dans «Accorder l'accès à un utilisateur pour s'abonner à une rubrique plus en profondeur dans l'arborescence», à la page 509.

Pourquoi et quand exécuter cette tâche

Dans «Accorder l'accès à un utilisateur pour s'abonner à une rubrique plus en profondeur dans l'arborescence», à la page 509, l'accès à la rubrique `"Price/Fruit/Apples"`a été refusé à USER2 . Cette rubrique vous explique comment accorder l'accès à cette rubrique, mais pas à d'autres rubriques.

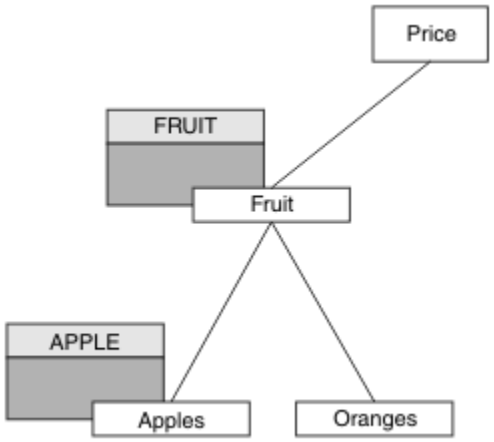


Figure 25. Octroi de l'accès à des rubriques spécifiques dans une arborescence de rubriques

Tableau 91. Exigences d'accès pour des exemples de rubriques et d'objets de rubrique

Topic	Accès à l'abonnement requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun
Prix / Fruits	USER1	fruit
Prix / Fruits / Pommes	USER1 et USER2	Apple
Prix / Fruits / Oranges	USER1	

Définissez un nouvel objet de rubrique comme suit:

Procédure

1. Exécutez la commande MQSC `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`.

2. Accordez l'accès comme suit:

- z/OS **z/OS** :

  Dans «Accorder l'accès à un utilisateur pour s'abonner à une rubrique plus en profondeur dans l'arborescence», à la page 509 , USER1 a été autorisé à s'abonner à la rubrique `"Price/Fruit/Apples"` en accordant à l'utilisateur l'accès au profil `hlq.SUBSCRIBE.FRUIT` .

  Ce profil unique a également accordé à USER1 l'accès pour s'abonner à `"Price/Fruit/Oranges"` `"Price/Fruit/#"` et cet accès reste même avec l'ajout du nouvel objet de rubrique et des profils qui lui sont associés.

  Accordez l'accès à USER2 pour vous abonner à la rubrique `"Price/Fruit/Apples"` en accordant à l'utilisateur l'accès au profil `hlq.SUBSCRIBE.APPLE` . Pour ce faire, utilisez les commandes RACF suivantes:

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
  ```

- Multi Multiplateformes:

  Dans «Accorder l'accès à un utilisateur pour s'abonner à une rubrique plus en profondeur dans l'arborescence», à la page 509 USER1 , un accès a été accordé pour s'abonner à la rubrique `"Price/Fruit/Apples"` en accordant à l'utilisateur un accès par abonnement au profil `FRUIT` .

  Ce profil unique a également accordé à USER1 l'accès pour s'abonner à `"Price/Fruit/Oranges"` et `"Price/Fruit/#"`, et cet accès reste même avec l'ajout du nouvel objet de rubrique et des profils qui lui sont associés.

  Accordez l'accès à USER2 pour vous abonner à la rubrique `"Price/Fruit/Apples"` en accordant à l'utilisateur l'accès par abonnement au profil `APPLE` . Pour ce faire, utilisez la commande d'autorisation pour la plateforme:

  ALW **AIX, Linux, and Windows systèmes**

  ```
  setmqaut -t topic -n APPLE -p USER2 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
  ```

Résultats

z/OS Sous z/OS, lorsque USER1 tente de s'abonner à la rubrique `"Price/Fruit/Apples"` , le premier contrôle de sécurité du profil `hlq.SUBSCRIBE.APPLE` échoue, mais lorsqu'il remonte l'arborescence, le profil `hlq.SUBSCRIBE.FRUIT` permet à USER1 de s'abonner. L'abonnement aboutit donc et aucun code retour n'est envoyé à l'appel MQSUB. Toutefois, un message RACF `ICH` est généré pour la première vérification:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

Lorsque USER2 tente de s'abonner à la rubrique `"Price/Fruit/Apples"` , le résultat est un succès car le contrôle de sécurité réussit sur le premier profil.

Lorsque USER2 tente de s'abonner à la rubrique `"Price/Fruit/Oranges"` , le résultat est un échec avec un message `MQRC_NOT_AUTHORIZED` , ainsi que:

- z/OS Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Sur les plateformes AIX, Linux, and Windows , l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i Sous IBMi, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

z/OS L'inconvénient de cette configuration est que, sous z/OS, vous recevez des messages ICH supplémentaires sur la console. Vous pouvez éviter cela si vous sécurisez l'arborescence de rubriques d'une manière différente.

Modifier le contrôle d'accès pour éviter les messages supplémentaires

Cette rubrique est la quatrième d'une liste de tâches qui vous indique comment accorder l'accès pour vous abonner à des rubriques par plusieurs utilisateurs et pour éviter des messages RACF ICH408I supplémentaires sur z/OS.

Avant de commencer

Cette rubrique améliore la configuration décrite dans «Accorder à un autre utilisateur l'accès permettant de s'abonner uniquement à la rubrique située plus en profondeur dans l'arborescence», à la page 511 afin d'éviter des messages d'erreur supplémentaires.

Pourquoi et quand exécuter cette tâche

Cette rubrique vous explique comment accorder l'accès à des rubriques plus en profondeur dans l'arborescence et comment supprimer l'accès à la rubrique située en bas de l'arborescence lorsqu'aucun utilisateur n'en a besoin.

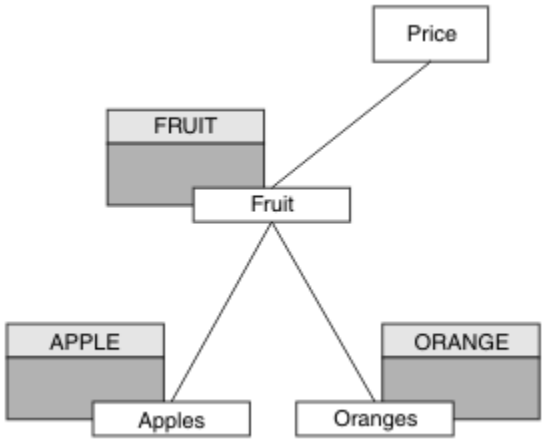


Figure 26. Exemple d'octroi de contrôle d'accès pour éviter des messages supplémentaires.

Définissez un nouvel objet de rubrique comme suit:

Procédure

1. Exécutez la commande MQSC `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`.
2. Accordez l'accès comme suit:

- z/OS **z/OS** :

  Définissez un nouveau profil et ajoutez l'accès à ce profil et aux profils existants. Pour ce faire, utilisez les commandes RACF suivantes:

```
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 Multiplateformes:

  Configurez l'accès équivalent à l'aide des commandes d'autorisation pour la plateforme:

  ALW **AIX, Linux, and Windows systèmes**

```
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)
```

Résultats

z/OS Sous z/OS, lorsque USER1 tente de s'abonner à la rubrique `"Price/Fruit/Apples"` , le premier contrôle de sécurité sur le profil `hlq.SUBSCRIBE.APPLE` aboutit.

Lorsque USER2 tente de s'abonner à la rubrique `"Price/Fruit/Apples"` , le résultat est un succès car le contrôle de sécurité réussit sur le premier profil.

Lorsque USER2 tente de s'abonner à la rubrique `"Price/Fruit/Oranges"` , le résultat est un échec avec un message `MQRC_NOT_AUTHORIZED` , ainsi que:

- z/OS Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Sous AIX, Linux, and Windows, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i Sous IBM i, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

Accorder l'accès à un utilisateur pour la publication dans une rubrique

Cette rubrique est la première d'une liste de tâches qui vous indique comment accorder l'accès à la publication de rubriques à plusieurs utilisateurs.

Pourquoi et quand exécuter cette tâche

Cette tâche suppose qu'aucun objet de rubrique d'administration n'existe à droite de l'arborescence de rubriques et qu'aucun profil n'a été défini pour la publication. L'hypothèse utilisée est que les diffuseurs utilisent uniquement la chaîne de rubrique.

Une application peut publier dans une rubrique en fournissant un objet de rubrique, une chaîne de rubrique ou une combinaison des deux. Quel que soit le mode de sélection de l'application, l'effet est la publication à un certain point de l'arborescence de rubriques. Si ce point de l'arborescence de rubriques est représenté par un objet de rubrique d'administration, un profil de sécurité est vérifié en fonction du nom de cet objet de rubrique. Exemple :

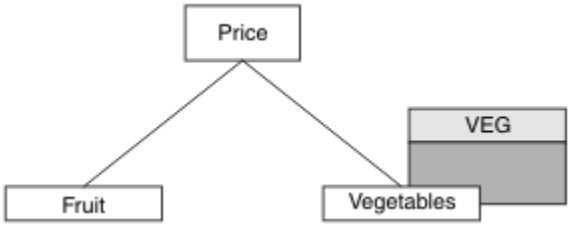


Figure 27. Octroi de l'accès en publication à une rubrique

Tableau 92. Exemple de conditions d'accès à la publication

Topic	Accès à la publication requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun

Tableau 92. Exemple de conditions d'accès à la publication (suite)		
Topic	**Accès à la publication requis**	**Objet de rubrique**
Prix / Légumes	USER1	VEG

Définissez un nouvel objet de rubrique comme suit:

Procédure

1. Exécutez la commande MQSC `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')`.
2. Accordez l'accès comme suit:

- z/OS **z/OS** :

  Accordez l'accès à USER1 pour publier dans la rubrique `"Price/Vegetables"` en accordant à l'utilisateur l'accès au profil `hlq.PUBLISH.VEG` . Pour ce faire, utilisez les commandes RACF suivantes:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
  PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
  ```

- Autres plateformes:

  Accordez l'accès à USER1 pour publier dans la rubrique `"Price/Vegetables"` en accordant à l'utilisateur l'accès au profil `VEG` . Pour ce faire, utilisez la commande d'autorisation pour la plateforme:

  ALW **AIX, Linux, and Windows systèmes**

  ```
  setmqaut -t topic -n VEG -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
  ```

Résultats

Lorsque USER1 tente de publier dans la rubrique `"Price/Vegetables"` , le résultat est un succès, c'est-à-dire que l'appel MQOPEN aboutit.

Lorsque USER2 tente de publier dans la rubrique `"Price/Vegetables"` , l'appel MQOPEN échoue avec un message `MQRC_NOT_AUTHORIZED` et:

- z/OS Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

  ```
  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.VEG ...

  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
  ```

- ALW Sur les autres plateformes, l'événement d'autorisation suivant:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
  ```

```
UserIdentifier        USER2
AdminTopicNames       VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

- IBM i Sous IBMi, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier       MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames       VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

Notez qu'il s'agit d'une illustration de ce que vous voyez ; pas de tous les champs.

Accorder l'accès à un utilisateur pour publier dans une rubrique plus en profondeur dans l'arborescence

Cette rubrique est la deuxième d'une liste de tâches qui vous indique comment accorder l'accès à la publication à des rubriques par plusieurs utilisateurs.

Avant de commencer

Cette rubrique utilise la configuration décrite dans «Accorder l'accès à un utilisateur pour la publication dans une rubrique», à la page 515.

Pourquoi et quand exécuter cette tâche

Si le point de l'arborescence de rubriques où l'application publie n'est pas représenté par un objet de rubrique d'administration, déplacez l'arborescence vers le haut jusqu'à ce que l'objet de rubrique d'administration parent le plus proche soit localisé. Le profil de sécurité est vérifié en fonction du nom de cet objet de rubrique.

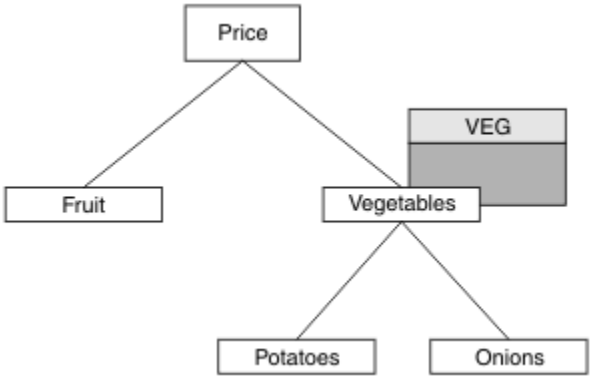


Figure 28. Octroi de l'accès en publication à une rubrique dans une arborescence de rubriques

Tableau 93. Exemple de conditions d'accès à la publication

Topic	Accès à l'abonnement requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun
Prix / Légumes	USER1	VEG
Prix / Légumes / Pommes de terre	USER1	
Prix / Légumes / Oignons	USER1	

Dans la tâche précédente, USER1 a été autorisé à publier la rubrique "Price/Vegetables/Potatoes" en lui accordant l'accès au profil hlq.PUBLISH.VEG sur z/OS ou l'accès en publication au profil VEG sur Multiplatforms. Ce profil unique accorde également à USER1 l'accès à la publication sur "Price/Vegetables/Onions".

Lorsque USER1 tente de publier dans la rubrique "Price/Vegetables/Potatoes" , le résultat est un succès, c'est-à-dire que l'appel MQOPEN aboutit.

Lorsque USER2 tente de s'abonner à la rubrique "Price/Vegetables/Potatoes" , le résultat est un échec, c'est-à-dire que l'appel MQOPEN échoue avec un message MQRC_NOT_AUTHORIZED , ainsi que:

- z/OS Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Multi Sur Multiplatforms, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

Notez ce qui suit :

- z/OS Les messages que vous recevez sur z/OS sont identiques à ceux reçus lors de la tâche précédente car les mêmes objets de rubrique et les mêmes profils contrôlent l'accès.
- Multi Le message d'événement que vous recevez sur Multiplatforms est similaire à celui reçu dans la tâche précédente, mais la chaîne de rubrique réelle est différente.

Accorder l'accès pour la publication et l'abonnement

Cette rubrique est la dernière d'une liste de tâches qui vous indique comment accorder l'accès à la publication et l'abonnement à des rubriques par plusieurs utilisateurs.

Avant de commencer

Cette rubrique utilise la configuration décrite dans «Accorder l'accès à un utilisateur pour publier dans une rubrique plus en profondeur dans l'arborescence», à la page 517.

Pourquoi et quand exécuter cette tâche

Dans une tâche précédente, USER1 a été autorisé à s'abonner à la rubrique "Price/Fruit". Cette rubrique vous indique comment accorder l'accès à cet utilisateur pour publier dans cette rubrique.

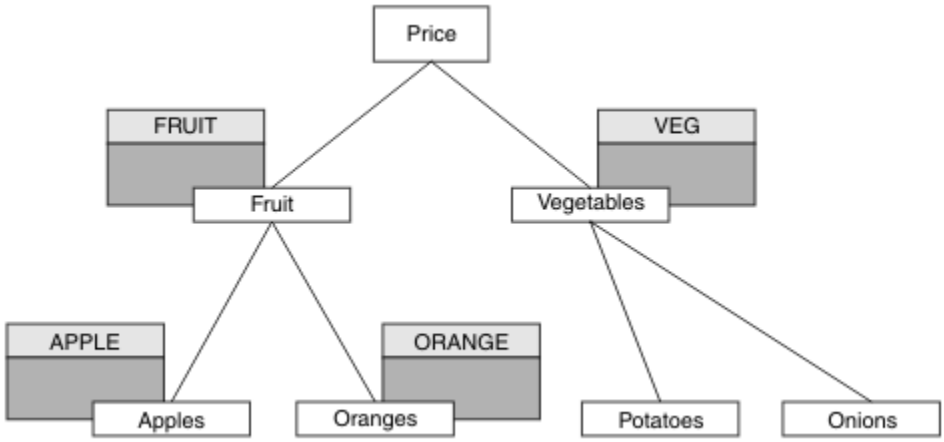


Figure 29. Octroi d'accès pour la publication et l'abonnement

Tableau 94. Exemple de conditions d'accès à la publication et à l'abonnement

Topic	Accès à l'abonnement requis	Accès à la publication requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun utilisateur	Aucun
Prix / Fruits	USER1	USER1	fruit
Prix / Fruits / Pommes	USER1 et USER2		Apple
Prix / Fruits / Oranges	USER1		ORANGE

Procédure

Accordez l'accès comme suit:

- z/OS **z/OS** :

  Dans une tâche antérieure, USER1 a été autorisé à s'abonner à la rubrique `"Price/Fruit"` en accordant à l'utilisateur l'accès au profil `hlq.SUBSCRIBE.FRUIT` .

  Pour publier dans la rubrique `"Price/Fruit"` , accordez l'accès à USER1 au profil `hlq.PUBLISH.FRUIT` . Pour ce faire, utilisez les commandes RACF suivantes:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi Multiplateformes:

  Accordez l'accès à USER1 pour publier dans la rubrique `"Price/Fruit"` en accordant à l'utilisateur l'accès en publication au profil FRUIT . Pour ce faire, utilisez la commande d'autorisation pour la plateforme:

  ALW **AIX, Linux, and Windows systèmes**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

IBM i

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
```

Résultats

Sous z/OS, lorsque USER1 tente de publier dans la rubrique `"Price/Fruit"` , le contrôle de sécurité de l'appel MQOPEN passe.

Lorsque USER2 tente de publier à la rubrique `"Price/Fruit"` , le résultat est un échec avec un message `MQRC_NOT_AUTHORIZED` , ainsi que:

- Sous z/OS, les messages suivants affichés sur la console indiquent le chemin de sécurité complet via l'arborescence de rubriques qui a été tentée:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Sur les plateformes AIX, Linux, and Windows , l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- Sous IBM i, l'événement d'autorisation suivant:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

En suivant l'ensemble complet de ces tâches, vous attribuez à USER1 et USER2 les droits d'accès suivants pour la publication et l'abonnement aux rubriques répertoriées:

Tableau 95. Liste complète des droits d'accès résultant d'exemples de sécurité

Topic	Accès à l'abonnement requis	Accès à la publication requis	Objet de rubrique
Prix	Aucun utilisateur	Aucun utilisateur	Aucun
Prix / Fruits	USER1	USER1	fruit
Prix / Fruits / Pommes	USER1 et USER2		Apple
Prix / Fruits / Oranges	USER1		ORANGE
Prix / Légumes		USER1	VEG

Tableau 95. Liste complète des droits d'accès résultant d'exemples de sécurité (suite)			
Topic	**Accès à l'abonnement requis**	**Accès à la publication requis**	**Objet de rubrique**
Prix / Légumes / Pommes de terre			
Prix / Légumes / Oignons			

z/OS Lorsque vous avez des exigences différentes en matière d'accès de sécurité à différents niveaux de l'arborescence de rubriques, une planification minutieuse garantit que vous ne recevez pas d'avertissements de sécurité superflus dans le journal de la console z/OS . La configuration de la sécurité au niveau approprié dans l'arborescence permet d'éviter les messages de sécurité trompeurs.

Sécurité des abonnements

MQSO_ALTERNATE_USER_AUTHORITY

La zone ID AlternateUsercontient un identificateur utilisateur à utiliser pour valider cet appel MQSUB. L'appel peut aboutir uniquement si cet ID AlternateUserest autorisé à s'abonner à la rubrique avec les options d'accès spécifiées, que l'ID utilisateur sous lequel l'application s'exécute soit autorisé ou non à le faire.

MQSO_SET_IDENTITY_CONTEXT

L'abonnement permet d'utiliser le jeton de comptabilité et les données d'identité d'application fournies dans les zones PubAccountingToken et PubApplIdentityData .

Si cette option est spécifiée, la même vérification d'autorisation est effectuée comme si la file d'attente de destination était accessible à l'aide d'un appel MQOPEN avec MQOO_SET_IDENTITY_CONTEXT, sauf dans le cas où l'option MQSO_MANAGED est également utilisée, auquel cas aucune vérification d'autorisation n'est effectuée sur la file d'attente de destination.

Si cette option n'est pas spécifiée, les informations de contexte par défaut sont associées aux publications envoyées à cet abonné comme suit:

Tableau 96. Informations de contexte de publication par défaut	
Zone dans MQMD	**Valeur utilisée**
UserIdentifier	ID utilisateur associé à l'abonnement (voir la zone SUBUSER sur DISPLAY SBSTATUS) au moment de la publication.
AccountingToken	Déterminé à partir de l'environnement si possible ; défini sur MQACT_NONE dans le cas contraire.
ApplIdentityData	Mettez à blanc.

Cette option est valide uniquement avec MQSO_CREATE et MQSO_ALTER. Si elles sont utilisées avec MQSO_RESUME, les zones PubAccountingToken et PubApplIdentityData sont ignorées, de sorte que cette option n'a aucun effet.

Si un abonnement est modifié sans utiliser cette option alors que l'abonnement avait précédemment fourni des informations de contexte d'identité, des informations de contexte par défaut sont générées pour l'abonnement modifié.

Si un abonnement permettant à différents ID utilisateur de l'utiliser avec l'option MQSO_ANY_USERID, est repris par un autre ID utilisateur, le contexte d'identité par défaut est généré pour le nouvel ID utilisateur propriétaire de l'abonnement et toutes les publications suivantes sont distribuées contenant le nouveau contexte d'identité.

AlternateSecurityId

Il s'agit d'un identificateur de sécurité qui est transmis avec l'ID AlternateUserau service d'autorisation pour permettre l'exécution des vérifications d'autorisation appropriées. L'ID AlternateSecurityest utilisé uniquement si MQSO_ALTERNATE_USER_AUTHORITY est spécifié et que la zone d'ID AlternateUsern'est pas entièrement vide jusqu'au premier caractère null ou jusqu'à la fin de la zone.

Option d'abonnement MQSO_ANY_USERID

Lorsque MQSO_ANY_USERID est spécifié, l'identité de l'abonné n'est pas limitée à un seul ID utilisateur. Cela permet à tout utilisateur de modifier ou de reprendre l'abonnement lorsqu'il dispose des droits appropriés. Un seul utilisateur peut disposer de l'abonnement à la fois. Une tentative de reprise de l'utilisation d'un abonnement actuellement utilisé par une autre application entraîne l'échec de l'appel avec MQRC_SUBSCRIPTION_IN_USE.

Pour ajouter cette option à un abonnement existant, l'appel MQSUB (à l'aide de MQSO_ALTER) doit provenir du même ID utilisateur que l'abonnement d'origine.

Si un appel MQSUB fait référence à un abonnement existant avec MQSO_ANY_USERID défini et que l'ID utilisateur est différent de l'abonnement d'origine, l'appel aboutit uniquement si le nouvel ID utilisateur est autorisé à s'abonner à la rubrique. Une fois l'opération terminée, les futures publications destinées à cet abonné sont placées dans la file d'attente de l'abonné avec le nouvel ID utilisateur défini dans la publication.

ID_UTILISATEUR_FIXE_MQSO_FIXE

Lorsque MQSO_FIXED_USERID est spécifié, l'abonnement ne peut être modifié ou repris que par un seul ID utilisateur propriétaire. Cet ID utilisateur est le dernier ID utilisateur à modifier l'abonnement qui définit cette option, supprimant ainsi l'option MQSO_ANY_USERID, ou si aucune modification n'a eu lieu, il s'agit de l'ID utilisateur qui a créé l'abonnement.

Si une instruction MQSUB fait référence à un abonnement existant avec MQSO_ANY_USERID défini et modifie l'abonnement (à l'aide de MQSO_ALTER) pour utiliser l'option MQSO_FIXED_USERID, l'ID utilisateur de l'abonnement est désormais corrigé au niveau de ce nouvel ID utilisateur. L'appel aboutit uniquement si le nouvel ID utilisateur est autorisé à s'abonner à la rubrique.

Si un ID utilisateur autre que celui enregistré comme propriétaire d'un abonnement tente de reprendre ou de modifier un abonnement MQSO_FIXED_USERID, l'appel échoue avec MQRC_IDENTITY_MISMATCH. L'ID utilisateur propriétaire d'un abonnement peut être affiché à l'aide de la commande DISPLAY SBSTATUS.

Si ni MQSO_ANY_USERID ni MQSO_FIXED_USERID n'est spécifié, la valeur par défaut est MQSO_FIXED_USERID.

Sécurité de publication / abonnement entre les gestionnaires de files d'attente

Les messages internes de publication / abonnement, tels que les abonnements de proxy et les publications, sont placés dans des files d'attente système de publication / abonnement à l'aide de règles de sécurité de canal normales. Les informations et les diagrammes de cette rubrique mettent en évidence les différents processus et ID utilisateur impliqués dans la distribution de ces messages.

Contrôle d'accès local

L'accès aux rubriques pour la publication et les abonnements est régi par des définitions et des règles de sécurité locales décrites dans Sécurité de publication / abonnement. Aucun objet de rubrique local n'est requis pour établir le contrôle d'accès. Les administrateurs peuvent choisir d'appliquer le contrôle d'accès aux objets de rubrique en cluster, qu'ils existent ou non dans le cluster.

Les administrateurs système sont responsables du contrôle d'accès sur leur système local. Ils doivent faire confiance aux administrateurs des autres membres de la hiérarchie ou des collectivités de cluster pour qu'ils soient responsables de leur stratégie de contrôle d'accès. Étant donné que le contrôle d'accès est défini pour chaque machine distincte, il risque d'être fastidieux si un contrôle de niveau fin est nécessaire. Il peut ne pas être nécessaire d'imposer un contrôle d'accès, ou le contrôle d'accès peut être défini sur des objets de haut niveau dans l'arborescence de rubriques. Un contrôle d'accès de niveau fin peut être défini pour chaque subdivision de l'espace de nom de sujet.

Création d'un abonnement de proxy

La confiance d'une organisation pour la connexion de son gestionnaire de files d'attente à votre gestionnaire de files d'attente est confirmée par des moyens d'authentification de canal normaux. Si cette organisation digne de confiance est également autorisée à effectuer une publication / abonnement distribué, une vérification des droits d'accès est effectuée. La vérification est effectuée lorsque le canal insère un message dans une file d'attente de publication / abonnement distribuée. Par exemple, si un message est inséré dans la file d'attente `SYSTEM.INTER.QMGR.CONTROL` . L'ID utilisateur pour la vérification des droits d'accès à la file d'attente dépend des valeurs PUTAUT du canal récepteur. Par exemple, l'ID utilisateur du canal, `MCAUSER`, le contexte de message, en fonction de la valeur et de la plateforme. Pour plus d'informations sur la sécurité des canaux, voir Sécurité des canaux.

Les abonnements de proxy sont effectués avec l'ID utilisateur de l'agent de publication / abonnement réparti sur le gestionnaire de files d'attente éloignées. Par exemple, `QM2` dans Figure 30, à la page 523. L'utilisateur est alors facilement autorisé à accéder aux profils d'objet de rubrique locaux, car cet ID utilisateur est défini dans le système et il n'y a donc pas de conflit de domaine.

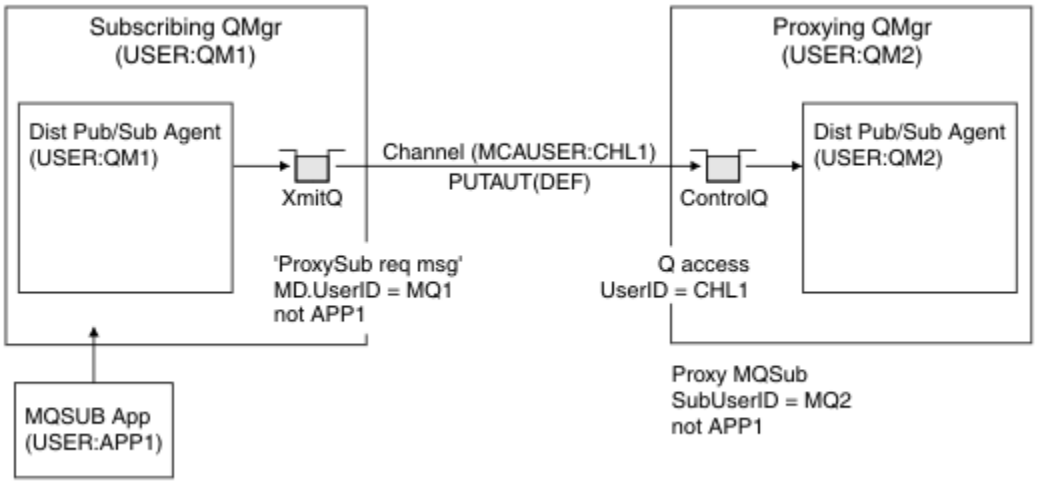


Figure 30. sécurité de l'abonnement de proxy, création d'un abonnement

Renvoi de publications distantes

Lorsqu'une publication est créée sur le gestionnaire de files d'attente de publication, une copie de la publication est créée pour tout abonnement de proxy. Le contexte de la publication copiée contient le contexte de l'ID utilisateur qui a effectué l'abonnement ; `QM2` dans Figure 31, à la page 524. L'abonnement proxy est créé avec une file d'attente de destination qui est une file d'attente éloignée, de sorte que le message de publication est résolu dans une file d'attente de transmission.

La confiance d'une organisation pour la connexion de son gestionnaire de files d'attente, QM2, à un autre gestionnaire de files d'attente, QM1, est confirmée par des moyens d'authentification de canal normaux. Si cette organisation digne de confiance est alors autorisée à effectuer une publication / abonnement distribué, une vérification des droits d'accès est effectuée lorsque le canal place le message de publication dans la file d'attente de publication / abonnement distribué SYSTEM.INTER.QMGR.PUBS. L'ID utilisateur pour la vérification des droits d'accès à la file d'attente dépend de la valeur PUTAUT du canal récepteur (par exemple, l'ID utilisateur du canal, MCAUSER, le contexte de message, etc., en fonction de la valeur et de la plateforme). Pour plus d'informations sur la sécurité des canaux, voir Sécurité des canaux.

Lorsque le message de publication atteint le gestionnaire de files d'attente d'abonnement, une autre opération MQPUT sur la rubrique est effectuée sous l'autorité de ce gestionnaire de files d'attente et le contexte contenant le message est remplacé par le contexte de chacun des abonnés locaux tels qu'ils reçoivent chacun le message.

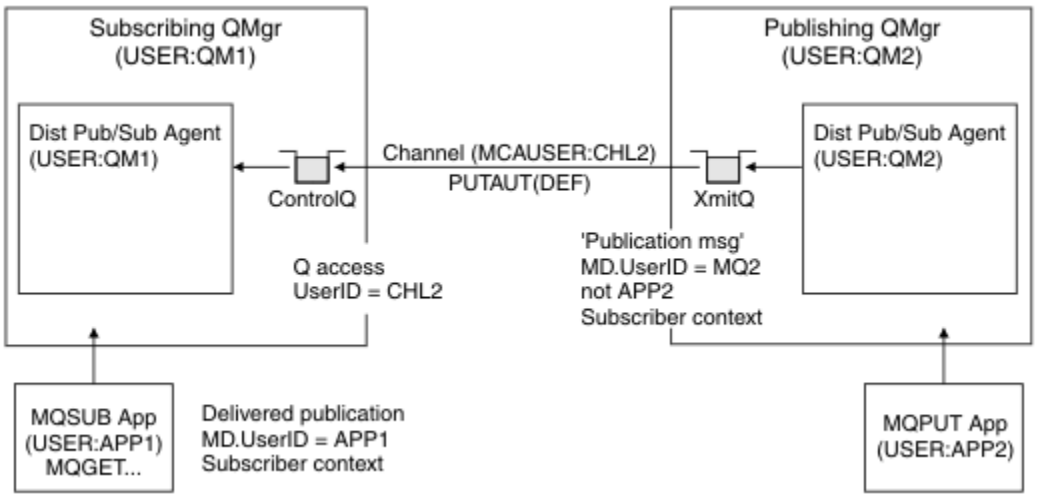


Figure 31. Sécurité des abonnements de proxy, transfert de publications

Sur un système où la sécurité est peu prise en compte, les processus de publication / abonnement distribué sont susceptibles de s'exécuter sous un ID utilisateur dans le groupe mqm , le paramètre MCAUSER sur un canal est vide (valeur par défaut) et les messages sont distribués aux différentes files d'attente système selon les besoins. Le système non sécurisé facilite la mise en place d'une preuve de concept pour illustrer la publication / l'abonnement distribué.

Sur un système où la sécurité est plus sérieusement prise en compte, ces messages internes sont soumis aux mêmes contrôles de sécurité que tout message passant par le canal.

Si le canal est configuré avec une valeur MCAUSER non vide et une valeur PUTAUT indiquant que MCAUSER doit être vérifié, l'accès aux files d'attente SYSTEM.INTER.QMGR.* doit être accordé à MCAUSER en question. S'il existe plusieurs gestionnaires de files d'attente éloignées, avec des canaux s'exécutant sous des ID MCAUSER différents, tous ces ID utilisateur doivent être autorisés à accéder aux files d'attente SYSTEM.INTER.QMGR.* . Des canaux s'exécutant sous des ID MCAUSER différents peuvent se produire, par exemple, lorsque plusieurs connexions hiérarchiques sont configurées sur un seul gestionnaire de files d'attente.

Si le canal est configuré avec une valeur PUTAUT spécifiant que le contexte du message est utilisé, l'accès aux files d'attente SYSTEM.INTER.QMGR.* est vérifié en fonction de l'ID utilisateur dans le message interne. Etant donné que tous ces messages sont insérés avec l'ID utilisateur de l'agent de publication / abonnement distribué à partir du gestionnaire de files d'attente qui envoie le message interne ou le message de publication (voir Figure 31, à la page 524), il ne s'agit pas d'un ensemble trop important d'ID utilisateur pour accorder l'accès aux différentes files d'attente système (une par gestionnaire de files d'attente éloignées), si vous souhaitez configurer votre sécurité de publication / abonnement distribué de cette manière. Il a toujours les mêmes problèmes que la sécurité de contexte de canal ; celui des différents domaines d'ID utilisateur et le fait que l'ID utilisateur dans le message peut ne pas être

défini sur le système récepteur. Cependant, il s'agit d'un moyen tout à fait acceptable de fonctionner si nécessaire.

z/OS La sécurité des files d'attente système fournit la liste des files d'attente et l'accès requis pour configurer de manière sécurisée votre environnement de publication / abonnement distribué. Si des messages internes ou des publications ne parviennent pas à être insérés en raison de violations de sécurité, le canal écrit un message dans le journal de manière normale et les messages peuvent être envoyés à la file d'attente des messages non livrés en fonction du traitement normal des erreurs du canal.

Toute la messagerie inter-gestionnaire de files d'attente pour les besoins de la publication / abonnement distribué s'exécute à l'aide de la sécurité de canal normale.

Pour plus d'informations sur la restriction des publications et des abonnements de proxy au niveau des rubriques, voir Sécurité de publication / abonnement.

Utilisation des ID utilisateur par défaut avec une hiérarchie de gestionnaires de files d'attente

Si vous disposez d'une hiérarchie de gestionnaires de files d'attente s'exécutant sur des plateformes différentes et que vous utilisez des ID utilisateur par défaut, notez que ces ID utilisateur par défaut diffèrent d'une plateforme à l'autre et peuvent ne pas être connus sur la plateforme cible. Par conséquent, un gestionnaire de files d'attente s'exécutant sur une plateforme rejette les messages reçus des gestionnaires de files d'attente sur d'autres plateformes avec le code anomalie `MQRC_NOT_AUTHORIZED`.

Pour éviter que des messages ne soient rejetés, au minimum, les droits suivants doivent être ajoutés aux ID utilisateur par défaut utilisés sur d'autres plateformes:

- Droit *PUT *GET sur le système SYSTEM.BROKER. queues
- *PUB *SUB droit sur SYSTEM.BROKER. rubriques
- Droit *ADMCRT *ADMDLT *ADMCHG sur le système SYSTEM.BROKER.CONTROL.QUEUE .

Les ID utilisateur par défaut avec une hiérarchie de gestionnaires de files d'attente sont les suivants:

Plateforme	**ID utilisateur par défaut**
Windows	`mqm`
Systèmes AIX and Linux	`mqm`
IBM i	`QMQM`
z/OS	ID utilisateur de l'espace adresse de l'initiateur de canal

Si des gestionnaires de files d'attente sur des plateformes autres que IBM i sont hiérarchiquement connectés à un gestionnaire de files d'attente sur IBM i, créez et accordez l'accès à l'ID utilisateur'qmqm'.

Si des gestionnaires de files d'attente sous IBM i ou z/OS sont connectés hiérarchiquement à un gestionnaire de files d'attente sous AIX, Linux, and Windows, créez et accordez l'accès à l'ID utilisateur'mqm'.

Si les gestionnaires de files d'attente sous Multiplateformes sont connectés de manière hiérarchique à un gestionnaire de files d'attente sous z/OS, créez et accordez l'accès à l'ID utilisateur de l'espace adresse de l'initiateur de canal z/OS .

Les ID utilisateur peuvent être sensibles à la casse. Le gestionnaire de files d'attente d'origine (sous Multiplateformes) force l'ID utilisateur à être en majuscules. Le gestionnaire de files d'attente de réception (sous AIX, Linux, and Windows) force l'ID utilisateur à être en minuscules. Par conséquent, tous les ID utilisateur créés sur les systèmes AIX and Linux doivent être créés en minuscules. Si un exit de message a été installé, la mise en majuscules ou en minuscules de l'ID utilisateur n'est pas forcée. Prenez soin de comprendre comment l'exit de message traite l'ID utilisateur.

Pour éviter les problèmes potentiels liés à la conversion des ID utilisateur:

- Sur les systèmes AIX, Linux, and Windows , vérifiez que les ID utilisateur sont spécifiés en minuscules.
- Sur les systèmes IBM i et z/OS , vérifiez que les ID utilisateur sont indiqués en majuscules.

Sécurité IBM MQ Console et REST API

La sécurité pour IBM MQ Console et REST API est configurée en éditant la configuration du serveur mqweb dans le fichier `mqwebuser.xml` .

Pourquoi et quand exécuter cette tâche

Vous pouvez suivre les actions utilisateur et auditer l'utilisation de IBM MQ Console et de REST API en examinant les fichiers journaux du serveur mqweb.

Les utilisateurs de IBM MQ Console et de REST API peuvent être authentifiés à l'aide des éléments suivants:

- Registre de base
- registre LDAP
- Registre du système d'exploitation local
- SAF sous z/OS
- Tout autre type de registre pris en charge par WebSphere Liberty

Des rôles peuvent être affectés à des utilisateurs IBM MQ Console et à des utilisateurs REST API pour déterminer le niveau d'accès qui leur est accordé aux objets IBM MQ . Par exemple, pour effectuer des opérations de messagerie, le rôle `MQWebUser` doit être affecté aux utilisateurs. Pour plus d'informations sur les rôles disponibles, voir «Rôles sur IBM MQ Console et REST API», à la page 538.

Une fois qu'un rôle a été affecté à un utilisateur, un certain nombre de méthodes peuvent être utilisées pour l'authentification de l'utilisateur. Avec IBM MQ Console, les utilisateurs peuvent se connecter avec un nom d'utilisateur et un mot de passe ou utiliser l'authentification par certificat client. Avec REST API, les utilisateurs peuvent utiliser l'authentification HTTP de base, l'authentification basée sur un jeton ou l'authentification par certificat client.

Procédure

1. Définissez le registre d'utilisateurs pour authentifier les utilisateurs et affectez à chaque utilisateur ou groupe un rôle pour autoriser les utilisateurs et les groupes à utiliser IBM MQ Console ou REST API. Pour plus d'informations, voir «Configuration des utilisateurs et des rôles», à la page 527
2. Choisissez comment les utilisateurs de IBM MQ Console s'authentifient auprès du serveur mqweb. Il n'est pas nécessaire d'utiliser la même méthode pour tous les utilisateurs:
   - Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur entre un ID utilisateur et un mot de passe à l'écran de connexion IBM MQ Console . Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Aucune configuration supplémentaire n'est requise pour utiliser cette option d'authentification, mais vous pouvez éventuellement configurer l'heure d'expiration du jeton LTPA. Pour plus d'informations, voir Configuration de l'intervalle d'expiration du jeton LTPA.
   - Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à IBM MQ Console, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.
3. Choisissez comment les utilisateurs de REST API s'authentifient auprès du serveur mqweb. Il n'est pas nécessaire d'utiliser la même méthode pour tous les utilisateurs:
   - Permet aux utilisateurs de s'authentifier à l'aide de l'authentification de base HTTP. Dans ce cas, un nom d'utilisateur et un mot de passe sont codés, mais non chiffrés, et envoyés avec chaque demande REST API pour authentifier et autoriser l'utilisateur pour cette demande. Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous

devez utiliser HTTPS. Pour plus d'informations, voir «Utilisation de l'authentification de base HTTP avec REST API», à la page 545.

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur fournit un ID utilisateur et un mot de passe à la ressource REST API `login` avec la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Pour plus d'informations, voir «Utilisation de l'authentification basée sur un jeton avec l'API REST», à la page 547.

  Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous devez utiliser HTTPS. Toutefois, si vous avez activé les connexions HTTP, vous pouvez autoriser l'utilisation d'un jeton LTPA émis pour une connexion HTTPS pour une connexion HTTP. Pour plus d'informations, voir Configuration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à REST API, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

4. Facultatif : Configurez le partage de ressources d'origine croisée pour REST API.

   Par défaut, un navigateur Web n'autorise pas les scripts, tels que JavaScript, à appeler REST API lorsque le script n'est pas de la même origine que le REST API. C'est-à-dire que les demandes d'origine croisée ne sont pas activées. Vous pouvez configurer le partage de ressources d'origine croisée (CORS) pour autoriser les demandes d'origine croisée à partir d'URL spécifiées. Pour plus d'informations, voir «Configuration de CORS pour REST API», à la page 549.
5. Facultatif : Configurez la validation de l'en-tête d'hôte pour IBM MQ Console et REST API.

   Vous pouvez configurer la validation de l'en-tête d'hôte et créer une liste autorisée de noms d'hôte et de ports pour vous assurer que seules les demandes contenant des en-têtes d'hôte spécifiques sont traitées par IBM MQ Console et REST API. Pour plus d'informations, voir «Configuration de la validation de l'en-tête d'hôte pour IBM MQ Console et REST API», à la page 550.

Configuration des utilisateurs et des rôles

Pour utiliser IBM MQ Console ou REST API, les utilisateurs doivent s'authentifier auprès d'un registre d'utilisateurs défini sur le serveur mqweb.

Pourquoi et quand exécuter cette tâche

Les utilisateurs authentifiés doivent être membres de l'un des groupes qui autorise l'accès aux fonctions de IBM MQ Console et REST API. Par défaut, le registre d'utilisateurs ne contient aucun utilisateur ; vous devez les ajouter en éditant le fichier `mqwebuser.xml` .

Lorsque vous configurez des utilisateurs et des groupes, vous devez d'abord configurer un registre d'utilisateurs pour authentifier les utilisateurs et les groupes. Ce registre d'utilisateurs est partagé entre IBM MQ Console et REST API. Vous pouvez contrôler si les utilisateurs et les groupes ont accès à IBM MQ Consoleet/ou à REST APIlorsque vous configurez des rôles pour vos utilisateurs et groupes.

Après avoir configuré le registre d'utilisateurs, vous configurez des rôles pour les utilisateurs et les groupes afin de leur accorder des autorisations. Plusieurs rôles sont disponibles, notamment des rôles spécifiques à l'utilisation de REST API for Managed File Transfer. Chaque rôle accorde un niveau d'accès différent. Pour plus d'informations, voir «Rôles sur IBM MQ Console et REST API», à la page 538.

Un certain nombre d'exemples de fichiers XML sont fournis avec le serveur mqweb pour simplifier la configuration des utilisateurs et des groupes. Les utilisateurs qui connaissent bien la configuration de la sécurité dans WebSphere Liberty (WLP) peuvent préférer ne pas utiliser les exemples. WLP fournit d'autres fonctions d'autorisation en plus de celles décrites ici.

Procédure

- Configurez les utilisateurs et les groupes avec un registre de base à l'aide du fichier `basic_registry.xml` .

Les noms d'utilisateur et les mots de passe du registre sont utilisés pour authentifier et autoriser les utilisateurs de IBM MQ Console et de REST API.

Pour configurer un registre de base à l'aide de l'exemple de fichier `basic_registry.xml` , voir «Configuration d'un registre de base pour IBM MQ Console et REST API», à la page 529.

- Configurez les utilisateurs et les groupes avec un registre LDAP à l'aide du fichier `ldap_registry.xml` .

  Les noms d'utilisateur et les mots de passe du registre LDAP sont utilisés pour authentifier et autoriser l'utilisation de IBM MQ Console et de REST API.

  Pour configurer un registre LDAP à l'aide de l'exemple de fichier `ldap_registry.xml` , voir «Configuration d'un registre LDAP pour IBM MQ Console et REST API», à la page 533.

- ALW

  Configurez les utilisateurs et les groupes avec un registre de système d'exploitation local à l'aide du fichier `local_os_registry.xml` .

  Les noms d'utilisateur et les mots de passe du registre du système d'exploitation sont utilisés pour authentifier et autoriser les utilisateurs du IBM MQ Console et du REST API.

  Pour configurer un registre de système d'exploitation local à l'aide de l'exemple de fichier `local_os_registry.xml` , voir «Configuration d'un registre de système d'exploitation local pour IBM MQ Console et REST API», à la page 532.

- z/OS

  Configurez les utilisateurs et les groupes avec l'interface SAF (System Authorization Facility) sous z/OS à l'aide du fichier `zos_saf_registry.xml` .

  Les profils RACF, ou tout autre produit de sécurité, sont utilisés pour accorder aux utilisateurs et aux groupes l'accès aux rôles. Les noms d'utilisateur et les mots de passe de la base de données RACF sont utilisés pour authentifier et autoriser les utilisateurs de IBM MQ Console et REST API.

  Pour configurer l'interface SAF à l'aide de l'exemple de fichier `zos_saf_registry.xml` , voir «Configuring a SAF registry for the IBM MQ Console and REST API», à la page 535.

- Désactivez la sécurité, y compris la possibilité d'accéder au IBM MQ Consoleou au REST API, à l'aide de HTTPS, à l'aide du fichier `no_security.xml` .

Que faire ensuite

Choisissez comment les utilisateurs s'authentifient:

IBM MQ Consoleoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur entre un ID utilisateur et un mot de passe à l'écran de connexion IBM MQ Console . Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Aucune configuration supplémentaire n'est requise pour utiliser cette option d'authentification, mais vous pouvez éventuellement configurer l'intervalle d'expiration pour le jeton LTPA. Pour plus d'informations, voir Configuration de l'intervalle d'expiration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à IBM MQ Console, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

REST APIoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification de base HTTP. Dans ce cas, un nom d'utilisateur et un mot de passe sont codés, mais non chiffrés, et envoyés avec chaque demande REST API pour authentifier et autoriser l'utilisateur pour cette demande. Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous devez utiliser HTTPS. Pour plus d'informations, voir «Utilisation de l'authentification de base HTTP avec REST API», à la page 545.

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur fournit un ID utilisateur et un mot de passe à la ressource REST API `login` avec la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Pour plus d'informations, voir «Utilisation de l'authentification basée sur un jeton avec l'API REST», à la page 547. Vous pouvez configurer l'intervalle d'expiration du jeton LTPA. Pour plus d'informations, voir Configuration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à REST API, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

Configuration d'un registre de base pour IBM MQ Console et REST API

Vous pouvez configurer un registre de base dans le fichier `mqwebuser.xml` . Les noms d'utilisateur, les mots de passe et les rôles du fichier XML sont utilisés pour authentifier et autoriser les utilisateurs du IBM MQ Console et du REST API.

Avant de commencer

- Lorsque vous configurez des utilisateurs dans le registre de base, vous devez affecter un rôle à chaque utilisateur. Chaque rôle fournit différents niveaux de privilèges pour accéder à IBM MQ Console et REST API, et détermine le contexte de sécurité qui est utilisé lorsqu'une opération autorisée est tentée. Vous devez comprendre ces rôles avant de configurer le registre de base. Pour plus d'informations sur chacun des rôles, voir «Rôles sur IBM MQ Console et REST API», à la page 538.
- Pour effectuer cette tâche, vous devez être un utilisateur disposant de privilèges suffisants pour éditer le fichier `mqwebuser.xml` :
  - z/OS Sous z/OS, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` .
  - Multi Sur tous les autres systèmes d'exploitation, vous devez être un utilisateur privilégié.
  - V 9.4.0 Linux Si le serveur mqweb fait partie d'une installation IBM MQ Web Server autonome, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` dans le répertoire de données IBM MQ Web Server .

Procédure

1. Copiez l'exemple de fichier XML `basic_registry.xml` à partir de l'un des chemins suivants:
   - Dans une installation IBM MQ :
     - ALW Sous AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS Sous z/OS: `PathPrefix /web/mq/samp/configuration`

       où `PathPrefix` est le chemin d'installation de IBM MQ for z/OS UNIX System Services Components .
   - V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     où *CHEMIN_INSTALLATION_MQ* est le répertoire dans lequel le fichier d'installation IBM MQ Web Server a été décompressé.
2. Placez l'exemple de fichier dans le répertoire approprié:
   - Dans une installation IBM MQ :
     - Linux AIX Sous AIX ou Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`

– Windows Sous Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb, où *MQ_DATA_PATH* est le chemin de données IBM MQ . Il s'agit du chemin de données sélectionné lors de l'installation de IBM MQ. Par défaut, ce chemin est C:\ProgramData\IBM\MQ.

– z/OS Sous z/OS : *WLP_user_directory*/servers/mqweb

où *WLP_user_directory* est le répertoire qui a été spécifié lors de l'exécution du script **crtmqweb** pour créer la définition de serveur mqweb.

- V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

  où *MQ_OVERRIDE_DATA_PATH* est le répertoire de données IBM MQ Web Server vers lequel la variable d'environnement **MQ_OVERRIDE_DATA_PATH** pointe.

3. Facultatif : Si vous avez modifié des paramètres de configuration dans mqwebuser.xml, copiez-les dans l'exemple de fichier.
4. Supprimez le fichier mqwebuser.xml existant et renommez l'exemple de fichier en mqwebuser.xml.
5. Editez le nouveau fichier mqwebuser.xml pour ajouter des utilisateurs et des groupes dans les balises **basicRegistry** .

   N'oubliez pas que tout utilisateur disposant du rôle MQWebUser peut effectuer uniquement les opérations que l'ID utilisateur est autorisé à effectuer sur le gestionnaire de files d'attente. Par conséquent, l'ID utilisateur défini dans le registre doit avoir un ID utilisateur identique sur le système sur lequel IBM MQ est installé. Ces ID utilisateur doivent être dans le même cas, sinon le mappage entre les ID utilisateur peut échouer.

   Pour plus d'informations sur la configuration des registres d'utilisateurs de base, voir Configuration d'un registre d'utilisateurs de base pour Liberty dans la documentation WebSphere Liberty .
6. Affectez des rôles aux utilisateurs et aux groupes en éditant le fichier mqwebuser.xml :

   Plusieurs rôles sont disponibles pour autoriser les utilisateurs et les groupes à utiliser le IBM MQ Consoleet le REST API. Chaque rôle accorde un niveau d'accès différent. Pour plus d'informations, voir «Rôles sur IBM MQ Console et REST API», à la page 538.

   - Pour affecter des rôles et accorder l'accès au IBM MQ Console, ajoutez vos utilisateurs et groupes entre les balises **security-role** appropriées dans les balises **<enterpriseApplication id="com.ibm.mq.console">** .
   - Pour affecter des rôles et accorder l'accès au REST API, ajoutez vos utilisateurs et groupes entre les balises **security-role** appropriées dans les balises **<enterpriseApplication id="com.ibm.mq.rest">** .

   Pour obtenir de l'aide sur le format des informations d'utilisateur et de groupe dans les balises **security-role** , voir les exemples.
7. Si vous avez fourni des mots de passe pour les utilisateurs dans mqwebuser.xml, vous devez les coder afin de les sécuriser à l'aide de la commande **securityUtility encoding** fournie par WebSphere Liberty. Pour plus d'informations, voir Liberty: commandesecurityUtility dans la documentation du produit WebSphere Liberty .

Exemple

Dans l'exemple suivant, le groupe MQWebAdminGroup est autorisé à accéder au IBM MQ Console avec le rôle MQWebAdmin. L'accès est accordé à l'utilisateur readeravec le rôle MQWebAdminROet à l'utilisateur guest avec le rôle MQWebUser:

```
<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>
```

Dans l'exemple suivant, les utilisateurs `reader` et `guest` sont autorisés à accéder à IBM MQ Console. L'utilisateur `user` a accès à REST APIet tous les utilisateurs du groupe `MQAdmin` ont accès à IBM MQ Console et à REST API. L'utilisateur `mftadmin` est autorisé à accéder à REST API for MFT :

```
<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>
```

Que faire ensuite

Choisissez comment les utilisateurs s'authentifient:

IBM MQ Consoleoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur entre un ID utilisateur et un mot de passe à l'écran de connexion IBM MQ Console . Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Aucune configuration supplémentaire n'est requise pour utiliser cette option d'authentification, mais vous pouvez éventuellement configurer l'intervalle d'expiration pour le jeton LTPA. Pour plus d'informations, voir Configuration de l'intervalle d'expiration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à IBM MQ Console, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

REST APIoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification de base HTTP. Dans ce cas, un nom d'utilisateur et un mot de passe sont codés, mais non chiffrés, et envoyés avec chaque demande REST API pour authentifier et autoriser l'utilisateur pour cette demande. Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous devez utiliser HTTPS. Pour plus d'informations, voir «Utilisation de l'authentification de base HTTP avec REST API», à la page 545.
- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur fournit un ID utilisateur et un mot de passe à la ressource REST API `login` avec la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Pour plus d'informations, voir «Utilisation de l'authentification

basée sur un jeton avec l'API REST», à la page 547. Vous pouvez configurer l'intervalle d'expiration du jeton LTPA. Pour plus d'informations, voir Configuration du jeton LTPA.

- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à REST API, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

ALW Configuration d'un registre de système d'exploitation local pour IBM MQ Console et REST API

Vous pouvez configurer un registre de système d'exploitation local dans le fichier `mqwebuser.xml` . Les noms d'utilisateur et les mots de passe du système d'exploitation local sont utilisés pour authentifier et autoriser les utilisateurs du IBM MQ Console et du REST API.

Avant de commencer

- Pour l'authentification par certificat client avec la fonction d'authentification du système d'exploitation local, l'identité de l'utilisateur est le nom usuel (CN) à partir du nom distinctif (DN) du certificat client. Si l'identité de l'utilisateur n'existe pas en tant qu'utilisateur du système d'exploitation, la connexion par certificat client échoue et l'authentification par mot de passe est réactivée.
- Pour effectuer cette tâche, vous devez être un utilisateur disposant des privilèges suffisants pour éditer le fichier `mqwebuser.xml` :
  - V 9.4.0 Linux Si le serveur mqweb fait partie d'une installation IBM MQ Web Server autonome, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` dans le répertoire de données IBM MQ Web Server .
  - Si le serveur mqweb fait partie d'une installation IBM MQ , vous devez être un utilisateur privilégié.

Pourquoi et quand exécuter cette tâche

Avec un registre de système d'exploitation local, les utilisateurs et les groupes reçoivent automatiquement un rôle:

- Tout utilisateur faisant partie du groupe'mqm'ou du groupe'QMQMADM'sur IBM ise voit attribuer les rôles `MQWebAdmin` et `MFTWebAdmin` .
- Tous les autres utilisateurs ont le rôle `MQWebUser` .

Pour plus d'informations sur ces rôles, voir «Rôles sur IBM MQ Console et REST API», à la page 538.

Un registre de système d'exploitation local ne peut être utilisé que sur AIX, Linux, and Windows.
z/OS La fonction équivalente est fournie sur z/OS en configurant un registre SAF. Pour plus d'informations, voir «Configuring a SAF registry for the IBM MQ Console and REST API», à la page 535.

Procédure

1. Copiez l'exemple de fichier XML `local_os_registry.xml` à partir de l'un des chemins suivants:
   - V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     où *CHEMIN_INSTALLATION_MQ* est le répertoire dans lequel le fichier d'installation IBM MQ Web Server a été décompressé.
   - Dans une installation IBM MQ : *MQ_INSTALLATION_PATH*`/web/mq/samp/configuration`
2. Placez l'exemple de fichier dans l'un des répertoires suivants:
   - V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

où *MQ_OVERRIDE_DATA_PATH* est le répertoire de données IBM MQ Web Server vers lequel la variable d'environnement **MQ_OVERRIDE_DATA_PATH** pointe.

- Dans une installation IBM MQ : *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

3. Facultatif : Si vous avez modifié des paramètres de configuration dans mqwebuser.xml, copiez-les dans l'exemple de fichier.
4. Supprimez le fichier mqwebuser.xml existant et renommez l'exemple de fichier en mqwebuser.xml.

Que faire ensuite

Choisissez comment les utilisateurs s'authentifient:

IBM MQ Consoleoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur entre un ID utilisateur et un mot de passe à l'écran de connexion IBM MQ Console . Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Aucune configuration supplémentaire n'est requise pour utiliser cette option d'authentification, mais vous pouvez éventuellement configurer l'intervalle d'expiration pour le jeton LTPA. Pour plus d'informations, voir Configuration de l'intervalle d'expiration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à IBM MQ Console, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

REST APIoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification de base HTTP. Dans ce cas, un nom d'utilisateur et un mot de passe sont codés, mais non chiffrés, et envoyés avec chaque demande REST API pour authentifier et autoriser l'utilisateur pour cette demande. Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous devez utiliser HTTPS. Pour plus d'informations, voir «Utilisation de l'authentification de base HTTP avec REST API», à la page 545.
- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur fournit un ID utilisateur et un mot de passe à la ressource REST API login avec la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Pour plus d'informations, voir «Utilisation de l'authentification basée sur un jeton avec l'API REST», à la page 547. Vous pouvez configurer l'intervalle d'expiration du jeton LTPA. Pour plus d'informations, voir Configuration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à REST API, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

Configuration d'un registre LDAP pour IBM MQ Console et REST API

Vous pouvez configurer un registre LDAP dans le fichier mqwebuser.xml . Les noms d'utilisateur et les mots de passe du registre LDAP sont utilisés pour authentifier et autoriser les utilisateurs de IBM MQ Console et de REST API.

Avant de commencer

- Lorsque vous configurez un registre LDAP, vous devez affecter un rôle à chaque utilisateur. Chaque rôle fournit différents niveaux de privilèges pour accéder à IBM MQ Console et REST API, et détermine le contexte de sécurité qui est utilisé lorsqu'une opération autorisée est tentée. Vous devez comprendre ces rôles avant de configurer le registre. Pour plus d'informations sur chacun des rôles, voir «Rôles sur IBM MQ Console et REST API», à la page 538.

N'oubliez pas que tout utilisateur disposant du rôle `MQWebUser` peut effectuer uniquement les opérations que l'ID utilisateur est autorisé à effectuer sur le gestionnaire de files d'attente. Par conséquent, l'ID utilisateur défini sur le serveur LDAP doit avoir un ID utilisateur identique sur le système sur lequel IBM MQ est installé. Ces ID utilisateur doivent être dans le même cas, sinon le mappage entre les ID utilisateur peut échouer.

- Pour effectuer cette tâche, vous devez être un utilisateur disposant de privilèges suffisants pour éditer le fichier `mqwebuser.xml` :
  - z/OS Sous z/OS, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` .
  - Multi Sur tous les autres systèmes d'exploitation, vous devez être un utilisateur privilégié.
  - V 9.4.0 Linux Si le serveur mqweb fait partie d'une installation IBM MQ Web Server autonome, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` dans le répertoire de données IBM MQ Web Server .

Procédure

1. Copiez l'exemple de fichier XML `ldap_registry.xml` à partir de l'un des chemins suivants:
   - Dans une installation IBM MQ :
     - ALW Sous AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS Sous z/OS: `PathPrefix /web/mq/samp/configuration`

       où `PathPrefix` est le chemin d'installation de IBM MQ for z/OS UNIX System Services Components .
   - V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     où *CHEMIN_INSTALLATION_MQ* est le répertoire dans lequel le fichier d'installation IBM MQ Web Server a été décompressé.
2. Placez l'exemple de fichier dans le répertoire approprié:
   - Dans une installation IBM MQ :
     - Linux AIX Sous AIX ou Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Sous Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`, où *MQ_DATA_PATH* est le chemin de données IBM MQ . Il s'agit du chemin de données sélectionné lors de l'installation de IBM MQ. Par défaut, ce chemin est `C:\ProgramData\IBM\MQ`.
     - z/OS Sous z/OS : *WLP_user_directory*`/servers/mqweb`

       où *WLP_user_directory* est le répertoire qui a été spécifié lors de l'exécution du script **crtmqweb** pour créer la définition de serveur mqweb.
   - V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     où *MQ_OVERRIDE_DATA_PATH* est le répertoire de données IBM MQ Web Server vers lequel la variable d'environnement **MQ_OVERRIDE_DATA_PATH** pointe.
3. Facultatif : Si vous avez modifié des paramètres de configuration dans `mqwebuser.xml`, copiez-les dans l'exemple de fichier.
4. Supprimez le fichier `mqwebuser.xml` existant et renommez l'exemple de fichier en `mqwebuser.xml`.

5. Editez le nouveau fichier `mqwebuser.xml` pour modifier les paramètres du registre LDAP dans les balises **ldapRegistry** et **idsLdapFilterProperties** .

   Pour plus d'informations sur la configuration des registres LDAP, voir Configuration des registres d'utilisateurs LDAP dans Liberty dans la documentation WebSphere Liberty .
6. Affectez des rôles aux utilisateurs et aux groupes en éditant le fichier `mqwebuser.xml` :

   Plusieurs rôles sont disponibles pour autoriser les utilisateurs et les groupes à utiliser le IBM MQ Consoleet le REST API. Chaque rôle accorde un niveau d'accès différent. Pour plus d'informations, voir «Rôles sur IBM MQ Console et REST API», à la page 538.

   - Pour affecter des rôles et accorder l'accès au IBM MQ Console, ajoutez vos utilisateurs et groupes entre les balises **security-role** appropriées dans les balises **<enterpriseApplication id="com.ibm.mq.console">** .
   - Pour affecter des rôles et accorder l'accès au REST API, ajoutez vos utilisateurs et groupes entre les balises **security-role** appropriées dans les balises **<enterpriseApplication id="com.ibm.mq.rest">** .

Que faire ensuite

Choisissez comment les utilisateurs s'authentifient:

IBM MQ Consoleoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur entre un ID utilisateur et un mot de passe à l'écran de connexion IBM MQ Console . Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Aucune configuration supplémentaire n'est requise pour utiliser cette option d'authentification, mais vous pouvez éventuellement configurer l'intervalle d'expiration pour le jeton LTPA. Pour plus d'informations, voir Configuration de l'intervalle d'expiration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à IBM MQ Console, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

REST APIoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification de base HTTP. Dans ce cas, un nom d'utilisateur et un mot de passe sont codés, mais non chiffrés, et envoyés avec chaque demande REST API pour authentifier et autoriser l'utilisateur pour cette demande. Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous devez utiliser HTTPS. Pour plus d'informations, voir «Utilisation de l'authentification de base HTTP avec REST API», à la page 545.
- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur fournit un ID utilisateur et un mot de passe à la ressource REST API `login` avec la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Pour plus d'informations, voir «Utilisation de l'authentification basée sur un jeton avec l'API REST», à la page 547. Vous pouvez configurer l'intervalle d'expiration du jeton LTPA. Pour plus d'informations, voir Configuration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à REST API, mais utilise le certificat client à la place. Pour plus d'informations, voir «Configuration de l'authentification par certificat client avec REST API et IBM MQ Console», à la page 542.

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 “Rôles sur IBM MQ Console et REST API” on page 538.
- 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 536.

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 536. 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 536. 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 536.

```
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 537. For more information about the roles, see “Rôles sur IBM MQ Console et REST API” on page 538.

    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 536.

```
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

Choisissez comment les utilisateurs s'authentifient:

IBM MQ Consoleoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur entre un ID utilisateur et un mot de passe à l'écran de connexion IBM MQ Console . Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Aucune configuration supplémentaire n'est requise pour utiliser cette option d'authentification, mais vous pouvez éventuellement configurer l'intervalle d'expiration pour le jeton LTPA. Pour plus d'informations, voir Configuration de l'intervalle d'expiration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à IBM MQ Console, mais utilise

le certificat client à la place. Pour plus d'informations, voir “Configuration de l'authentification par certificat client avec REST API et IBM MQ Console” on page 542.

REST APIoptions d'authentification

- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification de base HTTP. Dans ce cas, un nom d'utilisateur et un mot de passe sont codés, mais non chiffrés, et envoyés avec chaque demande REST API pour authentifier et autoriser l'utilisateur pour cette demande. Pour que cette authentification soit sécurisée, vous devez utiliser une connexion sécurisée. C'est-à-dire que vous devez utiliser HTTPS. Pour plus d'informations, voir “Utilisation de l'authentification de base HTTP avec REST API” on page 545.
- Permet aux utilisateurs de s'authentifier à l'aide de l'authentification par jeton. Dans ce cas, un utilisateur fournit un ID utilisateur et un mot de passe à la ressource REST API `login` avec la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur de rester connecté et autorisé pendant une durée définie. Pour plus d'informations, voir “Utilisation de l'authentification basée sur un jeton avec l'API REST” on page 547. Vous pouvez configurer l'intervalle d'expiration du jeton LTPA. Pour plus d'informations, voir Configuration du jeton LTPA.
- Permet aux utilisateurs de s'authentifier à l'aide de certificats client. Dans ce cas, l'utilisateur n'utilise pas d'ID utilisateur ou de mot de passe pour se connecter à REST API, mais utilise le certificat client à la place. Pour plus d'informations, voir “Configuration de l'authentification par certificat client avec REST API et IBM MQ Console” on page 542.

Rôles sur IBM MQ Console et REST API

Lorsque vous autorisez des utilisateurs et des groupes à utiliser IBM MQ Console ou REST API, vous devez affecter aux utilisateurs et aux groupes l'un des rôles disponibles: **MQWebAdmin**, **MQWebAdminRO**, **MQWebUser**, **MFTWebAdmin**et **MFTWebAdminRO**. Chaque rôle fournit différents niveaux de privilèges pour accéder à IBM MQ Console et REST API, et détermine le contexte de sécurité qui est utilisé lorsqu'une opération autorisée est tentée.

Remarque : A l'exception du rôle **MQWebUser** , l'ID utilisateur n'est pas sensible à la casse. Pour connaître les exigences spécifiques à ce rôle, voir «MQWebUser», à la page 539 .

MQWebAdmin
Un utilisateur ou un groupe auquel ce rôle est affecté peut effectuer toutes les opérations d'administration et fonctionne dans le contexte de sécurité de l'ID utilisateur du système d'exploitation utilisé pour démarrer le serveur mqweb.

Un utilisateur ou un groupe ayant ce rôle n'a pas accès aux services REST suivants:

- REST API pour MFT. Pour utiliser ces services, l'utilisateur ou le groupe doit également disposer du rôle **MFTWebAdmin** ou **MFTWebAdminRO** .
- messaging REST API. Pour utiliser le messaging REST API, le rôle **MQWebUser** doit être affecté à l'utilisateur.

MQWebAdminRO
Ce rôle permet d'accéder en lecture seule à IBM MQ Console ou REST API. Un utilisateur ou un groupe auquel ce rôle est affecté peut effectuer les opérations suivantes:

- Affichez et interrogez les opérations sur les objets IBM MQ tels que les files d'attente et les canaux.
- Parcourez les messages dans les files d'attente.

Un utilisateur ou un groupe auquel ce rôle est affecté fonctionne dans le contexte de sécurité de l'ID utilisateur du système d'exploitation utilisé pour démarrer le serveur mqweb.

Un utilisateur ou un groupe ayant ce rôle n'a pas accès aux services REST suivants:

- REST API pour MFT. Pour utiliser ces services, l'utilisateur ou le groupe doit également disposer du rôle **MFTWebAdmin** ou **MFTWebAdminRO** .
- messaging REST API. Pour utiliser le messaging REST API, le rôle **MQWebUser** doit être affecté à l'utilisateur.

MQWebUser
Un utilisateur ou un groupe auquel ce rôle est affecté peut effectuer toute opération que l'ID utilisateur est autorisé à effectuer sur le gestionnaire de files d'attente. Exemple :

- Opérations de démarrage et d'arrêt sur les objets IBM MQ tels que les canaux.
- Définissez et définissez des opérations sur des objets IBM MQ tels que des files d'attente et des canaux.
- Affichez et interrogez les opérations sur les objets IBM MQ tels que les files d'attente et les canaux.
- Insérez et extrayez des messages à l'aide de messaging REST API.

Un utilisateur ou un groupe auquel ce rôle est affecté agit dans le contexte de sécurité du principal et peut effectuer uniquement les opérations que l'ID utilisateur est autorisé à effectuer sur le gestionnaire de files d'attente.

Par conséquent, l'utilisateur ou le groupe défini dans le registre d'utilisateurs mqweb doit disposer des droits d'accès dans IBM MQ pour que cet utilisateur puisse effectuer des opérations. En utilisant ce rôle, vous pouvez contrôler finement quels utilisateurs ont quel type d'accès à des ressources IBM MQ spécifiques lorsqu'ils utilisent IBM MQ Console et REST API.

Remarque :

- La longueur maximale d'un ID utilisateur auquel ce rôle est affecté est de 12 caractères.
- La casse de l'ID utilisateur doit être la même dans le registre d'utilisateurs mqweb et sur le système IBM MQ . Si la casse de l'ID utilisateur est différente, l'utilisateur peut être authentifié par IBM MQ Console et REST API mais ne pas être autorisé à utiliser les ressources IBM MQ .

MFTWebAdmin
Un utilisateur ou un groupe auquel ce rôle a été affecté peut effectuer toutes les opérations REST MFT et fonctionne dans le contexte de sécurité de l'ID utilisateur du système d'exploitation utilisé pour démarrer le serveur `mqweb` .

Un utilisateur ou un groupe ayant ce rôle n'a accès à aucun des services IBM MQ REST API .
Pour utiliser ces services, l'utilisateur ou le groupe doit également disposer du rôle **MQWebAdmin**, **MQWebAdminRO**ou **MQWebUser** .

MFTWebAdminRO
Ce rôle permet d'accéder en lecture seule à REST API for MFT . Un utilisateur ou un groupe auquel ce rôle est affecté peut effectuer des opérations en lecture seule (demandes GET) telles que le transfert de liste et les agents de liste.

Un utilisateur ou un groupe auquel ce rôle est affecté fonctionne dans le contexte de sécurité de l'ID utilisateur du système d'exploitation utilisé pour démarrer le serveur mqweb.

Un utilisateur ou un groupe ayant ce rôle n'a accès à aucun des services IBM MQ REST API .
Pour utiliser ces services, l'utilisateur ou le groupe doit également disposer du rôle **MQWebAdmin**, **MQWebAdminRO**ou **MQWebUser** .

Pour plus d'informations sur la configuration des utilisateurs et des groupes pour utiliser ces rôles, voir «Configuration des utilisateurs et des rôles», à la page 527.

Chevauchement de rôles

Plusieurs rôles peuvent être affectés à un utilisateur ou à un groupe. Lorsqu'un utilisateur effectue une opération dans cette situation, le rôle de privilège le plus élevé applicable à l'opération est utilisé. Par exemple, si un utilisateur disposant des rôles **MQWebAdminRO** et **MQWebUser** effectue une opération d'interrogation de file d'attente, le rôle **MQWebAdminRO** est utilisé et l'opération est tentée dans le contexte de l'ID utilisateur système qui a démarré le serveur Web. Si ce même utilisateur effectue une opération de définition, le rôle **MQWebUser** est utilisé et l'opération est tentée dans le contexte du principal.

ALW Modification du certificat présenté par le IBM MQ Console dans votre navigateur

Vous pouvez configurer IBM MQ Console pour qu'il présente un certificat signé par une autorité de certification à des fins d'authentification. Si vous configurez IBM MQ Console pour qu'il présente un certificat signé par une autorité de certification, le navigateur ne présente plus d'avertissement de certificat autosigné lors de l'accès à IBM MQ Console .

Pourquoi et quand exécuter cette tâche

La sécurité du IBM MQ Console est fournie par le serveur mqweb qui exécute le IBM MQ Console. Pour modifier le certificat présenté par le serveur mqweb à votre navigateur, ajoutez d'abord le nouveau certificat au magasin de clés du serveur mqweb. Editez ensuite la configuration de sécurité dans le fichier `mqwebuser.xml` pour spécifier le certificat présenté par le serveur.

La procédure fait les hypothèses suivantes:

- Vous êtes un utilisateur privilégié.
- Vous utilisez un système AIX, Linuxou Windows .
- Votre fichier `mqwebuser.xml` est basé sur les exemples de fichier XML `basic_registry.xml`, `local_os_registry.xml`ou `ldap_registry.xml` .

Procédure

1. Facultatif : Modifiez le mot de passe par défaut du magasin de clés du serveur mqweb `key.jks` à l'aide de la commande **runmqktool** :

   ```
   runmqktool -storepasswd -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass oldPassword
            -new newPassword
   ```

   oldPassword
   Indique le mot de passe `key.jks` existant. Le mot de passe par défaut est `password`.

   newPassword
   Indique un nouveau mot de passe `key.jks` .

2. Créez une paire de clés et une demande de certificat à envoyer à l'autorité de certification:

   a) Créez la paire de clés à l'aide de la commande **runmqktool** :

   ```
   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
   ```

   password
   Indique le mot de passe du magasin de clés `key.jks` .

   label
   Indique le libellé du certificat. Par exemple, `MQWebConsole`.

   nom_distinctif
   Indique le nom distinctif X.500 du certificat. Placez le nom distinctif entre guillemets.

   Exemple: `"cn=MQWebConsole,o=myOrg,c=UK"`

   algorithme_signature
   Indique l'algorithme à utiliser pour signer le certificat. Pour plus d'informations, voir Algorithmes de signature

   b) Créez la demande de certificat à l'aide de la commande **runmqktool** :

   ```
   runmqktool -certreq -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass password -alias label
              -file filename
   ```

password
Indique le mot de passe du magasin de clés `key.jks` .

label
Indique le libellé de certificat de la sous-étape «2.a», à la page 540.

fichier
Indique le nom de fichier complet de la demande de certificat.

3. Envoyez le fichier de demande de certificat à une autorité de certification.
4. Lorsque vous disposez du certificat de l'autorité de certification, importez le certificat et tous les autres certificats de la chaîne de certificats, en commençant par le certificat de l'autorité de certification racine, dans le magasin de clés `keys.jks` à l'aide de la commande **runmqktool** :

```
runmqktool -importcert -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password
          -alias label -file filename
```

password
Indique le mot de passe du magasin de clés `key.jks` .

label
Indique le libellé de certificat de la sous-étape «2.a», à la page 540.

fichier
Indique le nom de fichier qualifié complet du certificat à importer.

5. Configurez le serveur mqweb pour qu'il présente le certificat de l'autorité de certification:
   a) Ouvrez le fichier `mqwebuser.xml`.

   Le fichier `mqwebuser.xml` se trouve dans le chemin suivant: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`

   b) Désactivez la configuration de sécurité par défaut en mettant en commentaire la ligne suivante:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   Si vous avez configuré le serveur mqweb pour utiliser l'authentification par certificat client, cette ligne du fichier xml est déjà mise en commentaire.

   c) Supprimez la mise en commentaire de la section du fichier `mqwebuser.xml` qui active la configuration de certificat personnalisé. La section contient le texte suivant:

```
<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"/>
```

   Si vous avez configuré le serveur mqweb pour utiliser l'authentification par certificat client, cette section du fichier xml est déjà supprimée de la mise en commentaire.

   d) Facultatif : Si vous avez modifié le mot de passe du magasin de clés `key.jks` à l'étape «1», à la page 540, remplacez la valeur de **password** dans les balises `defaultKeyStore` par une version codée du mot de passe que vous avez défini:

      i) Dans le répertoire *MQ_INSTALLATION_PATH*`/web/bin` , entrez la commande suivante:

```
securityUtility encode password
```

      ii) Placez la sortie de cette commande dans la zone **password** pour `defaultKeyStore`.

   e) Si vous n'utilisez pas l'authentification par certificat client, mettez en commentaire la ligne suivante:

```
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
```

f) Remplacez la valeur de **serverKeyAlias** `default` par la valeur du libellé de certificat de l'autorité de certification.

6. Arrêtez le serveur mqweb à l'aide de la commande **endmqweb** .
7. Démarrez le serveur mqweb à l'aide de la commande **strmqweb** .

Résultats

Lorsque le serveur Web démarre, accédez à votre IBM MQ Console et actualisez. Le certificat de l'autorité de certification est utilisé et vous accédez directement à la page de connexion.

ALW Configuration de l'authentification par certificat client avec REST API et IBM MQ Console

Vous pouvez mapper des certificats client à des principaux pour authentifier les utilisateurs IBM MQ Console et REST API .

Avant de commencer

- Configurez les utilisateurs, les groupes et les rôles pour qu'ils soient autorisés à utiliser IBM MQ Console et REST API. Pour plus d'informations, voir «Configuration des utilisateurs et des rôles», à la page 527.
- Lorsque vous utilisez le REST API, vous pouvez interroger les données d'identification de l'utilisateur en cours à l'aide de la méthode HTTP GET sur la ressource `login` , en fournissant le certificat client pour authentifier la demande. Cette demande renvoie des informations sur le nom d'utilisateur et les rôles affectés à l'utilisateur. Pour plus d'informations, voir GET `/login`.
- Lorsque vous mappez des certificats client à des principaux pour authentifier les utilisateurs, le nom distinctif du certificat client est utilisé pour établir une correspondance avec les utilisateurs du registre d'utilisateurs configuré:
  - Pour un registre de base, le nom usuel (CN) est comparé à l'utilisateur. Par exemple, `CN=Fred, O=IBM, C=GB` est comparé à un nom d'utilisateur `Fred`.
  - Pour un registre LDAP, par défaut, le nom distinctif complet est comparé à LDAP. Vous pouvez configurer des filtres et des mappages pour personnaliser la mise en correspondance. Pour plus d'informations, voir Liberty :LDAP certificate map mode dans la documentation WebSphere Liberty .

Pourquoi et quand exécuter cette tâche

Lorsqu'un utilisateur s'authentifie à l'aide d'un certificat client, le certificat est utilisé à la place d'un nom d'utilisateur et d'un mot de passe. Pour le REST API, le certificat client est fourni avec chaque demande REST pour authentifier l'utilisateur. Pour IBM MQ Console, lorsqu'un utilisateur se connecte avec un certificat, il ne peut pas être déconnecté.

ALW Sur les systèmes AIX, Linuxou Windows , la procédure suppose les informations suivantes:

- Votre fichier `mqwebuser.xml` est basé sur les exemples de fichier XML `basic_registry.xml`, `local_os_registry.xml`ou `ldap_registry.xml` .
- Que vous êtes un utilisateur privilégié.

z/OS Pour configurer l'authentification par certificat client avec un fichier de clés RACF sur les systèmes z/OS , suivez la procédure décrite dans «Configuring TLS for the REST API and IBM MQ Console on z/OS», à la page 555.

Remarque : La procédure suivante décrit les étapes nécessaires à l'utilisation des certificats client avec IBM MQ Console et REST API. Pour des raisons de commodité pour les développeurs, les étapes expliquent comment créer et utiliser des certificats autosignés. Toutefois, pour la production, utilisez des certificats obtenus auprès d'une autorité de certification.

Procédure

1. Créez un certificat à l'aide de la commande **runmqktool** :

```
runmqktool -genkeypair -keystore filename -storepass password -storetype PKCS12
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

fichier
Indique le nom du magasin de clés, par exemple `user.p12`. Si le magasin de clés n'existe pas, il est créé lors de l'exécution de la commande.

password
Indique le mot de passe du magasin de clés.

label
Indique le libellé du certificat. Par exemple, `user1`.

nom_distinctif
Indique le nom distinctif X.500 du certificat. Placez le nom distinctif entre guillemets.

Si vous utilisez un registre d'utilisateurs de base, entrez le nom d'un utilisateur de votre registre d'utilisateurs dans la partie Nom commun (CN) du nom distinctif. Par exemple, pour un utilisateur `mqadmin`, utilisez le nom distinctif `"CN=mqadmin"`.

Si vous utilisez un registre de système d'exploitation local, entrez le nom d'un ID utilisateur de système d'exploitation local dans la partie Nom commun (CN) du nom distinctif. Par exemple, pour un utilisateur `mqadmin`, utilisez le nom distinctif `"CN=mqadmin"`.

Si vous utilisez un registre d'utilisateurs LDAP, entrez un nom distinctif qui correspond au nom distinctif dans le registre LDAP.

algorithme_signature
Indique l'algorithme à utiliser pour signer le certificat. Pour plus d'informations, voir Algorithmes de signature

2. Facultatif : Obtenir un certificat d'une autorité de certification (CA). Sinon, pour utiliser un certificat autosigné, passez à l'étape «3», à la page 544.

a) Pour obtenir un certificat d'une autorité de certification, créez une demande de certificat à l'aide de la commande **runmqktool** :

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

fichier
Indique le nom du magasin de clés de l'étape «1», à la page 543.

password
Indique le mot de passe du magasin de clés.

label
Indique le libellé de certificat de l'étape «1», à la page 543.

fichier
Indique le nom de fichier complet de la demande de certificat.

b) Envoyez le fichier de demande de certificat à une autorité de certification.

c) Lorsque vous disposez du certificat de l'autorité de certification, importez le certificat dans votre magasin de clés à l'aide de la commande **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

fichier
Indique le nom du magasin de clés de l'étape «1», à la page 543.

password
Indique le mot de passe du magasin de clés.

label
Indique le libellé de certificat de l'étape «1», à la page 543.

fichier
Indique le nom de fichier complet du certificat de l'autorité de certification.

3. Extrayez la partie publique du certificat à l'aide de la commande **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass password
           -alias label -file filename -rfc
```

fichier
Indique le nom du magasin de clés de l'étape «1», à la page 543.

password
Indique le mot de passe du magasin de clés.

label
Indique le libellé de certificat de l'étape «1», à la page 543.

fichier
Indique le nom de fichier qualifié complet du certificat extrait.

4. Importez la partie publique du certificat dans le magasin de clés de confiance du serveur mqweb en tant que certificat de signataire afin que le serveur puisse valider le certificat client à l'aide de la commande **runmqktool** :

```
runmqktool -importcert -keystore  MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/trust.jks -storepass password
           -alias label -file filename
```

password
Indique le mot de passe du magasin de clés `trust.jks` . Vous pouvez spécifier un mot de passe pour un magasin de clés `trust.jks` existant ou un nouveau mot de passe pour un nouveau magasin de clés `trust.jks` .

label
Indique le libellé de certificat de l'étape «1», à la page 543.

fichier
Indique le nom de fichier qualifié complet du certificat extrait.

5. Configurez le serveur mqweb pour utiliser l'authentification par certificat client:
   a) Ouvrez le fichier `mqwebuser.xml`.

   Le fichier `mqwebuser.xml` se trouve dans le chemin suivant: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`

   b) Désactivez la configuration de sécurité par défaut en mettant en commentaire la ligne suivante:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   Si vous avez configuré le serveur mqweb pour présenter un certificat d'autorité de certification au navigateur, cette ligne est déjà mise en commentaire.

   c) Supprimez la mise en commentaire de la section du fichier `mqwebuser.xml` qui active l'authentification par certificat client. La section contient le texte suivant:

```
<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"/>
```

   Si vous avez configuré le serveur mqweb pour présenter un certificat d'autorité de certification au navigateur, la mise en commentaire de cette section est déjà annulée. Toutefois, vous devrez peut-être supprimer la mise en commentaire de la ligne **defaultTrustStore** .

d) Modifiez la valeur de **password** pour `defaultTrustStore` afin qu'elle corresponde au mot de passe du magasin de clés `trust.jks` :

i) Dans le répertoire *MQ_INSTALLATION_PATH*`/web/bin` , entrez la commande suivante:

```
securityUtility encode password
```

ii) Placez la sortie de cette commande dans la zone **password** de `defaultTrustStore`.

6. Arrêtez le serveur mqweb à l'aide de la commande **endmqweb** .
7. Démarrez le serveur mqweb à l'aide de la commande **strmqweb** .
8. Utilisez le certificat client pour l'authentification:

- Pour utiliser le certificat client avec IBM MQ Console, installez le certificat client dans le navigateur Web utilisé pour accéder au IBM MQ Console.
- Pour utiliser le certificat client avec REST API, indiquez le certificat client avec chaque demande REST. Lorsque vous utilisez les méthodes HTTP POST, PATCH ou DELETE, vous devez fournir une authentification supplémentaire avec le certificat client pour empêcher les attaques de falsification de requêtes entre sites. Autrement dit, l'authentification supplémentaire est utilisée pour confirmer que les données d'identification utilisées pour authentifier la demande sont utilisées par le propriétaire des données d'identification.

  Cette authentification supplémentaire est fournie par l'en-tête HTTP `ibm-mq-rest-csrf-token` . Définissez la valeur de l'en-tête `ibm-mq-csrf-token` sur n'importe quelle valeur, y compris une valeur vide, puis soumettez la demande.

Exemple

Important : Dans l'exemple, toutes les implémentations cURL ne prenant pas en charge les certificats autosignés, vous devez utiliser une implémentation cURL .

L'exemple cURL suivant montre comment créer une nouvelle file d'attente `Q1`, sur un gestionnaire de files d'attente `QM1`, avec authentification par certificat client. La configuration exacte de cette commande cURL dépend des bibliothèques avec lesquelles cURL a été généré. L'exemple est basé sur un système Windows avec cURL généré avec OpenSSL.

- Utilisez la méthode HTTP POST avec la ressource de file d'attente, en vous authentifiant avec le certificat client et en incluant l'en-tête HTTP `ibm-mq-rest-csrf-token` avec une valeur arbitraire. Cette valeur peut être n'importe quoi, y compris vide. L'indicateur `--cert-type` spécifie que le certificat est un certificat PKCS#12 . L'indicateur `--cert` spécifie l'emplacement du certificat, suivi d'un signe deux-points, puis du mot de passe du certificat:

```
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\"}"
```

Utilisation de l'authentification de base HTTP avec REST API

Les utilisateurs du REST API peuvent s'authentifier en fournissant leur ID utilisateur et leur mot de passe dans un en-tête HTTP. Pour utiliser cette méthode d'authentification avec des méthodes HTTP, telles que POST, PATCH et DELETE, l'en-tête HTTP `ibm-mq-rest-csrf-token` doit également être fourni, ainsi qu'un ID utilisateur et un mot de passe.

Avant de commencer

- Configurez les utilisateurs, les groupes et les rôles pour qu'ils soient autorisés à utiliser le REST API. Pour plus d'informations, voir «Configuration des utilisateurs et des rôles», à la page 527.

- Vérifiez que l'authentification de base HTTP est activée. Vérifiez que le code XML suivant est présent et qu'il n'est pas mis en commentaire dans le fichier `mqwebuser.xml` . Ce code XML doit se trouver dans les balises `<featureManager>` :

  ```
  <feature>basicAuthenticationMQ-1.0</feature>
  ```

  z/OS Sous z/OS, vous devez être un utilisateur disposant d'un accès en écriture à `mqwebuser.xml` pour pouvoir éditer ce fichier.

  Multi Sur tous les autres systèmes d'exploitation, vous devez être un utilisateur privilégié pour pouvoir éditer le fichier `mqwebuser.xml` .
- Vérifiez que vous utilisez une connexion sécurisée lorsque vous envoyez des demandes REST. Comme la combinaison du nom d'utilisateur et du mot de passe est codée, mais pas chiffrée, vous devez utiliser une connexion sécurisée (HTTPS) lorsque vous utilisez l'authentification de base HTTP avec REST API.
- Vous pouvez interroger les données d'identification de l'utilisateur en cours à l'aide de la méthode HTTP GET sur la ressource `login` , en fournissant les informations d'authentification de base pour authentifier la demande. Cette demande renvoie des informations sur le nom d'utilisateur et les rôles affectés à l'utilisateur. Pour plus d'informations, voir GET `/login`.

Procédure

1. Concaténez le nom d'utilisateur avec un signe deux-points et le mot de passe. Notez que le nom d'utilisateur est sensible à la casse.

   Par exemple, le nom d'utilisateur admin et le mot de passe admin deviennent la chaîne suivante:

   ```
   admin:admin
   ```
2. Codez ce nom d'utilisateur et cette chaîne de mot de passe dans le codage base64 .
3. Incluez ce nom d'utilisateur et ce mot de passe codés dans un en-tête HTTP `Authorization: Basic` .

   Par exemple, avec le nom d'utilisateur codé admin et le mot de passe admin, l'en-tête suivant est créé:

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```
4. Lorsque vous utilisez les méthodes HTTP POST, PATCH ou DELETE, vous devez fournir une authentification supplémentaire, ainsi qu'un nom d'utilisateur et un mot de passe.

   Cette authentification supplémentaire est fournie par l'en-tête HTTP `ibm-mq-rest-csrf-token` . L'en-tête HTTP `ibm-mq-rest-csrf-token` doit être présent dans la demande, mais sa valeur peut être quelconque, y compris vide.
5. Soumettez votre demande REST à IBM MQ avec les en-têtes appropriés.

Exemple

L'exemple suivant montre comment créer une nouvelle file d'attente Q1, sur le gestionnaire de files d'attente QM1, avec authentification de base, sur les systèmes Windows . L'exemple utilise cURL:

- Utilisez la méthode HTTP POST avec la ressource de file d'attente, en vous authentifiant avec l'authentification de base et en incluant l'en-tête HTTP `ibm-mq-rest-csrf-token` avec une valeur arbitraire. Cette valeur peut être n'importe quoi, y compris vide:

  ```
  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\"}"
  ```

Utilisation de l'authentification basée sur un jeton avec l'API REST

Les utilisateurs de REST API peuvent s'authentifier en fournissant un ID utilisateur et un mot de passe à la ressource REST API `login` à l'aide de la méthode HTTP POST. Un jeton LTPA est généré pour permettre à l'utilisateur d'authentifier les demandes futures. Ce jeton LTPA a le préfixe `LtpaToken2`. L'utilisateur peut se déconnecter à l'aide de la méthode HTTP DELETE et peut interroger les informations de connexion de l'utilisateur en cours à l'aide de la méthode HTTP GET.

Avant de commencer

- Configurez les utilisateurs, les groupes et les rôles pour qu'ils soient autorisés à utiliser le REST API. Pour plus d'informations, voir «Configuration des utilisateurs et des rôles», à la page 527.
- Par défaut, le nom du cookie qui inclut le jeton LTPA commence par `LtpaToken2`et inclut un suffixe qui peut être modifié lorsque le serveur mqweb est redémarré. Ce nom de cookie aléatoire permet à plusieurs serveurs mqweb de s'exécuter sur le même système. Toutefois, si vous souhaitez que le nom du cookie reste cohérent, vous pouvez spécifier le nom du cookie à l'aide de la commande **setmqweb** . Pour plus d'informations, voir Configuration du jeton LTPA.
- Par défaut, le cookie de jeton LTPA expire au bout de 120 minutes. Vous pouvez configurer l'heure d'expiration du cookie de jeton LTPA à l'aide de la commande **setmqweb** . Pour plus d'informations, voir Configuration du jeton LTPA.
- Vérifiez que vous utilisez une connexion sécurisée lorsque vous envoyez des demandes REST. Lorsque vous utilisez la méthode HTTP POST sur la ressource `login` , la combinaison de nom d'utilisateur et de mot de passe envoyée avec la demande n'est pas chiffrée. Par conséquent, vous devez utiliser une connexion sécurisée (HTTPS) lorsque vous utilisez l'authentification basée sur un jeton avec REST API. Par défaut, vous ne pouvez pas utiliser HTTP avec l'authentification par jeton LTPA. Vous pouvez activer le jeton LTPA à utiliser par les connexions HTTP non sécurisées en définissant **secureLTPA** sur `False`. Pour plus d'informations, voir Configuration du jeton LTPA.
- Vous pouvez interroger les données d'identification de l'utilisateur en cours à l'aide de la méthode HTTP GET sur la ressource `login` , en fournissant le jeton LTPA pour authentifier la demande. Cette demande renvoie des informations sur le nom d'utilisateur et les rôles affectés à l'utilisateur. Pour plus d'informations, voir GET `/login`.

Procédure

1. Connectez-vous à un utilisateur:
   a) Utilisez la méthode HTTP POST sur la ressource `login` :

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   Incluez le nom d'utilisateur et le mot de passe dans le corps de la demande JSON, au format suivant:

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) Stockez le jeton LTPA renvoyé par la demande dans le magasin de cookies local. Par défaut, ce jeton LTPA a le préfixe `LtpaToken2`.
2. Authentifiez les demandes REST avec le jeton LTPA stocké en tant que cookie avec chaque demande.
   Pour les demandes qui utilisent les méthodes HTTP PUT, PATCH ou DELETE, incluez un en-tête `ibm-mq-rest-csrf-token` . La valeur de cet en-tête peut être n'importe quoi, y compris vide.
3. Déconnecter un utilisateur:
   a) Utilisez la méthode HTTP DELETE sur la ressource `login` :

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

Vous devez fournir le jeton LTPA en tant que cookie pour authentifier la demande et inclure un en-tête `ibm-mq-rest-csrf-token` . La valeur de cet en-tête peut être n'importe quoi, y compris vide

b) Traitez l'instruction de suppression du jeton LTPA du magasin de cookies local.

Remarque : Si l'instruction n'est pas traitée et que le jeton LTPA reste dans le magasin de cookies local, le jeton LTPA peut être utilisé pour authentifier les futures demandes REST. C'est-à-dire que lorsque l'utilisateur tente de s'authentifier avec le jeton LTPA après la fin de la session, une nouvelle session est créée qui utilise le jeton existant.

Exemple

L'exemple cURL suivant montre comment créer une nouvelle file d'attente `Q1`, sur le gestionnaire de files d'attente QM1, avec authentification basée sur un jeton, sur les systèmes Windows :

- Connectez-vous et ajoutez le jeton LTPA avec le préfixe `LtpaToken2`au magasin de cookies local. Les informations de nom d'utilisateur et de mot de passe sont incluses dans le corps JSON. L'indicateur `-c` spécifie l'emplacement du fichier dans lequel stocker le jeton:

```
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
```

- Créez une file d'attente. Utilisez la méthode HTTP POST avec la ressource de file d'attente, en vous authentifiant avec le jeton LTPA. Le jeton LTPA avec le préfixe `LtpaToken2` est extrait du fichier `cookiejar.txt` à l'aide de l'indicateur `-b` . La protection CSRF est assurée par la présence de l'en-tête HTTP `ibm-mq-rest-csrf-token` :

```
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\"}"
```

- Déconnectez-vous et supprimez le jeton LTPA du magasin de cookies local. Le jeton LTPA est extrait du fichier `cookiejar.txt` à l'aide de l'indicateur `-b` . La protection CSRF est assurée par la présence de l'en-tête HTTP `ibm-mq-rest-csrf-token` . L'emplacement du fichier `cookiejar.txt` est spécifié par l'indicateur `-c` afin que le jeton LTPA soit supprimé du fichier:

```
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
```

Référence associée

POST `/login`
GET`/login`
Supprimer `/login`

Incorporation d'IBM MQ Console dans une trame d'information

L'élément HTML < iframe> peut être utilisé pour imbriquer une page Web dans une autre à l'aide d'un cadre en ligne (IFrame). Pour des raisons de sécurité, le IBM MQ Console ne peut pas être imbriqué dans un IFrame par défaut. Toutefois, vous pouvez activer un IFrame à l'aide de la propriété de configuration **mqConsoleFrameAncestors** sur le serveur mqweb.

Pourquoi et quand exécuter cette tâche

Le serveur mqweb gère une liste autorisée des origines des pages Web qui peuvent incorporer le IBM MQ Console à l'aide d'un IFrame. Une origine est une combinaison d'un schéma d'URL, d'un domaine et d'un port, par exemple, `https://example.com:1234`.

Vous pouvez utiliser la propriété de configuration **mqConsoleFrameAncestors** sur le serveur mqweb pour spécifier les entrées dans la liste.

Par défaut, **mqConsoleFrameAncestors** est vide, ce qui signifie que IBM MQ Console ne peut pas être imbriqué dans un IFrame.

Procédure

Spécifiez une liste d'origines de pages Web, qui peuvent imbriquer le IBM MQ Console dans un IFrame, en entrant la commande suivante:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

où *allowedOrigins* est une liste d'origines séparées par des virgules. Chaque origine doit se composer des éléments suivants:

- Un nom d'hôte ou une adresse IP
- Un schéma d'URL facultatif
- Numéro de port facultatif

Notez que le nom d'hôte peut commencer par le caractère générique (*) et que le numéro de port peut également utiliser le caractère générique (*).

Exemples d'origines:

```
https://example.com:1234
```

qui permet à n'importe quelle page Web servie à partir de `https://example.com:1234` d'imbriquer le IBM MQ Console dans un IFrame.

```
https://*.example.com:*
```

qui permet à toute page Web HTTPS avec un nom d'hôte se terminant par `example.com`, et utilisant n'importe quel port, d'imbriquer le IBM MQ Console dans un IFrame.

Exemple

L'exemple suivant permet à IBM MQ Console d'être imbriqué dans un IFrame à partir de pages Web servies à partir de `https://site2.example.com:1234` ou de `https://site2.example.com:1235`:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

Configuration de CORS pour REST API

Par défaut, un navigateur Web n'autorise pas les scripts, tels que JavaScript, à appeler REST API lorsque le script n'est pas de la même origine que le REST API. C'est-à-dire que les demandes d'origine croisée ne sont pas activées. Vous pouvez configurer le partage de ressources d'origine croisée (CORS) pour autoriser les demandes d'origine croisée provenant d'origines spécifiées.

Pourquoi et quand exécuter cette tâche

Vous pouvez accéder à REST API via un navigateur Web, par exemple via un script. Comme ces demandes proviennent d'une origine différente de celle de REST API, le navigateur Web refuse la demande car il s'agit d'une demande d'origine croisée. L'origine est différente si le domaine, le port ou le schéma n'est pas le même.

Par exemple, si vous disposez d'un script hébergé sur `http://localhost:1999/` , vous effectuez une demande inter-origine si vous émettez une requête HTTP GET sur un site Web hébergé sur `https://localhost:9443/`. Cette demande est une demande inter-origine car les numéros de port et le schéma (HTTP) sont différents.

Vous pouvez activer les demandes inter-origines en configurant CORS et en spécifiant les origines qui sont autorisées à accéder à REST API.

Pour plus d'informations sur CORS, voir https://www.w3.org/TR/cors/ et https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Procédure

1. Affichez la configuration en cours en entrant la commande suivante:

   ```
   dspmqweb properties -a
   ```

   L'entrée `mqRestCorsAllowedOrigins` indique les origines autorisées. L'entrée `mqRestCorsMaxAgeInSeconds` indique la durée, en secondes, pendant laquelle le navigateur Web peut mettre en cache les résultats des vérifications préalables à la mise en cache CORS.
2. Indiquez les origines autorisées à accéder à REST API en entrant la commande suivante:

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   où *allowedOrigins* indique l'origine à partir de laquelle vous souhaitez autoriser les demandes inter-origine. Vous pouvez utiliser un astérisque entre guillemets, "*", pour autoriser toutes les demandes d'origine croisée. Vous pouvez entrer plusieurs origines dans une liste séparée par des virgules, en les plaçant entre guillemets. Pour n'autoriser aucune demande d'origine croisée, entrez des guillemets vides comme valeur pour *allowedOrigins*.
3. Indiquez la durée, en secondes, pendant laquelle vous souhaitez autoriser un navigateur Web à mettre en cache les résultats des vérifications CORS préalables à la mise en cache en entrant la commande suivante:

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

Exemple

L'exemple suivant illustre les demandes d'origine croisée activées pour `http://localhost:9883`, `https://localhost:1999`et `https://localhost:9663`. L'âge maximal des résultats mis en cache des vérifications CORS avant vol est défini sur 90 secondes:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

Configuration de la validation de l'en-tête d'hôte pour IBM MQ Console et REST API

Vous pouvez configurer le serveur mqweb pour restreindre l'accès à IBM MQ Console et à REST API de sorte que seules les demandes envoyées avec un en-tête d'hôte correspondant à une liste blanche spécifiée soient traitées. Une erreur est renvoyée si une valeur d'en-tête d'hôte qui n'est pas dans la liste blanche est utilisée.

Pourquoi et quand exécuter cette tâche

Le serveur mqweb utilise des hôtes virtuels pour définir la liste autorisée des en-têtes d'hôte acceptables. Pour plus d'informations sur les hôtes virtuels, voir la documentation WebSphere Liberty : https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

Pour effectuer cette tâche, vous devez être un utilisateur disposant de privilèges suffisants pour éditer le fichier `mqwebuser.xml` :

- z/OS Sous z/OS, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` .
- Multi Sur tous les autres systèmes d'exploitation, vous devez être un utilisateur privilégié.

- V 9.4.0 Linux Si le serveur mqweb fait partie d'une installation IBM MQ Web Server autonome, vous devez disposer d'un accès en écriture au fichier `mqwebuser.xml` dans le répertoire de données IBM MQ Web Server .

Procédure

1. Ouvrez le fichier `mqwebuser.xml`. Ce fichier se trouve dans l'un des emplacements suivants:
   - Dans une installation IBM MQ :
     - Linux AIX Sous AIX ou Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Sous Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`, où *MQ_DATA_PATH* est le chemin de données IBM MQ . Il s'agit du chemin de données sélectionné lors de l'installation de IBM MQ. Par défaut, ce chemin est `C:\ProgramData\IBM\MQ`.
     - z/OS Sous z/OS : *WLP_user_directory*`/servers/mqweb`

       Où *WLP_user_directory* est le répertoire qui a été spécifié lors de l'exécution de la commande **crtmqweb** pour créer la définition de serveur mqweb.
   - V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     où *MQ_OVERRIDE_DATA_PATH* est le répertoire de données IBM MQ Web Server vers lequel la variable d'environnement **MQ_OVERRIDE_DATA_PATH** pointe.
2. Ajoutez ou supprimez la mise en commentaire du code suivant dans le fichier `mqwebuser.xml` :

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. Editez la zone **<hostAlias>** en insérant la combinaison de nom d'hôte et de port que vous souhaitez autoriser.

   Cette combinaison peut être le nom d'hôte et le nom de port que vous avez utilisés dans la configuration du serveur mqweb. Par exemple, si vous utilisez la configuration par défaut de `localhost:9443`, vous pouvez utiliser `localhost:9443` dans la zone **<hostAlias>** .

   Si nécessaire, vous pouvez ajouter plusieurs zones **<hostAlias>** dans les balises **<virtualHost>** pour autoriser davantage de combinaisons de nom d'hôte et de port. Par exemple, pour autoriser les en-têtes d'hôte qui utilisent un port HTTP ainsi que les en-têtes d'hôte qui utilisent le port HTTPS.

Audit

Les enregistrements d'audit des opérations effectuées dans IBM MQ Console et REST API peuvent être générés en activant les événements de commande et de configuration du gestionnaire de files d'attente, et sur AIX, Linux, and Windows les modifications d'état significatives sont enregistrées dans les fichiers journaux du serveur mqweb.

Changements d'état significatifs

ALW

Sous AIX, Linux, and Windows, IBM MQ Console enregistre les changements d'état significatifs sous forme de messages dans les journaux du serveur mqweb. Chaque message indique le nom du principal authentifié qui a demandé l'opération.

Les modifications d'état importantes, telles que la création, le démarrage, l'arrêt ou la suppression des gestionnaires de files d'attente, sont consignées dans les fichiers `messages.log` et `console.log`

du serveur mqweb au niveau de journalisation [AUDIT]. Chaque entrée de journal indique le nom du principal authentifié qui a demandé l'opération.

Les fichiers `messages.log` et `console.log` se trouvent à l'emplacement suivant:

- Dans une installation IBM MQ :
  - Linux AIX Sous AIX ou Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`
  - Windows Sous Windows: *`MQ_DATA_PATH`*`\web\installations\`*`installationName`*`\servers\mqweb\logs`, où *`MQ_DATA_PATH`* est le chemin de données IBM MQ . Il s'agit du chemin de données sélectionné lors de l'installation de IBM MQ. Par défaut, ce chemin est `C:\ProgramData\IBM\MQ`.
- V 9.4.0 Linux Dans une installation IBM MQ Web Server autonome: *`MQ_OVERRIDE_DATA_PATH`*`/web/installations/MQWEBINST/servers/mqweb/logs`

  où *`MQ_OVERRIDE_DATA_PATH`* est le répertoire de données IBM MQ Web Server vers lequel la variable d'environnement **`MQ_OVERRIDE_DATA_PATH`** pointe.

Pour plus d'informations sur la configuration des niveaux de journalisation du serveur mqweb, voir Configuration de la journalisation.

Evénements de commande et de configuration

Vous pouvez éventuellement activer des événements de commande et de configuration sur le gestionnaire de files d'attente pour fournir des informations sur la plupart des activités IBM MQ Console et REST API . Par exemple, la création de canaux et l'interrogation de files d'attente génèrent des événements de commande et de configuration. Pour plus d'informations sur l'activation des événements de commande et de configuration, voir Contrôle des événements de configuration, de commande et de consignateur.

Pour ces messages d'événement de commande et de configuration, la zone **`MQIACF_EVENT_ORIGIN`** est définie sur `MQEVO_REST` et la zone **`MQCACF_EVENT_APPL_IDENTITY`** signale les 32 premiers caractères du nom principal authentifié. Si un utilisateur a le rôle `MQWebAdmin` ou `MQWebAdminRO` , la zone **`MQCACF_EVENT_USER_ID`** indique l'ID utilisateur du serveur mqweb, et non le nom d'utilisateur du principal qui a émis la commande. Toutefois, si l'utilisateur a le rôle `MQWebUser` , **`MQCACF_EVENT_USER_ID`** indique le nom d'utilisateur du principal qui a émis la commande.

Concepts associés

«Audit», à la page 492
Vous pouvez vérifier les intrusions de sécurité ou les tentatives d'intrusion à l'aide de messages d'événement. Vous pouvez également vérifier la sécurité de votre système à l'aide de la IBM MQ Explorer.

z/OS Remarques relatives à la sécurité pour IBM MQ Console et REST API sur z/OS

Les IBM MQ Console et REST API disposent de fonctions de sécurité contrôlant si un utilisateur peut émettre, afficher ou modifier des commandes. Les commandes sont ensuite transmises au gestionnaire de files d'attente, puis la sécurité du gestionnaire de files d'attente est utilisée pour contrôler si l'utilisateur est autorisé à exécuter la commande sur ce gestionnaire de files d'attente spécifique.

Procédure

1. Vérifiez que l'ID utilisateur de la tâche démarrée du serveur mqweb dispose des droits appropriés pour émettre certaines commandes PCF et accéder à certaines files d'attente. Pour plus d'informations, voir «Authority required by the mqweb server started task user ID», à la page 553.
2. Vérifiez que tous les utilisateurs auxquels le rôle `MQWebUser` a été attribué disposent des droits appropriés.

Les utilisateurs IBM MQ Console et REST API affectés au rôle `MQWebUser` fonctionnent dans le contexte de sécurité du principal. Ces ID utilisateur peuvent uniquement effectuer des opérations que l'ID utilisateur est autorisé à effectuer sur le gestionnaire de files d'attente et doivent être autorisés à accéder aux mêmes files d'attente système que l'espace adresse du serveur mqweb.

L'ID utilisateur de la tâche démarrée du serveur mqweb doit disposer d'un autre accès utilisateur à tous les utilisateurs affectés au rôle `MQWebUser` .

Pour plus d'informations sur l'octroi des droits appropriés aux utilisateurs ayant le rôle `MQWebUser` , voir «Accès aux ressources IBM MQ requises pour utiliser IBM MQ Console ou REST API», à la page 553.

3. Facultatif : Configurez TLS pour IBM MQ Console et REST API. Pour plus d'informations, voir «Configuring TLS for the REST API and IBM MQ Console on z/OS», à la page 555.

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 237, “System queue security” on page 216, and “Profiles for context security” on page 226.
- 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 535 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 207.

z/OS Accès aux ressources IBM MQ requises pour utiliser IBM MQ Console ou REST API

Les opérations effectuées dans IBM MQ Consoleou REST APIpar un utilisateur ayant le rôle `MQWebUser` sont effectuées dans le contexte de sécurité de l'utilisateur.

Pourquoi et quand exécuter cette tâche

Pour plus d'informations sur les rôles dans IBM MQ Console et REST API, voir «Rôles sur IBM MQ Console et REST API», à la page 538 .

Utilisez la procédure suivante pour accorder à un utilisateur, dans le rôle `MQWebUser` , l'accès aux ressources de gestionnaire de files d'attente requises pour utiliser IBM MQ Console ou REST API.

Procédure

1. Accordez à l'ID utilisateur `mqweb server started task` un autre accès utilisateur à chaque ID utilisateur du rôle `MQWebUser` .

   Effectuez cette opération sur chaque gestionnaire de files d'attente que les utilisateurs administreront via IBM MQ Console ou REST API.

   Vous pouvez utiliser les exemples de commande RACF suivants pour accorder à l'ID utilisateur `mqweb server started task` un accès utilisateur de remplacement à un utilisateur ayant le rôle `MQWebUser` :

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   où :

   hlq
   Correspond au préfixe de profil, qui peut être le nom du gestionnaire de files d'attente ou le nom du groupe de partage de files d'attente

   userId
   Est l'utilisateur ayant le rôle `MQWebUser`

   mqwebUserId
   Est l'ID utilisateur `mqweb server started task`

   Remarque : Si vous utilisez la sécurité à casse mixte, utilisez la classe MXADMIN plutôt que la classe MQADMIN.
2. Accordez à chaque utilisateur du rôle `MQWebUser` l'accès aux files d'attente système nécessaires pour utiliser IBM MQ Console et REST API.

   Pour ce faire, pour les deux systèmes SYSTEM.ADMIN.COMMAND.QUEUE et SYSTEM.REST.REPLY.QUEUE, accordez à chaque utilisateur l'accès UPDATE aux classes MQQUEUE ou MXQUEUE, selon que la sécurité à casse mixte est utilisée ou non.

   Vous devez effectuer cette opération sur chaque gestionnaire de files d'attente que l'utilisateur administrera via REST API, y compris les gestionnaires de files d'attente éloignées gérés via la passerelle administrative REST API.
3. Pour permettre à un utilisateur ayant le rôle `MQWebUser` d'administrer des gestionnaires de files d'attente éloignées, accordez à l'utilisateur l'accès UPDATE au profil dans la classe MQQUEUE ou MXQUEUE, en protégeant la file d'attente de transmission utilisée pour envoyer des commandes au gestionnaire de files d'attente éloignées. Notez que vous devez accorder à l'utilisateur l'accès UPDATE sur le gestionnaire de files d'attente de passerelle.

   Sur le gestionnaire de files d'attente éloignées, accordez l'accès au même utilisateur pour l'insertion dans la file d'attente de transmission utilisée pour renvoyer les messages de réponse de commande au gestionnaire de files d'attente de passerelle.
4. Accordez aux utilisateurs du rôle `MQWebUser` l'accès à toutes les autres ressources requises pour effectuer les opérations prises en charge par IBM MQ Console et REST API.

   L'accès nécessaire pour:

   - L'exécution d'opérations dans le REST APIest décrite dans les sections *Exigences de sécurité* des ressources REST API individuelles

- L'exécution de commandes par le IBM MQ Console est décrite dans «IBM MQ Console - required command security profiles», à la page 237

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 Gestion des clés et des certificats sur AIX, Linux, and Windows

Sous AIX, Linux, and Windows, utilisez les commandes **runmqakm** et V 9.4.0 V 9.4.0 **runmqktool** pour gérer les clés, les certificats et les demandes de certificat.

Pourquoi et quand exécuter cette tâche

La commande **runmqakm** fournit des fonctions similaires à celles de **gskitcapicmd**.

V 9.4.0 V 9.4.0 La commande **runmqktool** fournit des fonctions similaires à celles de l'utilitaire de gestion de certificats Java **keytool** . Avant d'utiliser les commandes **runmqakm** ou **runmqktool** , vérifiez que les variables d'environnement système sont correctement configurées en exécutant la commande **setmqenv** .

La commande **runmqktool** requiert l'installation du composant IBM MQ JRE. Si ce composant n'est pas installé, vous pouvez utiliser la commande **runmqakm** à la place.

Si vous devez gérer les certificats TLS d'une manière conforme à la norme FIPS, utilisez la commande **runmqakm** . En effet, la commande **runmqakm** prend en charge un chiffrement renforcé.

Procédure

- Utilisez les commandes **runmqakm** et **runmqktool** pour effectuer les actions suivantes:
  - Créez un référentiel de clés CMS et PKCS #12 pris en charge par IBM MQ .

- Créez des demandes de certificat.
- Exportez des certificats.
- Importez des certificats personnels et des certificats de l'autorité de certification.
- Gérez les certificats autosignés.
- Créez, extrayez et ajoutez des clés secrètes.

Information associée
Keytool (utilitaire)

ALW Commandes `runmqakm` et `runmqktool` sous AIX, Linux, and Windows

Sur les systèmes AIX, Linux, and Windows , utilisez les commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool) pour gérer les clés et les certificats.

Remarque : V 9.4.0 V 9.4.0

Depuis la IBM MQ 9.4.0, les commandes **runmqckm** et **strmqikm** sont supprimées. La commande **runmqktool** peut être utilisée à la place de la commande **runmqckm** pour gérer les référentiels de clés PKCS #12 et JKS. Il n'y a pas de remplacement pour l'interface graphique **strmqikm** .

Les commandes **runmqckm** et **runmqktool** présentent les différences importantes suivantes:

- La commande **runmqktool** ne prend pas en charge les fichiers de dissimulation pour stocker les mots de passe de référentiel de clés. Le mot de passe permettant d'accéder à un référentiel de clés doit toujours être fourni à la commande **runmqktool** lors de son exécution, soit en tant que paramètre de la commande, soit en réponse à une invite émise par la commande.
- La commande **runmqktool** ne prend pas en charge les référentiels de clés CMS . Par conséquent, pour exporter un certificat d'un JKS vers un référentiel de clés CMS , vous devez effectuer les étapes suivantes:
  1. La commande **runmqktool -importkeystore** permet de copier le certificat du référentiel de clés JKS vers un référentiel de clés PKCS #12 intermédiaire. Pour plus d'informations sur l'exportation d'un certificat, voir «Exportation d'un certificat personnel à partir d'un référentiel de clés sous AIX, Linux, and Windows», à la page 568.
  2. Utilisez la commande **runmqakm -cert -import** pour importer le certificat du référentiel de clés PKCS #12 intermédiaire vers le référentiel de clés CMS . Pour plus d'informations sur l'importation d'un certificat, voir «Importation d'un certificat personnel dans un référentiel de clés sous AIX, Linux, and Windows», à la page 570.

Les commandes IBM MQ suivantes peuvent être utilisées pour gérer les clés et les certificats:

runmqakm

- Fournit des fonctions similaires à celles de **gskitcapicmd**.
- Prend en charge les référentiels de clés CMS et PKCS #12 .
- Prend en charge la création d'un fichier de dissimulation pour stocker le mot de passe chiffré du référentiel de clés.
- Certifié conforme à la norme FIPS 140-2, et peut être configuré pour fonctionner de manière conforme à la norme FIPS avec le paramètre **-fips** .

V 9.4.0 V 9.4.0 **runmqktool**

- Fournit des fonctions similaires à celles de la commande Java **keytool** .
- Prend en charge les référentiels de clés PKCS #12, JKS et JCEKS.
- Nécessite que le composant IBM MQ Java runtime environment (JRE) soit installé.

Si vous devez gérer les certificats d'une manière compatible FIPS, utilisez la commande **runmqakm** .

Pour plus d'informations sur la commande **runmqakm** , voir runmqakm.

V 9.4.0 V 9.4.0 Pour plus d'informations sur la commande **runmqktool** , voir runmqktool.

Les rubriques de cette section contiennent des exemples d'utilisation de ces commandes pour effectuer des tâches de gestion de certificats communes.

ALW Création d'un certificat personnel autosigné sur AIX, Linux, and Windows

Suivez cette procédure pour créer un certificat personnel autosigné dans un référentiel de clés.

Remarque : IBM MQ ne prend pas en charge les algorithmes SHA-3 ou SHA-5 . Vous pouvez utiliser les noms d'algorithme de signature numérique SHA384WithRSA et SHA512WithRSA car ces deux algorithmes sont membres de la famille SHA-2 .

Deprecated Les noms d'algorithme de signature numérique SHA3WithRSA et SHA5WithRSA sont obsolètes car ils sont de forme abrégée SHA384WithRSA et SHA512WithRSA respectivement.

Vous pouvez créer un certificat autosigné à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Pour plus d'informations sur la raison pour laquelle vous pouvez utiliser des certificats autosignés, voir Utilisation de certificats autosignés pour l'authentification mutuelle de deux gestionnaires de files d'attente.

Tous les certificats numériques ne peuvent pas être utilisés avec tous les CipherSpecs. Veillez à créer un certificat compatible avec les CipherSpecs que vous utilisez. IBM MQ prend en charge trois types différents de CipherSpec. Pour plus d'informations, voir «Interopérabilité de Elliptic Curve et de RSA CipherSpecs», à la page 50.

Pour utiliser les CipherSpecs de type 1 (ceux dont le nom commence par ECDHE_ECDSA_), vous devez utiliser la commande **runmqakm** pour créer le certificat et spécifier un paramètre d'algorithme de signature Elliptic Curve ECDSA. Par exemple, en spécifiant le paramètre **-sig_alg** EC_ecdsa_with_SHA384.

Utilisation runmqakm

Exécutez la commande suivante pour créer un certificat personnel autosigné à l'aide de la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat. Le libellé du certificat est sensible à la casse.

Le libellé d'un certificat TLS utilisé par IBM MQ est soit la valeur de l'attribut **CERTLABL** s'il est défini, soit la valeur par défaut ibmwebspheremq avec le nom du gestionnaire de files d'attente ou l'ID utilisateur IBM MQ MQI client ajouté, le tout en minuscules. Pour plus d'informations, voir «Labels de certificat numérique, compréhension des exigences», à la page 28.

-dn ***nom_distinctif***
Indique le nom distinctif X.500 entre guillemets. Au moins un attribut est requis dans le nom distinctif. Vous pouvez fournir plusieurs attributs d'unité organisationnelle et de centre de données.

Remarque : La commande **runmqakm** fait référence à l'attribut de code postal POSTALCODEet non PC. Spécifiez toujours POSTALCODE dans le paramètre **-dn** lorsque vous utilisez la commande **runmqakm** pour demander des certificats avec un code postal.

-size ***taille_clé***
Indique la taille de la clé. La valeur peut être 512, 1024ou 2048.

-x509version ***version***
Version du certificat X.509 à créer. La valeur peut être 1, 2 ou 3. La valeur par défaut est 3.

-expire ***Jours***
Délai d'expiration en jours du certificat. La valeur par défaut est 365 jours pour un certificat.

-fips
indique que la commande est exécutée en mode FIPS. Seul le composant FIPS IBM Crypto for C (ICC) est utilisé et ce composant doit être correctement initialisé en mode FIPS. En mode FIPS, le composant ICC utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

-sig_alg
Indique l'algorithme de hachage utilisé lors de la création du certificat. Cet algorithme de hachage est utilisé pour créer la signature associée au certificat. La valeur peut être 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_SHA384ou EC_ecdsa_with_SHA512.

La valeur par défaut est SHA1WithRSA.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour créer un certificat personnel autosigné à l'aide de la commande **runmqktool** :

```
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
```

où :

-keystore ***nom_fichier***
Indique le nom du référentiel de clés. Le référentiel de clés est créé s'il n'existe pas.

-storepass ***mot_de_passe***
Indique le mot de passe du référentiel de clés.

-storetype ***type_magasin***
Indique le type de référentiel de clés.

-alias ***libellé***
Indique le libellé du certificat. Le libellé de certificat est converti en minuscules.

-dname ***nom_distinctif***
Indique le nom distinctif X.500 du certificat entre guillemets.

-validité ***jours***
Indique le nombre de jours pendant lesquels le certificat est valide.

-keyalg *algorithme_clé*
Indique l'algorithme utilisé pour créer la paire de clés.

-keysize *taille_clé*
Indique la taille de la clé.

-sigalg *algorithme_signature*
Indique l'algorithme utilisé pour signer le certificat. Pour plus d'informations sur les algorithmes de signature pouvant être spécifiés, voir Algorithmes de signature.

Pour plus d'informations sur ces paramètres et sur les valeurs pouvant être spécifiées, voir genkeypair.

ALW Demande d'un certificat personnel sur AIX, Linux, and Windows

Suivez cette procédure pour créer une demande de certificat personnel.

Remarque : IBM MQ ne prend pas en charge les algorithmes SHA-3 ou SHA-5 . Vous pouvez utiliser les noms d'algorithme de signature numérique SHA384WithRSA et SHA512WithRSA car ces deux algorithmes sont membres de la famille SHA-2 .

Deprecated Les noms d'algorithme de signature numérique SHA3WithRSA et SHA5WithRSA sont obsolètes car ils sont de forme abrégée SHA384WithRSA et SHA512WithRSA respectivement.

Vous pouvez demander un certificat personnel à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Tous les certificats numériques ne peuvent pas être utilisés avec tous les CipherSpecs. Veillez à créer un certificat compatible avec les CipherSpecs que vous utilisez. IBM MQ prend en charge trois types différents de CipherSpec. Pour plus d'informations, voir «Interopérabilité de Elliptic Curve et de RSA CipherSpecs», à la page 50.

Pour utiliser les CipherSpecs de type 1 (ceux dont le nom commence par ECDHE_ECDSA_), vous devez utiliser la commande **runmqakm** pour créer le certificat et spécifier un paramètre d'algorithme de signature Elliptic Curve ECDSA. Par exemple, en spécifiant le paramètre **-sig_alg** EC_ecdsa_with_SHA384.

Si vous utilisez du matériel de cryptographie, voir «Demande d'un certificat personnel pour votre matériel PKCS #11», à la page 580.

Utilisation runmqakm

Exécutez la commande suivante pour créer une demande de certificat avec la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet d'un référentiel de clés. Le référentiel de clés doit déjà exister.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat. Le libellé du certificat est sensible à la casse.

Le libellé d'un certificat TLS utilisé par IBM MQ est soit la valeur de l'attribut **CERTLABL** s'il est défini, soit la valeur par défaut ibmwebspheremq avec le nom du gestionnaire de files d'attente ou l'ID utilisateur IBM MQ MQI client ajouté, le tout en minuscules. Pour plus d'informations, voir «Labels de certificat numérique, compréhension des exigences», à la page 28.

-dn *nom_distinctif*
Indique le nom distinctif X.500 entre guillemets. Au moins un attribut est requis dans le nom distinctif. Vous pouvez fournir plusieurs attributs d'unité organisationnelle et de centre de données.

Remarque : La commande **runmqakm** fait référence à l'attribut de code postal POSTALCODEet non PC. Spécifiez toujours POSTALCODE dans le paramètre **-dn** lorsque vous utilisez la commande **runmqakm** pour demander des certificats avec un code postal.

-size *taille_clé*
Indique la taille de la clé. La valeur peut être 512, 1024ou 2048.

-file *nom fichier*
Indique le nom de fichier de la demande de certificat.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

-sig_alg
Indique l'algorithme de hachage utilisé lors de la création de la demande de certificat. Cet algorithme de hachage est utilisé pour créer la signature associée à la demande de certificat. La valeur peut être 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_SHA384ou EC_ecdsa_with_SHA512.

La valeur par défaut est SHA1WithRSA.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm -certreq .

Utilisation runmqktool

V 9.4.0 V 9.4.0

Avant de pouvoir créer une demande de certificat à l'aide de la commande **runmqktool** , vous devez générer une paire de clés à l'aide de la commande **runmqktool -genkeypair** . Pour plus d'informations sur la commande **runmqktool -genkeypair** , voir «Création d'un certificat personnel autosigné sur AIX, Linux, and Windows», à la page 559.

Exécutez la commande suivante pour créer une demande de certificat avec la commande **runmqktool** :

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

où :

-keystore *nom_fichier*
Indique le nom du référentiel de clés.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat. Il s'agit du libellé de certificat qui a été spécifié lors de la génération de la paire de clés. Le libellé de certificat est insensible à la casse.

-file *nom fichier*
Indique le nom de fichier de la demande de certificat.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir certreq.

Etapes suivantes

Soumettez une demande de certificat à une autorité de certification. Lorsque vous recevez le certificat signé de l'autorité de certification, ajoutez le certificat signé dans le référentiel de clés. Pour plus d'informations, voir «Réception de certificats personnels dans un référentiel de clés sur AIX, Linux, and Windows», à la page 563.

ALW Renouvellement d'un certificat personnel existant sous AIX, Linux, and Windows

Un certificat personnel a une date d'expiration, après laquelle le certificat ne peut plus être utilisé. Suivez cette procédure pour renouveler un certificat personnel avant qu'il n'expire.

Vous pouvez renouveler un certificat personnel à l'aide de la commande **runmqakm** (GSKCapiCmd).

Si vous devez utiliser des tailles de clé plus grandes pour vos certificats personnels, vous ne pouvez pas renouveler un certificat existant. Vous devez remplacer votre clé existante en suivant les étapes décrites dans «Demande d'un certificat personnel sur AIX, Linux, and Windows», à la page 561 pour créer une nouvelle demande de certificat qui utilise les tailles de clé dont vous avez besoin.

Utilisation runmqakm

Exécutez la commande suivante pour créer une demande de certificat afin de renouveler un certificat personnel à l'aide de la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat. Le libellé du certificat est sensible à la casse.

-target *nom_fichier*
Indique le nom de fichier de la demande de certificat.

Etapes suivantes

Soumettez une demande de certificat à une autorité de certification. Lorsque vous recevez le certificat signé de l'autorité de certification, ajoutez le certificat signé dans le référentiel de clés. Pour plus d'informations, voir «Réception de certificats personnels dans un référentiel de clés sur AIX, Linux, and Windows», à la page 563.

ALW Réception de certificats personnels dans un référentiel de clés sur AIX, Linux, and Windows

Utilisez cette procédure pour recevoir un certificat personnel dans le référentiel de clés.

Une fois que l'autorité de certification vous a envoyé un nouveau certificat personnel, ajoutez-le au référentiel de clés à partir duquel vous avez généré la nouvelle demande de certificat. Si l'autorité de certification envoie le certificat dans le cadre d'un message électronique, copiez le certificat dans un fichier distinct.

Avant d'ajouter le certificat personnel signé par une autorité de certification au référentiel de clés, effectuez les étapes décrites dans «Ajout d'un certificat de l'autorité de certification ou de la partie publique d'un certificat digne de confiance dans un référentiel de clés sous AIX, Linux, and Windows», à la page 567 pour ajouter le certificat de l'autorité de certification au référentiel de clés.

Vous pouvez recevoir un certificat personnel dans un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Si vous utilisez du matériel de cryptographie, voir «Réception d'un certificat personnel dans votre matériel PKCS #11», à la page 581.

Utilisation runmqakm

Exécutez la commande suivante pour ajouter un certificat personnel à un référentiel de clés à l'aide de la commande **runmqakm** :

```
runmqakm -cert -receive -file filename -format format
         -db filename -pw password -fips
```

où :

-file *nom fichier*
Indique le nom de fichier qualifié complet du certificat personnel.

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister et doit être le même référentiel que celui dans lequel vous avez créé la demande de certificat.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-format *format*
Indique le format du certificat. La valeur peut-être `ascii` pour les données ASCII codées en base 64 ou `binary` pour les données Binary DER. La valeur par défaut est `ascii`.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour ajouter un certificat personnel à un référentiel de clés à l'aide de la commande **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

où :

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister et doit être le même référentiel que celui dans lequel vous avez créé la demande de certificat.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat qui a été utilisé pour créer la demande de certificat. Le libellé de certificat est converti en minuscules.

-file *nom fichier*
Indique le nom de fichier qualifié complet du certificat personnel.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir importcert.

Etapes suivantes

Si le certificat est ajouté au référentiel de clés TLS du gestionnaire de files d'attente, émettez la commande MQSC **REFRESH SECURITY TYPE(SSL)** pour actualiser le cache du référentiel de clés TLS du gestionnaire de files d'attente.

ALW Extraction d'un certificat de l'autorité de certification à partir d'un référentiel de clés sur AIX, Linux, and Windows

Suivez cette procédure pour extraire un certificat d'autorité de certification d'un référentiel de clés.

Vous pouvez extraire un certificat de l'autorité de certification d'un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Utilisation runmqakm

Exécutez la commande suivante pour extraire un certificat d'autorité de certification à l'aide de la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat de l'autorité de certification. Le libellé du certificat est sensible à la casse.

-target *nom_fichier*
Indique le nom de fichier qualifié complet du fichier cible.

-format *format*
Indique le format du certificat. La valeur peut-être `ascii` pour les données ASCII codées en base 64 ou `binary` pour les données Binary DER. La valeur par défaut est `ascii`.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour extraire un certificat d'autorité de certification à l'aide de la commande **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

où :

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat de l'autorité de certification. Le libellé de certificat est insensible à la casse.

-file *nom fichier*
Indique le nom de fichier qualifié complet du fichier cible.

-rfc
Indique que le fichier de sortie est au format ASCII Base64-encoded , tel que défini par la norme Internet RFC 1421. Si cette option n'est pas spécifiée, le fichier de sortie est au format binaire.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir exportcert.

ALW Extraction de la partie publique d'un certificat autosigné à partir d'un référentiel de clés sur AIX, Linux, and Windows

Procédez comme suit pour extraire la partie publique d'un certificat autosigné à partir d'un référentiel de clés.

Vous pouvez extraire la partie publique d'un certificat d'un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Utilisation runmqakm

Exécutez la commande suivante pour extraire la partie publique d'un certificat autosigné à l'aide de la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat de l'autorité de certification. Le libellé du certificat est sensible à la casse.

-target *nom_fichier*
Indique le nom de fichier qualifié complet du fichier cible.

-format *format*
Indique le format du certificat. La valeur peut-être `ascii` pour les données ASCII codées en base 64 ou `binary` pour les données Binary DER. La valeur par défaut est `ascii`.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour extraire la partie publique d'un certificat autosigné à l'aide de la commande **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

où :

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat de l'autorité de certification. Le libellé de certificat est insensible à la casse.

-file *nom fichier*
Indique le nom de fichier qualifié complet du fichier cible.

-rfc
Indique que le fichier de sortie est au format ASCII Base64-encoded , tel que défini par la norme Internet RFC 1421. Si cette option n'est pas spécifiée, le fichier de sortie est au format binaire.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir exportcert.

ALW Ajout d'un certificat de l'autorité de certification ou de la partie publique d'un certificat digne de confiance dans un référentiel de clés sous AIX, Linux, and Windows

Suivez cette procédure pour ajouter un certificat de l'autorité de certification ou la partie publique d'un certificat de confiance à un référentiel de clés.

Vous pouvez ajouter un certificat de l'autorité de certification ou la partie publique d'un certificat de confiance dans un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Si le certificat que vous souhaitez ajouter se trouve dans une chaîne de certificats, vous devez également ajouter tous les certificats se trouvant au-dessus de ce dernier dans la chaîne. Vous devez absolument ajouter les certificats dans l'ordre décroissant en commençant par la racine, puis par le certificat de l'autorité de certification situé immédiatement en-dessous dans la chaîne, etc.

Remarque :

- Vérifiez que le certificat est au codage ASCII (UTF-8) ou binaire (DER).
- En raison d'une restriction dans la commande IBM Java 8 **keytool** , **runmqktool** ne peut pas importer de certificats au format de codage imprimable (également appelé codage Base64) tel que défini par Internet RFC 1421 si le fichier contient des commentaires. Pour importer un certificat au format de codage imprimable, supprimez tous les commentaires du fichier. Le fichier doit commencer par une chaîne commençant par "----- BEGIN" et se terminer par une chaîne commençant par "----- END".

Utilisation runmqakm

Exécutez la commande suivante pour ajouter un certificat digne de confiance à un référentiel de clés à l'aide de la commande **runmqakm** :

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

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat. Le libellé du certificat est sensible à la casse.

-file *nom fichier*
Indique le nom du fichier contenant le certificat.

-format *ascii*
Indique le format du certificat. La valeur peut-être `ascii` pour les données ASCII codées en base 64 ou `binary` pour les données Binary DER. La valeur par défaut est `ascii`.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour ajouter un certificat digne de confiance à un référentiel de clés à l'aide de la commande **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

où :

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés est créé s'il n'existe pas.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat. Le libellé de certificat est converti en minuscules.

-file *nom fichier*
Indique le nom de fichier qualifié complet du certificat personnel.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir importcert.

ALW Exportation d'un certificat personnel à partir d'un référentiel de clés sous AIX, Linux, and Windows

Suivez cette procédure pour exporter un certificat personnel à partir d'un référentiel de clés.

L'exportation d'un certificat copie le certificat et ses clés publiques et privées associées dans un autre référentiel de clés.

Vous pouvez exporter un certificat à partir d'un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Utilisation `runmqakm`

Exécutez la commande suivante pour exporter un certificat à l'aide de la commande **runmqakm** :

```
runmqakm -cert -export -db filename -pw password -label label
         -target filename -target_pw password -target_type type
         -encryption strength -fips
```

où :

-db *nom_fichier*
Indique le nom de fichier complet du référentiel de clés qui contient le certificat.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés qui contient le certificat.

-label *Libellé*
Indique le libellé du certificat à exporter. Le libellé du certificat est sensible à la casse.

-target *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés de destination. Le référentiel de clés est créé s'il n'existe pas.

-target_pw *mot_de_passe*
Indique le mot de passe du référentiel de clés de destination.

-target_type *type*
Indique le type du référentiel de clés de destination. La valeur peut être `cms` ou `pkcs12`. La valeur par défaut est `cms`.

-encryption *puissance*
Indique le niveau de chiffrement utilisé dans la commande d'exportation de certificat. La valeur peut être `forte` ou `faible`. La valeur par défaut est `strong`.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm-cert .

Utilisation `runmqktool`

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour exporter un certificat à l'aide de la commande **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

où :

-srckeystore *nom_fichier*
Indique le nom de fichier complet du référentiel de clés qui contient le certificat.

-srcstorepass *mot_de_passe*
Indique le mot de passe du référentiel de clés qui contient le certificat.

-destkeystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés de destination. Le référentiel de clés est créé s'il n'existe pas.

-deststorepass *mot_de_passe*
Indique le mot de passe du référentiel de clés de destination.

-destkeypass *mot_de_passe*
Indique le mot de passe permettant de protéger la clé dans le référentiel de clés de destination. Si ce paramètre n'est pas spécifié, la clé est protégée par le mot de passe utilisé pour protéger la clé dans le référentiel de clés source.

-deststoretype *type*
Indique le type du référentiel de clés de destination.

-srcalias *libellé*
Indique le libellé du certificat à exporter. Le libellé de certificat est insensible à la casse.

-dessolias *libellé*
Indique le libellé du certificat dans le référentiel de clés de destination. Si ce paramètre n'est pas spécifié, le même libellé est affecté au certificat que dans le référentiel de clés source.

Le libellé de certificat est converti en minuscules.

-file *nom fichier*
Indique le nom de fichier qualifié complet du fichier cible.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir importkeystore.

ALW Importation d'un certificat personnel dans un référentiel de clés sous AIX, Linux, and Windows

Suivez cette procédure pour importer un certificat personnel dans un référentiel de clés.

L'importation d'un certificat copie le certificat et ses clés publiques et privées associées d'un référentiel de clés vers un autre référentiel de clés.

Avant d'importer un certificat personnel dans un référentiel de clés, vous devez d'abord ajouter la chaîne valide complète d'émission de certificats de l'autorité de certification au référentiel de clés. Pour plus d'informations, voir «Ajout d'un certificat de l'autorité de certification ou de la partie publique d'un certificat digne de confiance dans un référentiel de clés sous AIX, Linux, and Windows», à la page 567.

Vous pouvez importer un certificat dans un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Utilisation runmqakm

Exécutez la commande suivante pour importer un certificat à l'aide de la commande **runmqakm** :

```
runmqakm -cert -import -file filename -pw password -type type
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips
```

où :

-file *nom fichier*
Indique le nom de fichier complet du référentiel de clés qui contient le certificat.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés qui contient le certificat.

-type *type*
Indique le type du référentiel de clés qui contient le certificat. La valeur peut être cms ou pkcs12. La valeur par défaut est cms.

-target *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés de destination. Le référentiel de clés est créé s'il n'existe pas.

-target_pw *mot_de_passe*
Indique le mot de passe du référentiel de clés de destination.

-target_type *type*
Indique le type du référentiel de clés de destination. La valeur peut être cms ou pkcs12. La valeur par défaut est cms.

-label *Libellé*
Indique le libellé du certificat à importer à partir du référentiel de clés source. Le libellé du certificat est sensible à la casse.

-new_label *libellé*
Indique le libellé affecté au certificat dans le référentiel de clés cible. Si ce paramètre n'est pas spécifié, le même libellé est affecté au certificat que dans le référentiel de clés source.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm-cert .

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour importer un certificat à l'aide de la commande **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

où :

-srckeystore *nom_fichier*
Indique le nom de fichier complet du référentiel de clés qui contient le certificat.

-srcstorepass *mot_de_passe*
Indique le mot de passe du référentiel de clés qui contient le certificat.

-destkeystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés de destination. Le référentiel de clés est créé s'il n'existe pas.

-deststorepass *mot_de_passe*
Indique le mot de passe du référentiel de clés de destination.

-destkeypass *mot_de_passe*
Indique le mot de passe permettant de protéger la clé dans le référentiel de clés de destination. Si ce paramètre n'est pas spécifié, la clé est protégée par le mot de passe utilisé pour protéger la clé dans le référentiel de clés source.

Remarque : Pour un référentiel de clés PKCS #12 , la clé doit être protégée avec le même mot de passe que le référentiel de clés de destination.

-deststoretype *type*
Indique le type du référentiel de clés de destination.

-srcalias *libellé*
Indique le libellé du certificat dans le référentiel de clés source. Le libellé de certificat est insensible à la casse.

-dessolias *libellé*
Indique le libellé du certificat dans le référentiel de clés de destination. Si ce paramètre n'est pas spécifié, le même libellé est affecté au certificat que dans le référentiel de clés source.

Le libellé de certificat est converti en minuscules.

-file *nom fichier*
Indique le nom de fichier qualifié complet du fichier cible.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir importkeystore.

ALW Importer un certificat personnel depuis unMicrosoft Fichier .pfx

Suivez cette procédure pour importer un certificat depuis unMicrosoft Fichier .pfx surAIX, Linux, and Windows .

Un fichier .pfx peut contenir deux certificats relatifs à la même clé. Il s'agit d'un certificat personnel ou de site qui contient à la fois une clé publique et une clé privée. L'autre est un certificat de l'autorité de certification (signataire) qui contient uniquement une clé publique. Ces certificats ne pouvant pas coexister dans le même référentiel de clés CMS , un seul d'entre eux peut être importé.

Le libellé de certificat est associé uniquement au certificat de signataire. Le certificat personnel est identifié par un identificateur unique (UUID) généré par le système. Suivez cette procédure pour importer un certificat personnel à partir d'un fichier .pfx et définir le libellé de certificat personnel sur le libellé affecté au certificat de l'autorité de certification dans le fichier .pfx. Les certificats émis par l'autorité de certification doivent déjà être ajoutés à la base de données de clés cible.

Utilisation `runmqakm`

Exécutez la commande suivante pour importer un certificat à partir d'un fichier .pfx à l'aide de la commande **runmqakm** :

```
runmqakm -cert -import -file filename -pw password -type pkcs12
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips -pfx
```

où :

-file *nom fichier*
Indique le nom qualifié complet du fichier .pfx.

-pw *mot_de_passe*
Indique le mot de passe du fichier .pfx.

-type pkcs12
Indique le type du référentiel de clés.

-target *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés de destination. Le référentiel de clés est créé s'il n'existe pas.

-target_pw *mot_de_passe*
Indique le mot de passe du référentiel de clés de destination.

-target_type *type*
Indique le type du référentiel de clés de destination. La valeur peut être `cms` ou `pkcs12`. La valeur par défaut est `cms`.

-label *Libellé*
Indique le libellé du certificat à importer à partir du référentiel de clés source. Le libellé du certificat est sensible à la casse.

-new_label *libellé*
Indique le libellé affecté au certificat dans le référentiel de clés cible. Si ce paramètre n'est pas spécifié, le même libellé est affecté au certificat que dans le référentiel de clés source.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

-pfx
Indique que le référentiel de clés source utilise le format PFX.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm-cert .

ALW Importation d'un certificat personnel à partir d'un fichier PKCS #7

Suivez cette procédure pour importer un certificat à partir d'un fichier PKCS #7 sous AIX, Linux, and Windows.

Utilisez la commande **runmqakm** pour importer des certificats à partir d'un fichier PKCS #7 sous AIX, Linux, and Windows.

Ajout d'un certificat de l'autorité de certification ou de la partie publique d'un certificat digne de confiance

Exécutez la commande suivante pour ajouter un certificat de l'autorité de certification, ou la partie publique d'un certificat de confiance, à partir d'un fichier PKCS #7 :

```
runmqakm -cert -add -db filename -pw password -type type
         -label label -file filename
```

où :

-db *nom_fichier*
Indique le nom qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-type *type*
Indique le type du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat à ajouter. Le libellé du certificat est sensible à la casse.

Le libellé est affecté au premier certificat ajouté. Tous les autres certificats, s'ils sont présents, sont libellés avec leur nom de sujet.

-file *nom fichier*
Indique le nom qualifié complet du fichier PKCS #7 .

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Importation d'un certificat personnel

Exécutez la commande suivante pour importer un certificat personnel à partir d'un fichier PKCS #7 :

```
runmqakm -cert -import -file filename -pw password -type pkcs7
         -target filename -target_pw password -target_type type
         -label label -new_label label
```

où :

-file *nom fichier*
Indique le nom qualifié complet du fichier PKCS #7 .

-pw *mot_de_passe*
Indique le mot de passe du fichier PKCS #7 .

-type pkcs7
Indique le type du fichier PKCS #7 .

-target *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés de destination. Le référentiel de clés est créé s'il n'existe pas.

-target_pw *mot_de_passe*
Indique le mot de passe du référentiel de clés de destination.

-target_type *type*
Indique le type du référentiel de clés de destination. La valeur peut être `cms` ou `pkcs12`. La valeur par défaut est `cms`.

-label *Libellé*
Indique le libellé du certificat à importer à partir du fichier PKCS #7 . Le libellé du certificat est sensible à la casse.

-new_label *libellé*
Indique le libellé affecté au certificat dans le référentiel de clés cible. Si ce paramètre n'est pas spécifié, le même libellé est affecté au certificat que dans le référentiel de clés source.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

ALW Affichage de la liste des certificats dans un référentiel de clés sous AIX, Linux, and Windows

Utilisez cette procédure pour répertorier les certificats qui se trouvent dans un référentiel de clés.

Vous pouvez afficher des informations sur les certificats qui se trouvent dans un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool).

Utilisation runmqakm

- Exécutez la commande suivante pour répertorier les libellés des certificats dans un référentiel de clés à l'aide de la commande **runmqakm** :

```
runmqakm -cert -list -db filename -pw password
```

- Exécutez la commande suivante pour répertorier les détails d'un certificat dans un référentiel de clés à l'aide de la commande **runmqakm** :

```
runmqakm -cert -details -showOID -db filename -pw password
         -label label
```

où :

-file *nom fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat à répertorier. Le libellé du certificat est sensible à la casse.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -cert.

Utilisation runmqktool

V 9.4.0 V 9.4.0

- Exécutez la commande suivante pour répertorier les libellés des certificats dans un référentiel de clés à l'aide de la commande **runmqktool** :

```
runmqktool -list -keystore filename -storepass password
```

- Exécutez la commande suivante pour répertorier les détails d'un certificat dans un référentiel de clés à l'aide de la commande **runmqktool** :

```
runmqktool -list -keystore filename -storepass password -alias label -v
```

où :

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat à répertorier. Le libellé de certificat est insensible à la casse.

-v
Demande une sortie prolixe qui inclut les détails du certificat.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir liste.

ALW Suppression d'un certificat d'un référentiel de clés sur AIX, Linux, and Windows

Utilisez cette procédure pour supprimer un certificat personnel ou de l'autorité de certification d'un référentiel de clés.

Vous pouvez supprimer un certificat d'un référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Si vous devez gérer les certificats SSL ou TLS d'une manière compatible avec FIPS, utilisez la commande **runmqakm** .

Utilisation runmqakm

Exécutez la commande suivante pour supprimer un certificat à l'aide de la commande **runmqakm** :

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

où :

-file *nom fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé du certificat à supprimer. Le libellé du certificat est sensible à la casse.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes qui ont été validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm-cert .

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour supprimer un certificat à l'aide de la commande **runmqktool** :

```
runmqktool -delete -keystore filename -storepass password -alias label
```

où :

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-storepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-alias *libellé*
Indique le libellé du certificat à supprimer. Le libellé de certificat est insensible à la casse.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir delete.

ALW Conversion d'un référentiel de clés sous AIX, Linux, and Windows

Utilisez cette procédure pour convertir un référentiel de clés dans un autre type.

Vous pouvez convertir un mot de passe de référentiel de clés dans un autre type à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool).

Utilisation runmqakm

Exécutez la commande suivante pour convertir un référentiel de clés à l'aide de la commande **runmqakm** :

```
runmqakm -keydb -convert -db filename -pw password
         -new_db filename -new_pw password
         -old_format type -new_format type
```

où :

-file *nom fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-new_db *nom_fichier*
Indique le nom de fichier qualifié complet du nouveau référentiel de clés.

-new_pw *mot_de_passe*
Indique le mot de passe du nouveau référentiel de clés.

-old_format *type*
Indique le type en cours du référentiel de clés. Les valeurs suivantes peuvent être spécifiées :

- `pkcs12`
- `cms`

-new_format *type*
Indique le nouveau type du référentiel de clés. Les valeurs suivantes peuvent être spécifiées :

- `pkcs12`
- `cms`

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm -keydb .

Utilisation runmqktool

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour convertir un référentiel de clés à l'aide de la commande **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -destkeystore filename
           -srcstoretype type -deststoretype type
           -srcstorepass password -deststorepass password
```

où :

-all
Indique que le mot de passe est également modifié pour toutes les entrées qui sont protégées par le même mot de passe que le référentiel de clés.

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-destkeystore *nom_fichier*
Indique le nom de fichier qualifié complet du nouveau référentiel de clés.

-srcstoretype *type*
Indique le type de référentiel de clés.

-deststoretype *type*
Indique le nouveau type de référentiel de clés.

-srcstorepass *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-deststorepass *mot_de_passe*
Indique le mot de passe du nouveau référentiel de clés.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir importkeystore.

ALW Modification du mot de passe du référentiel de clés sous AIX, Linux, and Windows

Utilisez cette procédure pour modifier le mot de passe du référentiel de clés.

Vous pouvez modifier le mot de passe du référentiel de clés à l'aide des commandes **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool).

Remarque :

- V 9.4.0 V 9.4.0 La commande **runmqktool** permet de modifier le mot de passe du référentiel de clés indépendamment des mots de passe qui protègent les clés privées et secrètes individuelles. Pour les référentiels de clés PKCS #12 , le mot de passe du référentiel de clés et les mots de passe qui protègent toutes les clés du référentiel de clés doivent être identiques. Si la commande **runmqktool** est utilisée pour modifier le mot de passe du référentiel de clés, vérifiez que le paramètre **-all** est spécifié de sorte que les mots de passe de clé soient également modifiés.
- Si le mot de passe du référentiel de clés n'est pas stocké dans un fichier de dissimulation, vous devez également modifier le mot de passe stocké dans la configuration du gestionnaire de files d'attente ou dans toute application IBM MQ client qui accède au référentiel de clés. Pour plus d'informations, reportez-vous aux sections «Indication du mot de passe du référentiel de clés pour un gestionnaire de files d'attente sous AIX, Linux, and Windows», à la page 309 et «Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows», à la page 311.

Utilisation runmqakm

Exécutez la commande suivante pour modifier le mot de passe du référentiel de clés à l'aide de la commande **runmqakm** :

```
runmqakm -keydb -changepw -db filename -pw password -new_pw password -stash
```

où :

-file *nom fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe en cours pour le référentiel de clés.

-new_pw *mot_de_passe*
Indique le nouveau mot de passe du référentiel de clés.

-stash
Facultatif. Spécifiez cette option pour stocker le nouveau mot de passe du référentiel de clés dans un fichier de dissimulation. Vous n'avez pas besoin de stocker le mot de passe dans un fichier de dissimulation si vous chiffrez le mot de passe à l'aide du système de protection par mot de passe IBM MQ .

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir runmqakm -keydb.

Utilisation `runmqktool`

V 9.4.0 V 9.4.0

Exécutez la commande suivante pour modifier le mot de passe du référentiel de clés à l'aide de la commande **runmqktool** :

```
runmqktool -storepasswd -all -keystore filename -storepass password
           -new password
```

où :

-all
Indique que le mot de passe est également modifié pour toutes les entrées qui sont protégées par le même mot de passe que le référentiel de clés.

-keystore *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-storepass *mot_de_passe*
Indique le mot de passe en cours pour le référentiel de clés.

-new *mot_de_passe*
Indique le nouveau mot de passe du référentiel de clés.

Pour plus d'informations sur ces paramètres et les valeurs pouvant être spécifiées, voir storepasswd.

ALW Gestion des clés secrètes sur AIX, Linux, and Windows

Suivez cette procédure pour gérer les clés secrètes dans un référentiel de clés.

Vous pouvez gérer les clés secrètes à l'aide de la commande **runmqakm** (GSKCapiCmd). Les clés secrètes générées à l'aide de la commande **runmqktool** (keytool) ne peuvent pas être utilisées avec IBM MQ.

Création d'une clé secrète

Exécutez la commande suivante pour créer une clé secrète aléatoire à l'aide de la commande **runmqakm** :

```
runmqakm -secretkey -create -db filename -pw password
         -label label -size key_size
```

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé associé à la clé.

-size *taille_clé*
Indique la taille de la clé en octets.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm -clé secrète .

Extraction d'une clé secrète

Exécutez la commande suivante pour extraire une clé secrète à l'aide de la commande **runmqakm** :

```
runmqakm -secretkey -extract -db filename -pw password
         -label label -target filename -format format
```

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé de la clé à extraire.

-target *nom_fichier*
Indique le nom de fichier qualifié complet du fichier cible.

-format *format*
Indique le format de la clé dans le fichier de destination. La valeur peut être `ascii` pour Base64-encoded ASCII ou `binary` pour une copie binaire de la clé. La valeur par défaut est `ascii`.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm -clé secrète .

Ajout d'une clé secrète

Exécutez la commande suivante pour extraire une clé secrète à l'aide de la commande **runmqakm** :

```
runmqakm -secretkey -add -db filename -pw password
         -label label -file filename -format format
```

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés. Le référentiel de clés doit déjà exister.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-label *Libellé*
Indique le libellé associé à la clé.

-file *nom fichier*
Indique le nom du fichier contenant la clé.

-format *format*
Indique le format de la clé. La valeur peut être `ascii` pour Base64-encoded ASCII ou `binary` pour les données binaires. La valeur par défaut est `ascii`.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm -clé secrète .

ALW Gestion des certificats sur le matériel PKCS #11

Vous pouvez gérer des certificats numériques sur du matériel de cryptographie qui prend en charge l'interface PKCS #11 .

Vous devez créer un référentiel de clés pour préparer l'environnement IBM MQ , même si vous n'avez pas l'intention d'y stocker des certificats, mais que vous stockez tous vos certificats sur votre matériel de cryptographie. Un référentiel de clés est nécessaire pour que le gestionnaire de files d'attente y fasse référence dans son attribut **SSLKEYR** ou pour que l'application client y fasse référence dans la variable d'environnement MQSSLKEYR. Ce référentiel de clés est également requis si vous créez une demande de certificat.

Créez le référentiel de clés à l'aide de la commande **runmqakm** (GSKCapiCmd).

Exécutez la commande suivante pour créer un référentiel de clés avec la commande **runmqakm** :

```
runmqakm -keydb -create -db filename -pw password -type type -stash
```

où :

-db *nom_fichier*
Indique le nom de fichier qualifié complet du référentiel de clés.

-pw *mot_de_passe*
Indique le mot de passe du référentiel de clés.

-type *type*
Indique le type de base de données. La valeur doit être `cms` ou `pkcs12` pour un référentiel de clés utilisé par IBM MQ.

-stash
Facultatif. S'il est spécifié, le mot de passe du référentiel de clés chiffré est sauvegardé dans un fichier.

ALW *Demande d'un certificat personnel pour votre matériel PKCS #11*

Utilisez cette procédure pour demander un certificat personnel pour un gestionnaire de files d'attente ou un IBM MQ MQI client avec votre matériel de cryptographie.

Remarque : IBM MQ ne prend pas en charge les algorithmes SHA-3 ou SHA-5 . Vous pouvez utiliser les noms d'algorithme de signature numérique SHA384WithRSA et SHA512WithRSA car ces deux algorithmes sont membres de la famille SHA-2 .

Deprecated Les noms d'algorithme de signature numérique SHA3WithRSA et SHA5WithRSA sont obsolètes car ils sont de forme abrégée SHA384WithRSA et SHA512WithRSA respectivement.

Avant de créer une demande de certificat dans votre matériel de cryptographie, effectuez les étapes décrites dans «Gestion des certificats sur le matériel PKCS #11», à la page 579 pour créer un référentiel de clés.

Exécutez la commande suivante pour créer une demande de certificat avec la commande **runmqakm** (GSKCapiCmd):

```
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
```

où :

-crypto *nom_module*
Indique le nom qualifié complet de la bibliothèque PKCS #11 fournie avec le matériel de cryptographie.

-tokenlabel *jeton_matériel*
Indique le libellé du jeton de l'unité de chiffrement PKCS #11 .

-pw *mot_de_passe*
Indique le mot de passe permettant d'accéder au matériel de cryptographie.

-label *Libellé*
Indique le libellé du certificat.

Le libellé d'un certificat TLS utilisé par IBM MQ est soit la valeur de l'attribut **CERTLABL** s'il est défini, soit la valeur par défaut `ibmwebspheremq` avec le nom du gestionnaire de files d'attente ou l'ID utilisateur IBM MQ MQI client ajouté, le tout en minuscules. Pour plus d'informations, voir «Labels de certificat numérique, compréhension des exigences», à la page 28.

-dn *nom_distinctif*
Indique le nom distinctif X.500 entre guillemets. Au moins un attribut est requis dans le nom distinctif. Vous pouvez fournir plusieurs attributs d'unité organisationnelle et de centre de données.

Remarque : La commande **runmqakm** fait référence à l'attribut de code postal POSTALCODEet non PC. Spécifiez toujours POSTALCODE dans le paramètre **-dn** lorsque vous utilisez la commande **runmqakm** pour demander des certificats avec un code postal.

-size *taille_clé*
Indique la taille de la clé. La valeur peut être 512, 1024ou 2048.

-file *nom fichier*
Indique le nom de fichier de la demande de certificat.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

-sig_alg
Indique l'algorithme de hachage utilisé lors de la création de la demande de certificat. Cet algorithme de hachage est utilisé pour créer la signature associée à la demande de certificat. La valeur peut être 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_SHA384ou EC_ecdsa_with_SHA512.

La valeur par défaut est SHA1WithRSA.

Pour plus d'informations sur ces paramètres et les valeurs qui peuvent être spécifiées, voir runmqakm -certreq .

Etapes suivantes

Soumettez une demande de certificat à une autorité de certification. Lorsque vous recevez le certificat signé de l'autorité de certification, ajoutez le certificat signé dans le référentiel de clés. Pour plus d'informations, voir «Réception d'un certificat personnel dans votre matériel PKCS #11», à la page 581.

ALW *Réception d'un certificat personnel dans votre matériel PKCS #11*

Utilisez cette procédure pour recevoir un certificat personnel pour un gestionnaire de files d'attente ou un IBM MQ MQI client sur votre matériel de cryptographie.

Ajoutez le certificat de l'autorité de certification qui a signé le certificat personnel au matériel de cryptographie ou au référentiel de clés secondaire. Effectuez cette opération avant de recevoir le certificat signé dans le matériel de cryptographie. Pour ajouter un certificat d'autorité de certification à un fichier de référentiel de clés, suivez la procédure décrite dans «Ajout d'un certificat de l'autorité de certification ou de la partie publique d'un certificat digne de confiance dans un référentiel de clés sous AIX, Linux, and Windows», à la page 567.

Exécutez la commande suivante pour ajouter un certificat personnel à un référentiel de clés à l'aide de la commande **runmqakm** (GSKCapiCmd):

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

où :

-file *nom fichier*
Indique le nom de fichier complet du fichier contenant le certificat personnel.

-crypto *nom_module*
Indique le nom qualifié complet de la bibliothèque PKCS #11 fournie avec le matériel de cryptographie.

-tokenlabel *jeton_matériel*
Indique le libellé du jeton de l'unité de chiffrement PKCS #11 .

-pw *mot_de_passe_matériel*
Indique le mot de passe permettant d'accéder au matériel de cryptographie.

-format *format_certificat*
Indique le format du certificat. La valeur peut-être `ascii` pour les données ASCII codées en base 64 ou `binary` pour les données Binary DER. La valeur par défaut est ASCII.

-fips
indique que la commande est exécutée en mode FIPS. En mode FIPS, le composant IBM Crypto for C (ICC) utilise des algorithmes validés par la norme FIPS 140-2. Si le composant ICC ne s'initialise pas en mode FIPS, la commande **runmqakm** échoue.

-secondaryDB *nom_fichier*
Indique le nom de fichier qualifié complet du fichier de référentiel de clés utilisé pour stocker le certificat de l'autorité de certification.

-secondaryDBpw *mot_de_passe*
Indique le mot de passe du fichier de référentiel de clés utilisé pour stocker le certificat de l'autorité de certification.

Protection des mots de passe dans les fichiers de configuration du composant IBM MQ

Pour utiliser certaines fonctions d' IBM MQ, vous devrez peut-être fournir des mots de passe utilisés par la fonction. Les mots de passe fournis à IBM MQ peuvent être protégés à l'aide d'un système de protection par mot de passe.

La liste suivante décrit la terminologie utilisée pour chaque composant qui traite les mots de passe chiffrés:

Clé initiale
Clé de chiffrement utilisée pour protéger le mot de passe.

Clé initiale par défaut
Clé de chiffrement par défaut utilisée si vous ne fournissez pas de clé initiale lorsque le mot de passe est chiffré.

Chaîne de texte en clair
Chaîne chiffrée, généralement un mot de passe.

Chaîne de mot de passe chiffré
Chaîne contenant le mot de passe chiffré dans un format compris par IBM MQ .

Spécification de la clé initiale

Pour chaque composant, vous pouvez choisir de spécifier une clé initiale qui est utilisée pour chiffrer les mots de passe.

- Si vous ne spécifiez pas de clé initiale, la clé initiale par défaut du composant est utilisée. La clé initiale par défaut est la même pour toutes les installations IBM MQ . Cela signifie qu'un mot de passe chiffré avec la clé initiale par défaut n'est pas protégé de manière sécurisée car il peut être possible pour une autre installation de déchiffrer le mot de passe.
- Si vous fournissez votre propre clé initiale unique, seuls les utilisateurs ayant accès à la clé initiale que vous fournissez peuvent déchiffrer le mot de passe.

Avertissement : Pour fournir le niveau de sécurité le plus élevé pour les mots de passe stockés, indiquez une clé initiale unique pour chaque composant IBM MQ .

Si vous choisissez d'utiliser votre propre clé initiale, spécifiez une clé initiale unique pour chaque composant répertorié. La clé initiale est utilisée pour protéger les mots de passe stockés dans la configuration de ce composant. La même clé initiale doit également être mise à la disposition du composant pour que le mot de passe soit déchiffré.

La plupart des composants nécessitent que la clé initiale soit fournie dans un fichier. La clé initiale contenue dans le fichier de clés initiales doit répondre aux exigences suivantes:

- Il doit comporter au moins un caractère.
- Il doit s'agir d'une seule ligne de texte.

La longueur maximale de la clé initiale est illimitée et tous les caractères peuvent être spécifiés. Pour une sécurité adéquate, spécifiez une clé initiale d'au moins 16 caractères. Par exemple, votre fichier de clés initial peut contenir la chaîne suivante:

```
Th1sIs@n3Ncypt|onK$y
```

L'accès au fichier de clés initial doit être limité uniquement aux utilisateurs qui ont besoin d'accéder à la clé initiale à l'aide des droits d'accès au fichier du système d'exploitation.

Pour plus d'informations sur les avantages et les limitations de la protection par mot de passe, voir «Limites de la protection via le chiffrement de mot de passe», à la page 589.

Protection des mots de passe dans chaque composant IBM MQ

Plusieurs composants IBM MQ peuvent protéger les mots de passe stockés. Selon le composant, ces mots de passe peuvent être fournis à l'aide de l'un des mécanismes suivants:

- Fourni directement au gestionnaire de files d'attente IBM MQ ou à IBM MQ client.
- Spécifié dans une variable d'environnement.
- Stocké dans un fichier de configuration.

Chaque composant fournit une méthode de chiffrement des mots de passe. Dans la plupart des composants, les mots de passe doivent être chiffrés avant d'être fournis à IBM MQ ou stockés dans la configuration.

Important : Un mot de passe chiffré généré pour être utilisé avec un composant ne peut pas être copié dans le fichier de configuration d'un autre composant. Un mot de passe chiffré pour être utilisé par un composant particulier doit être protégé à l'aide de l'utilitaire fourni par le même composant.

Les détails de la protection des mots de passe pour chaque composant IBM MQ prenant en charge la protection par mot de passe sont répertoriés dans les sections suivantes:

- Advanced Message Security
- «Managed File Transfer», à la page 584
- «IBM MQ Internet Pass-Thru», à la page 585
- «IBM MQ clients qui utilisent du matériel de cryptographie», à la page 586
- «Gestionnaire de files d'attente IBM MQ», à la page 587
- «Applications client IBM MQ C», à la page 587
- V 9.4.0 «Configurations Native HA», à la page 588
- V 9.4.0 «Gestionnaire de files d'attente IBM MQ (sectionAuthToken du fichier qm.ini)», à la page 589

Advanced Message Security

Les clients Advanced Message Security (AMS) Java ont besoin d'accéder à un magasin de clés contenant les clés privées utilisées pour protéger les messages.

Les clients ou les gestionnaires de files d'attente Advanced Message Security (AMS) MQI configurés pour effectuer une interception MCA peuvent nécessiter un accès au matériel de cryptographie PKCS#11 ou aux fichiers PEM contenant les clés privées utilisées pour protéger les messages.

Pour accéder à ces référentiels de clés, un mot de passe doit être fourni dans le fichier de configuration AMS appelé `keystore.conf`. Utilisez la commande **runamscred** pour protéger les informations sensibles contenues dans le fichier `keystore.conf` . Exemple :

```
runamscred -f <keystore configuration file>
```

La commande **runamscred** protège les paramètres sensibles dans le fichier spécifié à l'aide du paramètre **-f** .

Deux commandes **runamscred** sont disponibles dans une installation IBM MQ :

- Une commande MQI **runamscred** située dans `<IBM MQ installation root>/bin`
- Une commande Java **runamscred** située dans `<IBM MQ installation root>/java/bin`

Avertissement : Pour garantir la compatibilité,

1. Utilisez la commande Java **runamscred** pour protéger les fichiers de configuration utilisés avec les clients Java AMS et la commande MQI **runamscred** pour protéger les fichiers de configuration pour les IBM MQ MQI clients qui utilisent AMS.
2. Vérifiez que toutes les informations sensibles nécessaires sont protégées après avoir exécuté la commande **runamscred** .
3. Indiquez le fichier contenant le mot de passe protégé comme normal pour les applications compatibles avec AMS .

Par défaut, la commande **runamscred** chiffre le mot de passe dans le fichier de configuration avec la clé initiale par défaut. Pour chiffrer les mots de passe avec une clé initiale spécifique, utilisez l'un des mécanismes suivants pour spécifier le nom du fichier contenant la clé initiale, par ordre de priorité:

1. Paramètre **-sf** de la commande **runamscred** .
2. Variable d'environnement **MQS_AMSCRED_KEYFILE** .
3. Le paramètre **amscred.keyfile** dans le fichier de configuration `keystore.conf` .

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Si vous spécifiez un fichier de clés initial lorsque vous exécutez la commande **runamscred** pour chiffrer les mots de passe dans la configuration AMS , vous devez également spécifier le même fichier de clés initial lors de l'exécution des applications AMS . Les mécanismes suivants peuvent être utilisés pour spécifier le nom du fichier de clés initial, par ordre de priorité:

1. Variable d'environnement **MQS_AMSCRED_KEYFILE** .
2. Le paramètre **amscred.keyfile** dans le fichier de configuration `keystore.conf` .

Par défaut, la commande **runamscred** protège les données d'identification avec un système de protection qui n'est pas compatible avec les versions d' AMS antérieures à IBM MQ 9.2. Pour protéger les fichiers de configuration avec le système de protection des données d'identification compatible avec les versions antérieures à IBM MQ 9.2, spécifiez le paramètre **-sp 0** lorsque la commande **runamscred** est exécutée.

Managed File Transfer

Managed File Transfer (MFT) stocke les données d'identification requises pour accéder aux gestionnaires de files d'attente et aux autres ressources dans les fichiers de propriétés XML suivants:

MQMFTCredentials.xml
Ce fichier contient les données d'identification suivantes:

- Données d'identification utilisées pour la connexion aux gestionnaires de files d'attente d'agent, de coordination et de commandes.
- Mots de passe utilisés pour accéder aux magasins de clés utilisés pour les communications sécurisées.

ProtocolBridgeCredentials.xml
Ce fichier contient les données d'identification utilisées pour se connecter aux serveurs de protocole, tels que FTP, SFTP et FTPS.

ConnectDirectCredentials.xml
Ce fichier contient les données d'identification utilisées par un agent Connect:Direct pour se connecter à un noeud Connect:Direct .

Pour protéger les informations sensibles stockées dans ces fichiers, utilisez la commande fteObfuscate . Indiquez le nom du fichier à protéger à l'aide de l'indicateur **-f** . Exemple :

```
fteObfuscate -f <File to protect>
```

Par défaut, la commande **fteObfuscate** protège les données d'identification avec la clé initiale par défaut. Pour protéger les données d'identification avec une clé initiale spécifique, utilisez le paramètre **-sf** pour spécifier le chemin d'accès au fichier qui contient la clé initiale. Exemple :

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Avertissement :

1. Vérifiez que toutes les informations sensibles sont protégées après avoir exécuté **fteObfuscate**.
2. Fournissez le fichier protégé normalement à MFT.

Si vous spécifiez un fichier de clés initial lorsque vous exécutez la commande **fteObfuscate** pour protéger les données d'identification dans la configuration MFT , vous devez également spécifier le même fichier de clés initial lorsque MFT démarre. Les mécanismes suivants peuvent être utilisés pour spécifier le nom du fichier de clés initial, par ordre de priorité:

1. Propriété système **com.ibm.wmqfte.cred.keyfile** Java .

   Remarque : Avant IBM MQ 9.3.1 et IBM MQ 9.3.0 Fix Pack 10, le nom de cette propriété système Java était mal orthographié en tant que **com.ibm.wqmfte.cred.keyfile**. Depuis IBM MQ 9.3.1 et IBM MQ 9.3.0 Fix Pack 10, Managed File Transfer utilise les deux versions de la propriété système Java pour maintenir la compatibilité avec les versions antérieures. Si les deux propriétés système Java sont définies, la valeur de la propriété correctement orthographiée **com.ibm.wmqfte.cred.keyfile** est utilisée.
2. Propriétés dans les fichiers de propriétés de l'agent, du consignateur, des commandes et de la coordination.
3. La propriété **commonCredentialsKeyFile** dans le fichier `installation.properties` .

Pour plus d'informations, voir «Chiffrement des données d'identification stockées dans MFT», à la page 592.

Par défaut, la commande **fteObfuscate** protège les données d'identification avec un système de protection qui n'est pas compatible avec les versions d' MFT antérieures à IBM MQ 9.2. Pour protéger les fichiers de configuration avec le système de protection des données d'identification compatible avec les versions antérieures à IBM MQ 9.2, spécifiez le paramètre **-sp 0** lorsque la commande **fteObfuscate** est exécutée.

IBM MQ Internet Pass-Thru

Le fichier de configuration IBM MQ Internet Pass-Thru (MQIPT) peut contenir des mots de passe utilisés pour accéder à diverses ressources.

Protégez les mots de passe dans le fichier de configuration MQIPT à l'aide de la commande **mqiptPW** .

La commande **mqiptPW** vous invite à entrer le mot de passe à chiffrer et renvoie le mot de passe chiffré. Copiez le mot de passe chiffré dans le fichier de configuration MQIPT .

Par défaut, la commande **mqiptPW** chiffre un mot de passe avec la clé initiale par défaut. Pour chiffrer le mot de passe avec une clé initiale spécifique, utilisez le paramètre **-sf** pour spécifier le chemin d'accès au fichier qui contient la clé initiale.

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Pour plus d'informations, voir Spécification de la clé de chiffrement de mot de passe.

Si vous spécifiez un fichier de clés initial lorsque vous chiffrez le mot de passe du référentiel de clés, vous devez également spécifier le même fichier de clés initial lorsque MQIPT démarre. Les mécanismes suivants peuvent être utilisés pour spécifier le nom du fichier de clés initial, par ordre de priorité:

1. Paramètre **-sf** de la commande utilisée pour démarrer MQIPT.
2. Variable d'environnement **MQS_MQIPTCRED_KEYFILE** .
3. Propriété **com.ibm.mq.ipt.cred.keyfile** Java .
4. Un fichier nommé `mqipt_cred.key` dans le répertoire de base MQIPT . Le répertoire de base MQIPT est le répertoire qui contient le fichier de configuration MQIPT .

Par défaut, la commande **mqiptPW** protège les données d'identification avec un système de protection qui n'est pas compatible avec les versions d' MQIPT antérieures à IBM MQ 9.2. Pour protéger les mots de passe avec le système de protection des données d'identification compatible avec les versions antérieures à IBM MQ 9.2, utilisez la syntaxe de commande **mqiptPW** qui est prise en charge dans les versions antérieures à IBM MQ 9.2.

IBM MQ clients qui utilisent du matériel de cryptographie

Vous pouvez configurer les clients IBM MQ pour qu'ils utilisent du matériel de cryptographie PKCS #11 afin de stocker les clés privées et les certificats utilisés dans les communications TLS. Pour accéder aux unités PKCS #11 , vous devez fournir un mot de passe dans la chaîne de configuration fournie à IBM MQ client.

Important : Les mots de passe fournis à l'aide de la zone **CryptoHardware** dans la structure MQSCO ou l'attribut **SSLCRYP** du gestionnaire de files d'attente ne peuvent pas être protégés à l'aide de ce mécanisme.

Vous pouvez protéger ce mot de passe à l'aide de la commande **runp11cred** , qui se trouve dans le dossier `bin` du répertoire d'installation de IBM MQ .

La commande **runp11cred** vous invite à entrer le mot de passe à chiffrer et renvoie le mot de passe chiffré. Le mot de passe chiffré doit être copié dans la chaîne de configuration du matériel de chiffrement.

Par exemple, si la chaîne de configuration du matériel de cryptographie est la suivante:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

Lorsque la commande **runp11cred** vous invite à entrer le mot de passe, entrez `Passw0rd`. La commande renvoie une chaîne similaire à la suivante:

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

Remplacez le mot de passe dans la chaîne de configuration du matériel de cryptographie par la chaîne renvoyée par la commande **runp11cred** , pour fournir la chaîne suivante qui contient le mot de passe chiffré:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

Lorsque l'application IBM MQ client s'exécute, spécifiez la chaîne de configuration matérielle cryptographique qui contient le mot de passe chiffré dans l'une des méthodes suivantes:

- L'attribut **SSLCryptoHardware** dans la strophe SSL du fichier de configuration du client.
- Variable d'environnement **MQSSLCRYP** .

Par défaut, la commande **runp11cred** chiffre un mot de passe avec une clé initiale par défaut. Pour protéger un mot de passe avec votre propre clé initiale, indiquez le nom du fichier qui contient la clé initiale à l'aide de l'un des mécanismes suivants, par ordre de priorité:

1. Paramètre **-sf** de la commande **runp11cred** .
2. Variable d'environnement **MQS_SSLCRYP_KEYFILE** .

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Si vous spécifiez un fichier de clés initial lorsque vous chiffrez le mot de passe du référentiel de clés, vous devez également indiquer le nom du fichier qui contient la clé initiale lors de l'exécution de IBM MQ client . Indiquez le nom du fichier de clés initial à l'aide de l'un des mécanismes suivants, par ordre de priorité:

1. Variable d'environnement **MQS_SSLCRYP_KEYFILE** .
2. L'attribut **SSLCryptoHardwareKeyFile** dans la section **SSL** du fichier de configuration du client.

Gestionnaire de files d'attente IBM MQ

Le gestionnaire de files d'attente IBM MQ stocke les mots de passe en interne dans plusieurs attributs. Par exemple, l'attribut **KEYRPWD** du gestionnaire de files d'attente. Le gestionnaire de files d'attente chiffre automatiquement le mot de passe avant qu'il ne soit stocké dans des fichiers sur le disque.

Le mot de passe du référentiel de clés TLS du gestionnaire de files d'attente peut être protégé à l'aide du système de protection par mot de passe IBM MQ ou d'un fichier de dissimulation de référentiel de clés. Pour plus d'informations sur ces deux méthodes, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Lorsque le gestionnaire de files d'attente chiffre un mot de passe, la clé initiale par défaut est utilisée sauf si vous spécifiez votre propre clé initiale. Pour utiliser votre propre clé initiale, définissez l'attribut **INITKEY** du gestionnaire de files d'attente sur une clé unique et fiable avant de définir des attributs de gestionnaire de files d'attente chiffrés.

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Avertissement : Si la clé initiale est modifiée après que vous avez défini la valeur des attributs qui sont chiffrés, les attributs chiffrés ne sont pas rechiffrés avec la nouvelle clé initiale. Par conséquent, si vous modifiez la clé initiale sans fournir à nouveau la phrase passe du référentiel de clés, IBM MQ ne peut pas déchiffrer la phrase passe du référentiel de clés et ne peut pas accéder au référentiel de clés.

Pour plus d'informations, voir INITKEY.

Applications client IBM MQ C

Les bibliothèques client IBM MQ C requièrent des mots de passe pour accéder à certaines ressources sécurisées. Par exemple, un référentiel de clés TLS pour les applications qui utilisent TLS pour se connecter au gestionnaire de files d'attente.

Le mot de passe du référentiel de clés peut être protégé à l'aide du système de protection par mot de passe IBM MQ ou d'un fichier de dissimulation de référentiel de clés. Pour plus d'informations sur ces deux méthodes, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Pour protéger les mots de passe avec le système de protection par mot de passe IBM MQ , utilisez la commande **runmqicred** . La commande se trouve dans le répertoire *MQ_INSTALLATION_PATH*/bin .

La commande **runmqicred** vous invite à entrer le mot de passe à chiffrer et renvoie le mot de passe chiffré. Le mot de passe chiffré peut être utilisé par l'application client à la place d'un mot de passe en texte en clair.

Par exemple, si vous choisissez de fournir un mot de passe de référentiel de clés TLS à l'aide de la variable d'environnement *MQKEYRPWD* et que votre mot de passe de magasin de clés TLS est Passw0rd. Lorsque vous exécutez **runmqicred**, entrez Passw0rd lorsque vous y êtes invité. La commande renvoie une chaîne similaire à la suivante:

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

Définissez cette chaîne comme valeur de la variable d'environnement *MQKEYRPWD* :

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

Par défaut, la commande **runmqicred** chiffre un mot de passe avec la clé initiale par défaut. Pour protéger un mot de passe avec votre propre clé initiale, utilisez l'un des mécanismes suivants pour spécifier le nom du fichier qui contient la clé, par ordre de priorité:

1. Paramètre **-sf** de la commande **runmqicred** .
2. Variable d'environnement *MQS_MQI_KEYFILE* .

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Si vous spécifiez un fichier de clés initial lorsque vous chiffrez le mot de passe, vous devez également mettre la clé initiale à la disposition de l'application client lors de son exécution.

Pour plus d'informations, voir «Indication du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows», à la page 311.

Configurations Native HA

V 9.4.0

Le trafic de réplication de journal Native HA entre les instances peut être chiffré à l'aide de TLS. Les certificats utilisés pour sécuriser le trafic de réplication de journal sont stockés dans un référentiel de clés spécifié dans la section **NativeHALocalInstance** du fichier qm.ini .

Le mot de passe du référentiel de clés peut être protégé à l'aide du système de protection par mot de passe IBM MQ ou d'un fichier de dissimulation de référentiel de clés. Pour plus d'informations sur ces deux méthodes, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Pour protéger le mot de passe du référentiel de clés Native HA avec le système de protection par mot de passe IBM MQ , utilisez la commande **runmqicred** .

La commande **runmqicred** vous invite à entrer le mot de passe à chiffrer et renvoie le mot de passe chiffré. Le mot de passe chiffré doit être utilisé à la place d'un mot de passe en texte en clair. Définissez la valeur de l'attribut **KeyRepositoryPassword** dans la section **NativeHALocalInstance** du fichier qm.ini sur le mot de passe chiffré renvoyé par la commande.

Par défaut, la commande **runmqicred** chiffre un mot de passe avec la clé initiale par défaut. Pour protéger un mot de passe avec votre propre clé initiale, utilisez l'un des mécanismes suivants pour spécifier le nom du fichier qui contient la clé, par ordre de priorité:

1. Paramètre **-sf** de la commande **runmqicred** .
2. Variable d'environnement *MQS_MQI_KEYFILE* .

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Si vous spécifiez un fichier de clés initial lorsque vous chiffrez le mot de passe du référentiel de clés, vous devez également spécifier le même fichier de clés initial à l'aide de l'attribut **InitialKeyFile** dans la section **NativeHALocalInstance** du fichier qm.ini .

Pour plus d'informations, voir la strophe d'instanceNativeHALocaldu fichier qm.ini.

Gestionnaire de files d'attente IBM MQ (sectionAuthToken du fichier qm.ini)

Depuis la IBM MQ 9.3.4, les IBM MQ MQI clients qui se connectent aux gestionnaires de files d'attente IBM MQ qui s'exécutent sur des systèmes AIX ou Linux peuvent utiliser des jetons d'authentification pour s'authentifier auprès du gestionnaire de files d'attente. Le gestionnaire de files d'attente doit être configuré pour accepter les jetons d'authentification et pouvoir accéder au certificat de clé publique de l'émetteur de jeton ou à la clé secrète utilisée pour signer le jeton. Le référentiel de clés qui contient les certificats de clé publique ou les clés secrètes de l'émetteur de confiance est sécurisé avec un mot de passe.

Le mot de passe du référentiel de clés peut être protégé à l'aide du système de protection par mot de passe IBM MQ ou d'un fichier de dissimulation de référentiel de clés. Pour plus d'informations sur ces deux méthodes, voir «Chiffrement des mots de passe du référentiel de clés sous AIX, Linux, and Windows», à la page 305.

Pour protéger le mot de passe du référentiel de clés de jeton d'authentification avec le système de protection par mot de passe IBM MQ , utilisez la commande **runqmcred** pour chiffrer le mot de passe.

La commande **runqmcred** vous invite à entrer le mot de passe à chiffrer et renvoie le mot de passe chiffré. Le mot de passe chiffré doit être utilisé à la place d'un mot de passe en texte en clair. Copiez le mot de passe chiffré dans un fichier et incluez le chemin d'accès au fichier dans l'attribut **KeyStorePwdFile** de la section **AuthToken** du fichier qm.ini .

Par défaut, la commande **runqmcred** chiffre un mot de passe avec la clé initiale par défaut. Pour chiffrer le mot de passe avec une clé initiale spécifique, utilisez le paramètre **-sf** pour spécifier le chemin d'accès au fichier qui contient la clé initiale.

ATTENTION : La clé initiale par défaut est la même pour toutes les installations IBM MQ . Pour protéger les mots de passe en toute sécurité, fournissez une clé initiale unique à votre installation lorsque vous chiffrez les mots de passe.

Important : Si vous fournissez une clé initiale lorsque vous chiffrez le mot de passe, la même clé initiale doit être spécifiée dans l'attribut **INITKEY** du gestionnaire de files d'attente afin que le gestionnaire de files d'attente puisse déchiffrer le mot de passe. Si l'attribut **INITKEY** du gestionnaire de files d'attente est déjà défini, utilisez la même clé initiale lorsque vous exécutez la commande **runqmcred** . Pour plus d'informations sur l'attribut **INITKEY** du gestionnaire de files d'attente, voir INITKEY.

Par exemple, pour chiffrer les mots de passe du magasin de clés de jeton d'authentification avec la clé initiale dans le fichier /home/initial.key, exécutez la commande suivante:

```
runqmcred -sf /home/initial.key
```

Pour plus d'informations, voir «Configuration d'un gestionnaire de files d'attente pour l'acceptation de jetons d'authentification à l'aide d'un magasin de clés local», à la page 343.

Limites de la protection via le chiffrement de mot de passe

IBM MQ prend en charge le chiffrement AES-128 pour les mots de passe stockés dans divers fichiers de configuration. Lorsque vous utilisez le chiffrement AES (Advanced Encryption Standard) pour protéger les mots de passe dans la configuration IBM MQ , vous devez comprendre les limites de la protection qu'il fournit.

Le chiffrement d'un mot de passe dans les fichiers de configuration IBM MQ ne signifie pas que le mot de passe est sécurisé ou protégé. Il empêche seulement que le mot de passe soit facilement récupéré par une personne qui peut accéder au mot de passe chiffré, mais ne connaît pas la clé de chiffrement. Les processus IBM MQ requièrent l'accès au mot de passe chiffré et à la clé de déchiffrement pour obtenir le mot de passe en texte clair à utiliser. Ces deux éléments de données doivent être stockés sur le système de fichiers dans un emplacement accessible à IBM MQ. Toute personne qui chiffre un mot de passe placé dans un fichier de configuration doit également accéder à la clé de chiffrement. Si un agresseur a accès au même ensemble de fichiers que IBM MQ, l'application du chiffrement AES au mot de passe n'offre donc qu'un niveau de protection minimal.

Néanmoins, le chiffrement des mots de passe au repos est important à considérer car il empêche la divulgation accidentelle des mots de passe et permet le partage des fichiers de configuration, si la clé de déchiffrement n'est pas également partagée.

En plus de vous assurer que le fichier qui contient la clé de déchiffrement n'est pas partagé, veillez à ce que le fichier soit protégé des autres utilisateurs du système. Alors que les fichiers de configuration IBM MQ peuvent être accessibles à tous les utilisateurs, limitez les droits d'accès au fichier contenant la clé de déchiffrement au minimum nécessaire. Les ID utilisateur traités par IBM MQ doivent être autorisés à lire le fichier contenant la clé de déchiffrement. Toutefois, il n'est pas nécessaire d'accorder l'accès en lecture au fichier à un groupe ou à tous les utilisateurs du système.

Protection des détails d'authentification de la base de données

Si vous utilisez l'authentification par nom d'utilisateur et mot de passe pour vous connecter au gestionnaire de base de données, vous pouvez les stocker dans le magasin de données d'identification MQ XA afin d'éviter de stocker le mot de passe en texte en clair dans le fichier `qm.ini` .

Mettez à jour XAOpenString pour le gestionnaire de ressources

Pour utiliser le magasin de données d'identification, vous devez modifier XAOpenString dans le fichier `qm.ini` . La chaîne est utilisée pour se connecter au gestionnaire de base de données. Vous spécifiez des zones remplaçables pour identifier où le nom d'utilisateur et le mot de passe sont remplacés dans la chaîne XAOpenString .

- La zone +USER+ est remplacée par la valeur de nom d'utilisateur stockée dans le magasin XACredentials.
- La zone +PASSWORD+ est remplacée par la valeur de mot de passe stockée dans le magasin XACredentials.

Les exemples suivants montrent comment modifier un XAOpenString afin d'utiliser le fichier de données d'identification pour se connecter à la base de données.

Connexion à une base de données Db2

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

Connexion à une base de données Oracle

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

Utilisation des données d'identification de la base de données dans le magasin de données d'identification XA MQ

Après avoir mis à jour le fichier `qm.ini` avec les chaînes de données d'identification remplaçables, vous devez ajouter le nom d'utilisateur et le mot de passe au magasin de données d'identification MQ à l'aide de la commande **setmqxacred** . Vous pouvez également utiliser **setmqxacred** pour modifier des données d'identification existantes, supprimer des données d'identification ou répertorier des données d'identification. Les exemples suivants présentent des cas d'utilisation typiques:

Ajout de données d'identification

La commande suivante sauvegarde de manière sécurisée le nom d'utilisateur et le mot de passe du gestionnaire de files d'attente `QM1` pour la ressource `mqdb2`.

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

Mise à jour des droits

Pour mettre à jour le nom d'utilisateur et le mot de passe utilisés pour la connexion à une base de données, émettez à nouveau la commande **setmqxacred** avec le nouveau nom d'utilisateur et le nouveau mot de passe:

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

Vous devez redémarrer le gestionnaire de files d'attente pour que les modifications soient prises en compte.

Suppression des données d'identification

La commande suivante supprime les données d'identification:

```
setmqxacred -m QM1 -x mydb2 -d
```

Liste des données d'identification

La commande suivante répertorie les données d'identification:

```
setmqxacred -m QM1 -l
```

Référence associée

setmqxacred

Sécurisation de Managed File Transfer

Juste après l'installation et si vous n'avez apporté aucune modification, Managed File Transfer présente un niveau de sécurité pouvant être adapté à des fins de test ou d'évaluation dans un environnement protégé. Toutefois, dans un environnement de production, vous devez envisager de contrôler de façon appropriée les utilisateurs pouvant démarrer des opérations de transfert de fichier et lire et écrire les fichiers transférés, et déterminer comment protéger l'intégrité des fichiers.

Tâches associées

Restriction des droits de groupe pour les ressources spécifiques à MFT

Droits de gestion pour les ressources spécifiques à MFT

«Utilisation de Advanced Message Security avec Managed File Transfer», à la page 660

Ce scénario explique comment configurer Advanced Message Security pour fournir la confidentialité des messages pour les données envoyées via un Managed File Transfer.

Référence associée

Droits d'accès de MFT aux systèmes de fichiers

Propriété commandPath MFT

Droits de publication du journal et des messages d'état des agents MFT

Chiffrement des données d'identification stockées dans MFT

Managed File Transfer (MFT) requiert plusieurs ID utilisateur et données d'identification, qui sont stockés dans deux fichiers XML, et vous pouvez les brouiller à l'aide de la commande **fteObfuscate** .

Fichiers de données d'identification

MQMFTCredentials.xml
Ce fichier contient l'ID utilisateur et les données d'identification pour la connexion aux agents et aux gestionnaires de files d'attente de coordination et de commandes. Les données d'identification permettant d'accéder aux magasins de clés pour les connexions sécurisées aux gestionnaires de files d'attente sont également stockées dans le même fichier.

Pour plus de détails sur les valeurs de propriété qui définissent l'emplacement du fichier `MQMFTCredentials.xml` , voir «Authentification de connexion MFT et IBM MQ», à la page 595 .

ProtocolBridgeCredentials.xml
Ce fichier contient l'ID utilisateur et les données d'identification pour la connexion aux serveurs de protocole.

Chiffrement des données d'identification à l'aide de la commande `fteObfuscate`

La commande **fteObfuscate** accepte les paramètres suivants:

- **-f** *credentials_file_name* (requis)

  Remarque : Deprecated Ce paramètre remplace le paramètre **-credentialsFile** qui est obsolète à partir de IBM MQ 9.2.0.
- **-sp** *mode_protection*
- **-sf** *fichier_clé_données d'identification*
- **-o** *nom_fichier_sortie*

Pour plus de détails sur les paramètres, voir **fteObfuscate** .

Si vous ne spécifiez pas le mode de protection ou un fichier de clés de données d'identification, la commande utilise le mode de protection par défaut et l'algorithme le plus récent, mais avec une clé fixe pour chiffrer les données d'identification.

Si vous spécifiez le mode de protection `0`et que vous ne spécifiez pas de fichier de clés de données d'identification, la commande fonctionne comme dans les versions précédentes du produit. Vous recevez un message d'avertissement sur la console indiquant l'utilisation de la protection obsolète.

Si vous spécifiez un mode de protection `0`et un fichier de clés de données d'identification, vous recevez une sortie d'erreur sur la console indiquant qu'il n'est pas valide de spécifier un fichier de clés lors de l'utilisation du mode de protection `0`.

Si vous spécifiez le mode de protection de `1`et que vous ne spécifiez pas de fichier de clés de données d'identification, la commande utilise l'algorithme le plus récent, mais avec une clé fixe pour chiffrer les données d'identification.

Si vous spécifiez le mode de protection de `1`et un fichier de clés de données d'identification, la commande chiffre les données d'identification avec l'algorithme le plus récent.

Si vous spécifiez le mode de protection de `1`ou que vous ne spécifiez pas le mode de protection et que vous spécifiez un fichier de clés de données d'identification qui n'existe pas, une erreur est générée sur la console indiquant que le fichier n'existe pas.

Si vous spécifiez le mode de protection de `1`ou si vous ne spécifiez pas le mode de protection et que vous spécifiez un fichier de clés de données d'identification illisible, une erreur est générée sur la console pour indiquer que le fichier est illisible.

Si vous spécifiez le mode de protection de 2et que vous ne spécifiez pas de fichier de clés de données d'identification, la commande utilise le mode de protection 2 pour chiffrer les données d'identification à l'aide de l'algorithme le plus récent et une clé fixe pour le chiffrement.

Si vous spécifiez le mode de protection de 2et un fichier de clés de données d'identification, la commande utilise le mode de protection 2 pour chiffrer les données d'identification à l'aide de l'algorithme le plus récent et une clé spécifiée par l'utilisateur pour le chiffrement.

Si vous spécifiez le mode de protection de 2ou que vous ne spécifiez pas le mode de protection et que vous spécifiez un fichier de clés de données d'identification qui n'existe pas, une erreur est générée sur la console indiquant que le fichier n'existe pas.

Si vous spécifiez le mode de protection de 2ou si vous ne spécifiez pas le mode de protection et que vous spécifiez un fichier de clés de données d'identification illisible, une erreur est générée sur la console pour indiquer que le fichier est illisible.

Déchiffrement des données d'identification

Vous pouvez spécifier le chemin d'accès au fichier de clés initial à différents endroits. Pour déchiffrer les données d'identification qui ont été chiffrées à l'aide d'une clé initiale autre que celle par défaut, le nom du fichier contenant la clé initiale doit être fourni à MFT de l'une des manières suivantes, dans cet ordre de priorité:

1. A l'aide d'une propriété système Java , par exemple:

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

   Remarque :

   - Avant IBM MQ 9.3.1 et IBM MQ 9.3.0 Fix Pack 10, le nom de cette propriété système Java était mal orthographié dans le code produit sous la forme `com.ibm.wqmfte.cred.keyfile`. Depuis IBM MQ 9.3.1 et IBM MQ 9.3.0 Fix Pack 10, l'orthographe du nom de propriété est corrigée pour être `com.ibm.wmqfte.cred.keyfile`. Managed File Transfer utilise les deux versions de la propriété système Java lorsqu'il vérifie si un utilisateur a spécifié un fichier contenant la clé initiale qui doit être utilisée pour le chiffrement et le déchiffrement des données d'identification. Cela vous permet d'utiliser l'orthographe correcte du nom de propriété, tout en conservant la compatibilité avec l'ancien nom mal orthographié. Notez que si les deux propriétés système Java sont définies, la valeur de la propriété correctement orthographiée `com.ibm.wmqfte.cred.keyfile` est utilisée.
   - Avant IBM MQ 9.3.1 et IBM MQ 9.3.0 Fix Pack 10, utilisez la propriété `com.ibm.wqmfte.cred.keyfile`.

   .
2. En définissant une propriété dans un fichier de propriétés d'agent, de commande, de coordination ou de consignateur. Le nom du fichier de propriétés et la propriété qui doit y être définie sont affichés dans le tableau suivant:

Fichier de propriétés	Nom de la propriété
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. Dans le fichier installation.properties .

   Au lieu d'ajouter des propriétés dans des fichiers de propriétés individuels, vous pouvez ajouter la propriété **commonCredentialsKeyFile** au fichier `installation.properties` commun existant, de sorte que l'agent, le consignateur et les commandes puissent utiliser la même propriété.

Si vous avez défini les différentes propriétés **CredentialsKeyFile** dans plusieurs emplacements:

- Le chemin du fichier de clés de données d'identification utilisé pour l'agent et le consignateur est consigné dans le fichier `output0.log` de cet agent ou consignateur.
- Le chemin du fichier de clés de données d'identification utilisé pour les commandes s'affiche sur la console.

La propriété système Java **com.ibm.wmqfte.cred.keyfile** remplace toutes les autres. Si la propriété système n'est pas définie, l'agent recherche le fichier `agent.properties` , suivi du fichier `installation.properties` pour le fichier de clés initial.

Si le fichier de clés initial est toujours introuvable et que vous avez défini le mode de protection sur la commande **fteObfuscate** sur `1`, l'agent consigne un message d'erreur dans le fichier `output0.log` .

Si vous avez défini le mode de protection sur `0` dans la commande **fteObfuscate** , un message d'avertissement est consigné pour indiquer l'obsolescence.

Le consignateur et les commandes suivent les mêmes étapes pour localiser le fichier de clés initial.

Protocol Bridge et Connect:Direct Bridge

Protocol Bridge utilise un fichier de propriétés, `ProtocolBridgeProperties.xml`, pour la connexion aux serveurs FTP, SFTP et FTPS. Ce fichier de propriétés contient les attributs de connexion requis pour la connexion à ces serveurs.

Un redémarrage de l'agent de pont est requis si vous modifiez la valeur des attributs **credentialsFile** ou **credentialsKeyFile** dans le fichier `ProtocolBridgeProperties.xml` .

L'un des attributs est **credentialsFile**et la valeur contient le chemin d'accès à un fichier XML contenant l'ID utilisateur, le mot de passe ou la clé requise pour la connexion à ces serveurs. La valeur par défaut de l'attribut est *ProtocolBridgeCredentials.xml* et le fichier se trouve dans votre répertoire de base, tout comme le fichier `MQMFTCredentials.xml` .

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

Tout comme *MQMFTCredentails.xml*, vous pouvez chiffrer *ProtocolBridgeCredentials.xml* à l'aide de la commande **fteObfuscate** . A des fins de déchiffrement, vous pouvez spécifier le chemin d'accès requis à un fichier de clés de données d'identification à l'aide de l'élément supplémentaire **credentialsKeyFile** , comme indiqué dans le texte suivant. Le chemin peut contenir des variables d'environnement.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Remarque : La spécification d'une valeur pour la propriété d'agent **agentCredentialsKeyFile** , la propriété **commonCredentialsKeyFile** dans `installation.properties`ou via la propriété système **com.ibm.wqmfte.cred.keyfile**, n'a aucun impact sur la valeur spécifiée pour l'attribut **credentialsKeyFile** .

De même, Connect:Direct Bridge utilise *ConnectDirectNodeProperties.xml* pour se connecter au serveur Connect:Direct . Le fichier XML contient les informations de connexion requises, ainsi qu'un attribut qui définit le chemin d'accès au fichier XML de données d'identification. Ce fichier XML de données d'identification contient l'ID utilisateur ou le mot de passe, ainsi que des informations supplémentaires requises pour la connexion au serveur Connect:Direct .

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

Tout comme le fichier *ProtocolBridgeCredentials.xml* , vous pouvez chiffrer *ConnectDirectCredentials.xml* à l'aide de la commande **fteObfuscate** . A des fins de déchiffrement, vous pouvez spécifier le chemin d'accès requis à un fichier de clés de données d'identification à l'aide de l'élément supplémentaire **credentialsKeyFile** , comme indiqué dans le texte suivant. Le chemin peut contenir des variables d'environnement.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Remarque : La spécification d'une valeur pour la propriété d'agent **agentCredentialsKeyFile** , la propriété **commonCredentialsKeyFile** dans `installation.propertiesou` via la propriété système **com.ibm.wqmfte.cred.keyfile** n'a aucun impact sur la valeur spécifiée pour l'attribut **credentialsKeyFile** .

Vous pouvez spécifier l'élément **credentialsKeyFile** sans spécifier l'élément **credentialsFile** dans le fichier *ProtocolBridgeProperties.xml* .

Si vous ne spécifiez pas l'élément **credentialsFile** , le fichier de données d'identification par défaut *ProtocolBridgeCredentials.xml* est utilisé par l'agent de pont de protocole et la valeur du fichier de clés spécifiée dans l'attribut **credentialsKeyFile** est utilisée pour déchiffrer le fichier de données d'identification.

De même, vous pouvez spécifier l'élément **credentialsKeyFile** sans spécifier l'élément **credentialsFile** dans le fichier *ConnectDirectNodeProperties.xml* .

Si vous ne spécifiez pas l'élément **credentialsFile** , le fichier de données d'identification par défaut *ConnectDirectCredentials.xml* est utilisé par le pont Connect:Direct et la valeur du fichier de clés spécifiée dans l'attribut **credentialsKeyFile** est utilisée pour déchiffrer le fichier de données d'identification.

Utilisation de la clé du fichier sous z/OS

z/OS

Sous z/OS, vous pouvez spécifier **MQMFTCredentials** et fournir le fichier de clés de données d'identification à l'aide d'un ensemble de données partitionnées étendu. Voir «Configuring MQMFTCredentials.xml on z/OS», à la page 598.

Référence associée

Quelle commande MFT se connecte à quel gestionnaire de files d'attente

Format du fichier de données d'identification MFT

fteObfuscate (chiffrement des données sensibles)

Authentification de connexion MFT et IBM MQ

L'authentification de connexion permet à un gestionnaire de files d'attente d'être configuré pour authentifier les applications à l'aide d'un ID utilisateur et d'un mot de passe fournis. Si la sécurité est activée pour le gestionnaire de files d'attente associé et que les données d'identification (ID utilisateur et mot de passe) sont requises, la fonction d'authentification de connexion doit être activée pour que la connexion à un gestionnaire de files d'attente puisse être établie. L'authentification de connexion peut être exécutée en mode compatibilité ou en mode d'authentification MQCSP.

Méthodes de fourniture des détails des données d'identification

De nombreuses commandes Managed File Transfer prennent en charge les méthodes suivantes pour fournir les détails des données d'identification:

Détails fournis par les arguments de ligne de commande.
Les détails des données d'identification peuvent être spécifiés à l'aide des paramètres **-mquserid** et **-mqpassword** . Si le **-mqpassword** n'est pas fourni, l'utilisateur est invité à indiquer le mot de passe dans lequel l'entrée n'est pas affichée.

Détails fournis depuis un fichier de données d'identification : `MQMFTCredentials.xml`.
Les données d'identification détaillées peuvent être prédéfinies dans un fichier `MQMFTCredentials.xml` sous forme de texte en clair ou brouillé.

Multi Pour plus d'informations sur la configuration d'un fichier MQMFTCredentials.xml sur IBM MQ for Multiplatforms , voir «Configuration de MQMFTCredentials.xml sur Multiplatforms», à la page 596.

z/OS Pour plus d'informations sur la configuration d'un fichier MQMFTCredentials.xml sur IBM MQ for z/OS , voir «Configuring MQMFTCredentials.xml on z/OS», à la page 598.

Priorité

L'ordre de priorité des méthodes pour déterminer les données d'identification détaillées est le suivant :

1. Argument de ligne de commande
2. Index `MQMFTCredentials.xml` par gestionnaire de files d'attente associé et utilisateur exécutant la commande
3. Index `MQMFTCredentials.xml` par gestionnaire de files d'attente associé
4. Mode de compatibilité amont par défaut dans lequel aucun détail de données d'identification n'est fourni pour permettre la compatibilité avec les versions précédentes de IBM MQou IBM WebSphere MQ

Remarques :

- Les commandes **`fteStartAgent`** et **`fteStartLogger`** ne prennent pas en charge l'argument de ligne de commande **`-mquserid`** ou **`-mqpassword`**, et les données d'identification détaillées ne peuvent être spécifiées qu'à l'aide du fichier `MQMFTCredentials.xml`.
- z/OS

  Sous z/OS, le mot de passe doit être en majuscules, même si le mot de passe de l'utilisateur est en minuscules. Par exemple, si le mot de passe de l'utilisateur est "motdepasse", il doit être entré sous la forme "MOTDEPASSE".

Référence associée

Quelle commande MFT se connecte à quel gestionnaire de files d'attente

Format du fichier de données d'identification MFT

Configuration de MQMFTCredentials.xml sur Multiplatforms

Si Managed File Transfer (MFT) est configuré avec la sécurité activée, l'authentification de connexion requiert toutes les commandes MFT qui se connectent à un gestionnaire de files d'attente pour fournir les données d'identification par ID utilisateur et mot de passe. De même, les consignateurs MFT peuvent être tenus de spécifier un ID utilisateur et un mot de passe lors de la connexion à une base de données. Ces données d'identification peuvent être stockées dans le fichier de données d'identification MFT .

Pourquoi et quand exécuter cette tâche

Les éléments du fichier `MQMFTCredentials.xml` doivent être conformes au schéma `MQMFTCredentials.xsd` . Pour plus d'informations sur le format de `MQMFTCredentials.xml`, voir Format de fichier des données d'identification MFT.

Vous trouverez un exemple de fichier de données d'identification dans le répertoire `MQ_INSTALLATION_PATH/mqft/samples/credentials` .

Vous pouvez disposer d'un fichier de données d'identification MFT pour le gestionnaire de files d'attente de coordination, d'un fichier pour le gestionnaire de files d'attente de commandes, d'un fichier pour chaque agent et d'un fichier pour chaque consignateur. Vous pouvez également disposer d'un fichier qui est utilisé par tous les éléments de votre topologie.

L'emplacement par défaut du fichier de données d'identification MFT est le suivant:

Linux AIX **AIX and Linux**

$HOME

Windows **Windows**

%USERPROFILE% ou %HOMEDRIVE%%HOMEPATH%

Si le fichier de données d'identification est stocké à un autre emplacement, vous pouvez utiliser les propriétés suivantes pour indiquer où les commandes doivent le rechercher:

Tableau 97. : Propriétés qui définissent l'emplacement du fichier MQMFTCredentials.xml pour diverses commandes.

Type de commande	Fichier de propriétés	Nom de la propriété
Commande qui se connecte au gestionnaire de file d'attente de coordination	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Commande qui se connecte au gestionnaire de files d'attente de commandes	connection.properties	connectionQMgrAuthenticationCredentialsFile
Commande qui se connecte à un processus d'agent	agent.properties	agentQMgrAuthenticationCredentialsFile
Commande qui se connecte à un processus de consignateur	logger.properties	loggerQMgrAuthenticationCredentialsFile

Tableau 98. : Propriétés qui définissent l'emplacement du fichier MQMFTCredentials.xml pour les agents et les processus de consignateur.

Type de commande	Fichier de propriétés	Nom de la propriété
Agents MFT	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

Pour plus de détails sur les commandes et les processus qui se connectent à quel gestionnaire de files d'attente, voir Quelles commandes et quels processus MFT se connectent à quel gestionnaire de files d'attente.

Au lieu d'ajouter des propriétés dans des fichiers de propriétés individuels, vous pouvez ajouter la propriété **commonCredentialsKeyFile** au fichier `installation.properties` commun existant, afin que l'agent, le consignateur et les commandes puissent utiliser la même propriété.

Etant donné que le fichier de données d'identification contient des informations d'ID utilisateur et de mot de passe, il requiert des droits spéciaux pour empêcher tout accès non autorisé à ce fichier:

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

Vérifiez que l'héritage n'est pas activé, puis supprimez tous les ID utilisateur à l'exception de ceux qui exécutent l'agent ou le consignateur qui utiliseront le fichier de données d'identification.

Les données d'identification utilisées pour la connexion à un gestionnaire de files d'attente de coordination MFT dans le plug-in IBM MQ Explorer Managed File Transfer dépendent du type de configuration:

Global (configuration sur disque local)

Une configuration globale utilise le fichier de données d'identification spécifié dans les propriétés de coordination et de commande.

Local (défini dans IBM MQ Explorer):
Une configuration locale utilise les propriétés des détails de connexion du gestionnaire de files d'attente associé dans IBM MQ Explorer.

Tâches associées

«Activation de l'authentification de connexion pour MFT», à la page 600
L'authentification de connexion du plug-in IBM MQ Explorer MFT se connectant à un gestionnaire de files d'attente de coordination ou à un gestionnaire de files d'attente de commandes et l'authentification de connexion pour un agent Managed File Transfer se connectant à un gestionnaire de files d'attente de coordination ou à un gestionnaire de files d'attente de commandes peuvent être exécutées en mode compatibilité ou en mode d'authentification MQCSP.

Création d'une structure de transfert de fichiers IBM MQ

Référence associée

Format du fichier de données d'identification MFT

Chiffrement des données d'identification stockées dans MFT

fteObfuscate: chiffrement des données sensibles

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.xmlfile:

```
<?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

“Configuration de MQMFTCredentials.xml sur Multiplatforms” on page 596
Si Managed File Transfer (MFT) est configuré avec la sécurité activée, l'authentification de connexion requiert toutes les commandes MFT qui se connectent à un gestionnaire de files d'attente pour fournir les données d'identification par ID utilisateur et mot de passe. De même, les consignateurs MFT peuvent être tenus de spécifier un ID utilisateur et un mot de passe lors de la connexion à une base de données. Ces données d'identification peuvent être stockées dans le fichier de données d'identification MFT .

Activation de l'authentification de connexion pour MFT

L'authentification de connexion du plug-in IBM MQ Explorer MFT se connectant à un gestionnaire de files d'attente de coordination ou à un gestionnaire de files d'attente de commandes et l'authentification de connexion pour un agent Managed File Transfer se connectant à un gestionnaire de files d'attente de coordination ou à un gestionnaire de files d'attente de commandes peuvent être exécutées en mode compatibilité ou en mode d'authentification MQCSP.

Pourquoi et quand exécuter cette tâche

Le mode d'authentification MQCSP est le mode par défaut.

Pour l'authentification de connexion pour le plug-in IBM MQ Explorer Managed File Transfer ou pour les agents Managed File Transfer qui se connectent à un gestionnaire de files d'attente à l'aide du transport CLIENT, les mots de passe de plus de 12 caractères sont uniquement pris en charge pour le mode d'authentification MQCSP. Si vous indiquez un mot de passe de plus de 12 caractères lors de l'autorisation en mode compatibilité, une erreur se produit et l'agent ne s'authentifie pas auprès du gestionnaire de files d'attente. Voir le message BFGAG0187E dans Messages de diagnostic: BFGAG0001 - BFGAG9999.

Procédure

- Pour sélectionner le mode d'authentification de connexion pour un gestionnaire de files d'attente de coordination ou un gestionnaire de files d'attente de commandes dans IBM MQ Explorer, procédez comme suit:
  a) Sélectionnez le gestionnaire de files d'attente auquel vous souhaitez vous connecter.
  b) Cliquez avec le bouton droit de la souris et sélectionnez **Détails de connexion-> Propriétés** dans le menu contextuel.
  c) Cliquez sur l'onglet **ID utilisateur**.
  d) Vérifiez que la case correspondant au mode d'authentification de connexion que vous souhaitez utiliser est cochée:
     – Par défaut, la case **User identification compatibility mode** est désélectionnée. Cela signifie que si la case **Activer l'identification de l'utilisateur** est cochée, IBM MQ Explorer utilisera l'authentification MQCSP lors de la connexion au gestionnaire de files d'attente. Si IBM MQ Explorer doit se connecter au gestionnaire de files d'attente à l'aide du mode de compatibilité au lieu de l'authentification MQCSP, vérifiez que les cases à cocher **Activer l'identification de l'utilisateur** et **Mode de compatibilité d'identification de l'utilisateur** sont sélectionnées.
- Pour activer ou désactiver le mode d'authentification MQCSP pour un agent Managed File Transfer à l'aide du fichier `MQMFTCredentials.xml` , ajoutez le paramètre **`useMQCSPAuthentication`** au fichier `MQMFTCredentials.xml` pour l'utilisateur approprié.

  Le paramètre **`useMQCSPAuthentication`** a les valeurs suivantes:

  Oui
  Le mode d'authentification MQCSP permet d'authentifier l'utilisateur auprès du gestionnaire de files d'attente.

`true` est la valeur par défaut. Si le paramètre **useMQCSPAuthentication** n'est pas spécifié, il est défini par défaut sur `true` et le mode d'authentification MQCSP est utilisé pour authentifier l'utilisateur avec le gestionnaire de files d'attente.

false
Le mode compatibilité est utilisé pour authentifier l'utilisateur auprès du gestionnaire de files d'attente.

L'exemple suivant montre comment définir le paramètre **useMQCSPAuthentication** dans le fichier `MQMFTCredentials.xml` :

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

Concepts associés

«Protection par mot de passe MQCSP», à la page 32
Les données d'authentification qui sont spécifiées dans la structure MQCSP peuvent être protégées à l'aide de la fonction de protection par mot de passe IBM MQ MQCSP ou chiffrées à l'aide du chiffrement TLS.

Référence associée

«Authentification de connexion MFT et IBM MQ», à la page 595
L'authentification de connexion permet à un gestionnaire de files d'attente d'être configuré pour authentifier les applications à l'aide d'un ID utilisateur et d'un mot de passe fournis. Si la sécurité est activée pour le gestionnaire de files d'attente associé et que les données d'identification (ID utilisateur et mot de passe) sont requises, la fonction d'authentification de connexion doit être activée pour que la connexion à un gestionnaire de files d'attente puisse être établie. L'authentification de connexion peut être exécutée en mode compatibilité ou en mode d'authentification MQCSP.

Format du fichier de données d'identification MFT

MFT bacs à sable

Vous pouvez restreindre la zone du système de fichiers à laquelle l'agent peut accéder dans le cadre d'un transfert. La zone à laquelle l'agent est limité est appelée le bac à sable. Vous pouvez appliquer des restrictions à l'agent ou à l'utilisateur qui demande un transfert.

Les bacs à sable ne sont pas pris en charge lorsque l'agent est un agent de pont de protocole ou un agent de pont Connect:Direct . Vous ne pouvez pas utiliser le bac à sable d'agent pour les agents qui doivent effectuer un transfert vers ou depuis des files d'attente IBM MQ .

Référence associée

«Utilisation des bacs à sable d'agent MFT», à la page 601
Pour ajouter un niveau de sécurité supplémentaire à Managed File Transfer, vous pouvez restreindre la zone d'un système de fichiers à laquelle un agent peut accéder.

«Utilisation des bacs à sable utilisateur MFT», à la page 603
Vous pouvez restreindre la zone du système de fichiers dans laquelle les fichiers peuvent être transférés en fonction du nom d'utilisateur MQMD qui demande le transfert.

Utilisation des bacs à sable d'agent MFT

Pour ajouter un niveau de sécurité supplémentaire à Managed File Transfer, vous pouvez restreindre la zone d'un système de fichiers à laquelle un agent peut accéder.

Vous ne pouvez pas utiliser le bac à sable d'agent pour les agents qui sont transférés vers ou depuis des files d'attente IBM MQ . La restriction de l'accès aux files d'attente IBM MQ avec bac à sable peut être implémentée à la place en utilisant le bac à sable utilisateur, qui est la solution recommandée pour toutes les exigences en matière de bac à sable. Pour plus d'informations sur le bac à sable utilisateur, voir «Utilisation des bacs à sable utilisateur MFT», à la page 603

Pour activer le bac à sable de l'agent, ajoutez la propriété suivante au fichier `agent.properties` de l'agent que vous souhaitez restreindre:

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

où :

- *restricted_directory_name* est un chemin de répertoire à autoriser ou à refuser.
- `!` est facultatif et indique que la valeur suivante pour *restricted_directory_name* est refusée (exclue). Si `!` n'est pas spécifié, *restricted_directory_name* est un chemin d'accès autorisé (inclus).
- *separator* est le séparateur spécifique à la plateforme.

Par exemple, si vous souhaitez restreindre l'accès de AGENT1 au répertoire `/tmp` uniquement, mais ne pas autoriser l'accès au sous-répertoire `private` , définissez la propriété comme suit dans le fichier `agent.properties` appartenant à AGENT1: `sandboxRoot=/tmp:!/tmp/private`.

La propriété sandboxRoot est décrite dans Propriétés avancées de l'agent.

Les bacs à sable d'agent et d'utilisateur ne sont pas pris en charge sur les agents de pont de protocole ou sur les agents de pont Connect:Direct .

Utilisation d'un bac à sable sur les plateformes AIX, Linux, and Windows

ALW Sur les plateformes AIX, Linux, and Windows , le bac à sable restreint les répertoires dans lesquels un Managed File Transfer Agent peut effectuer des opérations de lecture et d'écriture. Lorsque le bac à sable est activé, le Managed File Transfer Agent peut lire et écrire dans les répertoires spécifiés comme étant autorisés, ainsi que dans les sous-répertoires que les répertoires spécifiés contiennent, sauf si les sous-répertoires sont spécifiés comme étant refusés dans sandboxRoot. Le bac à sable Managed File Transfer n'est pas prioritaire sur la sécurité du système d'exploitation. L'utilisateur qui a démarré Managed File Transfer Agent doit disposer de l'accès de niveau système d'exploitation approprié à n'importe quel répertoire pour pouvoir lire ou écrire dans le répertoire. Un lien symbolique vers un répertoire n'est pas suivi si le répertoire auquel il est lié se trouve en dehors des répertoires sandboxRoot spécifiés (et des sous-répertoires).

Utilisation d'un bac à sable sous z/OS

z/OS Sous z/OS, le bac à sable restreint les qualificateurs de nom de fichier que le Managed File Transfer Agent peut lire et dans lesquels il peut écrire. L'utilisateur qui a démarré Managed File Transfer Agent doit disposer des droits d'accès appropriés au système d'exploitation pour tous les fichiers concernés. Si vous placez une valeur de qualificateur de nom de fichier sandboxRoot entre guillemets, la valeur suit la convention z/OS normale et est traitée comme qualifiée complète. Si vous omettez les guillemets, sandboxRoot est préfixé avec l'ID utilisateur en cours. Par exemple, si vous définissez la propriété sandboxRoot sur la valeur suivante: `sandboxRoot=//test`, l'agent peut accéder aux ensembles de données suivants (en notation z/OS standard) `//username.test.**` Au moment de l'exécution, si les niveaux initiaux du nom de fichier entièrement résolu ne correspondent pas à sandboxRoot, la demande de transfert est rejetée.

Utilisation d'un bac à sable sur des systèmes IBM i

IBM i Pour les fichiers du système de fichiers intégré sur les systèmes IBM i , le bac à sable restreint les répertoires dans lesquels un Managed File Transfer Agent peut effectuer des opérations de lecture et d'écriture. Lorsque le bac à sable est activé, le Managed File Transfer Agent peut lire et écrire dans les répertoires spécifiés comme étant autorisés, ainsi que dans les sous-répertoires que les répertoires spécifiés contiennent, sauf si les sous-répertoires sont spécifiés comme étant refusés dans sandboxRoot. Le bac à sable Managed File Transfer n'est pas prioritaire sur la sécurité du système d'exploitation. L'utilisateur qui a démarré Managed File Transfer Agent doit disposer de l'accès de niveau système d'exploitation approprié à n'importe quel répertoire pour pouvoir lire ou écrire dans le répertoire.

Un lien symbolique vers un répertoire n'est pas suivi si le répertoire auquel il est lié se trouve en dehors des répertoires sandboxRoot spécifiés (et des sous-répertoires).

Référence associée

«Vérifications supplémentaires pour les transferts de caractères génériques», à la page 606
Si un agent a été configuré avec un bac à sable d'utilisateur ou d'agent afin de restreindre les emplacements vers et depuis lesquels l'agent peut transférer des fichiers, vous pouvez indiquer que des vérifications supplémentaires doivent être effectuées sur les transferts de caractères génériques pour cet agent.

«Utilisation des bacs à sable d'agent MFT», à la page 601
Pour ajouter un niveau de sécurité supplémentaire à Managed File Transfer, vous pouvez restreindre la zone d'un système de fichiers à laquelle un agent peut accéder.

Le fichier MFT `agent.properties`

Utilisation des bacs à sable utilisateur MFT

Vous pouvez restreindre la zone du système de fichiers dans laquelle les fichiers peuvent être transférés en fonction du nom d'utilisateur MQMD qui demande le transfert.

Les bacs à sable utilisateur ne sont pas pris en charge lorsque l'agent est un agent de pont de protocole ou un agent de pont Connect:Direct .

Pour activer le bac à sable utilisateur, ajoutez la propriété suivante au fichier `agent.properties` de l'agent que vous souhaitez restreindre:

```
userSandboxes=true
```

Lorsque cette propriété est présente et définie sur true, l'agent utilise les informations du fichier *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name*/UserSandboxes.xml pour déterminer les parties du système de fichiers auxquelles l'utilisateur qui demande le transfert peut accéder.

Le XML `UserSandboxes.xml` est composé d'un élément <agent> qui contient zéro ou plusieurs éléments <sandbox> . Ces éléments décrivent quelles règles sont appliquées à quels utilisateurs. L'attribut `user` de l'élément <sandbox> est un modèle utilisé pour établir une correspondance avec l'utilisateur MQMD de la demande.

Le fichier `UserSandboxes.xml` est rechargé périodiquement par l'agent et toute modification valide apportée au fichier aura une incidence sur le comportement de l'agent. L'intervalle de rechargement par défaut est de 30 secondes. Vous pouvez modifier cet intervalle en spécifiant la propriété d'agent xmlConfigReloadInterval dans le fichier `agent.properties` .

Si vous spécifiez l'attribut ou la valeur `userPattern="regex"` , l'attribut `user` est interprété comme une expression régulière Java . Pour plus d'informations, voir Expressions régulières utilisées par MFT.

Si vous ne spécifiez pas l'attribut ou la valeur `userPattern="regex"` , l'attribut `user` est interprété comme un modèle avec les caractères génériques suivants:

- astérisque (*), qui représente zéro ou plusieurs caractères
- point d'interrogation (?), qui représente exactement un caractère

Les correspondances sont effectuées dans l'ordre dans lequel les éléments <sandbox> sont répertoriés dans le fichier. Seule la première correspondance est utilisée, toutes les correspondances potentielles suivantes dans le fichier sont ignorées. Si aucun des éléments <sandbox> spécifiés dans le fichier ne correspond à l'utilisateur MQMD associé au message de demande de transfert, le transfert ne peut pas accéder au système de fichiers. Lorsqu'une correspondance a été trouvée entre le nom d'utilisateur MQMD et un attribut `user` , la correspondance identifie un ensemble de règles dans un élément <sandbox> qui sont appliquées au transfert. Cet ensemble de règles est utilisé pour déterminer quels fichiers, ou ensembles de données, peuvent être lus ou écrits dans le cadre du transfert.

Chaque ensemble de règles peut spécifier un élément <read> , qui identifie les fichiers qui peuvent être lus, et un élément <write> , qui identifie les fichiers qui peuvent être écrits. Si vous omettez les

éléments <read> ou <write> d'un ensemble de règles, il est supposé que l'utilisateur associé à cet ensemble de règles n'est pas autorisé à effectuer des lectures ou des écritures, selon le cas.

Remarque : L'élément <read> doit être antérieur à l'élément <write> et l'élément <include> doit être antérieur à l'élément <exclude> , dans le fichier UserSandboxes.xml .

Chaque élément <read> ou <write> contient un ou plusieurs canevas utilisés pour déterminer si un fichier se trouve dans le bac à sable et peut être transféré. Spécifiez ces modèles à l'aide des éléments <include> et <exclude> . L'attribut name de l'élément <include> ou <exclude> spécifie le modèle à mettre en correspondance. Un attribut type facultatif indique si la valeur de nom est un fichier ou un modèle de file d'attente. Si l'attribut type n'est pas spécifié, l'agent traite le modèle comme un modèle de chemin de fichier ou de répertoire. Exemple :

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

Les modèles <include> et <exclude> name sont utilisés par l'agent pour déterminer si les fichiers, les ensembles de données, ou les files d'attente peuvent être lus ou écrits. Une opération est autorisée si le chemin de fichier canonique, l'ensemble de données ou le nom de file d'attente correspond à au moins un des modèles inclus et à exactement zéro des modèles exclus. Les modèles spécifiés à l'aide de l'attribut name des éléments <include> et <exclude> utilisent les séparateurs de chemin et les conventions appropriés à la plateforme sur laquelle l'agent s'exécute. Si vous spécifiez des chemins de fichier relatifs, les chemins sont résolus par rapport à la propriété transferRoot de l'agent.

Lors de la spécification d'une restriction de file d'attente, la syntaxe QUEUE@QUEUEMANAGER est prise en charge, avec les règles suivantes:

- Si le caractère at (@) est manquant dans l'entrée, le modèle est traité comme un nom de file d'attente accessible sur n'importe quel gestionnaire de files d'attente. Par exemple, si le modèle est name , il est traité de la même manière que name@**.
- Si le caractère arobase (@) est le premier caractère de l'entrée, le modèle est traité comme un nom de gestionnaire de files d'attente et toutes les files d'attente du gestionnaire de files d'attente sont accessibles. Par exemple, si le modèle est @name , il est traité de la même manière que **@name.

Les caractères génériques suivants ont une signification spéciale lorsque vous les spécifiez dans le cadre de l'attribut name des éléments <include> et <exclude> :

*****
: Un astérisque unique correspond à zéro ou plusieurs caractères dans un nom de répertoire ou dans un qualificateur d'un nom de fichier ou d'un nom de file d'attente .

?
: Un point d'interrogation correspond exactement à un caractère dans un nom de répertoire ou dans un qualificateur d'un nom de fichier ou d'un nom de file d'attente .

******
: Deux astérisques correspondent à zéro ou plusieurs noms de répertoire, ou à zéro ou plusieurs qualificatifs dans un nom de fichier ou un nom de file d'attente . En outre, les chemins qui se terminent par un séparateur de chemin ont un "**" implicite ajouté à la fin du chemin. Ainsi, /home/user/ est identique à /home/user/**.

Exemple :

- /**/test/** correspond à tout fichier dont le chemin contient un répertoire test
- /test/file? correspond à tout fichier du répertoire /test qui commence par la chaîne file suivie d'un caractère unique
- c:\test*.txt correspond à tout fichier du répertoire c:\test avec une extension .txt

- `c:\test\**\*.txt` correspond à n'importe quel fichier du répertoire `'c:\test` ou à l'un de ses sous-répertoires dont l'extension est `.txt`
- z/OS `//'TEST.*.DATA'` correspond à tout fichier dont le premier qualificateur est TEST, dont le second est un qualificateur et dont le troisième est DATA.
- *@QM1 correspond à toute file d'attente du gestionnaire de files d'attente QM1 comportant un qualificateur unique.
- `TEST.*.QUEUE@QM1` correspond à n'importe quelle file d'attente du gestionnaire de files d'attente QM1 qui possède le premier qualificateur TEST, un deuxième qualificateur et un troisième qualificateur QUEUE.
- **@QM1 correspond à n'importe quelle file d'attente du gestionnaire de files d'attente QM1.

Liens symboliques

Vous devez résoudre complètement tous les liens symboliques que vous utilisez dans les chemins de fichier du fichier `UserSandboxes.xml` en spécifiant des liens fixes dans les éléments `<include>` et `<exclude>` . Par exemple, si vous disposez d'un lien symbolique dans lequel `/var` est mappé à `/SYSTEM/var`, vous devez spécifier ce chemin en tant que `<tns:include name="/SYSTEM/var"/>`, sinon le transfert prévu échoue avec une erreur de sécurité du bac à sable de l'utilisateur.

Exemple

Cet exemple montre comment autoriser l'utilisateur avec le nom d'utilisateur MQMD `guest` à transférer un fichier depuis le répertoire `/home/user/public` ou l'un de ses sous-répertoires sur le système où l'agent AGENT_JUPITER est en cours d'exécution, en ajoutant l'élément `<sandbox>` suivant au fichier `UserSandboxes.xml` dans le répertoire de configuration d'AGENT_JUPITER:

```
<?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>
```

Exemple

Cet exemple montre comment autoriser tout utilisateur avec le nom d'utilisateur MQMD `account` suivi d'un chiffre unique, par exemple `account4`, à effectuer les actions suivantes:

- Transférez tout fichier à partir du répertoire `/home/account` ou de l'un de ses sous-répertoires, à l'exception du répertoire `/home/account/private` sur le système où l'agent AGENT_SATURN est en cours d'exécution
- Transférez tout fichier dans le répertoire `/home/account/output` ou dans l'un de ses sous-répertoires sur le système où l'agent AGENT_SATURN est en cours d'exécution.
- Lire les messages des files d'attente du gestionnaire de files d'attente local en commençant par le préfixe `ACCOUNT.` , sauf s'il commence par `ACCOUNT.PRIVATE.` (c'est-à-dire avec `PRIVATE` au deuxième niveau).
- Transférez les données dans les files d'attente en commençant par le préfixe `ACCOUNT.OUTPUT.` sur n'importe quel gestionnaire de files d'attente.

Pour permettre à un utilisateur doté du nom d'utilisateur MQMD `account` d'effectuer ces actions, ajoutez l'élément <sandbox> suivant au fichier `UserSandboxes.xml`, dans le répertoire de configuration d'AGENT_SATURN:

```
<?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>
```

Référence associée

«Vérifications supplémentaires pour les transferts de caractères génériques», à la page 606
Si un agent a été configuré avec un bac à sable d'utilisateur ou d'agent afin de restreindre les emplacements vers et depuis lesquels l'agent peut transférer des fichiers, vous pouvez indiquer que des vérifications supplémentaires doivent être effectuées sur les transferts de caractères génériques pour cet agent.

Le fichier MFT `agent.properties`

Vérifications supplémentaires pour les transferts de caractères génériques

Si un agent a été configuré avec un bac à sable d'utilisateur ou d'agent afin de restreindre les emplacements vers et depuis lesquels l'agent peut transférer des fichiers, vous pouvez indiquer que des vérifications supplémentaires doivent être effectuées sur les transferts de caractères génériques pour cet agent.

Propriété additionalWildcardSandboxChecking

Pour activer une vérification supplémentaire des transferts de caractères génériques, ajoutez la propriété suivante au fichier `agent.properties` de l'agent que vous souhaitez vérifier.

```
additionalWildcardSandboxChecking=true
```

Lorsque cette propriété est définie sur true et que l'agent effectue une demande de transfert qui tente de lire un emplacement qui se trouve en dehors du bac à sable défini pour la correspondance de fichier du caractère générique, le transfert échoue. S'il existe plusieurs transferts dans une même demande de transfert et que l'une de ces demandes échoue en raison d'une tentative de lecture d'un emplacement en dehors du bac à sable, le transfert complet échoue. Si la vérification échoue, la cause de l'échec est indiquée dans un message d'erreur.

Si la propriété additionalWildcardSandboxChecking est omise du fichier `agent.properties` d'un agent ou est définie sur false, aucune vérification supplémentaire n'est effectuée sur les transferts de caractères génériques pour cet agent.

Messages d'erreur pour la vérification des caractères génériques

Les messages qui sont signalés lorsqu'une demande de transfert générique est effectuée vers un emplacement en dehors d'un emplacement de bac à sable configuré sont les suivants.

Le message suivant apparaît lorsqu'un chemin de fichier générique dans une demande de transfert se trouve en dehors du bac à sable restreint:

```
BFGSS0077E: La tentative de lecture du chemin d'accès au fichier chemin a été refusée.
Le chemin d'accès au fichier se trouve hors du bac à sable de transfert restreint.
```

Le message suivant se produit lorsqu'un transfert au sein d'une demande de transfert multiple contient une demande de transfert générique dans laquelle le chemin se trouve en dehors du bac à sable restreint:

```
BFGSS0078E: La tentative de lecture du chemin d'accès au fichier: chemin a été ignorée car un
autre transfert
dans le transfert géré, tentative de lecture en dehors du bac à sable de transfert restreint.
```

Le message suivant s'affiche lorsqu'un fichier se trouve en dehors du bac à sable restreint:

```
BFGSS0079E: La tentative de lecture du fichier chemin d'accès au fichier a été refusée.
Le fichier se trouve en dehors du bac à sable de transfert restreint.
```

Le message suivant se produit dans une demande de transfert multiple où une autre demande de transfert générique a entraîné la non-prise en compte de cette demande:

```
BFGSS0080E: La tentative de lecture du fichier: chemin d'accès au fichier a été ignorée car un
autre transfert
dans le transfert géré, tentative de lecture en dehors du bac à sable de transfert restreint.
```

Dans le cas de transferts de fichiers uniques qui n'incluent pas de caractères génériques, le message qui est signalé lorsque le transfert implique un fichier situé en dehors du bac à sable est inchangé par rapport aux éditions précédentes:

```
Echec avec BFGIO0056E: La tentative de lecture du fichier "FILE" a été refusée.
Le fichier se trouve en dehors du bac à sable de transfert restreint.
```

Référence associée

«Utilisation des bacs à sable utilisateur MFT», à la page 603
Vous pouvez restreindre la zone du système de fichiers dans laquelle les fichiers peuvent être transférés en fonction du nom d'utilisateur MQMD qui demande le transfert.

«Utilisation des bacs à sable d'agent MFT», à la page 601
Pour ajouter un niveau de sécurité supplémentaire à Managed File Transfer, vous pouvez restreindre la zone d'un système de fichiers à laquelle un agent peut accéder.

Le fichier MFT `agent.properties`

Configuration du chiffrement SSL ou TLS pour MFT

Vous pouvez utiliser SSL ou TLS avec IBM MQ Managed File Transfer pour sécuriser les communications entre les agents et leurs gestionnaires de files d'attente d'agent, les commandes et les gestionnaires de files d'attente auxquels ils se connectent, ainsi que les différentes connexions entre les gestionnaires de files d'attente et les gestionnaires de files d'attente dans votre topologie.

Avant de commencer

Vous pouvez utiliser le chiffrement SSL ou TLS pour chiffrer les messages qui transitent par une topologie IBM MQ Managed File Transfer . Ces gestionnaires sont les suivants :

- Messages transmis entre un agent et son gestionnaire de files d'attente d'agent.
- Messages des commandes et des gestionnaires de files d'attente auxquels elles se connectent.
- Messages internes qui circulent entre les gestionnaires de files d'attente d'agent, les gestionnaires de files d'attente de commandes et le gestionnaire de files d'attente de coordination dans la topologie.

Pourquoi et quand exécuter cette tâche

Pour des informations générales sur l'utilisation de SSL avec IBM MQ, voir «Utilisation de SSL/TLS», à la page 283. En termes IBM MQ , Managed File Transfer est une application client Java standard.

Pour utiliser SSL avec Managed File Transfer, procédez comme suit:

Procédure

1. Créez un fichier de clés certifiées et éventuellement un fichier de clés (ces fichiers peuvent être identiques). Si vous n'avez pas besoin de l'authentification client (c'est-à-dire, SSLCAUTH=OPTIONAL sur les canaux), vous n'avez pas besoin de fournir un magasin de clés. Vous avez besoin d'un magasin de clés de confiance uniquement pour authentifier le certificat du gestionnaire de files d'attente.

   L'algorithme de clé utilisé pour créer des certificats pour le magasin de clés de confiance et les magasins de clés doit être RSA pour fonctionner avec IBM MQ.
2. Configurez votre gestionnaire de files d'attente IBM MQ pour utiliser SSL.

   Pour plus d'informations sur la configuration d'un gestionnaire de files d'attente pour utiliser SSL avec IBM MQ Explorer , par exemple, voir Configuration de SSL sur les gestionnaires de files d'attente.
3. Sauvegardez le fichier de clés certifiées et le fichier de clés (si vous en avez un) dans un emplacement approprié. Un emplacement suggéré est le répertoire *config_directory*/*coordination_qmgr*/agents/*agent_name* .
4. Définissez les propriétés SSL requises pour chaque gestionnaire de files d'attente SSL dans le fichier de propriétés Managed File Transfer approprié. Chaque ensemble de propriétés fait référence à un gestionnaire de files d'attente distinct (agent, coordination et commande), bien qu'un gestionnaire de files d'attente puisse exécuter deux ou plusieurs de ces rôles.

   L'une des propriétés **CipherSpec** ou **CipherSuite** est requise, sinon le client tente de se connecter sans SSL. Les propriétés **CipherSpec** ou **CipherSuite** sont fournies en raison des différences de terminologie entre IBM MQ et Java. Managed File Transfer accepte l'une ou l'autre propriété et effectue la conversion nécessaire. Vous n'avez donc pas besoin de définir les deux propriétés. Si vous spécifiez à la fois les propriétés **CipherSpec** ou **CipherSuite** , **CipherSpec** est prioritaire.

   La propriété **PeerName** est facultative. Vous pouvez définir la propriété sur le nom distinctif du gestionnaire de files d'attente auquel vous souhaitez vous connecter. Managed File Transfer rejette les connexions à un serveur SSL incorrect avec un nom distinctif qui ne correspond pas.

   Définissez les propriétés **SslTrustStore** et **SslKeyStore** sur des noms de fichier qui pointent vers les fichiers de clés certifiées et les fichiers de clés. Si vous configurez ces propriétés pour un agent déjà en cours d'exécution, arrêtez et redémarrez l'agent pour qu'il se reconnecte en mode SSL.

   Les fichiers de propriétés contiennent des mots de passe en texte en clair. Il est donc judicieux de définir les droits d'accès appropriés au système de fichiers.

   Pour plus d'informations sur les propriétés SSL, voir «Propriétés SSL/TLS pour MFT», à la page 608.
5. Si un gestionnaire de files d'attente d'agent utilise SSL, vous ne pouvez pas fournir les détails nécessaires lors de la création de l'agent. Procédez comme suit pour créer l'agent:
   a) Créez l'agent à l'aide de la commande **fteCreateAgent** . Vous recevez un avertissement indiquant que vous ne parvenez pas à publier l'existence de l'agent dans le gestionnaire de file d'attente de coordination.
   b) Editez le fichier `agent.properties` créé à l'étape précédente pour ajouter les informations SSL. Lorsque l'agent est correctement démarré, une nouvelle tentative de publication est effectuée.
6. Si des agents ou des instances de IBM MQ Explorer sont en cours d'exécution alors que les propriétés SSL du fichier `agent.properties` ou du fichier `coordination.properties` sont modifiées, vous devez redémarrer l'agent ou IBM MQ Explorer.

Référence associée

Le fichier MFT `agent.properties`

Propriétés SSL/TLS pour MFT

Certains fichiers de propriétés MFT incluent des propriétés SSL et TLS. Vous pouvez utiliser SSL ou TLS avec IBM MQ et Managed File Transfer pour empêcher les connexions non autorisées entre les agents et les gestionnaires de files d'attente et pour chiffrer le trafic des messages entre les agents et les gestionnaires de files d'attente.

Les fichiers de propriétés MFT suivants incluent des propriétés SSL:

- Propriétés SSL/TLS du fichier MFT `agent.properties`
- Propriétés SSL/TLS du fichier MFT `coordination.properties`
- Propriétés SSL/TLS du fichier MFT `command.properties`
- Propriétés SSL/TLS du fichier MFT `logger.properties`

Pour plus d'informations sur l'utilisation de SSL ou TLS avec Managed File Transfer, voir «Configuration du chiffrement SSL ou TLS pour MFT», à la page 607.

Depuis la IBM WebSphere MQ 7.5, vous pouvez utiliser des variables d'environnement dans certaines propriétés Managed File Transfer qui représentent des emplacements de fichier ou de répertoire. Cela permet aux emplacements des fichiers ou des répertoires utilisés lors de l'exécution de parties du produit de varier en fonction des changements d'environnement, tels que l'utilisateur qui exécute le processus. Pour plus d'informations, voir Utilisation des variables d'environnement dans les propriétés MFT.

Concepts associés

Options de configuration de MFT sur Multiplatforms

Référence associée

Utilisation des variables d'environnement dans les propriétés MFT

Connexion à un gestionnaire de files d'attente en mode client avec authentification de canal

IBM MQ utilise des enregistrements d'authentification de canal pour contrôler plus précisément l'accès au niveau d'un canal. Cela signifie que par défaut, les gestionnaires de files d'attente nouvellement créés rejettent les connexions client du composant Managed File Transfer .

Pour plus d'informations sur l'authentification de canal, voir «Enregistrements d'authentification de canal», à la page 54.

Si la configuration de l'authentification de canal pour le SVRCONN utilisé par Managed File Transfer spécifie un ID MCAUSER non privilégié, vous devez accorder des enregistrements de droits d'accès spécifiques pour le gestionnaire de files d'attente, les files d'attente et les rubriques, afin de permettre au Managed File Transfer Agent et aux commandes de fonctionner correctement. Utilisez la commande MQSC SET CHLAUTH ou la commande PCF Set Channel Authentication Record pour créer, modifier ou supprimer des enregistrements d'authentification de canal. Pour tous les agents Managed File Transfer que vous souhaitez connecter au gestionnaire de files d'attente IBM MQ , vous pouvez soit configurer un ID MCAUSER à utiliser pour tous vos agents, soit configurer un ID MCAUSER distinct pour chaque agent.

Accordez à chaque ID MCAUSER les droits suivants:

- Enregistrements de droits d'accès requis pour le gestionnaire de files d'attente:
  - connect
  - setid
  - inq
- Enregistrements de droits d'accès requis pour les files d'attente.

  Pour toutes les files d'attente spécifiques à l'agent, c'est-à-dire les noms de file d'attente se terminant par *nom_agent* dans la liste suivante, vous devez créer ces enregistrements de droits d'accès aux files d'attente pour chaque agent que vous souhaitez connecter au gestionnaire de files d'attente IBM MQ à l'aide d'une connexion client.
  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, browse (SYSTEM.FTE.COMMAND.*nom_agent*)
  - put, get (SYSTEM.FTE.DATA.*nom_agent*)
  - put, get (SYSTEM.FTE.REPLY.*nom_agent*)
  - put, get, inq, browse (SYSTEM.FTE.STATE.*nom_agent*)
  - put, get, browse (SYSTEM.FTE.EVENT.*nom_agent*)

  - put, get (SYSTEM.FTE)
- Enregistrements de droits d'accès requis pour les rubriques:
  - sub, pub (SYSTEM.FTE)
- Enregistrements de droits d'accès requis pour les transferts de fichiers.

  Si vous disposez d'ID MCAUSER distincts pour l'agent source et l'agent de destination, créez les enregistrements de droits d'accès dans les files d'attente des agents à la fois à la source et à la destination.

  Par exemple, si l'ID MCAUSER de l'agent source est **user1** et l'ID MCAUSER de l'agent cible est **user2**, définissez les droits suivants pour les utilisateurs de l'agent:

Utilisateur d'agent	File d'attente	droits requis
user1	SYSTEME SYSTEM.FTE.DATA.*nom_agent_destination*	put
user1	SYSTEME SYSTEM.FTE.COMMAND.*nom_agent_destination*	put
user2	SYSTEME SYSTEM.FTE.REPLY.*nom_agent_source*	put
user2	SYSTEME SYSTEM.FTE.COMMAND.*nom_agent_source*	put

Configuration de SSL ou TLS entre l'agent de pont Connect:Direct et le noeud Connect:Direct

Configurez l'agent de pont Connect:Direct et le noeud Connect:Direct pour qu'ils se connectent via le protocole SSL en créant un magasin de clés et un magasin de clés de confiance et en définissant les propriétés dans le fichier de propriétés de l'agent de pont Connect:Direct .

Pourquoi et quand exécuter cette tâche

Ces étapes incluent des instructions permettant d'obtenir vos clés signées par une autorité de certification. Si vous n'utilisez pas d'autorité de certification, vous pouvez générer un certificat autosigné. Pour plus d'informations sur la génération d'un certificat autosigné, voir «Utilisation de SSL/TLS sous AIX, Linux, and Windows», à la page 302.

Ces étapes incluent des instructions pour la création d'un magasin de clés et d'un magasin de clés de confiance pour l'agent de pont Connect:Direct . Si l'agent de pont Connect:Direct possède déjà un magasin de clés et un magasin de clés de confiance qu'il utilise pour se connecter de manière sécurisée aux gestionnaires de files d'attente IBM MQ , vous pouvez utiliser le magasin de clés et le magasin de clés de confiance existants lors de la connexion sécurisée au noeud Connect:Direct . Pour plus d'informations, voir «Configuration du chiffrement SSL ou TLS pour MFT», à la page 607.

Procédure

Pour le noeud Connect:Direct , procédez comme suit:

1. Générez une clé et un certificat signé pour le noeud Connect:Direct .

   Pour ce faire, utilisez l'outil IBM Key Management fourni avec IBM MQ. Pour plus d'informations, voir «Utilisation de SSL/TLS», à la page 283.
2. Envoyez une demande à une autorité de certification pour que la clé soit signée. Vous recevez un certificat en retour.
3. Créez un fichier texte, par exemple `/test/ssl/certs/CAcert`, qui contient la clé publique de votre autorité de certification.
4. Installez l'option Secure + sur le noeud Connect:Direct .

   Si le noeud existe déjà, vous pouvez installer l'option Secure + en exécutant à nouveau le programme d'installation, en indiquant l'emplacement de l'installation existante et en choisissant d'installer uniquement l'option Secure +.

5. Créez un nouveau fichier texte ; par exemple, `/test/ssl/cd/keyCertFile/`*node_name*`.txt`.
6. Copiez le certificat que vous avez reçu de votre autorité de certification et la clé privée, qui se trouve dans `/test/ssl/cd/privateKeys/`*node_name*`.key`, dans le fichier texte.

   Le contenu de `/test/ssl/cd/keyCertFile/`*node_name*`.txt` doit être au format suivant:

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHQjES
MBAGA1UECBMJSGFtcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xDjAMBgNVBAsTBU1RSVBUMQswCQYDVQQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEwlIYW1wc2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvgP1QIklU9ypSKD1XoODo1yk
EyMFXBOUpZRrDVxjoSEC0vtWNcJ199e+Vc4UpNybDyBu+NkDlMNofX4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnrwChe0MV3kjA84GKH/rOSVqtl984mu/lDyS819XcfSSn
cOOMsK1KbneVSCIV2XECAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0E
HxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFNXMIpSc
csBXUniW4A3UrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj4lVz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7klXa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/vl8+iv01OCL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIiEk9Dlz2U6WO9lICwn
l7PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qHO9ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRUOIQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDWoMNt5fj51v7aPmVeS60bOm+UlGre8B/Ze18JVj2O4K2Uh72rDCXE
5e6eFxSdUM207sQDy20euBVELJtM2kOkL1ROdoQQSlU3XQNgJw/t3ZIx5hPXWEQT
rjRQO64BEhb+PzzxPF8uwzZ9IrUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI8OGvBvWXO4fHyvIX5aslwhBoArXIS1AtNTrptPvoaP1zyIAeZ6OCVo/
SFo+A2UhmtEJeOJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWDyJBVhUOjXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+lYcX8OVZ6
sHFPNlHluCNy/riUcBy9iviVeodX8IomOchSyO5DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uUOAeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM5OoWXa6U5+AYuGUMg
/itPZmUmNrHjTk7ghT6i1IQOaBowXXKJBlMmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lzQ==
-----END RSA PRIVATE KEY-----
```

7. Démarrez l'outil d'administration Secure +.
   - Sur les systèmes AIX and Linux , exécutez la commande **spadmin.sh**.
   - Sur les systèmes Windows , cliquez sur **Démarrer** > **Programmes** > **Sterling Commerce Connect:Direct** > **CD Secure + Admin Tool**

   L'outil d'administration CD Secure + démarre.
8. Dans l'outil d'administration CD Secure +, cliquez deux fois sur **.Ligne** locale pour éditer les paramètres SSL ou TLS principaux.
   a) Sélectionnez **Activer le protocole SSL** ou **Activer le protocole TLS**, selon le protocole que vous utilisez.
   b) Sélectionnez **Désactiver la substitution**.
   c) Sélectionnez au moins une suite de chiffrement.
   d) Si vous souhaitez une authentification bidirectionnelle, remplacez la valeur de **Activer l'authentification client** par `Yes`.
   e) Dans la zone **Certificat racine accrédité** , entrez le chemin d'accès au fichier de certificat public de votre autorité de certification, `/test/ssl/certs/CAcert`.
   f) Dans la zone **Fichier de certificat de clé** , entrez le chemin d'accès au fichier que vous avez créé, `/test/ssl/cd/keyCertFile/`*node_name*`.txt`.
9. Cliquez deux fois sur le **.Ligne** du client pour éditer les paramètres SSL ou TLS principaux.
   a) Sélectionnez **Activer le protocole SSL** ou **Activer le protocole TLS**, selon le protocole que vous utilisez.
   b) Sélectionnez **Désactiver la substitution**.

Pour l'agent de pont Connect:Direct , procédez comme suit:

10. Créez un magasin de clés de confiance. Pour ce faire, vous pouvez créer une clé factice, puis la supprimer.

    Vous pouvez utiliser les commandes suivantes:

    ```
    keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
    ```

    ```
    keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
    ```

11. Importez le certificat public de l'autorité de certification dans le magasin de clés de confiance.

    Vous pouvez utiliser la commande suivante :

    ```
    keytool -import -trustcacerts -alias myCA
            -file /test/ssl/certs/CAcert
            -keystore /test/ssl/fte/stores/truststore.jks
    ```

12. Editez le fichier de propriétés de l'agent de pont Connect:Direct .

    Incluez les lignes suivantes n'importe où dans le fichier:

    ```
    cdNodeProtocol=protocol
    cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
    cdNodeTruststorePassword=password
    ```

    Dans l'exemple de cette étape, *protocol* est le protocole que vous utilisez, SSL ou TLS, et *password* est le mot de passe que vous avez spécifié lors de la création du magasin de clés de confiance.

13. Si vous souhaitez une authentification bidirectionnelle, créez une clé et un certificat pour l'agent de pont Connect:Direct .

    a) Créez un magasin de clés et une clé.

    Vous pouvez utiliser la commande suivante :

    ```
    keytool -genkey -keyalg RSA -alias agent_name
            -keystore /test/ssl/fte/stores/keystore.jks
            -storepass password -validity 365
    ```

    b) Générez une demande de signature.

    Vous pouvez utiliser la commande suivante :

    ```
    keytool -certreq -v -alias agent_name
            -keystore /test/ssl/fte/stores/keystore.jks -storepass password
            -file /test/ssl/fte/requests/agent_name.request
    ```

    c) Importez le certificat que vous avez reçu à l'étape précédente dans le magasin de clés. Le certificat doit être au format x.509 .

    Vous pouvez utiliser la commande suivante :

    ```
    keytool -import -keystore /test/ssl/fte/stores/keystore.jks
            -storepass password -file certificate_file_path
    ```

    d) Editez le fichier de propriétés de l'agent de pont Connect:Direct .

    Incluez les lignes suivantes n'importe où dans le fichier:

    ```
    cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
    cdNodeKeystorePassword=password
    ```

    Dans l'exemple de cette étape, *password* est le mot de passe que vous avez spécifié lors de la création du magasin de clés.

Tâches associées

Configuration du pont Connect:Direct

ALW Sécurisation des clients AMQP

Vous utilisez une série de mécanismes de sécurité pour sécuriser les connexions des clients AMQP et vous assurer que les données sont correctement protégées sur le réseau. Vous pouvez générer la sécurité dans vos applications MQ Light . Vous pouvez également utiliser les fonctions de sécurité existantes d' IBM MQ avec les clients AMQP, de la même manière que les fonctions sont utilisées pour d'autres applications.

Règles d'authentification de canal (CHLAUTH)

Vous pouvez utiliser des règles d'authentification de canal afin de restreindre les connexions TCP à un gestionnaire de files d'attente. Les canaux AMQP prennent en charge l'utilisation de règles d'authentification de canal que vous configurez pour votre gestionnaire de files d'attente. Si des règles d'authentification de canal sont définies avec un profil qui correspond à des canaux AMQP dans votre gestionnaire de files d'attente, elles sont appliquées à ces canaux. Par défaut, l'authentification de canal est activée sur les nouveaux gestionnaires de files d'attente IBM MQ . Vous devez donc effectuer au moins une configuration avant de pouvoir utiliser un canal AMQP.

Pour plus d'informations sur la configuration des règles d'authentification de canal pour autoriser les connexions AMQP à votre gestionnaire de files d'attente, voir Création et utilisation de canaux AMQP.

Authentification de connexion (CONNAUTH)

Vous pouvez utiliser l'authentification de connexion pour authentifier les connexions à un gestionnaire de files d'attente. Les canaux AMQP prennent en charge l'utilisation de l'authentification de connexion pour contrôler l'accès au gestionnaire de files d'attente depuis des applications AMQP.

Le protocole AMQP utilise l'infrastructure SASL (Simple Authentication and Security Layer) pour spécifier la façon dont une connexion est authentifiée. Il existe divers mécanismes SASL et IBM MQ en prend en charge deux : ANONYMOUS et PLAIN.

Dans le cas de ANONYMOUS, aucune donnée d'identification n'est transmise du client au gestionnaire de files d'attente pour l'authentification. Si l'objet IBM MQ AUTHINFO spécifié dans l'attribut **CONNAUTH** du gestionnaire de files d'attente a la valeur **CHCKCLNT** REQUIRED ou REQDADM (en cas de connexion en tant qu'administrateur), la connexion est refusée. Si la valeur de **CHCKCLNT** est NONE ou OPTIONAL, la connexion est acceptée.

Dans le cas de PLAIN, un nom d'utilisateur et un mot de passe sont transmis du client au gestionnaire de files d'attente pour l'authentification. Si l'objet IBM MQ AUTHINFO spécifié dans l'attribut **CONNAUTH** du gestionnaire de files d'attente a la valeur **CHCKCLNT** NONE, la connexion est refusée. Si la valeur de **CHCKCLNT** est OPTIONAL, REQUIREDou REQDADM (en cas de connexion en tant qu'administrateur), le nom d'utilisateur et le mot de passe sont vérifiés par le gestionnaire de files d'attente. Le gestionnaire de files d'attente vérifie le système d'exploitation (si l'objet AUTHINFO est de type IDPWOS) ou un référentiel LDAP (si l'objet AUTHINFO est de type IDPWLDAP).

Le tableau suivant présente un récapitulatif de ce comportement d'authentification :

Tableau 101. Récapitulatif des mécanismes SASL et de l'authentification de connexion		
Mécanisme SASL	**Données d'identification transmises du client au gestionnaire de files d'attente ?**	**Valeur CHKCLNT**
ANONYMOUS	Non	REQUIRED ou REQDADM - connexion refusée NONE ou OPTIONAL - connexion acceptée
PLAIN	Oui, nom d'utilisateur et mot de passe	REQUIRED, REQDADM ou OPTIONAL - nom d'utilisateur et mot de passe vérifiés par le gestionnaire de files d'attente NONE - connexion refusée

Si vous utilisez un client MQ Light, vous pouvez spécifier des données d'identification en les incluant dans l'adresse AMQP à laquelle vous vous connectez, par exemple :

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

Paramètre MCAUSER sur un canal

Les canaux AMQP possèdent un attribut MCAUSER que vous pouvez utiliser pour définir l'ID utilisateur IBM MQ sous lequel toutes les connexions à un canal particulier sont autorisées. Toutes les connexions depuis des clients AMQP à ce canal adoptent l'ID MCAUSER que vous avez configuré. Cet ID utilisateur est employé pour l'autorisation de la messagerie dans différentes rubriques.

Il est recommandé d'utiliser l'authentification de canal (CHLAUTH) pour sécuriser les connexions aux gestionnaires de files d'attente. Si vous utilisez l'authentification de canal, il est recommandé de configurer un utilisateur non privilégié comme valeur pour MCAUSER. Ainsi, si une connexion à un canal n'est pas mise en correspondance par une règle CHLAUTH, elle ne sera pas autorisée à effectuer des opérations de messagerie dans le gestionnaire de files d'attente.

prise en charge de SSL/TLS

Les canaux AMQP prennent en charge le chiffrement SSL/TLS à l'aide de clés provenant du référentiel de clés configuré pour votre gestionnaire de files d'attente. Les options de configuration de canal AMQP pour le chiffrement SSL/TLS prennent en charge les mêmes options que les autres types de canal MQ ; vous pouvez préciser une spécification de chiffrement et indiquer si le gestionnaire de files d'attente requiert des certificats des connexions client AMQP.

A l'aide des attributs FIPS du gestionnaire de files d'attente, vous pouvez contrôler les suites de chiffrement SSL/TLS que vous pouvez utiliser pour sécuriser les connexions depuis les clients AMQP.

Pour plus d'informations sur la configuration d'un référentiel de clés pour le gestionnaire de files d'attente, voir «Utilisation de SSL/TLS sous AIX, Linux, and Windows», à la page 302.

Pour plus d'informations sur la configuration de la prise en charge de SSL/TLS pour une connexion client AMQP, voir Création et utilisation de canaux AMQP.

V 9.4.0 V 9.4.0 Depuis la IBM MQ 9.4.0, le canal AMQP ne prend plus en charge les référentiels de clés CMS sur le gestionnaire de files d'attente. Vous pouvez utiliser la commande **runmqakm** pour convertir un référentiel de clés CMS au format PKCS #12 , qui est pris en charge. Par exemple, vous pouvez utiliser la commande suivante pour convertir un référentiel de clés nommé

`sslTest.kdb` du format CMS au format PKCS #12 . Le nouveau référentiel de clés est nommé `sslTest.p12`et protégé par le mot de passe `passw0rd`.

```
runmqakm -keydb -convert -type cms -db sslTest.kdb -stashed -new_format pkcs12 -target
sslTest.p12 -new_pw passw0rd
```

Service JAAS (Java Authentication and Authorization Service) (JAAS)

Si vous le souhaitez, vous pouvez configurer des canaux AMQP avec un module de connexion JAAS qui peut vérifier le nom d'utilisateur et le mot de passe indiqués par un client AMQP. Voir «Configuration de JAAS pour les canaux AMQP», à la page 616.

Tâches associées

Développement d'applications client AMQP

Création et utilisation de canaux AMQP

ALW Restriction de la reprise du client AMQP

Lorsqu'une connexion client AMQP dont l'identificateur de client est identique à celui d'une connexion client AMQP existante est établie, la connexion client existante est déconnectée par défaut. Toutefois, vous pouvez configurer le gestionnaire de files d'attente afin de restreindre le comportement de reprise de client pour que la reprise ne soit possible que lorsque certains critères sont satisfaits.

Par exemple, la déconnexion de la connexion client existante peut ne pas être appropriée si des applications AMQP sont développées par différentes équipes et qu'elles utilisent le même ID de client. Pour résoudre ce problème, vous pouvez restreindre la reprise de client reposant sur le nom du canal AMQP utilisé, l'adresse IP du client et l'ID utilisateur du client (lorsque l'authentification SASL est activée).

Utilisez les paramètres des attributs de gestionnaire de files d'attente **AdoptNewMCA** et **AdoptNewMCACheck** pour spécifier le niveau requis de la restriction de reprise du client, comme indiqué dans le tableau suivant:

*Tableau 102. Paramètres **AdoptNewMCA** et **AdoptNewMCACheck** pour restreindre la reprise du client*

AdoptNewMCA	**AdoptNewMCACheck**	**Critères vérifiés avant l'autorisation de la reprise de client**
NO ou indéfini	Non applicable	Néant. La reprise de client est autorisée pour toutes les connexions client qui sont authentifiées et qui satisfont toutes les règles CHLAUTH.
ALL (ou une valeur autre que NO)	QM ou indéfini	Néant. La reprise de client est autorisée pour toutes les connexions client qui sont authentifiées et qui satisfont toutes les règles CHLAUTH.
ALL (ou une valeur autre que NO)	NOM	ID utilisateur (lorsque SASL est activé) Nom du canal
ALL (ou une valeur autre que NO)	ADDRESS	ID utilisateur (lorsque SASL est activé) Adresse IP

*Tableau 102. Paramètres **AdoptNewMCA** et **AdoptNewMCACheck** pour restreindre la reprise du client (suite)*		
AdoptNewMCA	**AdoptNewMCACheck**	**Critères vérifiés avant l'autorisation de la reprise de client**
ALL (ou une valeur autre que NO)	TOUT	ID utilisateur (lorsque SASL est activé) Nom du canal Adresse IP

Les attributs de gestionnaire de files d'attente **AdoptNewMCA** et **AdoptNewMCACheck** font partie de la configuration de gestionnaire de files d'attente, qui est définie dans la strophe CHANNELS. Sur les systèmes IBM MQ for Windows et IBM MQ for Linux x86-64 , modifiez les informations de configuration à l'aide du IBM MQ Explorer. Sur les autres systèmes, modifiez les informations en éditant le fichier de configuration `qm.ini`. Pour des informations sur la modification des informations relatives aux canaux de gestionnaire de files d'attente, voir la rubrique relative aux attributs des canaux.

Tâches associées

Développement d'applications client AMQP

Création et utilisation de canaux AMQP

ALW

Configuration de JAAS pour les canaux AMQP

Les modules personnalisés du service d'authentification et d'autorisation Java (JAAS) peuvent être utilisés pour authentifier les données d'identification par nom d'utilisateur et mot de passe transmises à un canal AMQP par un client AMQP lorsqu'il se connecte.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser un module JAAS personnalisé si vous utilisez déjà des modules JAAS pour l'authentification dans d'autres systèmes Javaet que vous souhaitez réutiliser ces modules pour l'authentification des connexions AMQP à MQ. Vous pouvez aussi écrire un modèle JAAS personnalisé si les fonctions d'authentification intégrées à MQ ne prennent pas en charge le mécanisme d'authentification que vous utilisez.

La configuration de modules JAAS pour des canaux AMQP est effectuée au niveau du gestionnaire de files d'attente. Cela signifie que si vous configurez un module JAAS pour l'authentification des connexions AMQP au gestionnaire de files d'attente, le module est appliqué à tous les canaux AMQP. Le nom du canal qui a appelé le module JAAS est transmis au module, ce qui vous permet de coder des comportements de connexion JAAS différents pour des canaux différents.

D'autres informations sont également transmises au module JAAS :

- L'ID du client AMQP qui tente de s'authentifier.
- L'adresse réseau du client AMQP.
- Le nom du canal qui a appelé le module JAAS.

Procédure

Pour configurer un module de configuration JAAS pour les canaux AMQP, procédez comme suit :

1. Définissez un fichier `jaas.config` contenant une ou plusieurs strophes de configuration de module JAAS. La section doit spécifier le nom qualifié complet de la classe Java qui implémente l'interface JAAS `javax.security.auth.spi.LoginModule` .
   - Un fichier `jaas.config` par défaut est fourni avec le produit et se trouve dans *QM_data_directory*/amqp/jaas.config.

- Une strophe préconfigurée nommée `MQXRConfig` est déjà définie dans le fichier `jaas.config` par défaut.

2. Spécifiez le nom de la strophe à utiliser pour les canaux AMQP.
   - Linux AIX Ajoutez une propriété au fichier `amqp_unix.properties`.
   - Windows Ajoutez une propriété au fichier `amqp_win.properties`.

   Le format de la propriété est le suivant :

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   Par exemple :

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```

3. Configurez l'environnement de gestionnaire de files d'attente afin d'inclure la classe du module personnalisé. Le service AMQP doit avoir accès à la classe Java configurée dans la section de configuration JAAS .

   Pour ce faire, ajoutez le chemin d'accès à la classe JAAS dans le fichier `service.env` de MQ. Editez le fichier `service.env` dans le répertoire de configuration MQ (*MQ_config_directory*) ou dans le répertoire de configuration du gestionnaire de files d'attente (*QM_config_directory*) pour définir la variable CLASSPATH sur l'emplacement de la classe de module JAAS .

Que faire ensuite

Un exemple de module de connexion JAAS est fourni avec le produit dans le répertoire *mq_installation_directory*`/amqp/samples` . Il authentifie toutes les connexions client, quel que soit le nom d'utilisateur ou le mot de passe avec lequel le client se connecte.

Vous pouvez modifier le code source de l'exemple et le recompiler pour tenter de n'authentifier que des utilisateurs spécifiques qui se servent d'un mot de passe particulier. Pour configurer le canal AMQP sur un système UNIX afin d'utiliser l'exemple de module de connexion JAAS livré avec le produit :

1. Editez le fichier `/var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties` et définissez la propriété `com.ibm.mq.MQXR.JAASConfig=MQXRConfig`.
2. Editez le fichier /var/mqm/service.env et définissez la propriété CLASSPATH=*mq_installation_location*`/amqp/samples`

Le fichier `jaas.config` contient déjà une strophe nommée `MQXRConfig` qui spécifie l'exemple de classe `samples.JAASLoginModule` comme classe de module de connexion. Il n'est pas nécessaire de modifier le fichier `jaas.config` avant d'utiliser l'exemple de module.

Tâches associées

Développement d'applications client AMQP

Création et utilisation de canaux AMQP

Advanced Message Security

Advanced Message Security (AMS) est un composant de IBM MQ qui offre un niveau de protection élevé pour les données sensibles transitant par le réseau IBM MQ , sans affecter les applications finales.

Présentation des Advanced Message Security

Les applications IBM MQ peuvent utiliser Advanced Message Security pour envoyer des données sensibles, telles que des transactions financières à valeur élevée et des informations personnelles, avec différents niveaux de protection à l'aide d'un modèle de cryptographie à clé publique.

Concepts associés

«Interception MCA (Message Channel Agent) et AMS», à la page 671

L'interception MCA permet à un gestionnaire de files d'attente s'exécutant sous IBM MQ d'activer de manière sélective les règles à appliquer pour les canaux de connexion serveur.

Référence associée

Codes retour GSKit utilisés dans les messages AMS

Caractéristiques et fonctions de Advanced Message Security

Advanced Message Security développe les services de sécurité IBM MQ pour fournir la signature et le chiffrement des données au niveau des messages. Les services étendus garantissent que les données de message n'ont pas été modifiées entre le moment où elles sont placées à l'origine dans une file d'attente et le moment où elles sont extraites. En outre, AMS vérifie qu'un expéditeur de données de message est autorisé à placer des messages signés dans une file d'attente cible.

AMS fournit les fonctions suivantes:

- Sécurise les transactions sensibles ou à valeur élevée traitées par IBM MQ.
- Détecte et supprime les messages malveillants ou non autorisés avant qu'ils ne soient traités par une application de réception.
- Vérifie que les messages n'ont pas été modifiés lors de leur passage de la file d'attente à la file d'attente.
- Protège les données non seulement lorsqu'elles circulent sur le réseau, mais également lorsqu'elles sont placées dans une file d'attente.
- Sécurise les applications propriétaires et écrites par le client existantes pour IBM MQ.
- z/OS Depuis la IBM MQ 9.1.3, IBM MQ for z/OS offre la possibilité de supprimer et d'ajouter éventuellement une protection AMS à partir ou vers des messages qui transitent sur le réseau, respectivement. Il s'agit de l'exception *Server to Server Message Channel Agent (MCA) Interception.*.
- ALW Depuis IBM MQ 9.1.4 et IBM MQ 9.1.0 Fix Pack 4, une vérification est ajoutée au code de la bibliothèque IBM MQ qui s'exécute dans le programme d'application du client. La vérification s'exécute au début de son initialisation pour lire la valeur de la variable d'environnement *AMQ_AMS_FIPS_OFF* et, si elle est définie sur une valeur, le code IBM Global Security Kit (GSKit) est exécuté en mode non FIPS dans cette application.

Qualités de protection disponibles avec AMS

Il existe trois qualités de protection pour Advanced Message Security, `Integrity`, `Privacy`et `Confidentiality`.

La protection `Integrity` est assurée par la signature numérique, qui permet de savoir qui a créé le message et que le message n'a pas été modifié ou altéré.

La protection `Privacy` est assurée par une combinaison de signature numérique et de chiffrement. Le chiffrement garantit que les données de message ne sont visibles que par le ou les destinataires prévus. Même si des destinataires non autorisés obtiennent une copie des données de message chiffrées, ils ne peuvent pas afficher eux-mêmes les données de message réelles.

La protection `Confidentiality` est fournie par le chiffrement uniquement avec une réutilisation de clé facultative.

Effet sur les performances

AMS utilise une combinaison de routines cryptographiques symétriques et asymétriques pour fournir la signature numérique et le chiffrement. Etant donné que les opérations de clé symétrique sont très rapides par rapport aux opérations de clé asymétrique, qui consomment beaucoup d'UC, cela peut avoir un impact significatif sur les coûts de protection d'un grand nombre de messages avec AMS.

Routines cryptographiques asymétriques

Par exemple, lors de l'insertion d'un message signé, le hachage de message est signé à l'aide d'une opération de clé asymétrique.

Lors de l'obtention d'un message signé, une autre opération de clé asymétrique est utilisée pour vérifier le hachage signé.

Par conséquent, un minimum de deux opérations de clé asymétrique est requis par message pour signer et vérifier les données de message.

Routines cryptographiques asymétriques et symétriques

Lors de l'insertion d'un message chiffré, une clé symétrique est générée, puis chiffrée à l'aide d'une opération de clé asymétrique pour chaque destinataire prévu du message.

Les données de message sont ensuite chiffrées avec la clé symétrique. Lors de l'obtention du message chiffré, le destinataire prévu doit utiliser une opération de clé asymétrique pour reconnaître la clé symétrique utilisée pour le message.

Par conséquent, les trois qualités de protection contiennent des éléments différents des opérations de clés asymétriques à forte intensité d'UC, ce qui aura un impact significatif sur le débit de messagerie maximal réalisable pour les applications qui envoient et reçoivent des messages.

Toutefois, les règles `Confidentiality` permettent la réutilisation de clés symétriques sur une séquence de messages. Grâce à la réutilisation de clé symétrique, les politiques `Confidentiality` permettent des économies de coût importantes de l'UC. Ce mode de fonctionnement continue d'utiliser le format PKCS#7 pour partager une clé de chiffrement symétrique. Toutefois, il n'existe pas de signature numérique, ce qui élimine certaines des opérations de clé asymétrique par message. La clé symétrique doit quand même être chiffrée avec des opérations de clé asymétrique pour chaque destinataire, mais la clé symétrique peut éventuellement être réutilisée dans plusieurs messages destinés aux mêmes destinataires. Si la réutilisation de clé est autorisée par la politique, seul le premier message requiert des opérations de clé asymétrique. Les messages suivants doivent utiliser uniquement des opérations de clé symétrique.

Réutilisation de clé

Avec les règles `Confidentiality` , vous pouvez utiliser l'approche de réutilisation de clé symétrique pour réduire de manière significative les coûts liés au chiffrement d'un certain nombre de messages placés dans la même file d'attente et destinés au ou aux mêmes destinataires.

Par exemple, lors de l'insertion de 10 messages chiffrés dans le même ensemble de destinataires, une clé symétrique est générée, puis chiffrée pour le premier message, à l'aide d'une opération de clé asymétrique pour chaque destinataire prévu du message.

En fonction des limites contrôlées par la règle, la clé symétrique chiffrée peut ensuite être réutilisée par les messages ultérieurs destinés aux mêmes destinataires. Pour que la clé symétrique puisse être réutilisée par les messages suivants, l'application doit conserver la file d'attente ouverte après avoir placé un message dans la file d'attente. La clé symétrique ne peut pas être réutilisée par les opérations MQPUT1 . Une application qui obtient des messages chiffrés peut appliquer la même optimisation, dans la mesure où l'application peut détecter lorsqu'une clé symétrique n'a pas été modifiée et éviter les frais liés à l'extraction de la clé symétrique.

Dans cet exemple, 90% des opérations de clé asymétrique peuvent être évitées par les applications d'insertion et d'obtention en réutilisant la même clé.

Pour plus d'informations sur l'utilisation de la réutilisation des clés, voir:

- Commande MQSC SET POLICY
- Commande de contrôle setmqspl
- IBM i IBM i commande SETMQMSPL

Concepts clés dans AMS

Découvrez les concepts clés de Advanced Message Security pour comprendre comment l'outil fonctionne et comment le gérer efficacement.

Infrastructure à clés publiques et Advanced Message Security

L'infrastructure à clé publique (ICP) est un système d'installations, de politiques et de services qui appuient l'utilisation de la cryptographie à clé publique pour obtenir des communications sécurisées.

Il n'existe pas de norme unique qui définit les composants d'une infrastructure à clé publique, mais une ICP implique généralement l'utilisation de certificats de clé publique et comprend des autorités de certification (CA) et d'autres autorités d'enregistrement (RA) qui fournissent les services suivants:

- Emission de certificats numériques
- Validation des certificats numériques
- Révocation de certificats numériques
- Distribution de certificats

L'identité des utilisateurs et des applications est représentée par la zone **Nom distinctif (DN)** dans un certificat associé à des messages signés ou chiffrés. Advanced Message Security utilise cette identité pour représenter un utilisateur ou une application. Pour authentifier cette identité, l'utilisateur ou l'application doit avoir accès au magasin de clés dans lequel le certificat et la clé privée associée sont stockés. Chaque certificat est représenté par un libellé dans le magasin de clés.

Concepts associés

«Utilisation de magasins de clés et de certificats avec AMS», à la page 664
Pour fournir une protection cryptographique transparente aux applications IBM MQ , Advanced Message Security utilise le fichier de clés, dans lequel sont stockés les certificats de clé publique et une clé privée. Sous z/OS, un fichier de clés SAF est utilisé à la place d'un fichier de clés.

Certificats numériques dans AMS

Advanced Message Security associe les utilisateurs et les applications à des certificats numériques standard X.509 . Les certificats X.509 sont généralement signés par une autorité de certification de confiance et impliquent des clés privées et publiques qui sont utilisées pour le chiffrement et le déchiffrement.

Les certificats numériques offrent une protection contre l'usurpation d'identité en liant une clé publique à son propriétaire, qu'il s'agisse d'un individu, d'un gestionnaire de files d'attente ou d'une autre entité. Les certificats numériques sont également appelés certificats de clé publique, car ils vous donnent l'assurance de la propriété d'une clé publique lorsque vous utilisez un schéma de clé asymétrique. Ce schéma requiert la génération d'une clé publique et d'une clé privée pour une application. Les données chiffrées à l'aide de la clé publique ne peuvent être déchiffrées qu'à l'aide de la clé privée correspondante, tandis que les données chiffrées à l'aide de la clé privée ne peuvent être déchiffrées qu'à l'aide de la clé publique correspondante. La clé privée est stockée dans un fichier de base de données de clés protégé par mot de passe. Seul son propriétaire a accès à la clé privée utilisée pour déchiffrer les messages qui sont chiffrés à l'aide de la clé publique correspondante.

Si les clés publiques sont envoyées directement par leur propriétaire à une autre entité, il existe un risque que le message soit intercepté et que la clé publique soit remplacée par une autre. C'est ce qu'on appelle une attaque de type "man-in-the-middle". La solution consiste à échanger des clés publiques par l'intermédiaire d'un tiers de confiance, ce qui permet à l'utilisateur de s'assurer que la clé publique appartient à l'entité avec laquelle vous communiquez. Au lieu d'envoyer votre clé publique directement, vous demandez à un tiers de confiance de l'incorporer dans un certificat numérique. Le tiers de confiance qui émet des certificats numériques est appelé une autorité de certification (CA).

Pour plus d'informations sur les certificats numériques, voir Qu'est-ce qu'un certificat numérique?.

Un certificat numérique contient la clé publique d'une entité et indique que la clé publique appartient à cette entité:

- lorsqu'un certificat est destiné à une entité individuelle, il est appelé *certificat personnel* ou *certificat d'utilisateur*.
- lorsqu'un certificat est destiné à une autorité de certification, le certificat est appelé *certificat de l'autorité de certification* ou *certificat de signataire*.

Remarque : Advanced Message Security prend en charge les certificats autosignés dans les applications Java et natives

Concepts associés

«Cryptographie», à la page 11
La cryptographie est le processus de conversion entre du texte lisible, appelé *texte en clair*, et un format illisible, appelé *texte chiffré*.

Multi *Gestionnaire des droits d'accès aux objets et AMS*

Sur Multiplatforms, Object Authority Manager (OAM) est le composant de service d'autorisation fourni avec les produits IBM MQ .

L'accès aux entités Advanced Message Security est contrôlé par les groupes d'utilisateurs IBM MQ et la méthode d'accès aux objets (OAM). Les administrateurs peuvent utiliser l'interface de ligne de commande pour accorder ou révoquer des autorisations selon les besoins. Différents groupes d'utilisateurs peuvent avoir différents types de droits d'accès aux mêmes objets. Par exemple, un groupe peut effectuer à la fois des opérations PUT et GET pour une file d'attente spécifique, tandis qu'un autre groupe peut être autorisé uniquement à parcourir la file d'attente. De même, certains groupes peuvent disposer des droits GET et PUT sur une file d'attente, mais ils ne sont pas autorisés à modifier ou à supprimer la file d'attente.

Grâce à la méthode d'accès aux objets (OAM), vous pouvez contrôler:

- Accès aux objets Advanced Message Security via l'interface MQI (Message Queue Interface). Lorsqu'un programme d'application tente d'accéder à des objets, la méthode d'accès aux objets (OAM) vérifie si le profil utilisateur à l'origine de la demande possède l'autorisation pour l'opération demandée. Cela signifie que les files d'attente et les messages des files d'attente peuvent être protégés contre tout accès non autorisé.
- Droit d'utilisation des commandes PCF et MQSC.

Concepts associés

gestionnaire des droits d'accès aux objets

Présentation de l'interface de file d'attente de messages

Technologie prise en charge par Advanced Message Security

Advanced Message Security dépend de plusieurs composants technologiques pour fournir une infrastructure de sécurité.

Advanced Message Security prend en charge les interfaces de programme d'application (API) IBM MQ suivantes:

- interface de file d'attente de messages (MQI)
- IBM MQ Java Message Service (JMS) 1.0.2 et 1.1.
- IBM MQ Classes de base pour Java
- Classes IBM MQ pour. Net en mode non géré

Remarque : Advanced Message Security prend en charge les autorités de certification conformes à X.509 .

Limitations connues de AMS

Un certain nombre d'options IBM MQ ne sont pas prises en charge ou sont soumises à des limitations pour Advanced Message Security.

- Les options IBM MQ suivantes ne sont pas prises en charge ou sont soumises à des limitations:

  Publication/abonnement

  L'un des principaux avantages d'un modèle de messagerie de publication / abonnement sur un point à point est que les applications d'envoi et de réception n'ont pas besoin de se connaître les unes les autres pour que les données soient envoyées et reçues. Cet avantage est annulé par l'utilisation de règles Advanced Message Security qui doivent définir des destinataires ou des signataires autorisés. Il est possible pour une application de publier dans une rubrique via une définition de file d'attente

d'alias qui est protégée par une règle. Il est également possible pour une application d'abonnement d'obtenir des messages à partir d'une file d'attente protégée par une règle. Il n'est pas possible d'affecter une règle directement à une chaîne de rubrique. Les règles ne peuvent être affectées qu'à des définitions de file d'attente.

Conversion de données de canal
Le contenu protégé d'un message protégé Advanced Message Security est transmis au format binaire, ce qui garantit que la conversion des données sur un canal entre les applications n'invalide pas le résumé du message. Les applications qui extraient des messages d'une file d'attente protégée par des règles doivent demander la conversion de données, la conversion du contenu protégé sera tentée une fois les messages vérifiés et non protégés.

Listes de diffusion
Les règles Advanced Message Security peuvent être utilisées lors de la protection des applications qui placent des messages dans des listes de distribution, à condition qu'une règle identique soit définie pour chaque file d'attente de destination de la liste. Si des règles incohérentes sont identifiées lorsqu'une application ouvre une liste de distribution, l'opération d'ouverture échoue et une erreur de sécurité est renvoyée à l'application.

Segmentation des messages d'application
La taille des messages protégés par des règles augmente et les applications ne peuvent pas spécifier avec précision les limites de segment d'un message.

Applications utilisant IBM MQ classes for .NET en mode géré (connexions client)
Les applications utilisant IBM MQ classes for .NET en mode géré (connexions client) ne sont pas prises en charge.

Remarque : L'interception MCA peut être utilisée pour permettre aux clients non pris en charge d'utiliser AMS.

Client Message Service pour les applications .NET (XMS) en mode géré
Les applications du client Message Service pour .NET (XMS) en mode géré ne sont pas prises en charge.

Remarque : L'interception MCA peut être utilisée pour permettre aux clients non pris en charge d'utiliser AMS.

Files d'attente IBM MQ traitées par le pont IMS
Les files d'attente IBM MQ traitées par le pont IMS ne sont pas prises en charge.

Remarque : AMS est pris en charge sur les files d'attente de pont CICS . Vous devez utiliser le même ID utilisateur pour MQPUT (chiffrement) et MQGET (déchiffrement) sur les files d'attente de pont CICS .

Insertion dans la méthode d'accès get en attente
L'opération put to waiting getter n'est pas prise en charge pour les applications getter sur les files d'attente pour lesquelles des règles AMS sont définies.

z/OS **Interception MCA de serveur à serveur**
Depuis la IBM MQ for z/OS 9.1.3, l'interception MCA de serveur à serveur n'est prise en charge que pour les types de canal émetteur, serveur, récepteur et demandeur.

- Les utilisateurs doivent éviter de placer plusieurs certificats avec le même nom distinctif dans un même fichier de clés, car le choix du certificat à utiliser lors de la protection d'un message n'est pas défini.
- AMS n'est pas pris en charge dans JMS si la propriété **WMQ_PROVIDER_VERSION** est définie sur 6.
- L'intercepteur AMS n'est pas pris en charge pour les canaux AMQP ou MQTT.

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.

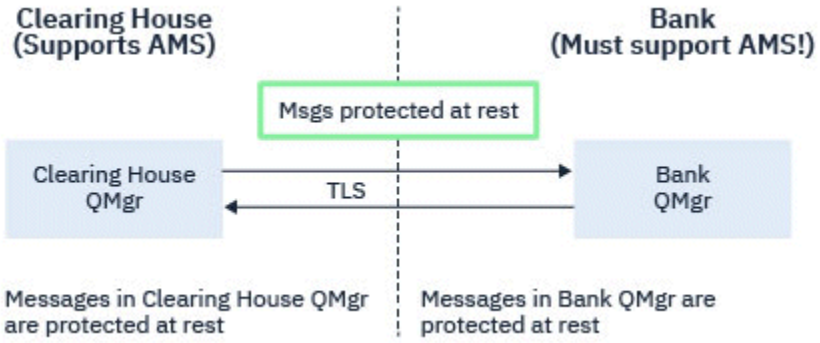


Figure 32. Usage of AMS without AMS interception

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.

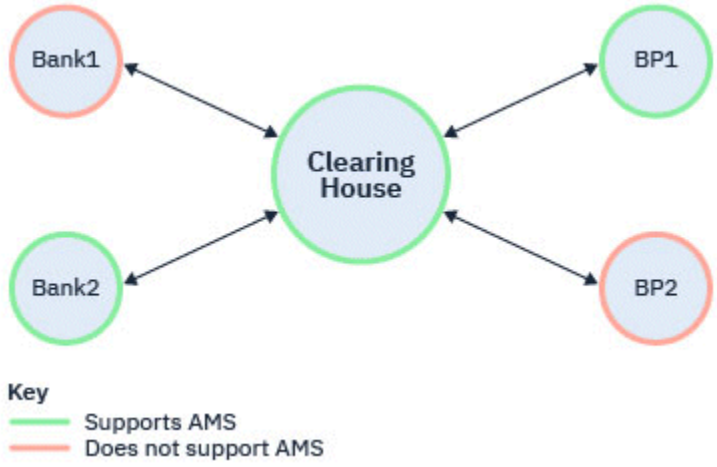


Figure 33. Some partners support AMS and some do not

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

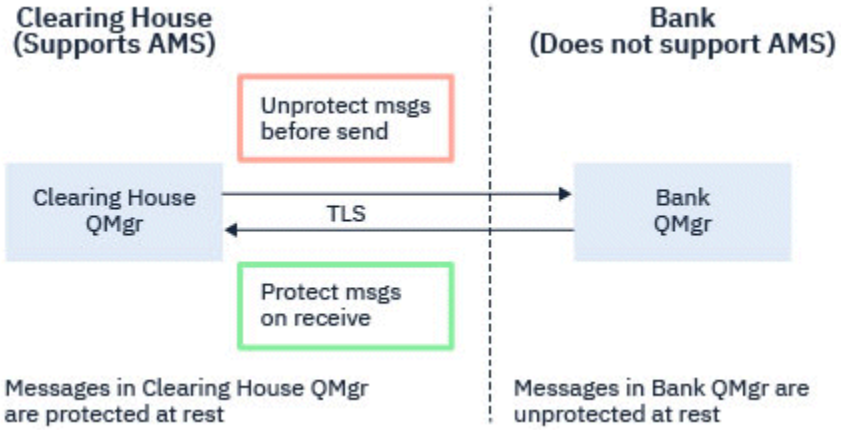


Figure 34. Message flow between business partners

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
: Transmission sans modification de tout message envoyé ou reçu par l'agent MCA pour ce canal.

  Cette valeur est valide pour les canaux avec un type de canal (**CHLTYPE**) de SDR, SVR, RCR ou RQSTR, et est la valeur par défaut.

REMOVE
: Retrait de toute protection AMS dans les messages extraits de la file d'attente de transmission par l'agent MCA et envoi des messages au partenaire.

  Lorsque l'agent MCA obtient un message de la file d'attente de transmission, si une stratégie AMS est définie pour la file d'attente de transmission, elle est appliquée afin de retirer toute protection AMS dans le message avant son envoi via le canal. Si aucune stratégie AMS n'est définie pour la file d'attente de transmission, le message est envoyé tel quel.

  Cette valeur est valide uniquement pour les canaux dont le type est SDR ou SVR.

ASPOLICY
: En fonction de la stratégie définie pour la file d'attente cible, application de la protection AMS aux messages entrants avant leur placement dans la file d'attente cible.

  Lorsque l'agent MCA reçoit un message entrant, si une stratégie AMS est définie pour la file d'attente cible, la protection AMS est appliquée au message avant son placement dans la file d'attente cible. Si

une règle AMS n'est pas définie pour la file d'attente cible, le message est placé dans la file d'attente cible comme c'est le cas.

Cette valeur est valide uniquement pour les canaux dont le type est RCVR ou RQSTR.

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

Traitement des erreurs pour AMS

IBM MQ Advanced Message Security définit une file d'attente de traitement des erreurs pour gérer les messages contenant des erreurs ou des messages qui ne peuvent pas être protégés.

Les messages défectueux sont traités comme des cas exceptionnels. Si un message reçu ne répond pas aux exigences de sécurité de la file d'attente dans laquelle il se trouve, par exemple, si le message est signé alors qu'il doit être chiffré, ou si le déchiffrement ou la vérification de la signature échoue, le message est envoyé à la file d'attente de traitement des erreurs. Un message peut être envoyé à la file d'attente de traitement des erreurs pour les raisons suivantes:

- Non-concordance de la qualité de protection-Il existe une non-concordance de la qualité de protection (QOP) entre le message reçu et la définition QOP dans la règle de sécurité.
- Erreur de déchiffrement-le message ne peut pas être déchiffré.
- Erreur d'en-tête PDMQ-L'en-tête de message Advanced Message Security (AMS) est inaccessible.
- Non-concordance de taille-la longueur d'un message après déchiffrement est différente de celle attendue.
- Non-concordance de la force de l'algorithme de chiffrement-l'algorithme de chiffrement du message est plus faible que requis.
- Erreur inconnue-une erreur inattendue s'est produite.

AMS utilise SYSTEM.PROTECTION.ERROR.QUEUE comme file d'attente de traitement des erreurs. Tous les messages insérés par IBM MQ AMS dans SYSTEM.PROTECTION.ERROR.QUEUE sont précédées d'un en-tête MQDLH.

Votre administrateur IBM MQ peut également définir le système SYSTEM.PROTECTION.ERROR.QUEUE en tant que file d'attente alias pointant vers une autre file d'attente.

z/OS Sous IBM MQ for z/OS, si l'interception de l'agent MCA (Message Channel Agent) de serveur à serveur est en cours d'utilisation:

- Si, pour l'une des raisons précédemment indiquées, IBM MQ AMS déplace les messages de la file d'attente de transmission vers la file d'attente de traitement des erreurs, l'agent MCA émetteur traite simplement le prochain message disponible dans la file d'attente de transmission.
- En général, les règles de canal existantes s'appliquent pour:
  - Insertion de messages dans la file d'attente des messages non livrés, et
  - Les actions effectuées si les insertions dans la file d'attente des messages non livrés doivent échouer.

Pour plus d'informations sur des scénarios spécifiques, voir «Messages non distribués pour AMS sous z/OS», à la page 626 .

z/OS *Messages non distribués pour AMS sous z/OS*

Scénarios spécifiques liés à l'interception de l'agent de canal de message serveur à serveur sur IBM MQ for z/OS.

Sous IBM MQ for z/OS, si l'interception de l'agent MCA (Message Channel Agent) de serveur à serveur est en cours d'utilisation:

- Si, après avoir reçu et protégé un message, l'agent MCA émetteur ne parvient pas à distribuer un message pour une raison quelconque, par exemple parce que le message est trop volumineux pour le canal, si l'attribut de canal émetteur USEDLQ est défini sur YES, l'agent MCA émetteur déplace le message vers la file d'attente des messages non livrés (DLQ) locale.

  Si SYSTEM.DEAD.LETTER.QUEUE est utilisée en tant que file d'attente DLQ locale, le message est placé sans protection.

  Remarque : IBM MQ AMS ne prend pas en charge la protection des messages insérés dans les files d'attente du système.

  Si une file d'attente DLQ nommée est utilisée comme file d'attente DLQ locale, le message sera placé protégé si vous avez défini une stratégie IBM MQ AMS portant le même nom que la file d'attente DLQ nommée et non protégée si vous n'avez pas défini de stratégie appropriée.
- Si un message ne peut pas être inséré dans le DLQ local pour une raison quelconque, si NPMSPEED du canal est défini sur NORMAL ou que le message est un message persistant, le lot de messages en cours est annulé et le canal passe à l'état RETRY. Sinon, le message est supprimé et l'agent MCA émetteur continue à traiter le message suivant dans la file d'attente de transmission.
- Etant donné que les règles de sécurité n'ont aucun effet sur SYSTEM.DEAD.LETTER.QUEUE, ou les autres files d'attente SYSTEM répertoriées dans «Protection des files d'attente système dans AMS», à la page 702, si SYSTEM.DEAD.LETTER.QUEUE est en cours d'utilisation, les messages insérés dans cette file d'attente par les MCM sont placés en l'état. Autrement dit, si des messages ont déjà été protégés, ils sont placés protégés ; dans le cas contraire, ils sont placés non protégés.

  Si l'attribut DEADQ du gestionnaire de files d'attente a été défini sur le nom d'une autre file d'attente de rebut (non système) et qu'il n'existe pas de règle AMS portant le même nom, les messages placés dans cette file d'attente par les agents MCA sont placés en l'état. Autrement dit, si des messages ont déjà été protégés, ils sont placés protégés ; dans le cas contraire, ils sont placés non protégés.

  Si l'attribut DEADQ du gestionnaire de files d'attente a été défini sur le nom d'une autre file d'attente de rebut (non système) et qu'il existe une règle AMS portant le même nom que la file d'attente des messages non livrés, cette règle est utilisée pour protéger les messages insérés dans cette file d'attente par les agents MCA. Si le message a déjà été protégé, il n'est pas protégé à nouveau ; cela permet

d'éviter une double protection. S'il n'existe pas de règle AMS portant le même nom, les messages sont placés en tant que tels.

- S'il existe une règle pour le DLQ avec l'option de tolérance dans la commande setmqspl définie sur off, c'est-à-dire'**-t** O', l'insertion dans le DLQ échoue si le message n'est pas AMS protégé et qu'il n'a donc pas d'en-tête PDMQ. Cela se produit si le message arrive au destinataire sans en-tête PDMQ. C'est-à-dire que le putter d'origine du message n'avait pas de règle pour la destination et que le récepteur n'a pas de SPLPROT (ASPOLICY) défini.
- Il se peut qu'un agent MCA n'arrive pas à insérer un message dans la file d'attente des messages non livrés si la règle AMS définie pour la file d'attente des messages non livrés n'autorise pas l'ID utilisateur sous lequel l'initiateur de canal s'exécute pour protéger le message.
- Les canaux récepteurs placent généralement les messages non distribués dans le DLQ local, tandis que les canaux émetteurs placent généralement les messages qui ne peuvent pas être traités pour une raison quelconque, par exemple, un message trop volumineux pour la file d'attente ou un en-tête MQXQH incorrect, et ainsi de suite dans le DLQ local.
- Les gestionnaires DLQ ne regardent généralement que l'en-tête DLQ (DLH) et non la charge de message elle-même. Par conséquent, le fait que la charge de message puisse être protégée n'empêche pas les gestionnaires de déterminer la raison pour laquelle le message a été placé sur la file d'attente des messages non livrés.
- Si un DLQ n'est pas défini, le canal:
  - Se termine de manière anormale (et passe à l'état de relance) si un message persistant ne peut pas être distribué.
  - Supprime un message non persistant non distribué et continue à s'exécuter.

Concepts associés

«Traitement des erreurs pour AMS», à la page 625
IBM MQ Advanced Message Security définit une file d'attente de traitement des erreurs pour gérer les messages contenant des erreurs ou des messages qui ne peuvent pas être protégés.

Scénarios utilisateur pour AMS

Familiarisez-vous avec les scénarios possibles pour comprendre les objectifs métier que vous pouvez atteindre avec Advanced Message Security.

Windows *Guide de démarrage rapide pour AMS sur les plateformes Windows*

Utilisez ce guide pour configurer rapidement Advanced Message Security (AMS) afin d'assurer la sécurité des messages sur les plateformes Windows . Lorsque vous l'aurez terminé, vous aurez créé une base de données de clés pour vérifier les identités utilisateur et défini des règles de signature / chiffrement pour votre gestionnaire de files d'attente.

Avant de commencer

Au moins les fonctions suivantes doivent être installées sur votre système:

- serveur
- Kit d'outils de développement (pour les exemples de programme)
- Advanced Message Security (AMS)

Pour plus d'informations, voir FonctionsIBM MQ pour les systèmes Windows .

Pour plus d'informations sur l'utilisation de la commande **setmqenv** pour initialiser l'environnement en cours afin que les commandes IBM MQ appropriées puissent être localisées et exécutées par le système d'exploitation, voir setmqenv (set IBM MQ environment).

1. Création d'un gestionnaire de files d'attente et d'une file d'attente

Pourquoi et quand exécuter cette tâche

Tous les exemples suivants utilisent une file d'attente nommée TEST.Q pour la transmission de messages entre les applications. Advanced Message Security utilise des intercepteurs pour signer et chiffrer les messages lorsqu'ils entrent dans l'infrastructure IBM MQ via l'interface IBM MQ standard. La configuration de base est effectuée dans IBM MQ et est configurée dans les étapes suivantes.

Vous pouvez utiliser IBM MQ Explorer pour créer le gestionnaire de files d'attente QM_VERIFY_AMS et sa file d'attente locale appelée TEST.Q à l'aide de tous les paramètres de l'assistant par défaut, ou vous pouvez utiliser les commandes disponibles dans `C:\Program Files\IBM\MQ\bin`. N'oubliez pas que vous devez être membre du groupe d'utilisateurs `mqm` pour exécuter les commandes d'administration suivantes.

Procédure

1. Création d'un gestionnaire de files d'attente

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. Démarrer le gestionnaire de files d'attente

   ```
   strmqm QM_VERIFY_AMS
   ```

3. Créez une file d'attente appelée TEST.Q en entrant la commande suivante dans **runmqsc** pour le gestionnaire de files d'attente QM_VERIFY_AMS

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

Résultats

Si la procédure est terminée, la commande entrée dans **runmqsc** affiche les détails relatifs à TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Création et autorisation d'utilisateurs

Pourquoi et quand exécuter cette tâche

Deux utilisateurs apparaissent dans cet exemple: `alice`, l'expéditeur et bob, le destinataire. Pour utiliser la file d'attente d'application, ces utilisateurs doivent être autorisés à l'utiliser. De même, pour utiliser correctement les règles de protection que nous définirons, ces utilisateurs doivent être autorisés à accéder à certaines files d'attente du système. Pour plus d'informations sur la commande **setmqaut** , voir **setmqaut**.

Procédure

1. Créez les deux utilisateurs et assurez-vous que HOMEPATH et HOMEDRIVE sont définis pour ces deux utilisateurs.
2. Autoriser les utilisateurs à se connecter au gestionnaire de files d'attente et à utiliser la file d'attente

   ```
   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. Vous devez également autoriser les deux utilisateurs à parcourir la file d'attente de règles système et à placer des messages dans la file d'attente d'erreurs.

```
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
```

Avertissement : IBM MQ optimise les performances en mettant en cache les règles de sorte que vous n'ayez pas à parcourir les enregistrements pour obtenir des détails sur les règles dans SYSTEM.PROTECTION.POLICY.QUEUE dans tous les cas.

IBM MQ ne met pas en cache toutes les règles disponibles. S'il existe un nombre élevé de règles, IBM MQ met en cache un nombre limité de règles. Par conséquent, si le nombre de règles définies pour le gestionnaire de files d'attente est faible, il n'est pas nécessaire de fournir l'option de navigation à SYSTEM.PROTECTION.POLICY.QUEUE.

Toutefois, vous devez accorder le droit de navigation à cette file d'attente, au cas où un nombre élevé de règles serait défini, ou si vous utilisez d'anciens clients. SYSTEM.PROTECTION.ERROR.QUEUE est utilisé pour placer les messages d'erreur générés par le code AMS. Le droit d'insertion sur cette file d'attente est vérifié uniquement lorsque vous tentez d'insérer un message d'erreur dans la file d'attente. Votre droit d'insertion sur la file d'attente n'est pas vérifié lorsque vous tentez d'insérer ou d'extraire un message d'une file d'attente protégée AMS.

Résultats

Les utilisateurs sont maintenant créés et les droits requis leur sont accordés.

Que faire ensuite

Pour vérifier si les étapes ont été effectuées correctement, utilisez les exemples `amqsput` et `amqsget` comme décrit dans la section «7. Test de la configuration», à la page 632.

3. Création d'une base de données de clés et de certificats

Pourquoi et quand exécuter cette tâche

L'intercepteur requiert la clé publique des utilisateurs qui l'envoient pour chiffrer le message. Par conséquent, la base de données de clés des identités utilisateur mappées aux clés publiques et privées doit être créée. Dans le système réel, où les utilisateurs et les applications sont dispersés sur plusieurs ordinateurs, chaque utilisateur dispose de son propre magasin de clés privé. De même, dans ce guide, nous créons des bases de données de clés pour `alice` et bob et nous partageons les certificats d'utilisateur entre eux.

Remarque : Dans ce guide, nous utilisons des exemples d'applications écrits en C se connectant à l'aide de liaisons locales. Si vous prévoyez d'utiliser des applications Java à l'aide de liaisons client, vous devez créer un magasin de clés JKS et des certificats à l'aide de la commande Java **`keytool`** V 9.4.0 V 9.4.0 ou de la commande IBM MQ **`runmqktool`**. Pour plus d'informations, voir «Guide de démarrage rapide pour AMS avec les clients Java», à la page 650. Pour tous les autres langages et pour les applications Java utilisant des liaisons locales, les étapes de ce guide sont correctes.

Procédure

1. Créez une nouvelle base de données de clés pour l'utilisateur `alice`.
   Par exemple, exécutez la commande suivante pour créer la nouvelle base de données de clés:

```
runmqakm -keydb -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -type cms -pw
passw0rd -stash
```

Remarque :

- Utilisez un mot de passe fiable pour sécuriser la base de données.
- Incluez le paramètre **-stash** pour stocker le mot de passe chiffré de la base de données de clés dans un fichier.

2. Créez un certificat autosigné pour identifier l'utilisateur `alice` à utiliser dans le chiffrement.
Par exemple, exécutez la commande suivante pour créer un nouveau certificat autosigné:

```
runmqakm -cert -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -stashed
-label Alice_Cert -dn "CN=alice, O=IBM, C=GB"
```

Remarque :

- Pour les besoins de ce guide, nous utilisons un certificat autosigné qui peut être créé sans utiliser d'autorité de certification. Pour les systèmes de production, il est conseillé d'utiliser des certificats signés par une autorité de certification.
- Le paramètre **-label** indique le nom du certificat, que les intercepteurs recherchent pour recevoir les informations nécessaires.
- Le paramètre **-dn** spécifie les détails du nom distinctif (DN) du certificat. Le nom distinctif doit être unique pour chaque utilisateur.

3. Répétez les étapes «1», à la page 629 et «2», à la page 630 pour l'utilisateur bob.

Résultats

Les deux utilisateurs `alice` et bob possèdent chacun un certificat autosigné.

4. Création de keystore.conf

Pourquoi et quand exécuter cette tâche

Vous devez pointer les intercepteurs Advanced Message Security vers le répertoire dans lequel se trouvent les bases de données de clés et les certificats. Cette opération est effectuée via le fichier `keystore.conf` , qui contient ces informations sous forme de texte en clair. Chaque utilisateur doit disposer d'un fichier `keystore.conf` distinct dans le dossier `.mqs` . Cette étape doit être effectuée pour `alice` et bob.

Le contenu de `keystore.conf` doit être au format suivant:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

Exemple

Pour ce scénario, le contenu de `keystore.conf` sera le suivant:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

Remarque :

- Le chemin d'accès au fichier de clés doit être fourni sans inclure l'extension de fichier.
- Le label de certificat peut inclure des espaces, ainsi "Alice_Cert" et "Alice_Cert" (avec un espace à la fin) par exemple, sont reconnus comme des libellés de deux certificats différents. Cependant, pour éviter toute confusion, il est préférable de ne pas utiliser d'espaces dans le nom de l'étiquette.

- Il existe les formats de fichier de clés suivants: CMS (Cryptographic Message Syntax), JKS (Java Keystore) et JCEKS (Java Cryptographic Extension Keystore). Pour plus d'informations, voir «Structure du fichier de configuration du magasin de clés (keystore.conf) pour AMS», à la page 665.
- *%HOMEDRIVE%\%HOMEPATH%*\.mqs\keystore.conf (par exemple, C:\Documents and Settings\alice\.mqs\keystore.conf) est l'emplacement par défaut où Advanced Message Security recherche le fichier keystore.conf . Pour plus d'informations sur l'utilisation d'un emplacement autre que celui par défaut pour le keystore.conf, voir «Utilisation de magasins de clés et de certificats avec AMS», à la page 664.
- Pour créer le répertoire .mqs , vous devez utiliser l'invite de commande.

5. Partage de certificats

Pourquoi et quand exécuter cette tâche

Partagez les certificats entre les deux bases de données de clés afin que chaque utilisateur puisse identifier l'autre. Cette opération est effectuée en extrayant le certificat public de chaque utilisateur dans un fichier, qui est ensuite ajouté à la base de données de clés de l'autre utilisateur.

Remarque : Prenez soin d'utiliser l'option *extract* et non l'option *export* . *Extract* obtient la clé publique de l'utilisateur, tandis que *export* obtient à la fois la clé publique et la clé privée. L'utilisation de *export* par erreur compromettrait complètement votre application en transmettant sa clé privée.

Procédure

1. Extrayez le certificat identifiant alice dans un fichier externe:

```
runmqakm -cert -extract -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. Ajoutez le certificat au magasin de clés bob's :

```
runmqakm -cert -add -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. Répétez les étapes pour bob:

```
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
```

Résultats

Les deux utilisateurs alice et bob sont désormais en mesure de s'identifier mutuellement en ayant créé et partagé des certificats autosignés.

Que faire ensuite

Vérifiez qu'un certificat se trouve dans le magasin de clés en le parcourant à l'aide de l'interface graphique ou en exécutant les commandes suivantes qui impriment ses détails:

```
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. Définition de la règle de file d'attente

Pourquoi et quand exécuter cette tâche

Avec le gestionnaire de files d'attente créé et les intercepteurs préparés pour intercepter les messages et les clés de chiffrement d'accès, nous pouvons commencer à définir des règles de protection sur `QM_VERIFY_AMS` à l'aide de la commande `setmqspl` . Pour plus d'informations sur cette commande, voir setmqspl . Chaque nom de règle doit être identique au nom de la file d'attente à laquelle il doit être appliqué.

Exemple

Voici un exemple de règle définie pour la file d'attente `TEST.Q` . Dans l'exemple, les messages sont signés avec l'algorithme Deprecated SHA1 et chiffrés avec l'algorithme AES256 . `alice` est le seul émetteur valide et bob est le seul récepteur des messages de cette file d'attente:

```
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"
```

Remarque : Les noms distinctifs correspondent exactement à ceux spécifiés dans le certificat de l'utilisateur respectif de la base de données de clés.

Que faire ensuite

Pour vérifier la règle que vous avez définie, exécutez la commande suivante:

```
dspmqspl -m QM_VERIFY_AMS
```

Pour imprimer les détails de la règle sous la forme d'un ensemble de commandes `setmqspl` , utilisez l'indicateur `-export` . Cela permet de stocker des règles déjà définies:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Test de la configuration

Pourquoi et quand exécuter cette tâche

En exécutant différents programmes sous différents utilisateurs, vous pouvez vérifier si l'application a été correctement configurée.

Procédure

1. Changez d'utilisateur pour qu'il s'exécute en tant qu'utilisateur `alice`

   Cliquez avec le bouton droit de la souris sur `cmd.exe` et sélectionnez **Exécuter en tant que ...**. Lorsque vous y êtes invité, connectez-vous en tant que `alice`.
2. En tant qu'utilisateur `alice` , placez un message à l'aide d'un exemple d'application:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. Entrez le texte du message, puis appuyez sur Entrée.
4. Changez d'utilisateur pour qu'il s'exécute en tant qu'utilisateur bob

   Ouvrez une autre fenêtre en cliquant avec le bouton droit de la souris sur `cmd.exe` et en sélectionnant **Exécuter en tant que ...**. Lorsque vous y êtes invité, connectez-vous en tant que bob.
5. En tant qu'utilisateur bob , obtenez un message à l'aide d'un exemple d'application:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

Résultats

Si l'application a été correctement configurée pour les deux utilisateurs, le message de l'utilisateur `alice` s'affiche lorsque bob exécute l'application d'obtention.

8. Test du chiffrement

Pourquoi et quand exécuter cette tâche

Pour vérifier que le chiffrement est effectué comme prévu, créez une file d'attente alias qui fait référence à la file d'attente d'origine TEST.Q. Cette file d'attente alias n'ayant pas de règle de sécurité, aucun utilisateur ne dispose des informations permettant de déchiffrer le message. Par conséquent, les données chiffrées sont affichées.

Procédure

1. A l'aide de la commande **runmqsc** sur le gestionnaire de files d'attente QM_VERIFY_AMS, créez une file d'attente alias.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Accordez à bob l'accès pour parcourir la file d'attente alias

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. En tant qu'utilisateur `alice`, placez un autre message à l'aide d'un exemple d'application comme précédemment:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. En tant qu'utilisateur bob, parcourez le message à l'aide d'un exemple d'application via la file d'attente alias cette fois:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. En tant qu'utilisateur bob, obtenez le message à l'aide d'un exemple d'application à partir de la file d'attente locale:

```
amqsget TEST.Q QM_VERIFY_AMS
```

Résultats

La sortie de l'application `amqsbcg` affiche les données chiffrées qui se trouvent dans la file d'attente prouvant que le message a été chiffré.

Linux AIX *Guide de démarrage rapide pour AMS sur AIX and Linux*

Utilisez ce guide pour configurer rapidement Advanced Message Security afin de fournir la sécurité des messages sur AIX and Linux. Lorsque vous l'aurez terminé, vous aurez créé une base de données de clés pour vérifier les identités utilisateur et défini des règles de signature / chiffrement pour votre gestionnaire de files d'attente.

Avant de commencer

Au moins les composants suivants doivent être installés sur votre système:

- Environnement d'exécution
- serveur
- Exemples de programme

- IBM Global Security Kit (GSKit)
- Advanced Message Security

Reportez-vous aux rubriques suivantes pour connaître les noms de composant sur chaque plateforme spécifique:

- Linux ComposantsIBM MQ pour les systèmes Linux
- AIX ComposantsIBM MQ pour les systèmes AIX

1. Création d'un gestionnaire de files d'attente et d'une file d'attente

Pourquoi et quand exécuter cette tâche

Tous les exemples suivants utilisent une file d'attente nommée TEST.Q pour la transmission de messages entre les applications. Advanced Message Security utilise des intercepteurs pour signer et chiffrer les messages lorsqu'ils entrent dans l'infrastructure IBM MQ via l'interface IBM MQ standard. La configuration de base est effectuée dans IBM MQ et est configurée dans les étapes suivantes.

Vous pouvez utiliser IBM MQ Explorer pour créer le gestionnaire de files d'attente QM_VERIFY_AMS et sa file d'attente locale appelée TEST.Q à l'aide de tous les paramètres de l'assistant par défaut, ou vous pouvez utiliser les commandes disponibles dans *MQ_INSTALLATION_PATH*/bin. N'oubliez pas que vous devez être membre du groupe d'utilisateurs mqm pour exécuter les commandes d'administration suivantes.

Procédure

1. Création d'un gestionnaire de files d'attente

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. Démarrer le gestionnaire de files d'attente

   ```
   strmqm QM_VERIFY_AMS
   ```

3. Créez une file d'attente appelée TEST.Q en entrant la commande suivante dans **runmqsc** pour le gestionnaire de files d'attente QM_VERIFY_AMS

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

Résultats

Si la procédure a abouti, la commande suivante entrée dans **runmqsc** affiche les détails relatifs à TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Création et autorisation d'utilisateurs

Pourquoi et quand exécuter cette tâche

Deux utilisateurs apparaissent dans cet exemple: alice, l'expéditeur et bob, le destinataire. Pour utiliser la file d'attente d'application, ces utilisateurs doivent être autorisés à l'utiliser. De même, pour utiliser correctement les règles de protection que nous définirons, ces utilisateurs doivent être autorisés à accéder à certaines files d'attente du système. Pour plus d'informations sur la commande **setmqaut** , voir **setmqaut**.

Procédure

1. Créer les deux utilisateurs

```
useradd alice
```

```
useradd bob
```

2. Autoriser les utilisateurs à se connecter au gestionnaire de files d'attente et à utiliser la file d'attente

```
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. Vous devez également autoriser les deux utilisateurs à parcourir la file d'attente de règles système et à placer des messages dans la file d'attente d'erreurs.

```
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
```

Avertissement : IBM MQ optimise les performances en mettant en cache les règles de sorte que vous n'ayez pas à parcourir les enregistrements pour obtenir des détails sur les règles dans SYSTEM.PROTECTION.POLICY.QUEUE dans tous les cas.

IBM MQ ne met pas en cache toutes les règles disponibles. S'il existe un nombre élevé de règles, IBM MQ met en cache un nombre limité de règles. Par conséquent, si le nombre de règles définies pour le gestionnaire de files d'attente est faible, il n'est pas nécessaire de fournir l'option de navigation à SYSTEM.PROTECTION.POLICY.QUEUE.

Toutefois, vous devez accorder le droit de navigation à cette file d'attente, au cas où un nombre élevé de règles serait défini, ou si vous utilisez d'anciens clients. SYSTEM.PROTECTION.ERROR.QUEUE est utilisé pour placer les messages d'erreur générés par le code AMS. Le droit d'insertion sur cette file d'attente est vérifié uniquement lorsque vous tentez d'insérer un message d'erreur dans la file d'attente. Votre droit d'insertion sur la file d'attente n'est pas vérifié lorsque vous tentez d'insérer ou d'extraire un message d'une file d'attente protégée AMS.

Résultats

Des groupes d'utilisateurs sont maintenant créés et les droits requis leur sont accordés. Ainsi, les utilisateurs affectés à ces groupes auront également le droit de se connecter au gestionnaire de files d'attente et d'insérer et d'extraire de la file d'attente.

Que faire ensuite

Pour vérifier si les étapes ont été effectuées correctement, utilisez les exemples `amqsput` et `amqsget` comme décrit dans la section «8. Test du chiffrement», à la page 639.

3. Création d'une base de données de clés et de certificats

Pourquoi et quand exécuter cette tâche

Pour chiffrer le message, l'intercepteur requiert la clé privée de l'utilisateur émetteur et la ou les clés publiques du ou des destinataires. Par conséquent, la base de données de clés des identités utilisateur mappées aux clés publiques et privées doit être créée. Dans le système réel, où les utilisateurs et les applications sont dispersés sur plusieurs ordinateurs, chaque utilisateur dispose de son propre magasin de clés privé. De même, dans ce guide, nous créons des bases de données de clés pour `alice` et bob et nous partageons les certificats d'utilisateur entre eux.

Remarque : Dans ce guide, nous utilisons des exemples d'applications écrits en C se connectant à l'aide de liaisons locales. Si vous prévoyez d'utiliser des applications Java à l'aide de liaisons client, vous devez créer un magasin de clés JKS et des certificats à l'aide de la commande **keytool** , qui fait partie de l'environnement d'exécution Java (voir «Guide de démarrage rapide pour AMS avec les clients Java», à la page 650 pour plus de détails). Pour tous les autres langages et pour les applications Java utilisant des liaisons locales, les étapes de ce guide sont correctes.

Procédure

1. Créer une nouvelle base de données de clés pour l'utilisateur `alice`

   ```
   mkdir /home/alice/.mqs -p
   ```

   ```
   runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
   ```

   Remarque :

   - Il est conseillé d'utiliser un mot de passe fiable pour sécuriser la base de données.
   - Le paramètre **stash** stocke le mot de passe dans le fichier `key.sth` , que les intercepteurs peuvent utiliser pour ouvrir la base de données.
2. Vérifiez que la base de données de clés est lisible

   ```
   chmod +r /home/alice/.mqs/alicekey.kdb
   ```
3. Créez un certificat identifiant l'utilisateur `alice` à utiliser dans le chiffrement

   ```
   runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -dn
   "cn=alice,O=IBM,c=GB" -default_cert yes
   ```

   Remarque :

   - Pour les besoins de ce guide, nous utilisons un certificat autosigné qui peut être créé sans utiliser d'autorité de certification. Pour les systèmes de production, il est conseillé de ne pas utiliser de certificats autosignés, mais de s'appuyer sur des certificats signés par une autorité de certification.
   - Le paramètre **label** indique le nom du certificat, que les intercepteurs recherchent pour recevoir les informations nécessaires.
   - Le paramètre **DN** spécifie les détails du **nom distinctif** (DN), qui doit être unique pour chaque utilisateur.
4. Maintenant que nous avons créé la base de données de clés, nous devons en définir la propriété et nous assurer qu'elle est illisible par tous les autres utilisateurs.

   ```
   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. Répétez les étapes 1 à 4 pour l'utilisateur bob

Résultats

Les deux utilisateurs `alice` et bob possèdent chacun un certificat autosigné.

4. Création de keystore.conf

Pourquoi et quand exécuter cette tâche

Vous devez pointer les intercepteurs Advanced Message Security vers le répertoire dans lequel se trouvent les bases de données de clés et les certificats. Cette opération est effectuée via le fichier `keystore.conf` , qui contient ces informations sous forme de texte en clair. Chaque utilisateur doit disposer d'un fichier `keystore.conf` distinct dans le dossier `.mqs` . Cette étape doit être effectuée pour `alice` et bob.

Le contenu de `keystore.conf` doit être au format suivant:

```
cms.keystore = dir/keystore_file
```

```
cms.certificate = certificate_label
```

Exemple

Pour ce scénario, le contenu de `keystore.conf` sera le suivant:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

Remarque :

- Le chemin d'accès au fichier de clés doit être fourni sans inclure l'extension de fichier.
- Il existe les formats de fichier de clés suivants: CMS (Cryptographic Message Syntax), JKS (Java Keystore) et JCEKS (Java Cryptographic Extension Keystore). Pour plus d'informations, voir «Structure du fichier de configuration du magasin de clés (keystore.conf) pour AMS», à la page 665.
- `HOME/.mqs/keystore.conf` est l'emplacement par défaut où Advanced Message Security recherche le fichier `keystore.conf` . Pour plus d'informations sur l'utilisation d'un emplacement autre que celui par défaut pour le `keystore.conf`, voir «Utilisation de magasins de clés et de certificats avec AMS», à la page 664.

5. Partage de certificats

Pourquoi et quand exécuter cette tâche

Partagez les certificats entre les deux bases de données de clés afin que chaque utilisateur puisse identifier l'autre. Cette opération est effectuée en extrayant le certificat public de chaque utilisateur dans un fichier, qui est ensuite ajouté à la base de données de clés de l'autre utilisateur.

Remarque : Prenez soin d'utiliser l'option *extract* et non l'option *export* . *Extract* obtient la clé publique de l'utilisateur, tandis que *export* obtient à la fois la clé publique et la clé privée. L'utilisation de *export* par erreur compromettrait complètement votre application en transmettant sa clé privée.

Procédure

1. Extrayez le certificat identifiant `alice` dans un fichier externe:

```
runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
-target alice_public.arm
```

2. Ajoutez le certificat au magasin de clés bob ' s :

```
runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
alice_public.arm
```

3. Répétez l'étape pour bob:

```
runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
bob_public.arm
```

4. Ajoutez le certificat pour bob au magasin de clés `alice` ' s :

```
runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
bob_public.arm
```

Résultats

Les deux utilisateurs `alice` et bob sont désormais en mesure de s'identifier mutuellement en ayant créé et partagé des certificats autosignés.

Que faire ensuite

Vérifiez qu'un certificat se trouve dans le magasin de clés en exécutant les commandes suivantes qui impriment ses détails:

```
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. Définition de la règle de file d'attente

Pourquoi et quand exécuter cette tâche

Avec le gestionnaire de files d'attente créé et les intercepteurs préparés pour intercepter les messages et les clés de chiffrement d'accès, nous pouvons commencer à définir des règles de protection sur `QM_VERIFY_AMS` à l'aide de la commande `setmqspl` . Pour plus d'informations sur cette commande, voir setmqspl . Chaque nom de règle doit être identique au nom de la file d'attente à laquelle il doit être appliqué.

Exemple

Voici un exemple de règle définie pour la file d'attente `TEST.Q` . Dans cet exemple, les messages sont signés par l'utilisateur `alice` à l'aide de l'algorithme Deprecated SHA1 et chiffrés à l'aide de l'algorithme AES 256 bits. `alice` est le seul émetteur valide et bob est le seul récepteur des messages de cette file d'attente:

```
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"
```

Remarque : Les noms distinctifs correspondent exactement à ceux spécifiés dans le certificat de l'utilisateur respectif de la base de données de clés.

Que faire ensuite

Pour vérifier la règle que vous avez définie, exécutez la commande suivante:

```
dspmqspl -m QM_VERIFY_AMS
```

Pour imprimer les détails de la règle sous la forme d'un ensemble de commandes `setmqspl` , utilisez l'indicateur `-export` . Cela permet de stocker des règles déjà définies:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Test de la configuration

Pourquoi et quand exécuter cette tâche

En exécutant différents programmes sous différents utilisateurs, vous pouvez vérifier si l'application a été correctement configurée.

Procédure

1. Accédez au répertoire contenant les exemples. Si MQ est installé dans un emplacement autre que celui par défaut, il se peut qu'il se trouve à un autre emplacement.

   ```
   cd /opt/mqm/samp/bin
   ```

2. Changez d'utilisateur pour qu'il s'exécute en tant qu'utilisateur `alice`

   ```
   su alice
   ```

3. En tant qu'utilisateur `alice`, placez un message à l'aide d'un exemple d'application:

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. Entrez le texte du message, puis appuyez sur Entrée.
5. Arrêt de l'exécution en tant qu'utilisateur `alice`

   ```
   exit
   ```

6. Changez d'utilisateur pour qu'il s'exécute en tant qu'utilisateur bob

   ```
   su bob
   ```

7. En tant qu'utilisateur bob, obtenez un message à l'aide d'un exemple d'application:

   ```
   ./amqsget TEST.Q QM_VERIFY_AMS
   ```

Résultats

Si l'application a été correctement configurée pour les deux utilisateurs, le message de l'utilisateur `alice` s'affiche lorsque bob exécute l'application d'obtention.

8. Test du chiffrement

Pourquoi et quand exécuter cette tâche

Pour vérifier que le chiffrement est effectué comme prévu, créez une file d'attente alias qui fait référence à la file d'attente d'origine TEST.Q. Cette file d'attente alias n'ayant pas de règle de sécurité, aucun utilisateur ne dispose des informations permettant de déchiffrer le message. Par conséquent, les données chiffrées sont affichées.

Procédure

1. A l'aide de la commande **runmqsc** sur le gestionnaire de files d'attente QM_VERIFY_AMS, créez une file d'attente alias.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Accordez à bob l'accès pour parcourir la file d'attente alias

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. En tant qu'utilisateur `alice`, placez un autre message à l'aide d'un exemple d'application comme précédemment:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. En tant qu'utilisateur bob, parcourez le message à l'aide d'un exemple d'application via la file d'attente alias cette fois:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. En tant qu'utilisateur bob, obtenez le message à l'aide d'un exemple d'application à partir de la file d'attente locale:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Résultats

La sortie de l'application `amqsbcg` affichera les données chiffrées qui se trouvent dans la file d'attente prouvant que le message a été chiffré.

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 *Mise en file d'attente locale des messages protégés par la confidentialité pour AMS sur z/OS*

Cet exemple détaille les politiques et les certificats Advanced Message Security nécessaires pour envoyer et extraire des messages protégés par la confidentialité vers et depuis une file d'attente, locale aux applications d'insertion et d'extraction. Les messages protégés par la confidentialité sont à la fois signés et chiffrés.

Les exemples de gestionnaire de files d'attente et de file d'attente locale sont les suivants:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

Ces utilisateurs sont utilisés:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Les étapes de configuration de ce scénario sont les suivantes:

Créer les certificats d'utilisateur

Dans cet exemple, deux certificats d'utilisateur sont requis. Il s'agit du certificat de l'utilisateur émetteur qui est nécessaire pour signer les messages et du certificat de l'utilisateur destinataire qui est nécessaire pour chiffrer et déchiffrer les données du message. L'utilisateur émetteur est'TELLER5'et l'utilisateur destinataire est'FINADM2'.

Le certificat de l'autorité de certification est également requis.Le certificat de l'autorité de certification est le certificat de l'autorité qui a émis le certificat de l'utilisateur. Il peut s'agir d'une chaîne de certificats. Si tel est le cas, tous les certificats de la chaîne sont requis dans le fichier de clés de l'utilisateur de la tâche Advanced Message Security , dans ce cas l'utilisateur WMQBNK6.

Un certificat d'autorité de certification peut être créé à l'aide de la commande RACDCERT RACF . Ce certificat est utilisé pour émettre des certificats d'utilisateur. Exemple :

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Cette commande RACDCERT crée un certificat d'autorité de certification qui peut ensuite être utilisé pour émettre des certificats d'utilisateur pour les utilisateurs'TELLER5'et'FINADM2'. Exemple :

```
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)
```

Votre installation comporte des procédures pour choisir ou créer un certificat d'autorité de certification, ainsi que des procédures pour émettre des certificats et les distribuer aux systèmes appropriés.

Lors de l'exportation et de l'importation de ces certificats, Advanced Message Security requiert:

- Certificat de l'autorité de certification (chaîne).
- Certificat d'utilisateur émetteur et sa clé privée.
- Certificat d'utilisateur du destinataire et sa clé privée.

Si vous utilisez RACF, la commande RACDCERT EXPORT peut être utilisée pour exporter des certificats vers un fichier et la commande RACDCERT ADD peut être utilisée pour importer des certificats à partir du fichier. Pour plus d'informations sur ces commandes et d'autres commandes RACDCERT, voir RACDCERT (Manage RACF digital certificates) dans le manuel *z/OS: Security Server RACF Command Language Reference*.

Dans ce cas, les certificats sont requis sur le système z/OS exécutant le gestionnaire de files d'attente BNK6.

Lorsque les certificats ont été importés sur le système z/OS exécutant BNK6, les certificats d'utilisateur requièrent l'attribut TRUST. La commande RACDCERT ALTER peut être utilisée pour ajouter l'attribut TRUST au certificat. Exemple :

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Connexion de certificats à des fichiers de clés pertinents

Lorsque les certificats requis ont été créés ou importés et définis comme étant de confiance, ils doivent être connectés aux fichiers de clés utilisateur appropriés sur le système z/OS exécutant BNK6. Pour créer les fichiers de clés, utilisez la commande RACDCERT ADDRING:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Cela crée un fichier de clés pour l'utilisateur de la tâche Advanced Message Security et des fichiers de clés pour les utilisateurs d'envoi et de destinataire. Notez que le nom du fichier de clés `drq.ams.keyring` est obligatoire et qu'il est sensible à la casse.

Une fois les fichiers de clés créés, les certificats appropriés peuvent être connectés.

```
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))
```

Les certificats d'utilisateur d'envoi et de destinataire doivent être connectés sous la forme DEFAULT. Si l'un des utilisateurs possède plusieurs certificats dans son fichier drq.ams.keyring, le certificat par défaut est utilisé à des fins de signature et de déchiffrement.

Le certificat de l'utilisateur destinataire doit également être connecté au fichier de clés de l'utilisateur de la tâche Advanced Message Security avec USAGE (SITE). En effet, la tâche Advanced Message Security a besoin de la clé publique du destinataire lors du chiffrement des données de message. USAGE (SITE) empêche la clé privée d'être accessible dans le fichier de clés.

La création et la modification de certificats ne sont pas reconnues par Advanced Message Security tant que le gestionnaire de files d'attente n'est pas arrêté et redémarré ou que la commande z/OS **MODIFY** n'est pas utilisée pour actualiser la configuration de certificat Advanced Message Security . Exemple :

```
F BNK6AMSM,REFRESH KEYRING
```

Création de la règle Advanced Message Security

Dans cet exemple, les messages protégés par la confidentialité sont placés dans la file d'attente FIN.XFER.Q8 par une application s'exécutant en tant qu'utilisateur'TELLER5'et extraite de la même file d'attente par une application s'exécutant en tant qu'utilisateur'FINADM2', de sorte qu'une seule règle Advanced Message Security est requise.

Les règles Advanced Message Security sont créées à l'aide de l'utilitaire CSQ0UTIL documenté à l'adresse The message security policy utility (CSQ0UTIL).

Utilisez l'utilitaire CSQ0UTIL pour exécuter la commande suivante:

```
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
```

Dans cette règle, le gestionnaire de files d'attente est identifié comme BNK6. Le nom de la règle et la file d'attente associée sont FIN.XFER.Q8. L'algorithme utilisé pour générer la signature de l'expéditeur est Deprecated SHA1et le nom distinctif (DN) de l'utilisateur émetteur est'CN=Teller5,O=BCO,C=US'et l'utilisateur destinataire est'CN=FinAdm2,O=BCO,C=US'. L'algorithme utilisé pour chiffrer les données de message est Deprecated 3DES.

Après avoir défini la règle, redémarrez le gestionnaire de files d'attente BNK6 ou utilisez la commande z/OS **MODIFY** pour actualiser la configuration de la règle Advanced Message Security . Exemple :

```
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 *Mise en file d'attente à distance des messages protégés par la confidentialité pour AMS on z/OS*

Cet exemple détaille les règles et les certificats Advanced Message Security nécessaires pour envoyer et extraire des messages protégés par la confidentialité vers et depuis des files d'attente gérées par deux gestionnaires de files d'attente différents. Les deux gestionnaires de files d'attente peuvent être exécutés sur le même système z/OS ou sur des systèmes z/OS différents, ou un gestionnaire de files d'attente peut être exécuté sur un système réparti exécutant Advanced Message Security.

Les exemples de gestionnaires de files d'attente et de files d'attente sont les suivants:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Remarque: dans cet exemple, BNK6 et BNK7 sont des gestionnaires de files d'attente s'exécutant sur des systèmes z/OS différents portant le même nom.

Ces utilisateurs sont utilisés:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

Les étapes de configuration de ce scénario sont les suivantes:

Créer les certificats d'utilisateur

Dans cet exemple, deux certificats d'utilisateur sont requis. Il s'agit du certificat de l'utilisateur émetteur qui est nécessaire pour signer les messages et du certificat de l'utilisateur destinataire qui est nécessaire pour chiffrer et déchiffrer les données du message. L'utilisateur émetteur est'TELLER5'et l'utilisateur destinataire est'FINADM2'.

Le certificat de l'autorité de certification est également requis.Le certificat de l'autorité de certification est le certificat de l'autorité qui a émis le certificat de l'utilisateur. Il peut s'agir d'une chaîne de certificats. Si tel est le cas, tous les certificats de la chaîne sont requis dans le fichier de clés de l'utilisateur de la tâche Advanced Message Security , dans ce cas l'utilisateur WMQBNK7.

Un certificat d'autorité de certification peut être créé à l'aide de la commande RACDCERT RACF . Ce certificat est utilisé pour émettre des certificats d'utilisateur. Exemple :

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Cette commande RACDCERT crée un certificat d'autorité de certification qui peut ensuite être utilisé pour émettre des certificats d'utilisateur pour les utilisateurs'TELLER5'et'FINADM2'. Exemple :

```
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)
```

Votre installation comporte des procédures pour choisir ou créer un certificat d'autorité de certification, ainsi que des procédures pour émettre des certificats et les distribuer aux systèmes appropriés.

Lors de l'exportation et de l'importation de ces certificats, Advanced Message Security requiert:

- Certificat de l'autorité de certification (chaîne).
- Certificat d'utilisateur émetteur et sa clé privée.
- Certificat d'utilisateur du destinataire et sa clé privée.

Si vous utilisez RACF, la commande RACDCERT EXPORT peut être utilisée pour exporter des certificats vers un fichier et la commande RACDCERT ADD peut être utilisée pour importer des certificats à partir du fichier.

Pour plus d'informations sur ces commandes et d'autres commandes RACDCERT, voir RACDCERT (Manage RACF digital certificates) dans le manuel *z/OS: Security Server RACF Command Language Reference*.

Dans ce cas, les certificats sont requis sur le système z/OS exécutant le gestionnaire de files d'attente BNK6 et BNK7.

Dans cet exemple, les certificats d'envoi et de destinataire doivent être importés sur le système z/OS exécutant BNK6, et les certificats d'autorité de certification et de destinataire doivent être importés sur le système z/OS exécutant BNK7. Lorsque les certificats ont été importés, les certificats d'utilisateur requièrent l'attribut TRUST. La commande RACDCERT ALTER peut être utilisée pour ajouter l'attribut TRUST au certificat. Exemple :

Sur BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Sur BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Connexion de certificats à des fichiers de clés pertinents

Une fois que les certificats requis ont été créés ou importés et définis comme certificats de confiance, ils doivent être connectés aux fichiers de clés utilisateur appropriés sur les systèmes z/OS exécutant BNK6 et BNK7.

Pour créer les fichiers de clés, utilisez la commande RACDCERT ADDRING:

Sur BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Cela crée un fichier de clés pour l'utilisateur de la tâche Advanced Message Security et un fichier de clés pour l'utilisateur émetteur sur BNK6. Notez que le nom de fichier de clés drq.ams.keyring est obligatoire et qu'il est sensible à la casse.

Sur BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Cela crée un fichier de clés pour l'utilisateur de la tâche Advanced Message Security et un fichier de clés pour l'utilisateur destinataire sur BNK7.

Une fois les fichiers de clés créés, les certificats appropriés peuvent être connectés.

Sur BNK6:

```
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))
```

Sur BNK7:

```
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))
```

Les certificats d'utilisateur d'envoi et de destinataire doivent être connectés sous la forme DEFAULT. Si l'un des utilisateurs possède plusieurs certificats dans son fichier de clés drq.ams.keyring, le certificat par défaut est utilisé à des fins de signature et de chiffrement / déchiffrement.

Sur BNK6, le certificat de l'utilisateur destinataire doit également être connecté au fichier de clés de l'utilisateur de la tâche Advanced Message Security avec USAGE (SITE). En effet, la tâche Advanced Message Security a besoin de la clé publique du destinataire lors du chiffrement des données de message. USAGE (SITE) empêche la clé privée d'être accessible dans le fichier de clés.

La création et la modification de certificats ne sont pas reconnues par Advanced Message Security tant que le gestionnaire de files d'attente n'est pas arrêté et redémarré ou que la commande z/OS **MODIFY** n'est pas utilisée pour actualiser la configuration de certificat Advanced Message Security . Exemple :

Sur BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

Sur BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Création des règles Advanced Message Security

Dans cet exemple, les messages protégés par la confidentialité sont placés dans la file d'attente éloignée FIN.XFER.Q7 sur BNK6 par une application s'exécutant en tant qu'utilisateur'TELLER5'et extraite de la file d'attente locale FIN.RCPT.Q7 sur BNK7 par une application s'exécutant en tant qu'utilisateur'FINADM2', deux règles Advanced Message Security sont requises.

Les règles Advanced Message Security sont créées à l'aide de l'utilitaire CSQ0UTIL documenté à l'adresse The message security policy utility (CSQ0UTIL).

Utilisez l'utilitaire CSQ0UTIL pour exécuter la commande suivante afin de définir une règle de confidentialité pour la file d'attente éloignée sur BNK6:

```
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
```

Dans cette règle, le gestionnaire de files d'attente est identifié comme BNK6. Le nom de la règle et la file d'attente associée sont FIN.XFER.Q7. L'algorithme utilisé pour générer la signature de l'expéditeur est Deprecated SHA1, le nom distinctif (DN) de l'utilisateur émetteur est'CN=Teller5,O=BCO,C=US'et l'utilisateur destinataire est'CN=FinAdm2,O=BCO,C=US'. L'algorithme utilisé pour chiffrer les données de message est Deprecated 3DES.

Utilisez également l'utilitaire CSQ0UTIL pour exécuter la commande suivante afin de définir une règle de confidentialité pour la file d'attente locale sur BNK7:

```
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
```

Dans cette règle, le gestionnaire de files d'attente est identifié comme BNK7. Le nom de la règle et la file d'attente associée sont FIN.RCPT.Q7. L'algorithme attendu pour la signature de l'expéditeur est Deprecated SHA1, le nom distinctif (DN) de l'utilisateur émetteur est'CN=Teller5,O=BCO,C=US'et l'utilisateur destinataire est'CN=FinAdm2,O=BCO,C=US'. L'algorithme utilisé pour déchiffrer les données de message est Deprecated 3DES.

Après avoir défini les deux règles, redémarrez les gestionnaires de files d'attente BNK6 et BNK7 ou utilisez la commande z/OS **MODIFY** pour actualiser la configuration des règles Advanced Message Security . Exemple :

Sur BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

Sur BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

Guide de démarrage rapide pour AMS avec les clients Java

Utilisez ce guide pour configurer rapidement Advanced Message Security afin de garantir la sécurité des messages pour les applications Java qui se connectent à l'aide de liaisons client. Lorsque vous l'aurez terminé, vous aurez créé un magasin de clés pour vérifier les identités utilisateur et défini des règles de signature / chiffrement pour votre gestionnaire de files d'attente.

Avant de commencer

Vérifiez que les composants appropriés sont installés, comme décrit dans «Guide de démarrage rapide pour AMS sur les plateformes Windows», à la page 627 ou «Guide de démarrage rapide pour AMS sur AIX and Linux», à la page 633.

1. Création d'un gestionnaire de files d'attente et d'une file d'attente

Pourquoi et quand exécuter cette tâche

Tous les exemples suivants utilisent une file d'attente nommée TEST.Q pour la transmission de messages entre les applications. Advanced Message Security utilise des intercepteurs pour signer et chiffrer les messages lorsqu'ils entrent dans l'infrastructure IBM MQ via l'interface IBM MQ standard. La configuration de base est effectuée dans IBM MQ et est configurée dans les étapes suivantes.

Procédure

1. Création d'un gestionnaire de files d'attente

```
crtmqm QM_VERIFY_AMS
```

2. Démarrer le gestionnaire de files d'attente

```
strmqm QM_VERIFY_AMS
```

3. Créez et démarrez un programme d'écoute en entrant les commandes suivantes dans **runmqsc** pour le gestionnaire de files d'attente QM_VERIFY_AMS

```
DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

```
START LISTENER(AMS.LSTR)
```

4. Créez un canal via lequel nos applications se connectent en entrant la commande suivante dans **runmqsc** pour le gestionnaire de files d'attente QM_VERIFY_AMS

```
DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
```

5. Créez une file d'attente appelée TEST.Q en entrant la commande suivante dans **runmqsc** pour le gestionnaire de files d'attente QM_VERIFY_AMS

```
DEFINE QLOCAL(TEST.Q)
```

Résultats

Si la procédure a abouti, la commande suivante entrée dans **runmqsc** affiche les détails relatifs à TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Création et autorisation d'utilisateurs

Pourquoi et quand exécuter cette tâche

Deux utilisateurs apparaissent dans ce scénario: alice, l'expéditeur, et bob, le destinataire. Pour utiliser la file d'attente d'application, ces utilisateurs doivent être autorisés à l'utiliser. De même, pour utiliser correctement les règles de protection définies dans ce scénario, ces utilisateurs doivent être autorisés à accéder à certaines files d'attente système. Pour plus d'informations sur la commande **setmqaut** , voir **setmqaut**.

Procédure

1. Créez les deux utilisateurs comme décrit dans le **Guide de démarrage rapide** (Windows ou AIX and Linux) pour votre plateforme.
2. Autoriser les utilisateurs à se connecter au gestionnaire de files d'attente et à utiliser la file d'attente

```
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. Vous devez également autoriser les deux utilisateurs à parcourir la file d'attente de règles système et à placer des messages dans la file d'attente d'erreurs.

```
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
```

Avertissement : IBM MQ optimise les performances en mettant en cache les règles de sorte que vous n'ayez pas à parcourir les enregistrements pour obtenir des détails sur les règles dans SYSTEM.PROTECTION.POLICY.QUEUE dans tous les cas.

IBM MQ ne met pas en cache toutes les règles disponibles. S'il existe un nombre élevé de règles, IBM MQ met en cache un nombre limité de règles. Par conséquent, si le nombre de règles définies pour le gestionnaire de files d'attente est faible, il n'est pas nécessaire de fournir l'option de navigation à SYSTEM.PROTECTION.POLICY.QUEUE.

Toutefois, vous devez accorder le droit de navigation à cette file d'attente, au cas où un nombre élevé de règles serait défini, ou si vous utilisez d'anciens clients. SYSTEM.PROTECTION.ERROR.QUEUE est utilisé pour placer les messages d'erreur générés par le code AMS. Le droit d'insertion sur cette file d'attente est vérifié uniquement lorsque vous tentez d'insérer un message d'erreur dans la file d'attente. Votre droit d'insertion sur la file d'attente n'est pas vérifié lorsque vous tentez d'insérer ou d'extraire un message d'une file d'attente protégée AMS.

Résultats

Les utilisateurs sont maintenant créés et les droits requis leur sont accordés.

Que faire ensuite

Pour vérifier si les étapes ont été effectuées correctement, utilisez les exemples `JmsProducer` et `JmsConsumer` comme décrit dans la section «7. Test de la configuration», à la page 655.

3. Création d'une base de données de clés et de certificats

Pourquoi et quand exécuter cette tâche

Pour chiffrer le message à l'intercepteur, la clé publique des utilisateurs qui l'envoient est requise. Par conséquent, la base de données de clés des identités utilisateur mappées aux clés publiques et privées doit être créée. Dans le système réel, où les utilisateurs et les applications sont dispersés sur plusieurs ordinateurs, chaque utilisateur dispose de son propre magasin de clés privé. De même, dans ce guide, nous créons des bases de données de clés pour `alice` et bob et nous partageons les certificats d'utilisateur entre eux.

Remarque : Dans ce guide, nous utilisons des exemples d'applications écrits en Java se connectant à l'aide de liaisons client. Si vous prévoyez d'utiliser des applications Java à l'aide de liaisons locales ou

d'applications C, vous devez créer un magasin de clés et des certificats CMS à l'aide de la commande **runmqakm** . Pour plus d'informations, voir «Guide de démarrage rapide pour AMS sur les plateformes Windows», à la page 627 et «Guide de démarrage rapide pour AMS sur AIX and Linux», à la page 633.

Procédure

1. Créez un répertoire dans lequel créer votre magasin de clés, par exemple `/home/alice/.mqs`. Vous souhaiterez peut-être le créer dans le même répertoire que celui utilisé par le guide de démarrage rapide de votre plateforme. Pour plus d'informations, reportez-vous aux sections «Guide de démarrage rapide pour AMS sur les plateformes Windows», à la page 627 et «Guide de démarrage rapide pour AMS sur AIX and Linux», à la page 633.

   Remarque : Ce répertoire est appelé *keystore-dir* dans les étapes suivantes
2. Création d'un fichier de clés et d'un certificat identifiant l'utilisateur `alice` à utiliser dans le chiffrement

   Remarque : La commande **keytool** fait partie de l'environnement d'exécution Java.

   ```
   keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks
   -storepass passw0rd
   -dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
   ```

   Remarque :

   - Si votre *keystore-dir* contient des espaces, vous devez placer des guillemets autour du nom complet de votre magasin de clés
   - Il est conseillé d'utiliser un mot de passe fiable pour sécuriser le magasin de clés.
   - Pour les besoins de ce guide, nous utilisons un certificat autosigné qui peut être créé sans utiliser d'autorité de certification. Pour les systèmes de production, il est conseillé de ne pas utiliser de certificats autosignés, mais de s'appuyer sur des certificats signés par une autorité de certification.
   - Le paramètre **alias** indique le nom du certificat, que les intercepteurs recherchent pour recevoir les informations nécessaires.
   - Le paramètre **dname** spécifie les détails du **nom distinctif** (DN), qui doit être unique pour chaque utilisateur.
3. Sous AIX and Linux, vérifiez que le magasin de clés est lisible

   ```
   chmod +r keystore-dir/keystore.jks
   ```
4. Répétez l' step1-4 pour l'utilisateur bob

Résultats

Les deux utilisateurs `alice` et bob possèdent chacun un certificat autosigné.

4. Création de keystore.conf

Pourquoi et quand exécuter cette tâche

Vous devez pointer les intercepteurs Advanced Message Security vers le répertoire dans lequel se trouvent les bases de données de clés et les certificats. Cette opération est effectuée via le fichier `keystore.conf` , qui contient ces informations sous forme de texte en clair. Chaque utilisateur doit disposer d'un fichier `keystore.conf` distinct. Cette étape doit être effectuée pour `alice` et bob.

Exemple

Pour ce scénario, le contenu de `keystore.conf` for `alice` est le suivant:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
```

```
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Pour ce scénario, le contenu de `keystore.conf` for bob est le suivant:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Remarque :

- Le chemin d'accès au fichier de clés doit être fourni sans inclure l'extension de fichier.
- Si vous disposez déjà d'un fichier `keystore.conf` car vous avez suivi les instructions du guide de démarrage rapide (Windows ou AIX and Linux), vous pouvez éditer le fichier existant pour ajouter ces lignes.
- Pour plus d'informations, voir «Structure du fichier de configuration du magasin de clés (keystore.conf) pour AMS», à la page 665.

5. Partage de certificats

Pourquoi et quand exécuter cette tâche

Partagez les certificats entre les deux magasins de clés afin que chaque utilisateur puisse identifier l'autre. Cette opération est effectuée en extrayant le certificat de chaque utilisateur et en l'important dans le magasin de clés de l'autre utilisateur.

Important : Les termes *extract* et *export* sont utilisés différemment par les différentes commandes de gestion des certificats.

- La commande IBM Global Security Kit (GSKit) **runmqakm** utilise le terme *extract* pour désigner le processus de copie uniquement de la partie publique d'un certificat à partir d'un magasin de clés, et le terme *export* pour désigner le processus de copie des certificats et de leurs clés publiques et privées associées d'un magasin de clés à un autre.
- La commande Java **keytool** , V 9.4.0 V 9.4.0 et la commande IBM MQ **runmqktool**, utilisent le terme *export* pour désigner le processus de copie uniquement de la partie publique d'un certificat à partir d'un magasin de clés.

Cette distinction est importante car l'utilisation incorrecte de *export* peut compromettre votre application en exposant sa clé privée. Etant donné que la distinction est si importante, la documentation IBM MQ utilise ces termes de manière cohérente. Pour ces raisons, la procédure suivante fait référence à l'*extraction* de certificats à l'aide de l'option `exportcert` de la commande **keytool** .

Procédure

1. Extrayez le certificat identifiant `alice`.

   ```
   keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
   -alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
   ```

2. Importez le certificat identifiant `alice` dans le magasin de clés que bob utilisera. Lorsque vous y êtes invité, indiquez que vous ferez confiance à ce certificat.

   ```
   keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
   -keystore bob-keystore-dir/keystore.jks -storepass passw0rd
   ```

3. Répétez les étapes pour bob

Résultats
Les deux utilisateurs `alice` et bob sont désormais en mesure de s'identifier mutuellement en ayant créé et partagé des certificats autosignés.

Que faire ensuite
Vérifiez qu'un certificat se trouve dans le magasin de clés en exécutant les commandes suivantes qui impriment ses détails:

```
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. Définition de la règle de file d'attente

Pourquoi et quand exécuter cette tâche
Avec le gestionnaire de files d'attente créé et les intercepteurs préparés pour intercepter les messages et les clés de chiffrement d'accès, nous pouvons commencer à définir des règles de protection sur `QM_VERIFY_AMS` à l'aide de la commande `setmqspl` . Pour plus d'informations sur cette commande, voir setmqspl . Chaque nom de règle doit être identique au nom de la file d'attente à laquelle il doit être appliqué.

Exemple
Voici un exemple de règle définie dans la file d'attente `TEST.Q` , signée par l'utilisateur `alice` à l'aide de l'algorithme Deprecated SHA1 et chiffrée à l'aide de l'algorithme AES 256 bits pour l'utilisateur bob:

```
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"
```

Remarque : Les noms distinctifs correspondent exactement à ceux spécifiés dans le certificat de l'utilisateur respectif de la base de données de clés.

Que faire ensuite
Pour vérifier la règle que vous avez définie, exécutez la commande suivante:

```
dspmqspl -m QM_VERIFY_AMS
```

Pour imprimer les détails de la règle sous la forme d'un ensemble de commandes `setmqspl` , utilisez l'indicateur `-export` . Cela permet de stocker des règles déjà définies:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Test de la configuration

Avant de commencer
Vérifiez que les fichiers de règles JCE sans restriction sont installés dans la version de Java que vous utilisez.

Remarque : La version de Java fournie dans l'installation IBM MQ contient déjà ces fichiers de règles. Il se trouve dans *MQ_INSTALLATION_PATH*`/java/bin`.

Pourquoi et quand exécuter cette tâche

En exécutant différents programmes sous différents utilisateurs, vous pouvez vérifier si l'application a été correctement configurée. Pour plus d'informations sur l'exécution de programmes sous différents utilisateurs, voir «Guide de démarrage rapide pour AMS sur les plateformes Windows», à la page 627 et «Guide de démarrage rapide pour AMS sur AIX and Linux», à la page 633.

Procédure

1. Pour exécuter ces modèles d'application JMS , utilisez le paramètre CLASSPATH pour votre plateforme, comme indiqué dans la rubrique Variables d'environnement utilisées par IBM MQ classes for JMS pour vous assurer que le répertoire des exemples est inclus.
2. En tant qu'utilisateur `alice`, placez un message à l'aide d'un exemple d'application, en vous connectant en tant que client:

   ```
   java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

3. En tant qu'utilisateur bob, obtenez un message à l'aide d'un exemple d'application, en vous connectant en tant que client:

   ```
   java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

Résultats

Si l'application a été correctement configurée pour les deux utilisateurs, le message de l'utilisateur `alice` s'affiche lorsque bob exécute l'application d'obtention.

Protection des files d'attente distantes sous AMS

Pour protéger complètement les files d'attente éloignées, des règles doivent être définies sur la file d'attente éloignée et la file d'attente locale à laquelle les messages sont transmis.

Lorsqu'un message est inséré dans une file d'attente éloignée, Advanced Message Security intercepte l'opération et traite le message conformément à un ensemble de règles pour la file d'attente éloignée. Par exemple, pour une règle de chiffrement, le message est chiffré avant d'être transmis à IBM MQ pour le gérer. Une fois que Advanced Message Security a traité le message inséré dans une file d'attente éloignée, IBM MQ le place dans la file d'attente de transmission associée et le transmet au gestionnaire de files d'attente cible et à la file d'attente cible.

Lorsqu'une opération GET est effectuée sur la file d'attente locale, Advanced Message Security tente de décoder le message en fonction de l'ensemble de règles de la file d'attente locale. Pour que l'opération aboutisse, la règle utilisée pour déchiffrer le message doit être identique à celle utilisée pour le chiffrer. Toute différence provoquera le rejet du message.

Si, pour une raison quelconque, les deux règles ne peuvent pas être définies en même temps, une prise en charge du déploiement par étapes est fournie. La règle peut être définie sur une file d'attente locale avec l'indicateur de tolérance activé, qui indique qu'une règle associée à une file d'attente peut être ignorée lorsqu'une tentative d'extraction d'un message de la file d'attente implique un message pour lequel la règle de sécurité n'est pas définie. Dans ce cas, GET tente de déchiffrer le message, mais autorise la distribution de messages non chiffrés. De cette manière, les règles des files d'attente éloignées peuvent être définies une fois que les files d'attente locales ont été protégées (et testées).

A faire : Supprimez l'indicateur de tolérance une fois le déploiement de Advanced Message Security terminé.

Référence associée

setmqspl (définition de la règle de sécurité)

Routage des messages protégés avec AMS à l'aide de IBM Integration Bus

Advanced Message Security peut protéger les messages dans une infrastructure où IBM Integration Busou WebSphere Message Broker 8.0.0.1 (ou version ultérieure) est installé. Vous devez comprendre la nature des deux produits avant d'appliquer la sécurité dans l'environnement IBM Integration Bus .

Pourquoi et quand exécuter cette tâche

Advanced Message Security fournit une sécurité de bout en bout de la charge de message. Cela signifie que seules les parties spécifiées comme expéditeurs et destinataires valides d'un message sont capables de le produire ou de le recevoir. Cela implique que pour sécuriser les messages transitant par IBM Integration Bus, vous pouvez autoriser IBM Integration Bus à traiter les messages sans connaître leur contenu (Scénario 1) ou en faire un utilisateur autorisé à recevoir et à envoyer des messages (Scénario 2).

Scénario 1- Integration Bus ne peut pas voir le contenu des messages

Avant de commencer

IBM Integration Bus doit être connecté à un gestionnaire de files d'attente existant. Remplacez *QMgrName* par ce nom de gestionnaire de files d'attente existant dans les commandes qui suivent.

Pourquoi et quand exécuter cette tâche

Dans ce scénario, Alice place un message protégé dans une file d'attente d'entrée `QIN`. En fonction de la propriété de message `routeTo`, le message est acheminé vers *Bob's* (`QBOB`),[1](`QCECIL`) ou la file d'attente par défaut (`QDEF`). Le routage est possible car Advanced Message Security protège uniquement la charge de message et non ses en-têtes et propriétés qui restent non protégés et peuvent être lus par IBM Integration Bus. Advanced Message Security est utilisé uniquement par *alice*, *bob* et *cecil*. Il n'est pas nécessaire de l'installer ou de le configurer pour IBM Integration Bus.

IBM Integration Bus reçoit le message protégé de la file d'attente d'alias non protégée afin d'éviter toute tentative de déchiffrement du message. S'il devait utiliser directement la file d'attente protégée, le message serait placé dans la file d'attente DEAD LETTER comme impossible à déchiffrer. Le message est acheminé par IBM Integration Bus et arrive dans la file d'attente cible sans modification. Par conséquent, il est toujours signé par l'auteur d'origine (*bob* et *cecil* n'acceptent que les messages envoyés par *alice*) et protégé comme précédemment (seuls *bob* et *cecil* peuvent le lire). IBM Integration Bus place le message acheminé dans un alias non protégé. Les destinataires extraient le message d'une file d'attente en sortie protégée où AMS déchiffre le message de manière transparente.

Procédure

1. Configurez *alice*, *bob* et *cecil* pour utiliser Advanced Message Security comme décrit dans le **Guide de démarrage rapide** (Windows ou AIX).

   Vérifiez que les étapes suivantes sont effectuées:

   - Création et autorisation d'utilisateurs
   - Création de la base de données de clés et des certificats
   - Création de keystore.conf
2. Fournissez le certificat *alice* à *bob* et *cecil*, de sorte que *alice* puisse être identifié par eux lors de la vérification des signatures numériques sur les messages.

   Pour ce faire, extrayez le certificat identifiant *alice* dans un fichier externe, puis ajoutez le certificat extrait aux magasins de clés *Bob's* et *Cecil's* . Il est important d'utiliser la méthode décrite dans la tâche 5 de **. Partage de certificats** dans le **Guide de démarrage rapide** (Windows ou AIX).
3. Fournissez les certificats *bob* et *cecil* à *alice*, de sorte que *alice* puisse envoyer des messages chiffrés pour *bob* et *cecil*.

   Effectuez cette opération à l'aide de la méthode spécifiée à l'étape précédente.

[1] Cecil's

4. Sur votre gestionnaire de files d'attente, définissez des files d'attente locales appelées QIN, QBOB, QCECIL et QDEF.

```
DEFINE QLOCAL(QIN)
```

5. Définissez la règle de sécurité pour la file d'attente QIN sur une configuration éligible. Utilisez la configuration identique pour les files d'attente QBOB, QCECIL et QDEF .

```
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"
```

   Ce scénario suppose la règle de sécurité dans laquelle *alice* est le seul expéditeur autorisé et *bob* et *cecil* sont les destinataires.

6. Définissez des files d'attente alias AIN, ABOB et ACECIL référençant des files d'attente locales QIN, QBOB et QCECIL respectivement.

```
DEFINE QALIAS(AIN) TARGET(QIN)
```

7. Vérifiez que la configuration de sécurité pour les alias spécifiés à l'étape précédente n'est pas présente ; sinon, définissez sa règle sur NONE.

```
dspmqspl -m QMgrName -p AIN
```

8. Dans IBM Integration Bus , créez un flux de messages pour acheminer les messages arrivant dans la file d'attente alias AIN vers le noeud BOB, CECIL ou DEF en fonction de la propriété routeTo du message. Pour ce faire :
   a) Créez un noeud MQInput appelé IN et affectez l'alias AIN comme nom de file d'attente.
   b) Créez des noeuds MQOutput appelés BOB, CECIL et DEFet affectez des files d'attente alias ABOB, ACECIL et ADEF comme noms de file d'attente respectifs.
   c) Créez un noeud de route et appelez-le TEST.
   d) Connectez le noeud IN au terminal d'entrée du noeud TEST .
   e) Créez des terminaux de sortie bobet cecil pour le noeud TEST .
   f) Connectez le terminal de sortie bob au noeud BOB .
   g) Connectez le terminal de sortie cecil au noeud CECIL .
   h) Connectez le noeud DEF au terminal de sortie par défaut.
   i) Appliquez les règles suivantes:

```
$Root/MQRFH2/usr/routeTo/text()="bob"
```

```
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. Déployez le flux de messages dans le composant d'exécution IBM Integration Bus .
10. L'exécution en tant qu'utilisateur Alice a inséré un message qui contient également une propriété de message appelée routeTo avec la valeur bob ou cecil. L'exécution du modèle d'application **amqsstm** vous permet d'effectuer cette opération.

```
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. L'exécution en tant qu'utilisateur *bob* extrait le message de la file d'attente QBOB à l'aide du modèle d'application **amqsget**.

Résultats

Lorsque *alice* place un message dans la file d'attente QIN , le message est protégé. Il est extrait sous forme protégée par IBM Integration Bus à partir de la file d'attente d'alias AIN . IBM Integration Bus décide où acheminer le message en lisant la propriété routeTo qui est, comme toutes les propriétés, non chiffrée. IBM Integration Bus place le message sur l'alias non protégé approprié en évitant sa protection supplémentaire. Lorsqu'il est reçu par *bob* ou *cecil* de la file d'attente, le message est déchiffré et la signature numérique est vérifiée.

Scénario 2- Integration Bus peut voir le contenu des messages

Pourquoi et quand exécuter cette tâche

Dans ce scénario, un groupe d'individus est autorisé à envoyer des messages à IBM Integration Bus. Un autre groupe est autorisé à recevoir les messages créés par IBM Integration Bus. La transmission entre les parties et IBM Integration Bus ne peut pas être espionner.

N'oubliez pas que IBM Integration Bus lit les règles de protection et les certificats uniquement lorsqu'une file d'attente est ouverte. Vous devez donc recharger le groupe d'exécution après avoir apporté des mises à jour aux règles de protection pour que les modifications soient prises en compte.

```
mqsireload execution-group-name
```

Si IBM Integration Bus est considéré comme une partie autorisée à lire ou à signer la charge de message, vous devez configurer Advanced Message Security pour l'utilisateur qui démarre le service IBM Integration Bus . Sachez que ce n'est pas nécessairement le même utilisateur qui insère / extrait les messages dans les files d'attente, ni l'utilisateur qui crée et déploie les applications IBM Integration Bus .

Procédure

1. Configurez *alice*, *bob*, *cecil* et *dave* et l'utilisateur du service IBM Integration Bus pour utiliser Advanced Message Security comme décrit dans le **Guide de démarrage rapide** (Windows ou AIX).

   Vérifiez que les étapes suivantes sont effectuées:

   - Création et autorisation d'utilisateurs
   - Création de la base de données de clés et des certificats
   - Création de keystore.conf
2. Fournissez les certificats *alice*, *bob*, *cecil* et *dave* à l'utilisateur du service IBM Integration Bus .

   Pour ce faire, extrayez chacun des certificats identifiant *alice*, *bob*, *cecil* et *dave* dans des fichiers externes, puis ajoutez les certificats extraits au magasin de clés IBM Integration Bus . Il est important d'utiliser la méthode décrite dans la tâche 5 de **. Partage de certificats** dans le **Guide de démarrage rapide** (Windows ou AIX).
3. Fournissez le certificat de l'utilisateur de service IBM Integration Bus à *alice*, *bob*, *cecil* et *dave*.

   Effectuez cette opération à l'aide de la méthode spécifiée à l'étape précédente.

   Remarque : *Alice* et *bob* ont besoin du certificat de l'utilisateur du service IBM Integration Bus pour chiffrer correctement les messages. L'utilisateur du service IBM Integration Bus a besoin des certificats *alice* et *bob* pour vérifier les auteurs des messages. L'utilisateur du service IBM Integration Bus a besoin des certificats *cecil* et *dave* pour chiffrer les messages qui lui sont destinés. *cecil* et *dave* ont besoin du certificat de l'utilisateur du service IBM Integration Bus pour vérifier si le message provient de IBM Integration Bus.

4. Définissez une file d'attente locale nommée IN et définissez la règle de sécurité avec *alice* et *bob* spécifiés comme auteurs, ainsi que l'utilisateur de service pour le IBM Integration Bus spécifié comme destinataire:

```
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. Définissez une file d'attente locale nommée OUTet définissez la règle de sécurité avec l'utilisateur de service pour IBM Integration Bus spécifié en tant qu'auteur et *cecil* et *dave* spécifié en tant que destinataires:

```
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. Dans IBM Integration Bus , créez un flux de messages avec un noeud MQInput et MQOutput . Configurez le noeud MQInput pour qu'il utilise la file d'attente IN et le noeud MQOutput pour qu'il utilise la file d'attente OUT .
7. Déployez le flux de messages dans le composant d'exécution IBM Integration Bus .
8. L'exécution en tant qu'utilisateur *alice* ou *bob* a inséré un message dans la file d'attente IN à l'aide du modèle d'application **amqsput**.
9. L'exécution en tant qu'utilisateur *cecil* ou *dave* extrait le message de la file d'attente OUT à l'aide du modèle d'application **amqsget**.

Résultats

Les messages envoyés par *alice* ou *bob* à la file d'attente d'entrée IN sont chiffrés, ce qui permet uniquement à IBM Integration Bus de le lire. IBM Integration Bus n'accepte que les messages de *alice* et *bob* et rejette les autres. Les messages acceptés sont traités de manière appropriée, puis signés et chiffrés avec les clés *cecil* et *dave* avant d'être placés dans la file d'attente de sortie OUT. Seuls *cecil* et *dave* sont capables de le lire, les messages non signés par IBM Integration Bus sont rejetés.

Utilisation de Advanced Message Security avec Managed File Transfer

Ce scénario explique comment configurer Advanced Message Security pour fournir la confidentialité des messages pour les données envoyées via un Managed File Transfer.

Avant de commencer

Vérifiez que le composant Advanced Message Security est installé sur l'installation IBM MQ hébergeant les files d'attente utilisées par Managed File Transfer que vous souhaitez protéger.

Si vos agents Managed File Transfer se connectent en mode liaisons, assurez-vous que le composant IBM Global Security Kit (GSKit) est également installé sur leur installation locale.

Pourquoi et quand exécuter cette tâche

Lorsque le transfert de données entre deux agents Managed File Transfer est interrompu, il est possible que des données confidentielles restent non protégées dans les files d'attente IBM MQ sous-jacentes utilisées pour gérer le transfert. Ce scénario explique comment configurer et utiliser Advanced Message Security pour protéger ces données dans les files d'attente Managed File Transfer .

Dans ce scénario, nous considérons une topologie simple comprenant une machine avec deux files d'attente Managed File Transfer et deux agents, AGENT1 et AGENT2, partageant un seul gestionnaire de files d'attente, comme décrit dans le scénario Managed File Transfer. Les deux agents se connectent de la même manière, soit en mode liaisons, soit en mode client.

1. Création de certificats

Avant de commencer

Ce scénario utilise un modèle simple dans lequel un utilisateur `ftagent` d'un groupe FTAGENTS est utilisé pour exécuter les processus Managed File Transfer Agent . Si vous utilisez vos propres noms d'utilisateur et de groupe, modifiez les commandes en conséquence.

Pourquoi et quand exécuter cette tâche

Advanced Message Security utilise la cryptographie à clé publique pour signer et / ou chiffrer les messages dans les files d'attente protégées.

Remarque :

- Si vos agents Managed File Transfer s'exécutent en mode liaisons, les commandes que vous utilisez pour créer un magasin de clés CMS (Cryptographic Message Syntax) sont détaillées dans le **Guide de démarrage rapide** (Windows ou AIX) pour votre plateforme.
- Si vos agents Managed File Transfer s'exécutent en mode client, les commandes dont vous aurez besoin pour créer un fichier de clés JKS (Java Keystore) sont détaillées dans «Guide de démarrage rapide pour AMS avec les clients Java», à la page 650.

Procédure

1. Créez un certificat autosigné pour identifier l'utilisateur `ftagent` comme indiqué dans le guide de démarrage rapide approprié.
   Utilisez un nom distinctif (DN) comme suit:

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. Créez un fichier `keystore.conf` pour identifier l'emplacement du magasin de clés et le certificat qu'il contient, comme indiqué dans le guide de démarrage rapide approprié.

2. Configuration de la protection des messages

Pourquoi et quand exécuter cette tâche

Vous devez définir une règle de sécurité pour la file d'attente de données utilisée par AGENT2, à l'aide de la commande **`setmqspl`** . Dans ce scénario, le même utilisateur est utilisé pour démarrer les deux agents et, par conséquent, le nom distinctif du signataire et du récepteur sont identiques et correspondent au certificat que nous avons généré.

Procédure

1. Arrêtez les agents Managed File Transfer en vue de leur protection à l'aide de la commande **`fteStopAgent`** .
2. Créez une règle de sécurité pour protéger la file d'attente SYSTEM.FTE.DATA.AGENT2 .

   ```
   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. Vérifiez que l'utilisateur exécutant le processus Managed File Transfer Agent a accès à la file d'attente de la règle système et placez les messages dans la file d'attente d'erreurs.

   ```
   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. Redémarrez vos agents Managed File Transfer à l'aide de la commande **fteStartAgent** .
5. Confirmez que vos agents ont été redémarrés avec succès à l'aide de la commande **fteListAgents** et vérifiez qu'ils sont à l'état READY .

Résultats

Vous pouvez maintenant soumettre des transferts depuis AGENT1 vers AGENT2et le contenu du fichier sera transmis de manière sécurisée entre les deux agents.

Présentation de l'installation de Advanced Message Security

Installez le composant Advanced Message Security sur différentes plateformes.

Procédure

- Multi
  Installez Advanced Message Security sur Multiplatforms.
- z/OS
  Installez IBM MQ Advanced for z/OS.
- z/OS
  Installez IBM MQ Advanced for z/OS Value Unit Edition.

Tâches associées

Désinstallation de Advanced Message Security

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 663:

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.

Table 103. CSQ0USMF optional parameters (continued)		
Parameter	**Value**	**Description**
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.

Utilisation de magasins de clés et de certificats avec AMS

Pour fournir une protection cryptographique transparente aux applications IBM MQ , Advanced Message Security utilise le fichier de clés, dans lequel sont stockés les certificats de clé publique et une clé privée. Sous z/OS, un fichier de clés SAF est utilisé à la place d'un fichier de clés.

Dans Advanced Message Security, les utilisateurs et les applications sont représentés par des identités PKI (Public Key Infrastructure). Ce type d'identité est utilisé pour signer et chiffrer les messages. L'identité PKI est représentée par la zone **Nom distinctif (DN)** du sujet dans un certificat associé à des messages signés et chiffrés. Pour qu'un utilisateur ou une application puisse chiffrer ses messages, il doit avoir accès au fichier de clés dans lequel sont stockés les certificats et les clés privées et publiques associées.

ALW Sous AIX, Linux, and Windows, l'emplacement du magasin de clés est fourni dans le fichier de configuration du magasin de clés, qui est `keystore.conf` par défaut. Chaque utilisateur Advanced Message Security doit disposer du fichier de configuration de magasin de clés qui pointe vers un fichier de magasin de clés. Advanced Message Security accepte le format suivant des fichiers de clés: `.kdb`, `.jceks`, `.jks`.

L'emplacement par défaut du fichier `keystore.conf` est:

- Linux IBM i AIX Sous IBM i, AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows Sous Windows : *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

Si vous utilisez un nom de fichier de clés et un emplacement spécifiés, vous devez le spécifier à l'aide de la variable d'environnement **MQS_KEYSTORE_CONF** , comme illustré dans les exemples de commande suivants:

- Pour Java: `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- Pour un client et un serveur C:
  - Linux AIX Sous AIX and Linux : `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windows Sous Windows : `set MQS_KEYSTORE_CONF=`*`path\filename`*

    Remarque : Le chemin d'accès sous Windows peut et doit spécifier l'identificateur d'unité si plusieurs lettres d'unité sont disponibles.

Protection des informations sensibles dans le fichier `keystore.conf`

Pour accéder aux informations sensibles du fichier de clés, telles que les mots de passe, vous devez fournir des jetons afin que IBM MQ Advanced Message Security (AMS) puisse accéder au fichier de clés et signer et chiffrer les messages.

Vous devez protéger les informations sensibles contenues dans le fichier de configuration du magasin de clés à l'aide de la commande **runamscred** fournie avec AMS. Pour plus d'informations sur la protection des fichiers de configuration, voir «Configuration de la protection par mot de passe AMS pour les fichiers de configuration», à la page 684 .

Lors de la protection des mots de passe, vous devez utiliser une clé de chiffrement renforcé personnalisée. Pour accéder aux mots de passe lors de l'exécution, cette clé de chiffrement doit être fournie à AMS.

Il existe deux méthodes pour fournir l'emplacement du fichier de clé de chiffrement, qui sont, via:

- Propriété de configuration **amscred.keyfile** dans le fichier `keystore.conf`
- Variable d'environnement **MQS_AMSCRED_KEYFILE**

L'ordre de priorité est **MQS_AMSCRED_KEYFILE**, suivi de **amscred.keyfile**, puis de la clé par défaut.

Concepts associés

«Noms distinctifs d'expéditeur dans AMS», à la page 693
Les noms distinctifs d'expéditeur identifient les utilisateurs autorisés à placer des messages dans une file d'attente. Un expéditeur utilise son certificat pour signer un message, avant de le placer dans une file d'attente.

«Noms distinctifs des destinataires dans AMS», à la page 694
Les noms distinctifs des destinataires identifient les utilisateurs qui sont autorisés à extraire des messages d'une file d'attente.

Structure du fichier de configuration du magasin de clés (keystore.conf) pour AMS

Le fichier de configuration du magasin de clés (`keystore.conf`) pointe Advanced Message Security vers l'emplacement du magasin de clés approprié.

Chacun des types de fichier de configuration suivants possède un préfixe:

AMSCRED
Paramètres liés au système de protection par mot de passe.

CMS
Certificate Management System, les entrées de configuration sont préfixées avec: `cms.`

PKCS#11
Public Key Cryptography Standard #11, les entrées de configuration sont préfixées avec: `pkcs11.`

IBM i **marque d"erreur d"impression**
Format Privacy Enhanced Mail, les entrées de configuration sont préfixées avec: `pem.`

JKS
Java KeyStore, les entrées de configuration sont préfixées avec: `jks.`

JCEKS
Java Chiffrement cryptographique KeyStore, les entrées de configuration sont préfixées avec: `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
Java Chiffrement cryptographique RACF keyring KeyStore, les entrées de configuration sont préfixées avec: `jceracfks`.

Important : A partir de IBM MQ 9.0 , les valeurs `JCEKS.provider` et `JKS.provider` sont ignorées. Le fournisseur Bouncy Castle est utilisé, en conjonction avec la disposition JCE/JCE fournie par l'environnement d'exécution Java utilisé. Pour plus d'informations, voir «Prise en charge des environnements d'exécution Java nonIBM avec AMS», à la page 669.

Exemples de structures pour les magasins de clés:

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 marque d"erreur d"impression

```
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
```

Tableau 104. Récapitulatif des paramètres requis pour chaque type de fichier de configuration

Paramètres	Obligatoire	Type de fichier de configuration				
		Java (PKCS#11, JKS, JCEKS et JCERACFKS)	**IBM i marque d"erreur d"impression**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			

Tableau 104. Récapitulatif des paramètres requis pour chaque type de fichier de configuration (suite)

Paramètres	Obligatoire	Type de fichier de configuration				
		Java (PKCS#11, JKS, JCEKS et JCERACFKS)	**IBM i marque d"erreur d"impression**	**PKCS#11**	**CMS**	**AMSCRED**
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`						✓ Vous

Notez que vous pouvez ajouter des commentaires à l'aide du symbole # .

Les paramètres du fichier de configuration sont définis comme suit:

keystore

Configuration CMS et Java uniquement.

Chemin d'accès au fichier de clés pour la configuration CMS, JKS et JCEKS.

z/OS MQ Adv. VUE URI du fichier de clés RACF pour la configuration de JCERACFKS.

Important :

- Le chemin d'accès au fichier de clés ne doit pas inclure l'extension de fichier.
- z/OS MQ Adv. VUE L'URI du fichier de clés RACF doit être au format suivant:

```
safkeyring://user/keyring
```

où :

- *user* est l'ID utilisateur propriétaire du fichier de clés
- *keyring* est le nom du fichier de clés.

IBM i **private**
Configuration PEM uniquement.

Nom de fichier d'un fichier contenant une clé privée et un certificat au format PEM.

IBM i **public**
Configuration PEM uniquement.

Nom de fichier d'un fichier contenant des certificats publics de confiance au format PEM.

IBM i **password**
Configuration PEM uniquement.

Mot de passe utilisé pour déchiffrer une clé privée chiffrée.

Vous devez protéger cette zone à l'aide de l'outil de protection par mot de passe AMS natif ; voir «Protection des mots de passe», à la page 669

library
PKCS#11 uniquement.

Nom de chemin de la bibliothèque PKCS#11 .

certificate
Configuration CMS, PKCS#11 et Java uniquement.

Label de certificat

token
PKCS#11 uniquement.

Libellé de jeton.

token_pin
PKCS#11 uniquement.

Code PIN pour déverrouiller le jeton.

Pour les opérations Java uniquement ; vous devez protéger cette zone à l'aide de l'outil de protection par mot de passe Java AMS ; voir «Protection des mots de passe», à la page 669.

Pour les opérations natives uniquement ; vous devez protéger cette zone à l'aide de l'outil de protection par mot de passe AMS natif ; voir «Protection des mots de passe», à la page 669.

secondary_keystore
PKCS#11 uniquement.

Nom de chemin du magasin de clés CMS , fourni sans l'extension `.kdb` , qui contient les certificats d'ancrage (certificats racine) requis par les certificats stockés sur le jeton PKCS #11 . Le magasin de clés secondaire peut également contenir des certificats intermédiaires dans la chaîne de confiance, ainsi que des certificats de destinataire définis dans la politique de sécurité de confidentialité. Ce magasin de clés CMS doit être accompagné d'un fichier de dissimulation qui doit se trouver dans le même répertoire que le magasin de clés secondaire.

Pour les environnements Java , un magasin de clés JKS est requis et vous devez fournir un **secondary_keystore_password**.

secondary_keystore_password
Java PKCS#11 uniquement.

Mot de passe du magasin de clés JKS fourni via la propriété `secondary_keystore` . Vous devez protéger cette zone à l'aide de l'outil de protection par mot de passe Java AMS ; voir «Protection des mots de passe», à la page 669.

encrypted

Java et, depuis IBM MQ 9.3.0, PKCS#11 et 

PEM uniquement.

Statut du mot de passe.

keystore_pass

Configuration de Java uniquement.

Mot de passe du fichier de clés.

Pour les opérations Java uniquement. Vous devez protéger cette zone à l'aide de l'outil de protection par mot de passe Java AMS ; voir «Protection des mots de passe», à la page 669.

key_pass

Configuration de Java uniquement.

Mot de passe de la clé privée de l'utilisateur.

Pour les opérations Java uniquement ; vous devez protéger cette zone à l'aide de l'outil de protection par mot de passe Java AMS ; voir «Protection des mots de passe», à la page 669.

keyfile

Fournit l'emplacement de la clé initiale à utiliser lors de la protection ou du déchiffrement des mots de passe contenus dans ce fichier de configuration ; voir «Protection des mots de passe», à la page 669

provider

Configuration de Java uniquement.

Fournisseur de sécurité Java qui implémente les algorithmes de cryptographie requis par le certificat du magasin de clés.

Important : Les informations stockées dans le magasin de clés sont essentielles pour le flux sécurisé des données envoyées à l'aide de IBM MQ. Les administrateurs de sécurité doivent accorder une attention particulière lorsqu'ils affectent des droits d'accès à ces fichiers.

Protection des mots de passe

Vous devez protéger les mots de passe et les autres informations sensibles contenues dans le fichier `keystore.conf` . Pour plus d'informations, voir **runamscred**.

Exemple de fichier `keystore.conf` :

```
# 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
```

Tâches associées

«Configuration de la protection par mot de passe AMS pour les fichiers de configuration», à la page 684
Le stockage des mots de passe de clés et de clés privées sous forme de texte en clair représente un risque pour la sécurité. Par conséquent, Advanced Message Security fournit un outil qui peut brouiller ces mots de passe à l'aide de la clé d'un utilisateur.

Prise en charge des environnements d'exécution Java nonIBM avec AMS

IBM MQ classes for Java et IBM MQ classes for JMS prennent en charge l'opération Advanced Message Security lors de l'exécution avec des environnements d'exécution Java nonIBM .

Advanced Message Security (AMS) implémente Cryptographic Message Syntax (CMS). La syntaxe CMS est utilisée pour signer numériquement, condenser, authentifier ou chiffrer du contenu de message arbitraire.

Depuis la IBM MQ 9.0, le support Advanced Message Security dans IBM MQ classes for Java et IBM MQ classes for JMS utilise les packages open source Bouncy Castle pour prendre en charge CMS. Cela signifie que ces classes peuvent prendre en charge l'opération Advanced Message Security lors de l'exécution avec des environnements d'exécution Java nonIBM .

Avant IBM MQ 9.0, Advanced Message Security n'était pas pris en charge dans les environnements d'exécution Java nonIBM dans les clients Java . La prise en charge de Advanced Message Security dans IBM MQ classes for Java et IBM MQ classes for JMS dépendait de la prise en charge de CMS spécifiquement fournie par l'implémentation IBM de JCE (Java Cryptography Extensions). En raison de cette restriction, la fonctionnalité n'était disponible que lors de l'utilisation d'un environnement Java runtime environment (JRE) incluant le fournisseur JCE Java .

Emplacement et numérotation des versions pour les fichiers JAR de Bouncy Castle

Les fichiers JAR Bouncy Castle qui sont requis pour la prise en charge des environnements d'exécution Java nonIBM sont inclus dans le package d'installation IBM MQ classes for Java et IBM MQ classes for JMS .

Les fichiers JAR Bouncy Castle utilisés sont les suivants:

Le fichier JAR du fournisseur, qui est fondamental pour les opérations Bouncy Castle.

V 9.4.0 Depuis IBM MQ 9.4.0, ce fichier JAR est appelé `bcprov-jdk18on.jar`.

Le fichier JAR "PKIX", qui contient la prise en charge des opérations CMS utilisées par Advanced Message Security.

V 9.4.0 Depuis IBM MQ 9.4.0, ce fichier JAR est appelé `bcpkix-jdk18on.jar`.

Le fichier JAR "util", qui contient les classes utilisées par les autres fichiers JAR Bouncy Castle.

V 9.4.0 Depuis IBM MQ 9.4.0, ce fichier JAR est appelé `bcutil-jdk18on.jar`.

Dépendances

Les classes IBM MQ 9.1 et ultérieures ont été testées avec des environnements d'exécution Java IBM et des environnements d'exécution Java Oracle . Ils sont également susceptibles de s'exécuter avec succès dans n'importe quel environnement d'exécution Java J2SE-compliant . Toutefois, vous devez noter les dépendances suivantes:

- Aucune modification n'a été apportée à la configuration de Advanced Message Security .
- Les classes Bouncy Castle sont utilisées uniquement pour les opérations CMS . Toutes les autres opérations liées à la sécurité, par exemple l'accès au magasin de clés, le chiffrement réel des données et le calcul des totaux de contrôle de signature utilisent la fonctionnalité fournie par l'environnement d'exécution Java.

  Important : Pour cette raison, l'environnement d'exécution Java utilisé doit inclure une implémentation de fournisseur JCE.
- Pour utiliser des algorithmes de chiffrement *fort* , vous devrez peut-être installer les fichiers de règles *sans restriction* pour l'implémentation JCE de l'environnement d'exécution Java.

  Pour plus de détails, reportez-vous à la documentation de l'environnement d'exécution Java.
- Si vous avez activé la sécurité Java :
  - Ajoutez `java.security.SecurityPermissioninsertProvider.BC` à l'application pour que les classes Bouncy Castle puissent être utilisées comme fournisseur de sécurité.
  - Accordez `java.security.AllPermission` aux fichiers JAR Bouncy Castle.

    V 9.4.0 Depuis IBM MQ 9.4.0, ces fichiers sont les suivants:

    mq_install_dir`/java/lib/bcutil-jdk18on.jar`
    mq_install_dir`/java/lib/bcpkix-jdk18on.jar`
    mq_install_dir`/java/lib/bcprov-jdk18on.jar`

Concepts associés

Ce qui est installé pour IBM MQ classes for JMS

Ce qui est installé pour les classes IBM MQ pour Java

Multi Interception MCA (Message Channel Agent) et AMS

L'interception MCA permet à un gestionnaire de files d'attente s'exécutant sous IBM MQ d'activer de manière sélective les règles à appliquer pour les canaux de connexion serveur.

L'interception MCA permet aux clients qui restent en dehors de AMS d'être toujours connectés à un gestionnaire de files d'attente et de chiffrer et déchiffrer leurs messages.

L'interception MCA est destinée à fournir la fonction AMS lorsque AMS ne peut pas être activé sur le client. Notez que l'utilisation de l'interception MCA et d'un client compatible avec AMSentraîne une double protection des messages qui peut être problématique pour la réception des applications. Pour plus d'informations, voir «Désactivation d'Advanced Message Security sur le client», à la page 674.

Remarque : Les intercepteurs MCA ne sont pas pris en charge pour les canaux AMQP ou MQTT.

Fichier de configuration du magasin de clés

Par défaut, le fichier de configuration du magasin de clés pour l'interception MCA est `keystore.conf` et se trouve dans le répertoire `.mqs` du répertoire HOME de l'utilisateur qui a démarré le gestionnaire de files d'attente ou le programme d'écoute. Le magasin de clés peut également être configuré à l'aide de la variable d'environnement MQS_KEYSTORE_CONF. Pour plus d'informations sur la configuration du magasin de clés AMS , voir «Utilisation de magasins de clés et de certificats avec AMS», à la page 664.

Pour activer l'interception MCA, vous devez fournir le nom d'un canal que vous souhaitez utiliser dans le fichier de configuration du magasin de clés. Pour l'interception MCA, seul un type de magasin de clés CMS peut être utilisé.

Voir «Exemple d'interception MCA pour AMS», à la page 671 pour un exemple de configuration de l'interception MCA.

Avertissement : Vous devez effectuer l'authentification et le chiffrement des clients sur les canaux sélectionnés, par exemple, en utilisant SSL et SSLPEER ou CHLAUTH TYPE (SSLPEERMAP), pour vous assurer que seuls les clients autorisés peuvent se connecter et utiliser cette fonction.

IBM i

Si votre entreprise utilise IBM iet que vous avez sélectionné une autorité de certification commerciale pour signer votre certificat, le Certificate Manager numérique crée une demande de certificat au format PEM (Privacy-Enhanced Mail). Vous devez transmettre la demande à l'autorité de certification choisie.

Pour ce faire, vous devez utiliser la commande suivante pour sélectionner le certificat approprié pour le canal spécifié dans `channelname`:

```
pem.certificate.channel.channelname
```

Exemple d'interception MCA pour AMS

Exemple de tâche de configuration d'une interception MCA AMS .

Avant de commencer

Avertissement : Vous devez effectuer l'authentification et le chiffrement des clients sur les canaux sélectionnés, par exemple, en utilisant SSL et SSLPEER ou CHLAUTH TYPE (SSLPEERMAP), pour vous assurer que seuls les clients autorisés peuvent se connecter et utiliser cette fonction.

Si votre entreprise utilise IBM iet que vous avez sélectionné une autorité de certification commerciale pour signer votre certificat, le Certificate Manager numérique crée une demande de certificat au format PEM (Privacy-Enhanced Mail). Vous devez transmettre la demande à l'autorité de certification choisie.

Pourquoi et quand exécuter cette tâche

Cette tâche vous guide tout au long du processus de configuration de votre système pour utiliser l'interception MCA, puis de vérification de la configuration.

Remarque : IBM MQinclut les intercepteurs AMS et les active de manière dynamique dans les environnements d'exécution client et serveur MQ .

Avertissement :

- Remplacez `userID` dans le code par votre ID utilisateur.
- La procédure suivante ne fonctionne pas comme prévu dans IBM MQ sauf si l'interception AMS est désactivée sur le client.

Procédure

1. Créez la base de données de clés et les certificats à l'aide des commandes suivantes pour créer un script shell.

   Modifiez également les paramètres **INSTLOC** et **KEYSTORELOC** ou exécutez les commandes requises. Notez que vous n'avez peut-être pas besoin de créer le certificat pour bob.

```
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. Partagez les certificats entre les deux bases de données de clés afin que chaque utilisateur puisse identifier l'autre.

   Il est important d'utiliser la méthode décrite pour le partage de certificats dans le *Guide de démarrage rapide*, pour la plateforme utilisée par votre entreprise:

   Windows
   Certificats de partage de la tâche 5

   AIX and Linux
   Certificats de partage de la tâche 5

   Java clients
   Certificats de partage de la tâche 5

3. Créez `keystore.conf` avec la configuration suivante: `Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

   Avertissement :

   a. Le magasin de clés doit se trouver sur le système où se trouve le gestionnaire de files d'attente.

   b. Vous devez spécifier un canal spécifique pour `cms.certificate` afin d'activer l'intervention MCA, puis le gestionnaire de files d'attente exécute des opérations AMS sur les applications qui se connectent via ce canal à des files d'attente avec des règles définies.

4. Création et démarrage du gestionnaire de files d'attente `AMSQMGR1`

5. Définissez un programme d'écoute TCP à l'aide d'un numéro de port disponible sous le contrôle QMGR.

   Exemple :

   ```
   DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
   ```

6. Démarrez le programme d'écoute et vérifiez qu'il a démarré correctement.

   Exemple :

   ```
   START  LISTENER(MY.LISTENER)
   DISPLAY LSSTATUS(MY.LISTENER) PORT
   ```

7. Arrêtez le gestionnaire de files d'attente.
8. Définissez le magasin de clés:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. Démarrez le gestionnaire de files d'attente sur le même interpréteur de commandes, de sorte que la variable d'environnement MQS_KEYSTORE_CONF soit disponible pour le gestionnaire de files d'attente.
10. Définissez la stratégie de sécurité et vérifiez:

```
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
```

    Pour plus d'informations, voir setmqspl et dspmqspl .

11. Définissez la variable d'environnement *MQSERVER* :

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. Supprimez la règle de sécurité et vérifiez le résultat:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

13. Parcourez la file d'attente à partir de votre installation IBM MQ 9.4 :

```
/opt/mq93/samp/bin/amqsbcg TESTQ AMSQMGR1
```

    La sortie de navigation affiche les messages au format chiffré.

14. Définissez la règle de sécurité et vérifiez le résultat:

```
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. Exécutez **amqsgetc** à partir de votre installation IBM MQ 9.4 :

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

Concepts associés

«Structure du fichier de configuration du magasin de clés (keystore.conf) pour AMS», à la page 665
Le fichier de configuration du magasin de clés (`keystore.conf`) pointe Advanced Message Security vers l'emplacement du magasin de clés approprié.

Référence associée

«Limitations connues de AMS», à la page 621
Un certain nombre d'options IBM MQ ne sont pas prises en charge ou sont soumises à des limitations pour Advanced Message Security.

Désactivation d'Advanced Message Security sur le client

Vous devez désactiver IBM MQ Advanced Message Security (AMS) si vous utilisez un client IBM MQ pour vous connecter à un gestionnaire de files d'attente à partir d'une version antérieure du produit et qu'une erreur `2085 (MQRC_UNKNOWN_OBJECT_NAME)` est signalée.

Pourquoi et quand exécuter cette tâche

IBM MQ Advanced Message Security (AMS) étant automatiquement activé dans un client IBM MQ , le client tente par défaut de vérifier les règles de sécurité des objets dans le gestionnaire de files d'attente.

Si cette erreur est signalée, lorsque vous tentez de vous connecter à un gestionnaire de files d'attente à partir d'une version antérieure du produit, vous pouvez désactiver AMS comme suit:

- Pour les clients Java, de l'une des façons suivantes :
  - En définissant une variable d'environnement **AMQ_DISABLE_CLIENT_AMS**.
  - En définissant la propriété système Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS.
  - A l'aide de la propriété **DisableClientAMS** , sous la section Sécurité du fichier `mqclient.ini` .
- Pour les clients C, en définissant une variable d'environnement **MQS_DISABLE_ALL_INTERCEPT**.

Remarque : Vous ne pouvez pas utiliser la variable d'environnement **AMQ_DISABLE_CLIENT_AMS** pour les clients C. Vous devez utiliser la variable d'environnement **MQS_DISABLE_ALL_INTERCEPT** à la place.

Procédure

- Pour désactiver AMS sur le client, utilisez l'une des options suivantes:

  Variable d'environnement AMQ_DISABLE_CLIENT_AMS
  Vous devez définir cette variable dans les cas suivants:
  - Si vous utilisez un environnement Java runtime environment (JRE) autre que IBM Java runtime environment (JRE)
  - Si vous utilisez un client IBM MQ IBM MQ classes for JMS ou IBM MQ classes for Java .

  Créez la variable d'environnement **AMQ_DISABLE_CLIENT_AMS** et définissez-la sur TRUE dans l'environnement où l'application s'exécute. Exemple :

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  Propriété système Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  Pour les clients IBM MQ classes for JMS et IBM MQ classes for Java , vous pouvez définir la propriété système Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS sur la valeur TRUE pour l'application Java .

  Par exemple, vous pouvez définir la propriété système Java en tant qu'option `-D` lorsque la commande Java est appelée:

  ```
  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
  ```

  Vous pouvez également spécifier la propriété système Java dans un fichier de configuration JMS , `jms.config`, si l'application utilise ce fichier.

  variable d'environnement MQS_DISABLE_ALL_INTERCEPT
  Vous devez définir cette variable d'environnement si vous utilisez IBM MQ avec des clients natifs et que vous devez désactiver AMS sur le client.

Créez la variable d'environnement **MQS_DISABLE_ALL_INTERCEPT** et définissez-la sur TRUE dans l'environnement où le client s'exécute. Exemple :

```
export MQS_DISABLE_ALL_INTERCEPT =TRUE
```

Vous pouvez utiliser la variable d'environnement **MQS_DISABLE_ALL_INTERCEPT** pour les clients C uniquement. Pour les clients Java , vous devez utiliser la variable d'environnement **AMQ_DISABLE_CLIENT_AMS** à la place.

Propriété DisableClientAMS dans le fichier mqclient.ini

Vous pouvez utiliser cette option pour les clients IBM MQ classes for JMS et IBM MQ classes for Java et pour les clients C.

Ajoutez le nom de propriété DisableClientAMS sous la section **Security** du fichier mqclient.ini , comme illustré dans l'exemple suivant:

```
Security:
DisableClientAMS=Yes
```

Vous pouvez également activer AMS comme illustré dans l'exemple suivant:

```
Security:
DisableClientAMS=No
```

Que faire ensuite

Pour plus d'informations sur les problèmes liés à l'ouverture de files d'attente protégées AMS , voir Problèmes liés à l'ouverture de files d'attente protégées lors de l'utilisation de AMS avec JMS.

Concepts associés

«Interception MCA (Message Channel Agent) et AMS», à la page 671
L'interception MCA permet à un gestionnaire de files d'attente s'exécutant sous IBM MQ d'activer de manière sélective les règles à appliquer pour les canaux de connexion serveur.

Tâches associées

fichier de configuration IBM MQ MQI client , mqclient.ini

Référence associée

Fichier de configuration IBM MQ classes for JMS

Exigences de certificat pour AMS

Les certificats doivent disposer d'une clé publique RSA pour pouvoir être utilisés avec Advanced Message Security.

Pour plus d'informations sur les différents types de clé publique et pour savoir comment les créer, voir «Certificats numériques et compatibilité CipherSpec dans IBM MQ», à la page 49.

Extensions d'utilisation de clé

Les extensions d'utilisation de clé imposent des restrictions supplémentaires sur la façon dont un certificat peut être utilisé.

Dans Advanced Message Security, l'utilisation des clés des certificats X.509 v3 doit être définie conformément à la spécification RFC 5280.

Pour la qualité de l'intégrité de la protection, si des extensions d'utilisation de clé de certificat sont définies, cet ensemble doit inclure au moins l'une des deux suivantes:

- **nonRepudiation**
- **digitalSignature**

Pour la qualité de la confidentialité de la protection, si des extensions d'utilisation de clé de certificat sont définies, cet ensemble doit inclure:

- **keyEncipherment**

Pour la qualité de la confidentialité de la protection, si des extensions d'utilisation de clé de certificat sont définies, cet ensemble doit inclure:

- **dataEncipherment**

L'utilisation étendue des clés permet d'affiner davantage les extensions d'utilisation des clés. Pour toutes les qualités de protection, si l'utilisation de la clé étendue du certificat est définie, l'ensemble doit inclure:

- **emailProtection**

Concepts associés

«Qualité de protection dans AMS», à la page 696
Les règles de protection des données Advanced Message Security impliquent une qualité de protection (QOP).

Méthodes de validation de certificat dans AMS

Vous pouvez utiliser Advanced Message Security pour détecter et rejeter les certificats révoqués afin que les messages de vos files d'attente ne soient pas protégés à l'aide de certificats qui ne répondent pas aux normes de sécurité.

AMS vous permet de vérifier la validité d'un certificat à l'aide du protocole OCSP (Online Certificate Status Protocol) ou de la liste de révocation de certificat (CRL).

AMS peut être configuré pour la vérification OCSP et/ou CRL. Si les deux méthodes sont activées, AMS utilise d'abord le protocole OCSP pour le statut de révocation pour des raisons de performances. Si le statut de révocation d'un certificat est indéterminé après la vérification OCSP, AMS utilise la vérification CRL.

Notez que la vérification OCSP et CRL sont activées par défaut.

Concepts associés

«Online Certificate Status Protocol (OCSP) dans AMS», à la page 676
Le protocole OCSP (Online Certificate Status Protocol) détermine si un certificat a été révoqué et, par conséquent, permet de déterminer si le certificat est digne de confiance. OCSP est activé par défaut.

«Listes de révocation de certificat (CRL) dans AMS», à la page 678
Les listes CRL contiennent une liste de certificats qui ont été marqués par l'autorité de certification comme n'étant plus dignes de confiance pour diverses raisons, par exemple, la clé privée a été perdue ou compromise.

Online Certificate Status Protocol (OCSP) dans AMS

Le protocole OCSP (Online Certificate Status Protocol) détermine si un certificat a été révoqué et, par conséquent, permet de déterminer si le certificat est digne de confiance. OCSP est activé par défaut.

OCSP n'est pas pris en charge sur les sytems IBM i .

Activation de la vérification OCSP dans les intercepteurs natifs de Advanced Message Security

La restitution du protocole OCSP (Online Certificate Status Protocol) dans Advanced Message Security est activée par défaut, en fonction des informations contenues dans les certificats utilisés.

Procédure

Ajoutez les options suivantes au fichier de configuration du magasin de clés :

Remarque : Toutes les strophes OCSP sont facultatives et peuvent être spécifiées indépendamment.

Option	Description
ocsp.enable=off	Activez la vérification OCSP si le certificat vérifié possède une extension AIA (Authority Info Access) avec une méthode d'accès PKIX_AD_OCSP contenant un URI de l'emplacement du répondeur OCSP. Valeurs possibles: on ou off.
ocsp.url=*responder_URL*	Adresse URL du canal répondeur OCSP. Si cette option est omise, la vérification OCSP non AIA est désactivée.
ocsp.http.proxy.host=*OCSP_proxy*	Adresse URL du serveur proxy OCSP. Si cette option est omise, un proxy n'est pas utilisé pour les vérifications de certificats en ligne non AIA.
ocsp.http.proxy.port=*port_number*	Numéro de port du serveur proxy OCSP. Si cette option est omise, le port par défaut 8080 est utilisé.
ocsp.nonce.generation=on/off	Génère une valeur nonce lors de l'interrogation d'OCSP. La valeur par défaut est off.
ocsp.nonce.check=on/off	Vérifie la valeur nonce après la réception d'une réponse d'OCSP. La valeur par défaut est off.
ocsp.nonce.size=8	Taille de la valeur nonce en octets.
ocsp.http.get=on/off	Spécifie HTTP GET comme méthode d'interrogation. Si cette option est définie sur off, HTTP POST est utilisé. La valeur par défaut est off.
ocsp.max_response_size=20480	Taille maximale de la réponse (en octets) du canal répondeur OCSP fourni.
ocsp.cache_size=100	Active la mise en cache de la réponse OCSP interne et définit la limite du nombre d'entrées du cache.
ocsp.timeout=30	Temps d'attente d'une réponse serveur (en secondes) après laquelle Advanced Message Security expire.
ocsp.unknown=ACCEPT	Définit le comportement lorsqu'un serveur OCSP ne peut pas être atteint dans un délai imparti. Valeurs possibles : • ACCEPT Permet le certificat • WARN Permet le certificat et consigne un avertissement • REJECT Empêche l'utilisation du certificat et consigne une erreur

Activation de la restitution OCSP dans Java dans AMS
Pour activer la vérification OCSP pour Java dans Advanced Message Security, modifiez le fichier java.security ou le fichier de configuration du magasin de clés.

Pourquoi et quand exécuter cette tâche

Il existe deux façons d'activer la restitution OCSP dans Advanced Message Security:

Utilisation de java.security
Vérifiez si votre certificat contient une extension de certificat AIA (Authority Information Access).

Procédure

1. Si AIA n'est pas configuré ou si vous souhaitez remplacer votre certificat, éditez le fichier `$JAVA_HOME/lib/security/java.security` avec les propriétés suivantes:

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   et activez la vérification OCSP en éditant le fichier `$JAVA_HOME/lib/security/java.security` avec la ligne suivante:

   ```
   ocsp.enable=true
   ```

2. Si AIA est configuré, activez la vérification OCSP en éditant le fichier `$JAVA_HOME/lib/security/java.security` avec la ligne suivante:

   ```
   ocsp.enable=true
   ```

Que faire ensuite

Si vous utilisez Java Security Manager, effectuez également la configuration, ajoutez le droit Java suivant à `lib/security/java.policy`

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

Utilisation de keystore.conf

Procédure

Ajoutez l'attribut suivant au fichier de configuration:

```
ocsp.enable=true
```

Important : La définition de cet attribut dans le fichier de configuration remplace les paramètres java.security .

Que faire ensuite

Pour terminer la configuration, ajoutez les droits Java suivants à `lib/security/java.policy`:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

Listes de révocation de certificat (CRL) dans AMS

Les listes CRL contiennent une liste de certificats qui ont été marqués par l'autorité de certification comme n'étant plus dignes de confiance pour diverses raisons, par exemple, la clé privée a été perdue ou compromise.

Pour valider les certificats, Advanced Message Security construit une chaîne de certificats qui se compose du certificat du signataire et de la chaîne de certificats de l'autorité de certification jusqu'à un point d'ancrage digne de confiance. Un point d'ancrage digne de confiance est un fichier de clés certifiées qui

contient un certificat digne de confiance ou un certificat racine digne de confiance utilisé pour vérifier la confiance d'un certificat. AMS vérifie le chemin du certificat à l'aide d'un algorithme de validation PKIX. Lorsque la chaîne est créée et vérifiée, AMS effectue la validation de certificat qui inclut la validation de la date d'émission et d'expiration de chaque certificat de la chaîne par rapport à la date en cours, en vérifiant si l'extension d'utilisation de clé est présente dans le certificat d'entité finale. Si l'extension est ajoutée au certificat, AMS vérifie si **digitalSignature** ou **nonRepudiation** sont également définis. Si ce n'est pas le cas, le MQRC_SECURITY_ERROR est signalé et consigné. Ensuite, AMS télécharge les CRL à partir de fichiers ou de LDAP en fonction des valeurs spécifiées dans le fichier de configuration. Seules les listes de révocation de certificat codées au format DER sont prises en charge par AMS. Si aucune configuration liée à la liste de révocation de certificat n'est trouvée dans le fichier de configuration du magasin de clés, AMS n'effectue aucune vérification de validité de la liste de révocation de certificat. Pour chaque certificat de l'autorité de certification, AMS demande à LDAP des listes de révocation de certificat à l'aide des noms distinctifs d'une autorité de certification pour trouver sa liste de révocation de certificat. Les attributs suivants sont inclus dans la requête LDAP:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

Remarque : deltaRevocationList est pris en charge uniquement lorsqu'il est spécifié en tant que points de distribution.

Activation de la prise en charge de la validation de certificat et de la liste de révocation de certificat dans les intercepteurs natifs
Vous devez modifier le fichier de configuration du magasin de clés afin que Advanced Message Security puisse télécharger des CLR à partir du serveur LDAP (Lightweight Directory Access Protocol).

Pourquoi et quand exécuter cette tâche

IBM i L'activation de la prise en charge de la validation de certificat et de la liste de révocation de certificat dans les intercepteurs natifs n'est pas prise en charge pour Advanced Message Security sur IBM i.

Procédure

Ajoutez les options suivantes au fichier de configuration:

Remarque : Toutes les sections CRL sont facultatives et peuvent être spécifiées indépendamment.

Option	Description
crl.ldap.host=*host_name*	Nom d'hôte du serveur LDAP.
crl.ldap.port=*port_number*	Numéro de port du serveur LDAP. Vous pouvez spécifier jusqu'à 11 serveurs. Plusieurs hôtes LDAP sont utilisés pour garantir une reprise en ligne transparente en cas d'échec de la connexion LDAP. Tous les serveurs LDAP doivent être des répliques et contenir les mêmes données. Lorsque l'intercepteur AMS Java parvient à se connecter à un serveur LDAP, il ne tente pas de télécharger des listes de révocation de certificat à partir des serveurs restants fournis.

Option	Description
`crl.cdp=off`	Utilisez cette option pour vérifier ou utiliser les extensions CRLDistributionPoints dans les certificats.
`crl.ldap.version=3`	Numéro de version du protocole LDAP. Valeurs possibles: 2 ou 3.
`crl.ldap.user=cn=`*`username`*	Connectez-vous au serveur LDAP. Si cette valeur n'est pas spécifiée, les attributs CRL dans LDAP doivent être lisibles par tous
`crl.ldap.pass=`*`password`*	Mot de passe du serveur LDAP.
`crl.ldap.encrypted=`*`no/yes`*	Indique si le `clr.ldap.pass` est chiffré ou non. Pour plus d'informations, voir Protection des mots de passe dans les fichiers de configuration AMS .
`crl.ldap.cache_lifetime=0`	Durée de vie du cache LDAP en secondes. Valeurs possibles: 0 à 86400.
`crl.ldap.cache_size=50`	Taille du cache LDAP. Cette option ne peut être spécifiée que si la valeur `crl.ldap.cache_lifetime` est supérieure à `0`.
`crl.http.proxy.host=some.host.com`	Port du serveur proxy HTTP pour l'extraction de la liste de révocation de certificat CDP.
`crl.http.proxy.port=8080`	Numéro de port du serveur proxy HTTP.
`crl.http.max_response_size=204800`	Taille maximale de la CRL, en octets, qui peut être extraite d'un serveur HTTP accepté par IBM Global Security Kit (GSKit).
`crl.http.timeout=30`	Délai d'attente d'une réponse du serveur, en secondes, après lequel AMS arrive à expiration.
`crl.http.cache_size=0`	Taille du cache HTTP, en octets.
`crl.unknown=ACCEPT`	Définit le comportement lorsqu'un serveur CRL ne peut pas être atteint dans un délai imparti. Valeurs possibles : • `ACCEPT` Permet le certificat • `WARN` Permet le certificat et consigne un avertissement • `REJECT` Empêche l'utilisation du certificat et consigne une erreur

Activation de la prise en charge de la liste de révocation de certificat dans Java dans AMS
Pour activer la prise en charge de CRL dans Advanced Message Security, vous devez modifier le fichier de configuration du magasin de clés afin de permettre à AMS de télécharger des CRL à partir du serveur LDAP (Lightweight Directory Access Protocol) et de configurer le fichier java.security .

Procédure

1. Ajoutez les options suivantes au fichier de configuration:

En-tête	Description
`crl.ldap.host=`*`host_name`*	Nom d'hôte LDAP.

En-tête	Description
`crl.ldap.port=`*`port_number`*	Numéro de port du serveur LDAP. Vous pouvez spécifier jusqu'à 11 serveurs. Plusieurs hôtes LDAP sont utilisés pour garantir une reprise en ligne transparente en cas d'échec de la connexion LDAP. Tous les serveurs LDAP doivent être des répliques et contenir les mêmes données. Lorsque l'intercepteur AMS Java parvient à se connecter à un serveur LDAP, il ne tente pas de télécharger des listes de révocation de certificat à partir des serveurs restants fournis. Java n'utilise pas les valeurs `crl.ldap.user` et crl.ldaworldp.pass . Il n'utilise pas d'utilisateur ni de mot de passe lors de la connexion à un serveur LDAP. Par conséquent, les attributs CRL dans LDAP doivent être lisibles par tous.
`crl.cdp=on/off`	Utilisez cette option pour vérifier ou utiliser les extensions CRLDistributionPoints dans les certificats.

2. Modifiez le fichier JRE/`lib/security/java.security` avec les propriétés suivantes:

Nom de la propriété	Description
`com.ibm.security.enableCRLDP`	Cette propriété prend les valeurs suivantes: `true`, `false`. S'il est défini sur `true`, lors de la vérification de la révocation de certificat, les listes de révocation de certificat sont localisées à l'aide de l'URL de l'extension des points de distribution de liste de révocation de certificat du certificat. S'il est défini sur `false` ou s'il n'est pas défini, la vérification de la liste de révocation de certificat à l'aide de l'extension des points de distribution de liste de révocation de certificat est désactivée.
`ibm.security.certpath.ldap.cache.lifetime`	Cette propriété peut être utilisée pour définir la durée de vie des entrées dans le cache mémoire de CertStore LDAP sur une valeur en secondes. La valeur 0 désactive le cache ; -1 signifie une durée de vie illimitée. S'il n'est pas défini, la durée de vie par défaut est de 30 secondes.

Nom de la propriété	Description
`com.ibm.security.enableAIAEXT`	Cette propriété prend les valeurs suivantes: `true`, `false`. Si la valeur est `true`, toutes les extensions d'accès aux informations d'autorité trouvées dans les certificats du chemin de certificat en cours de génération sont examinées afin de déterminer si elles contiennent des URI LDAP. Pour chaque URI LDAP trouvé, un objet LDAPCertStore est créé et ajouté à la collection CertStores qui est utilisée pour localiser les autres certificats requis pour générer le chemin de certificat. S'il est défini sur `false` ou s'il n'est pas défini, des objets LDAPCertStore supplémentaires ne sont pas créés.

z/OS *Activation des listes de révocation de certificats (CRL) sous z/OS*

Advanced Message Security prend en charge la vérification CRL (Certificate Revocation List) des certificats numériques utilisés pour protéger les messages de données

Pourquoi et quand exécuter cette tâche

Lorsque cette option est activée, Advanced Message Security valide les certificats de destinataire lorsque les messages sont placés dans une file d'attente protégée contre la confidentialité et valide les certificats d'expéditeur lorsque les messages sont extraits d'une file d'attente protégée (intégrité ou confidentialité). Dans ce cas, la validation comprend la vérification que les certificats pertinents ne sont pas enregistrés dans une LCR pertinente.

Advanced Message Security utilise les services IBM System SSL pour valider les certificats d'expéditeur et de destinataire. Vous pouvez trouver une documentation détaillée concernant la validation du certificat SSL système dans lez/OS Programmation de la couche de sockets sécurisés du système de services cryptographiques manuel.

Pour activer la vérification CRL, vous devez spécifier l'emplacement d'un fichier de configuration CRL via la définition de données CRLFILE dans le JCL de la tâche démarrée pour l'espace adresse AMS. Un exemple de fichier de configuration CRL pouvant être personnalisé est fourni dans *thlqual*.SCSQPROC (CSQ40CRL). Les paramètres autorisés dans ce fichier sont les suivants:

Tableau 105. Variables de configuration CRL Advanced Message Security

Variable	Valeur valides	Description
crl.ldap.host[.n]	*hostname -or-hostname: port*	Adresse IP / nom d'hôte de votre serveur LDAP qui héberge les CRL de vos certificats d'émetteur. Si vous n'indiquez pas de numéro de port pour votre serveur LDAP, le numéro de port spécifié par crl.ldap.port est utilisé.
crl.ldap.port	*port*	Numéro de port TCP/IP de votre serveur LDAP.
crl.ldap.user	*ldap_user*	Nom d'utilisateur LDAP à utiliser lors de la connexion au serveur LDAP.

Tableau 105. Variables de configuration CRL Advanced Message Security (suite)		
Variable	**Valeur valides**	**Description**
crl.ldap.pass	*mot_de_passe_ldap*	Mot de passe LDAP associé à crl.ldap.user.

Vous pouvez spécifier plusieurs noms d'hôte et ports de serveur LDAP comme suit:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

Vous pouvez spécifier jusqu'à 10 noms d'hôte. Si vous n'indiquez pas de numéro de port pour vos serveurs LDAP, le numéro de port spécifié par crl.ldap.port est utilisé. Chaque serveur LDAP doit utiliser la même combinaison crl.ldap.user/password pour l'accès.

Lorsque la définition de données CRLFILE est spécifiée, la configuration est chargée lors de l'initialisation de l'espace adresse Advanced Message Security et la vérification de la liste de révocation de certificat est activée. Si la définition de données CRLFILE n'est pas spécifiée, ou si le fichier de configuration de la liste de révocation de certificat n'est pas disponible ou n'est pas valide, la vérification de la liste de révocation de certificat est désactivée.

AMS effectue une vérification CRL à l'aide des services de validation de certificat SSL IBM System comme suit:

Tableau 106. Vérifications de la liste de révocation de certificat Advanced Message Security		
Opération	**Qualité de protection**	**Certificat (s) vérifié (s)**
PUT	Confidentialité	Destinataire(s)
GET	Intégrité / Confidentialité	Expéditeur

Si une opération de message échoue, une vérification CRL Advanced Message Security effectue les actions suivantes:

Tableau 107. Comportement d'échec de vérification de la liste de révocation de certificat Advanced Message Security	
Opération	**Echec de la vérification CRL**
PUT	Le message n'est pas inséré dans la file d'attente cible. Le code achèvement MQCC_FAILED et le code anomalie MQRC_SECURITY_ERROR sont renvoyés à l'application.
GET	Le message est supprimé de la file d'attente cible et déplacé vers la file d'attente d'erreurs de protection du système. Le code achèvement MQCC_FAILED et le code anomalie MQRC_SECURITY_ERROR sont renvoyés à l'application.

AMS for z/OS utilise les services IBM System SSL pour valider les certificats, ce qui inclut la liste de révocation de certificat et la vérification de la confiance.

IBM MQutilise un paramètre de sécurité dans lequel la validation du certificat nécessite que le serveur LDAP soit joignable, mais ne nécessite pas la définition d'une CRL.

Remarque : Il incombe aux administrateurs de s'assurer que les services LDAP appropriés sont disponibles et de gérer les entrées de liste de révocation de certificat pour les autorités de certification concernées.

Configuration de la protection par mot de passe AMS pour les fichiers de configuration

Le stockage des mots de passe de clés et de clés privées sous forme de texte en clair représente un risque pour la sécurité. Par conséquent, Advanced Message Security fournit un outil qui peut brouiller ces mots de passe à l'aide de la clé d'un utilisateur.

Avant de commencer

Le propriétaire du fichier `keystore.conf` doit s'assurer que seul le propriétaire du fichier est autorisé à lire et à écrire dans le fichier. La protection par mot de passe décrite dans cette rubrique n'est qu'une mesure de protection supplémentaire. En outre, vous devez effectuer cette procédure sur un système sécurisé.

Veillez à utiliser la variante **runamscred** appropriée pour le type de client AMS qui va lire le fichier de configuration. Si le client AMS est un:

- Client Java , vous devez utiliser la commande Java **runamscred** , qui se trouve dans <IBM MQ installation root>/java/bin
- Client MQI, vous devez utiliser la commande MQI **runmqascred** qui se trouve dans <IBM MQ installation root>/bin

Procédure

1. Editez les fichiers `keystore.conf` pour inclure toutes les informations requises, y compris les mots de passe qui doivent être protégés.

```
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. Placez la clé de chiffrement pour chiffrer les mots de passe dans un fichier accessible à l'utilisateur qui protège le fichier `keystore.conf` .

   Cette clé doit être la même que celle qui sera utilisée par le client AMS ultérieurement:

```
ThisIsAnExampleEncryptionKey
```

3. Exécutez la commande **runamscred** pour protéger le fichier `keystore.conf` fournissant le fichier de clé de chiffrement.

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. Vérifiez que le fichier `keystore.conf` a été protégé et qu'il contient des mots de passe chiffrés.

Exemple

L'exemple suivant montre à quoi ressemble un fichier `keystore.conf` protégé:

```
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
```

Information associée

runamscred: protéger les mots clés AMS

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 690 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 690, "AMS" indicates " Advanced Message Security".

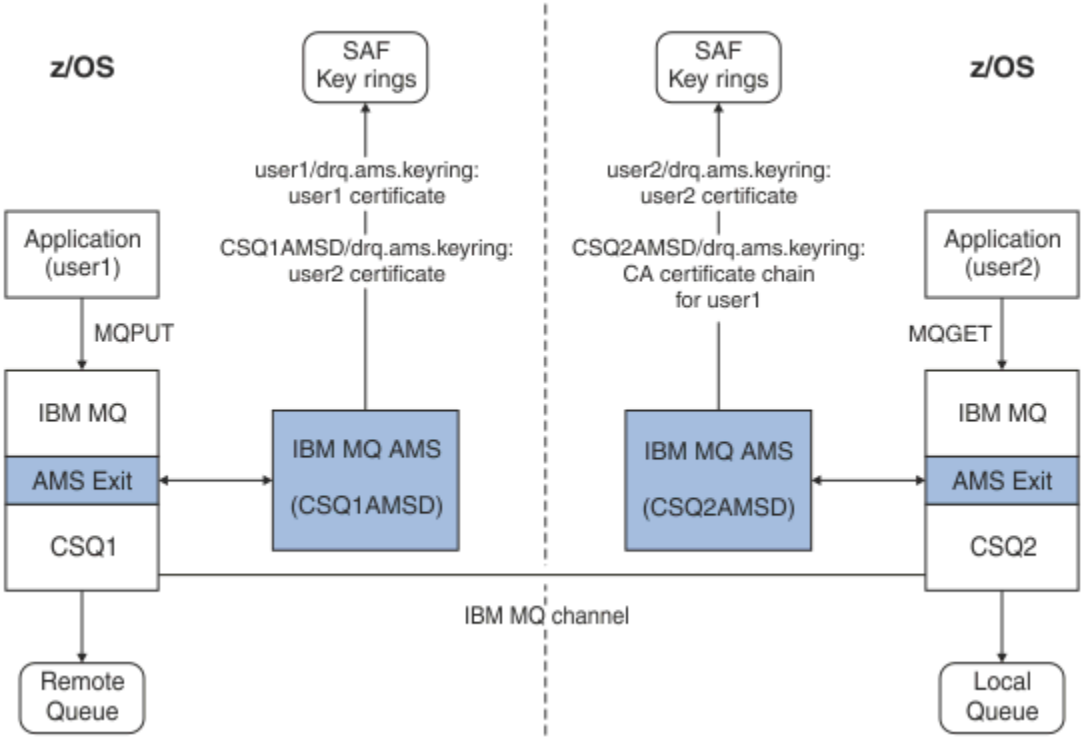


Figure 35. Application and certificate relationships

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.

Administration des règles de sécurité Advanced Message Security

Advanced Message Security utilise des règles de sécurité pour spécifier les algorithmes de chiffrement cryptographique et de signature pour le chiffrement et l'authentification des messages qui transitent par les files d'attente.

Présentation des stratégies de sécurité pour AMS

Les règles de sécurité Advanced Message Security sont des objets conceptuels qui décrivent la façon dont un message est chiffré et signé de manière cryptographique.

Pour plus de détails sur les attributs de stratégie de sécurité, voir les sous-rubriques suivantes:

Concepts associés

«Qualité de protection dans AMS», à la page 696
Les règles de protection des données Advanced Message Security impliquent une qualité de protection (QOP).

«Attributs de stratégie de sécurité dans AMS», à la page 695
Vous pouvez utiliser Advanced Message Security pour sélectionner un algorithme ou une méthode spécifique afin de protéger les données.

Noms de règle dans AMS

Le nom de règle est un nom unique qui identifie une règle Advanced Message Security spécifique et la file d'attente à laquelle elle s'applique.

Le nom de la règle doit être identique au nom de la file d'attente à laquelle elle s'applique. Il existe un mappage un à un entre un Advanced Message Security (AMS) et une file d'attente.

En créant une règle portant le même nom qu'une file d'attente, vous activez la règle pour cette file d'attente. Les files d'attente sans noms de règle correspondants ne sont pas protégées par AMS.

La portée de la règle est pertinente pour le gestionnaire de files d'attente local et ses files d'attente. Les gestionnaires de files d'attente éloignées doivent disposer de leurs propres règles définies en local pour les files d'attente qu'ils gèrent.

Algorithme de signature dans AMS

L'algorithme de signature indique l'algorithme qui doit être utilisé lors de la signature des messages de données.

Les valeurs valides sont les suivantes :

- MD5
- SHA-1
- SHA-2 Famille :
  - SHA256
  - SHA384 (longueur de clé minimale acceptable-768 bits)
  - SHA512 (longueur de clé minimale acceptable-768 bits)

Une règle qui ne spécifie pas d'algorithme de signature, ou qui spécifie un algorithme de NONE, implique que les messages placés dans la file d'attente associée à la règle ne sont pas signés.

Remarque : La qualité de protection utilisée pour les fonctions d'insertion et d'obtention de message doit correspondre. S'il existe une non-concordance de la qualité de protection de la règle entre la file d'attente et le message dans la file d'attente, le message n'est pas accepté et est envoyé à la file d'attente de traitement des erreurs. Cette règle s'applique aux files d'attente locales et éloignées.

Algorithme de chiffrement dans AMS

L'algorithme de chiffrement indique l'algorithme qui doit être utilisé lors du chiffrement des messages de données placés dans la file d'attente associée à la règle.

Les valeurs valides sont les suivantes :

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Une règle qui ne spécifie pas d'algorithme de chiffrement ou qui spécifie un algorithme de NONE implique que les messages placés dans la file d'attente associée à la règle ne sont pas chiffrés.

Notez qu'une règle qui spécifie un algorithme de chiffrement autre que NONE doit également spécifier au moins un nom distinctif de destinataire et un algorithme de signature car les messages chiffrés Advanced Message Security sont également signés.

Important : La qualité de protection utilisée pour les fonctions d'insertion et d'obtention de message doit correspondre. S'il existe une non-concordance de la qualité de protection de la règle entre la file d'attente et le message dans la file d'attente, le message n'est pas accepté et est envoyé à la file d'attente de traitement des erreurs. Cette règle s'applique aux files d'attente locales et éloignées.

Tolérance dans AMS

L'attribut toleration indique si Advanced Message Security peut accepter les messages pour lesquels aucune règle de sécurité n'est spécifiée.

Lors de l'extraction d'un message d'une file d'attente avec une règle de chiffrement des messages, si le message n'est pas chiffré, il est renvoyé à l'application appelante. Les valeurs valides sont les suivantes :

0
Non (**par défaut**).

1
Oui

Une règle qui ne spécifie pas de valeur de tolérance ou qui indique 0 implique que les messages placés dans la file d'attente associée à la règle doivent correspondre aux règles de la règle.

La tolérance est facultative et existe pour faciliter le déploiement de la configuration, où les règles ont été appliquées aux files d'attente, mais ces dernières contiennent déjà des messages pour lesquels aucune règle de sécurité n'est spécifiée.

Noms distinctifs d'expéditeur dans AMS

Les noms distinctifs d'expéditeur identifient les utilisateurs autorisés à placer des messages dans une file d'attente. Un expéditeur utilise son certificat pour signer un message, avant de le placer dans une file d'attente.

Advanced Message Security (AMS) ne vérifie pas si un message a été placé dans une file d'attente protégée par des données par un utilisateur valide jusqu'à ce que le message soit extrait. A ce stade, si la règle stipule un ou plusieurs expéditeurs valides et que l'utilisateur qui a placé le message dans la file d'attente ne figure pas dans la liste des expéditeurs valides, AMS renvoie une erreur à l'application de réception et place le message dans la file d'attente d'erreurs AMS .

Une règle peut avoir zéro ou plusieurs noms distinctifs d'expéditeurs spécifiés. Si aucun nom distinctif d'expéditeur n'est spécifié pour la règle, tout expéditeur peut placer des messages protégés par des données dans la file d'attente à condition que le certificat de l'expéditeur soit digne de confiance. Le certificat d'un expéditeur est digne de confiance en ajoutant le certificat public à un magasin de clés disponible pour l'application de réception.

Les noms distinctifs des expéditeurs se présentent sous la forme suivante :

```
CN=Common Name,O=Organization,C=Country
```

Important :

- Tous les noms de composant de nom distinctif doivent être en majuscules. Tous les identificateurs de nom de composant du nom distinctif doivent être indiqués dans l'ordre indiqué dans le tableau suivant:

Nom de composant	Valeur
CN	Nom usuel de l'objet de ce nom distinctif, tel qu'un nom complet ou la finalité prévue d'un périphérique.
OU	Unité au sein de l'organisation à laquelle l'objet du nom distinctif est affilié, telle qu'une division d'entreprise ou un nom de produit.
O	Organisation à laquelle l'objet du nom distinctif est affilié, telle qu'une société.
L	La localité (ville ou municipalité) où se trouve l'objet du nom distinctif.
ST	Nom de l'état ou de la province où se trouve l'objet du nom distinctif.
C	Pays dans lequel se trouve l'objet du nom distinctif (DN).

- Si un ou plusieurs noms distinctifs d'expéditeur sont spécifiés pour la règle, seuls ces utilisateurs peuvent placer des messages dans la file d'attente associée à la règle.
- Les noms distinctifs d'expéditeur, lorsqu'ils sont spécifiés, doivent correspondre exactement aux noms distinctifs contenus dans le certificat numérique associé à l'utilisateur plaçant le message.
- AMS prend en charge les noms distinctifs dont les valeurs proviennent uniquement du jeu de caractères Latin-1 . Pour créer des noms distinctifs avec des caractères de l'ensemble, vous devez d'abord créer un certificat avec un nom distinctif créé en UTF-8 à l'aide de AIX and Linux avec le codage UTF-8 activé. Vous devez ensuite créer une règle à partir d'une plateforme Linux ou AIX avec le codage UTF-8 activé, ou utiliser le plug-in AMS dans IBM MQ.
- La méthode utilisée par AMS, pour convertir le nom de l'expéditeur du format x.509 au format DN, utilise toujours ST = pour la valeur Etat ou Province.
- Les caractères spéciaux suivants nécessitent des caractères d'échappement:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- Si le nom distinctif contient des blancs imbriqués, vous devez placer le nom distinctif entre guillemets.

Concepts associés

«Noms distinctifs des destinataires dans AMS», à la page 694
Les noms distinctifs des destinataires identifient les utilisateurs qui sont autorisés à extraire des messages d'une file d'attente.

Noms distinctifs des destinataires dans AMS

Les noms distinctifs des destinataires identifient les utilisateurs qui sont autorisés à extraire des messages d'une file d'attente.

Une règle peut avoir zéro ou plusieurs noms distinctifs de destinataires spécifiés. Les noms distinctifs des destinataires se présentent sous la forme suivante:

```
CN=Common Name,O=Organization,C=Country
```

Important :

- Tous les noms de composant de nom distinctif doivent être en majuscules. Tous les identificateurs de nom de composant du nom distinctif doivent être indiqués dans l'ordre indiqué dans le tableau suivant:

Nom de composant	Valeur
CN	Nom usuel de l'objet de ce nom distinctif, tel qu'un nom complet ou la finalité prévue d'un périphérique.
OU	Unité au sein de l'organisation à laquelle l'objet du nom distinctif est affilié, telle qu'une division d'entreprise ou un nom de produit.
O	Organisation à laquelle l'objet du nom distinctif est affilié, telle qu'une société.
L	La localité (ville ou municipalité) où se trouve l'objet du nom distinctif.
ST	Nom de l'état ou de la province où se trouve l'objet du nom distinctif.
C	Pays dans lequel se trouve l'objet du nom distinctif (DN).

- Si aucun nom distinctif de destinataire n'est spécifié pour la règle, tous les utilisateurs peuvent récupérer des messages de la file d'attente associée aux règles.
- Si un ou plusieurs noms distinctifs de destinataire est (sont) spécifié(s) pour la règle, seuls ces utilisateurs peuvent récupérer des messages de la file d'attente associée aux règles.
- Les noms distinctifs de destinataire, lorsqu'ils sont spécifiés, doivent correspondre exactement au nom distinctif contenu dans le certificat numérique associé à l'utilisateur récupérant le message.
- Advanced Message Security prend en charge les noms distinctifs dont les valeurs proviennent uniquement du jeu de caractères Latin-1 . Pour créer des noms distinctifs avec des caractères de l'ensemble, vous devez d'abord créer un certificat avec un nom distinctif créé en UTF-8 à l'aide du codage AIX ou Linux avec le codage UTF-8 activé. Vous devez ensuite créer une règle à partir d'une plateforme Linux ou AIX avec le codage UTF-8 activé ou utiliser le plug-in Advanced Message Security dans IBM MQ.

Concepts associés

«Noms distinctifs d'expéditeur dans AMS», à la page 693
Les noms distinctifs d'expéditeur identifient les utilisateurs autorisés à placer des messages dans une file d'attente. Un expéditeur utilise son certificat pour signer un message, avant de le placer dans une file d'attente.

Attributs de stratégie de sécurité dans AMS

Vous pouvez utiliser Advanced Message Security pour sélectionner un algorithme ou une méthode spécifique afin de protéger les données.

Une règle de sécurité est un objet conceptuel qui décrit la façon dont un message est chiffré et signé de manière cryptographique.

Tableau 109. Attributs de stratégie de sécurité dans AMS	
Attributs	**Description**
Nom de la règle	Nom unique de la règle d'un gestionnaire de files d'attente.
Algorithme de signature	Algorithme de cryptographie utilisé pour signer les messages avant l'envoi.
Algorithme de chiffrement	Algorithme de cryptographie utilisé pour chiffrer les messages avant leur envoi.
Liste des destinataires	Liste des noms distinctifs (DN) de certificats des destinataires potentiels d'un message.
Liste de contrôle Nom distinctif de signature	Liste des noms distinctifs de signature à valider lors de l'extraction de message.

Dans Advanced Message Security, les messages sont chiffrés avec une clé symétrique et la clé symétrique est chiffrée avec les clés publiques des destinataires. Les clés publiques sont chiffrées avec l'algorithme RSA, avec des clés d'une longueur effective jusqu'à 2048 bits. Le chiffrement de la clé asymétrique dépend de la longueur de la clé de certificat.

Les algorithmes de clé symétrique pris en charge sont les suivants:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Advanced Message Security prend également en charge les fonctions de hachage cryptographique suivantes:

- Deprecated MD5
- Deprecated SHA-1
- SHA-2 Famille :
  - SHA256
  - SHA384 (longueur de clé minimale acceptable-768 bits)
  - SHA512 (longueur de clé minimale acceptable-768 bits)

Remarque : La qualité de protection utilisée pour les fonctions d'insertion et d'obtention de message doit correspondre. S'il existe une non-concordance de la qualité de protection de la règle entre la file d'attente et le message dans la file d'attente, le message n'est pas accepté et est envoyé à la file d'attente de traitement des erreurs. Cette règle s'applique aux files d'attente locales et éloignées.

Qualité de protection dans AMS

Les règles de protection des données Advanced Message Security impliquent une qualité de protection (QOP).

Les trois niveaux de qualité de protection dans Advanced Message Security sont complétés par un quatrième niveau dans IBM MQ 9.0 et les versions ultérieures, et dépendent tous des algorithmes de cryptographie utilisés pour signer et chiffrer le message:

- Les messages de confidentialité placés dans la file d'attente doivent être signés et chiffrés.
- Intégrité-Les messages placés dans la file d'attente doivent être signés par l'expéditeur.

- Confidentialité-les messages placés dans la file d'attente doivent être chiffrés. Pour plus d'informations, voir «Qualités de protection disponibles avec AMS», à la page 618
- Aucune-aucune protection des données n'est applicable.

Une règle qui stipule que les messages doivent être signés lorsqu'ils sont placés dans une file d'attente possède un QOP INTEGRITY. Une QOP d'INTEGRITY signifie qu'une règle stipule un algorithme de signature, mais pas un algorithme de chiffrement. Les messages protégés contre l'intégrité sont également appelés "SIGNED".

Une règle qui stipule que les messages doivent être signés et chiffrés lorsqu'ils sont placés dans une file d'attente a un QOP de type PRIVACY. Une QOP de PRIVACY signifie que lorsqu'une règle stipule un algorithme de signature et un algorithme de chiffrement. Les messages protégés par la protection de la vie privée sont également appelés "SCELLÉS".

Une règle qui stipule que les messages doivent être chiffrés lorsqu'ils sont placés dans une file d'attente a une qualité de protection des données (QOP) de type CONFIDENTIALITÉ. Une QOP de CONFIDENTIALITÉ signifie qu'une règle spécifie un algorithme de chiffrement.

Une règle qui ne stipule pas d'algorithme de signature ou de chiffrement a un QOP de NONE. Advanced Message Security ne fournit pas de protection des données pour les files d'attente dont la règle est associée à la valeur NONE.

Gestion des règles de sécurité dans AMS

Une règle de sécurité est un objet conceptuel qui décrit la façon dont un message est chiffré et signé de manière cryptographique.

L'emplacement à partir duquel toutes les tâches d'administration liées aux règles de sécurité sont exécutées dépend de la plateforme que vous utilisez.

- ALW Sous AIX, Linux, and Windows, vous utilisez les commandes DELETE POLICY, DISPLAY POLICYet SET POLICY (ou une commande PCF équivalente) pour gérer vos règles de sécurité.
  - Linux AIX Sous AIX and Linux, les tâches d'administration peuvent être exécutées à partir de *MQ_INSTALLATION_PATH*/bin.
  - Windows Sur les plateformes Windows , les tâches d'administration peuvent être exécutées à partir de n'importe quel emplacement car la variable d'environnement PATH est mise à jour lors de l'installation.
- IBM i Sous IBM i, les commandes DSPMQMSPL, SETMQMSPLet WRKMQMSPL sont installées dans la bibliothèque système QSYS pour la langue principale du système lorsque IBM MQ est installé.

  Des versions de langue nationale supplémentaires sont installées dans les bibliothèques QSYS29xx en fonction du chargement des fonctions de langue. Par exemple, une machine avec l'anglais américain comme langue principale et le coréen comme langue secondaire a les commandes d'anglais américain installées dans QSYS et le chargement de la langue secondaire coréenne dans QSYS2962 comme 2962 est le chargement de la langue pour le coréen.
- z/OS Sous z/OS, les commandes d'administration sont exécutées à l'aide de l'utilitaire de règles de sécurité des messages (CSQ0UTIL). Lorsque des règles sont créées, modifiées ou supprimées dans z/OS, les modifications ne sont pas reconnues par Advanced Message Security tant que le gestionnaire de files d'attente n'est pas arrêté et redémarré ou que la commande z/OS MODIFY n'est pas utilisée pour actualiser la configuration des règles Advanced Message Security . Exemple :

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

Tâches associées

«Création de stratégies de sécurité dans AMS», à la page 698
Les stratégies de sécurité définissent la façon dont un message est protégé lorsque le message est inséré, ou la façon dont un message doit avoir été protégé lorsqu'un message est reçu.

«Modification des règles de sécurité dans AMS», à la page 699
Vous pouvez utiliser Advanced Message Security pour modifier les détails des règles de sécurité que vous avez déjà définies.

«Affichage et vidage des règles de sécurité dans AMS», à la page 699
La commande **dspmqspl** permet d'afficher la liste de toutes les règles de sécurité ou les détails d'une règle nommée en fonction des paramètres de ligne de commande que vous indiquez.

«Suppression de règles de sécurité dans AMS», à la page 701
Pour supprimer des règles de sécurité dans Advanced Message Security, vous devez utiliser la commande `setmqspl` .

FonctionnementAdvanced Message Security

Référence associée

L'utilitaire de règles de sécurité des messages (CSQ0UTIL)

Création de stratégies de sécurité dans AMS

Les stratégies de sécurité définissent la façon dont un message est protégé lorsque le message est inséré, ou la façon dont un message doit avoir été protégé lorsqu'un message est reçu.

Avant de commencer

Certaines conditions d'entrée doivent être remplies lors de la création de règles de sécurité:

- Il doit être en cours d'exécution.
- Le nom d'une règle de sécurité doit respecter les règles de dénomination des objets IBM MQ.
- Vous devez disposer des droits nécessaires pour vous connecter au gestionnaire de files d'attente et créer une règle de sécurité:
  - z/OS Sous z/OS, accordez les droits documentés dans L'utilitaire de règles de sécurité des messages (CSQ0UTIL).
  - Multi Sur Multiplatforms, vous devez accorder les droits `+connect`, `+inq` et `+chg` nécessaires à l'aide de la commande **setmqaut** .

  Pour plus d'informations sur la configuration de la sécurité, voir «Configuration de la sécurité», à la page 139.
- z/OS Sous z/OS, vérifiez que les objets système requis ont été définis conformément aux définitions dans CSQ4INSM.

Exemple

Voici un exemple de création d'une règle sur le gestionnaire de files d'attente QMGR. La règle spécifie que les messages doivent être signés à l'aide de l'algorithme SHA256 et chiffrés à l'aide de l'algorithme AES256 pour les certificats avec le nom distinctif: CN=joe, O=IBM, C=US et DN: CN=jane, O=IBM, C = US. Cette règle est associée à `MY.QUEUE`:

```
setmqspl -m QMGR -p MY.QUEUE -s SHA256 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

Voici un exemple de création de règles sur le gestionnaire de files d'attente QMGR. La règle indique que les messages doivent être chiffrés à l'aide de l'algorithme 3DES pour les certificats avec DN: CN=john, O=IBM, C=US et CN=jeff, O=IBM, C=US et signés avec l'algorithme SHA256 pour les certificats avec DN: 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
```

Remarque :

- La qualité de la protection utilisée pour l'insertion et l'extraction de message doit correspondre. Si la qualité de protection de la règle définie pour le message est plus faible que celle définie pour une file d'attente, le message est envoyé à la file d'attente de traitement des erreurs. Cette règle est valide pour les files d'attente locales et éloignées.

Référence associée

Liste complète des attributs de la commande setmqspl

Modification des règles de sécurité dans AMS

Vous pouvez utiliser Advanced Message Security pour modifier les détails des règles de sécurité que vous avez déjà définies.

Avant de commencer

- Le gestionnaire de files d'attente sur lequel vous souhaitez travailler doit être en cours d'exécution.
- Vous devez disposer des droits nécessaires pour vous connecter au gestionnaire de files d'attente et créer une règle de sécurité.
  - z/OS Sous z/OS, accordez les droits documentés dans L'utilitaire de règles de sécurité des messages (CSQ0UTIL).
  - Multi Sur Multiplatforms, vous devez accorder les droits `+connect`, `+inq` et `+chg` nécessaires à l'aide de la commande **`setmqaut`** .

  Pour plus d'informations sur la configuration de la sécurité, voir «Configuration de la sécurité», à la page 139.

Pourquoi et quand exécuter cette tâche

Pour modifier les règles de sécurité, appliquez la commande `setmqspl` à une règle existante fournissant de nouveaux attributs.

Exemple

Voici un exemple de création d'une règle nommée MYQUEUE sur un gestionnaire de files d'attente nommé QMGR, spécifiant que les messages doivent être chiffrés à l'aide de l'algorithme 3DES pour les auteurs (**-a**) ayant des certificats avec le nom distinctif CN=alice, O=IBM, C=US et signés avec l'algorithme SHA256 pour les destinataires (**-r**) ayant des certificats avec le nom distinctif CN=jeff, O=IBM, C = US.

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Pour modifier cette règle, exécutez la commande `setmqspl` avec tous les attributs de l'exemple en modifiant uniquement les valeurs que vous souhaitez modifier. Dans cet exemple, la règle créée précédemment est associée à une nouvelle file d'attente et son algorithme de chiffrement est remplacé par AES256:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Référence associée

setmqspl (définition de la règle de sécurité)

Affichage et vidage des règles de sécurité dans AMS

La commande **dspmqspl** permet d'afficher la liste de toutes les règles de sécurité ou les détails d'une règle nommée en fonction des paramètres de ligne de commande que vous indiquez.

Avant de commencer

- Pour afficher les détails des règles de sécurité, le gestionnaire de files d'attente doit exister et être en cours d'exécution.

- Vous devez disposer des droits nécessaires pour vous connecter au gestionnaire de files d'attente et créer une règle de sécurité.
  - z/OS Sous z/OS, accordez les droits documentés dans L'utilitaire de règles de sécurité des messages (CSQ0UTIL).
  - Multi Sur Multiplatforms, vous devez accorder les droits `+connect`, `+inq` et `+chg` nécessaires à l'aide de la commande **setmqaut** .

  Pour plus d'informations sur la configuration de la sécurité, voir «Configuration de la sécurité», à la page 139.

Pourquoi et quand exécuter cette tâche

Voici la liste des indicateurs de commande **dspmqspl** :

*Tableau 110. Indicateurs de commande **dspmqspl** .*

Indicateur de commande	Explication
-m	Nom du gestionnaire de files d'attente (obligatoire).
-p	Nom de la politique.
-export	L'ajout de cet indicateur génère une sortie qui peut facilement être appliquée à un autre gestionnaire de files d'attente.

Exemple

L'exemple suivant montre comment créer deux règles de sécurité pour `venus.queue.manager`:

```
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
```

Cet exemple illustre une commande qui affiche les détails de toutes les règles définies pour `venus.queue.manager` et la sortie qu'elle génère:

```
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
```

Cet exemple illustre une commande qui affiche les détails d'une règle de sécurité sélectionnée définie pour `venus.queue.manager` et la sortie qu'elle génère:

```
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
```

Dans l'exemple suivant, nous créons d'abord une règle de sécurité, puis nous exportons la règle à l'aide de l'indicateur **-export** :

```
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 Sous z/OS, les informations de règle exportées sont écrites par CSQ0UTIL dans la définition de données EXPORT.

Multi Sur Multiplatforms, redirigez la sortie vers un fichier, par exemple:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

Pour importer une règle de sécurité:

- Linux AIX Sous AIX and Linux :
  1. Connectez-vous en tant qu'utilisateur appartenant au groupe d'administration `mqm` IBM MQ .
  2. Exécutez `. policies.sh`.
- Windows Sous Windows, exécutez `policies.bat`.
- z/OS Sous z/OS , utilisez l'utilitaire CSQ0UTIL , en indiquant à SYSIN le fichier contenant les informations de règle exportées.

Référence associée

Liste complète des attributs de la commande dspmqspl

Suppression de règles de sécurité dans AMS

Pour supprimer des règles de sécurité dans Advanced Message Security, vous devez utiliser la commande `setmqspl` .

Avant de commencer

Certaines conditions d'entrée doivent être remplies lors de la gestion des règles de sécurité:

- Il doit être en cours d'exécution.
- Vous devez disposer des droits nécessaires pour vous connecter au gestionnaire de files d'attente et créer une règle de sécurité.
  - z/OS Sous z/OS, accordez les droits documentés dans L'utilitaire de règles de sécurité des messages (CSQ0UTIL).
  - Multi Sur Multiplatforms, vous devez accorder les droits `+connect`, `+inq` et `+chg` nécessaires à l'aide de la commande **setmqaut** .

  Pour plus d'informations sur la configuration de la sécurité, voir «Configuration de la sécurité», à la page 139.

Pourquoi et quand exécuter cette tâche

Utilisez la commande **setmqspl** avec l'option **-remove** .

Exemple

Voici un exemple de suppression d'une règle:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

Référence associée

Liste complète des attributs de la commande setmqspl

Protection des files d'attente système dans AMS

Les files d'attente système permettent la communication entre IBM MQ et ses applications auxiliaires. Chaque fois qu'un gestionnaire de files d'attente est créé, une file d'attente système est également créée pour stocker les messages et les données internes IBM MQ . Vous pouvez protéger les files d'attente système avec Advanced Message Security afin que seuls les utilisateurs autorisés puissent y accéder ou les déchiffrer.

La protection des files d'attente système suit le même modèle que la protection des files d'attente standard. Voir «Création de stratégies de sécurité dans AMS», à la page 698.

Windows Pour utiliser la protection de file d'attente système sous Windows, copiez le fichier `keystore.conf` dans le répertoire suivant:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS Sous z/OS, pour protéger `SYSTEM.ADMIN.COMMAND.QUEUE`, le serveur de commandes doit avoir accès à `keystore` et à `keystore.conf`, qui contiennent des clés et une configuration permettant au serveur de commandes d'accéder aux clés et aux certificats. Toutes les modifications apportées à la règle de sécurité de `SYSTEM.ADMIN.COMMAND.QUEUE` nécessitent le redémarrage du serveur de commandes.

Tous les messages envoyés et reçus à partir de la file d'attente de commandes sont signés ou signés et chiffrés en fonction des paramètres de règle. Si un administrateur définit des signataires autorisés, les messages de commande qui ne passent pas la vérification du nom distinctif (DN) du signataire ne sont pas exécutés par le serveur de commandes et ne sont pas acheminés vers la file d'attente de traitement d'erreurs Advanced Message Security . Les messages envoyés en tant que réponses à des files d'attente dynamiques temporaires IBM MQ Explorer ne sont pas protégés par AMS.

Les règles de sécurité n'ont pas d'effet sur les files d'attente SYSTEM suivantes:

- 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 Files d'attente de flux et AMS

Il est possible de diffuser en flux des messages protégés Advanced Message Security (AMS) en double.

Si une règle AMS est définie pour une file d'attente qui entraîne la signature et / ou le chiffrement des messages insérés dans cette file d'attente, vous pouvez également configurer l'attribut **STREAMQ** de la file d'attente pour placer une copie de chaque message protégé dans une seconde file d'attente. Le message en double diffusé est signé et / ou chiffré à l'aide de la même règle que celle qui a été configurée pour la file d'attente d'origine.

Dans l'exemple suivant, vous configurez deux files d'attente, QUEUE1 et QUEUE2. QUEUE1 a son attribut **STREAMQ** configuré pour insérer des messages en continu dans QUEUE2:

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

Les messages protégés AMS sont placés dans QUEUE1 par un utilisateur avec le certificat `CN=bob,O=IBM,C=GB`.

Une application avec le certificat `CN=alice,O=IBM,C=GB` va consommer les messages de QUEUE1. Une application distincte avec le certificat `CN=fred,O=IBM,C=GB` va consommer les messages de QUEUE2.

La règle de confidentialité AMS suivante est appliquée à QUEUE1 :

```
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)
```

Si un algorithme de chiffrement a été configuré dans la règle pour QUEUE1, les destinataires répertoriés dans la règle doivent inclure à la fois les destinataires des messages d'origine provenant de QUEUE1et les destinataires qui vont consommer des messages en double provenant de QUEUE2.

Lorsque l'application tente de consommer des messages à partir de QUEUE2 , elle effectue des vérifications d'intégrité et/ou déchiffre le message en fonction de la règle qui a été définie sur QUEUE2. Si une application souhaite consommer des messages diffusés en continu à partir de QUEUE2, vous devez définir une règle appropriée sur QUEUE2 qui permet de vérifier l'intégrité des messages et de les déchiffrer correctement.

En particulier, l'algorithme de signature, le signataire et l'algorithme de chiffrement doivent être identiques à la règle appliquée à QUEUE1. Les destinataires de la règle QUEUE2 doivent inclure l'identité du destinataire qui consomme le message de QUEUE2.

Remarque : Il n'est pas nécessaire que la règle appliquée à QUEUE2 répertorie tous les destinataires nommés dans le jeu de règles sur QUEUE1.

Par exemple, la règle suivante peut être définie sur QUEUE2 pour permettre à une application avec le nom distinctif de certificat `CN=fred,O=IBM,C=GB` de lire les messages protégés par AMSà partir de celle-ci:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

Concepts associés

Files d'attente de flux

Octroi de droits OAM dans AMS

Les droits d'accès aux fichiers autorisent tous les utilisateurs à exécuter les commandes `setmqspl` et `dspmqspl` . Toutefois, Advanced Message Security s'appuie sur le gestionnaire des droits d'accès aux objets (OAM) et toute tentative d'exécution de ces commandes par un utilisateur qui n'appartient pas au groupe mqm, qui est le groupe d'administration IBM MQ , ou qui ne dispose pas des droits permettant de lire les paramètres de règles de sécurité accordés, génère une erreur.

Procédure

Pour accorder les droits nécessaires à un utilisateur, exécutez:

```
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
```

Remarque : Vous n'avez besoin de définir ces droits OAM que si vous prévoyez de connecter des clients au gestionnaire de files d'attente à l'aide de Advanced Message Security 7.0.1.

Avertissement : Parcourez les droits d'accès à SYSTEM.PROTECTION.POLICY.QUEUE n'est pas obligatoire dans toutes les situations. IBM MQ optimise les performances en mettant en cache les règles de sorte que vous n'ayez pas à parcourir les enregistrements pour obtenir des détails sur les règles dans SYSTEM.PROTECTION.POLICY.QUEUE dans tous les cas.

IBM MQ ne met pas en cache toutes les règles disponibles. S'il existe un nombre élevé de règles, IBM MQ met en cache un nombre limité de règles. Par conséquent, si le nombre de règles définies pour le gestionnaire de files d'attente est faible, il n'est pas nécessaire de fournir l'option de navigation à SYSTEM.PROTECTION.POLICY.QUEUE.

Toutefois, vous devez accorder le droit de navigation à cette file d'attente, au cas où un nombre élevé de règles serait défini, ou si vous utilisez d'anciens clients. SYSTEM.PROTECTION.ERROR.QUEUE est utilisé pour placer les messages d'erreur générés par le code AMS. Le droit d'insertion sur cette file d'attente est vérifié uniquement lorsque vous tentez d'insérer un message d'erreur dans la file d'attente. Votre droit d'insertion sur la file d'attente n'est pas vérifié lorsque vous tentez d'insérer ou d'extraire un message d'une file d'attente protégée AMS.

Octroi de droits de sécurité dans AMS

Lors de l'utilisation de la sécurité des ressources de commande, vous devez définir des droits pour permettre à Advanced Message Security de fonctionner. Cette rubrique utilise les commandes RACF dans les exemples. Si votre entreprise utilise un gestionnaire de sécurité externe (ESM) différent, vous devez utiliser les commandes équivalentes pour ce gestionnaire.

L'octroi de droits de sécurité comporte trois aspects:

- «Espace adresse AMSM», à la page 705
- «CSQ0UTIL», à la page 706
- «Utilisation de files d'attente pour lesquelles une règle Advanced Message Security est définie», à la page 706

Remarques : Les exemples de commande utilisent les variables suivantes.

1. *QMgrName* -nom du gestionnaire de files d'attente.

   z/OS Sous z/OS, cette valeur peut également être le nom d'un groupe de partage de files d'attente.
2. *username* -il peut s'agir d'un nom de groupe.
3. Les exemples montrent la classe MQQUEUE. z/OS Il peut également s'agir de MXQUEUE, GMQQUEUE ou GMXQUEUE. Pour plus d'informations, voir «Profiles for queue security», à la page 210.

De plus, si le profil existe déjà, vous n'avez pas besoin de la commande RDEFINE.

Espace adresse AMSM

Vous devez fournir une sécurité IBM MQ au nom d'utilisateur sous lequel l'espace adresse Advanced Message Security s'exécute.

- Pour la connexion par lots au gestionnaire de files d'attente, émettez

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
           PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Permet d'accéder à SYSTEM.PROTECTION.POLICY.QUEUE, problème:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

L'utilitaire qui permet aux utilisateurs d'exécuter les commandes **setmqspl** et **dspmqspl** requiert les droits suivants, où le nom d'utilisateur correspond à l'ID utilisateur du travail:

- Pour la connexion par lots au gestionnaire de files d'attente, émettez:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Permet d'accéder à SYSTEM.PROTECTION.POLICY.QUEUE, requis pour la commande **setmqpol** , exécutez:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
             PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- Permet d'accéder à SYSTEM.PROTECTION.POLICY.QUEUE, requis pour la commande **dspmqpol** , exécutez:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
               PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

Utilisation de files d'attente pour lesquelles une règle Advanced Message Security est définie

Lorsqu'une application utilise des files d'attente pour lesquelles une règle est définie, elle requiert des droits supplémentaires pour permettre à Advanced Message Security de protéger les messages.

L'application requiert:

- Accès en lecture à SYSTEM.PROTECTION.POLICY.QUEUE. Pour ce faire, exécutez la commande suivante:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- Placez l'accès à SYSTEM.PROTECTION.ERROR.QUEUE. Pour ce faire, exécutez la commande suivante:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i Configuration des certificats et du fichier de configuration du magasin de clés pour AMS sous IBM i

Votre première tâche lors de la configuration de la protection Advanced Message Security consiste à créer un certificat et à l'associer à votre environnement. L'association est configurée via un fichier stocké dans le système de fichiers intégré (IFS).

Procédure

1. Pour créer un certificat autosigné à l'aide des outils OpenSSL fournis avec IBM i, exécutez la commande suivante à partir de QShell:

```
/QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
$HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
```

La commande vous invite à indiquer divers attributs de nom distinctif pour un nouveau certificat autosigné, notamment:

- Nom usuel (CN =)
- Organisation (O =)
- Pays (C =)

Cela crée une clé privée non chiffrée et un certificat correspondant, au format PEM (Privacy Enhanced Mail).

Par souci de simplicité, entrez simplement des valeurs pour le nom usuel, l'organisation et le pays. Ces attributs et valeurs sont importants lors de la création d'une règle.

Des invites et des attributs supplémentaires peuvent être personnalisés en spécifiant un fichier de configuration openssl personnalisé sur la ligne de commande avec le paramètre **-config** . Pour plus de détails sur la syntaxe du fichier de configuration, voir la documentation OpenSSL .

Par exemple, la commande suivante ajoute des extensions de certificat X.509 v3 supplémentaires:

```
/QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
-keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
```

où myconfig.cnf est un fichier de flux ASCII qui contient les éléments suivants:

```
[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 requiert que le certificat et la clé privée soient conservés dans le même fichier. Exécutez la commande suivante pour ce faire:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

Le fichier `private.pem` dans $HOME contient désormais une clé privée et un certificat correspondants, tandis que le fichier `mycert.pem` contient tous les certificats publics pour lesquels vous pouvez chiffrer les messages et valider les signatures.

Les deux fichiers doivent être associés à votre environnement en créant un fichier de configuration de magasin de clés, `keystore.conf`, dans votre emplacement par défaut.

Par défaut, AMS recherche la configuration du magasin de clés dans un sous-répertoire `.mqs` de votre répertoire de base.

3. Dans QShell, créez le fichier `keystore.conf` :

```
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 *Création d'une règle pour AMS sur IBM i*

Avant de créer une règle, vous devez créer une file d'attente pour stocker les messages protégés.

Procédure

1. A l'invite de ligne de commande, entrez ;

   ```
   CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
   ```

   où mqmname est le nom de votre gestionnaire de files d'attente.

   Utilisez la commande DSPMQM pour vérifier que le gestionnaire de files d'attente est capable d'utiliser des règles de sécurité. Vérifiez que **Security Policy Capability** affiche **YES*.

   La règle la plus simple que vous pouvez définir est une règle d'intégrité, qui est obtenue en créant une règle avec un algorithme de signature numérique mais pas d'algorithme de chiffrement.

   Les messages sont signés mais non chiffrés. Si les messages doivent être chiffrés, vous devez spécifier un algorithme de chiffrement et un ou plusieurs destinataires de message prévus.

   Un certificat dans le magasin de clés public d'un destinataire de message est identifié par un nom distinctif.
2. Affichez les noms distinctifs des certificats dans le magasin de clés public, mycert.pem dans $HOME, à l'aide de la commande suivante dans QShell:

   ```
   /QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
   ```

   Vous devez entrer le nom distinctif comme destinataire prévu et le nom de la règle doit correspondre au nom de la file d'attente à protéger.
3. Dans une invite de commande CL, entrez, par exemple:

   ```
   SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
   O=.., C=..')
   ```

   où mqmname est le nom de votre gestionnaire de files d'attente.

   Une fois la règle créée, tous les messages insérés, consultés ou supprimés de façon destructive via ce nom de file d'attente sont soumis à la règle AMS .

Référence associée

Afficher le gestionnaire de files d'attente de messages (DSPMQM)

Définir une règle de sécurité MQM (SETMQMSPL)

IBM i *Test d'une règle pour AMS on IBM i*

Utilisez les exemples d'application fournis avec le produit pour tester vos stratégies de sécurité.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser les exemples d'application fournis avec IBM MQ , tels que AMQSPUT4, AMQSGET4, AMQSGBR4et des outils tels que WRKMQMMSG pour insérer, parcourir et extraire des messages à l'aide du nom de file d'attente PROTECTED.

Si tout a été configuré correctement, le comportement de l'application ne doit pas être différent de celui d'une file d'attente non protégée pour cet utilisateur.

Un utilisateur non configuré pour Advanced Message Securityou un utilisateur qui ne dispose pas de la clé privée requise pour déchiffrer le message ne pourra toutefois pas afficher le message. L'utilisateur reçoit le code achèvement RCFAIL, équivalent à MQCC_FAILED (2) et le code anomalie RC2063 (MQRC_SECURITY_ERROR).

Pour vérifier que la protection AMS est activée, placez des messages de test dans la file d'attente PROTECTED, par exemple à l'aide de AMQSPUT0. Vous pouvez ensuite créer une file d'attente alias pour parcourir les données protégées brutes alors qu'elles sont au repos.

Procédure

Pour accorder les droits nécessaires à un utilisateur, exécutez:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

L'exploration à l'aide du nom de file d'attente ALIAS, par exemple à l'aide de AMQSBCG4 ou WRKMQMMSG, doit révéler des messages `scrambled` plus volumineux dans lesquels une exploration de la file d'attente PROTECTED affiche des messages en clair.

Les messages brouillés sont visibles, mais le texte en clair d'origine n'est pas déchiffrable à l'aide de la file d'attente ALIAS, car il n'existe pas de règle permettant à AMS d'imposer la mise en correspondance de ce nom. Par conséquent, les données protégées brutes sont renvoyées.

Référence associée

Définir une règle de sécurité MQM (SETMQMSPL)

Gestion des messages MQ (WRKMQMMSG)

Evénements de commande et de configuration pour AMS

Avec Advanced Message Security, vous pouvez générer des messages d'événement de commande et de configuration, qui peuvent être consignés et servir d'enregistrement des changements de règles à des fins d'audit.

Les événements de commande et de configuration générés par IBM MQ sont des messages au format PCF envoyés aux files d'attente dédiées sur le gestionnaire de files d'attente où l'événement se produit.

Les messages d'événements de configuration sont envoyés à SYSTEM.ADMIN.CONFIG.EVENT EVENT.

Les messages d'événements de commande sont envoyés à SYSTEM.ADMIN.COMMAND.EVENT EVENT.

Les événements sont générés indépendamment des outils que vous utilisez pour gérer les règles de sécurité Advanced Message Security .

Dans Advanced Message Security, il existe quatre types d'événements générés par différentes actions sur les règles de sécurité:

- «Création de stratégies de sécurité dans AMS», à la page 698, qui génère deux messages d'événement IBM MQ :
  - Un événement de configuration
  - Un événement de commande
- «Modification des règles de sécurité dans AMS», à la page 699, qui génère trois messages d'événement IBM MQ :
  - Evénement de configuration contenant d'anciennes valeurs de règles de sécurité
  - Evénement de configuration contenant de nouvelles valeurs de règle de sécurité
  - Un événement de commande
- «Affichage et vidage des règles de sécurité dans AMS», à la page 699, qui génère un message d'événement IBM MQ :
  - Un événement de commande

- «Suppression de règles de sécurité dans AMS», à la page 701, qui génère deux messages d'événement IBM MQ :
  - Un événement de configuration
  - Un événement de commande

Activation et désactivation de la journalisation des événements pour AMS

Vous pouvez contrôler les événements de commande et de configuration à l'aide des attributs de gestionnaire de files d'attente **CONFIGEV** et **CMDEV**. Pour activer ces événements, définissez l'attribut de gestionnaire de files d'attente approprié sur ENABLED. Pour désactiver ces événements, définissez l'attribut de gestionnaire de files d'attente approprié sur DISABLED.

Procédure

Evénements de configuration

Pour activer les événements de configuration, définissez **CONFIGEV** sur ENABLED. Pour désactiver les événements de configuration, définissez **CONFIGEV** sur DISABLED. Par exemple, vous pouvez activer des événements de configuration à l'aide de la commande MQSC suivante:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Evénements Commande

Pour activer les événements de commande, définissez **CMDEV** sur ENABLED. Pour activer les événements de commande pour les commandes à l'exception des commandes **DISPLAY MQSC** et des commandes Inquire PCF, définissez **CMDEV** sur NODISPLAY. Pour désactiver les événements de commande, définissez **CMDEV** sur DISABLED. Par exemple, vous pouvez activer des événements de commande à l'aide de la commande MQSC suivante:

```
ALTER QMGR CMDEV (ENABLED)
```

Tâches associées

Contrôle des événements de configuration, de commande et de consignateur dans IBM MQ

Format de message d'événement de commande pour AMS

Le message d'événement de commande se compose de la structure MQCFH et des paramètres PCF qui la suivent.

Voici les valeurs MQCFH sélectionnées:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

Remarque : La valeur de ParameterCount est deux car il existe toujours deux paramètres de type MQCFGR (groupe). Chaque groupe est constitué de paramètres appropriés. Les données d'événement se composent de deux groupes, CommandContext et CommandData.

CommandContext contient:

EventUserId

Description : ID utilisateur qui a émis la commande ou l'appel qui a généré l'événement. (Il s'agit du même ID utilisateur que celui utilisé pour vérifier les droits d'émission de la commande ou de l'appel ; pour les commandes reçues d'une file d'attente, il s'agit également de l'ID utilisateur (UserIdentifier) du MD du message de commande).

Identificateur : MQCACF_EVENT_USER_ID.

Type de données : MQCFST.

Longueur maximale : MQ_USER_ID_LENGTH.

Renvoyé: Toujours.

EventOrigin

Description : Origine de l'action à l'origine de l'événement.

Identificateur : MQIACF_EVENT_ORIGIN.

Type de données : MQCFIN.

Valeurs :

MQEVO_CONSOLE
Ligne de commande de la console.

MQEVO_MSG
Message de commande du plug-in IBM MQ Explorer .

Renvoyé: Toujours.

EventQMgr

Description : Gestionnaire de files d'attente dans lequel la commande ou l'appel a été entré. (Le gestionnaire de files d'attente dans lequel la commande est exécutée et qui génère l'événement se trouve dans le descripteur du message d'événement).

Identificateur : MQCACF_EVENT_Q_MGR.

Type de données : MQCFST.

Longueur maximale : MQ_Q_MGR_NAME_LENGTH.

Renvoyé: Toujours.

EventAccountingToken

Description : Pour les commandes reçues sous forme de message (MQEVO_MSG), jeton de comptabilité (AccountingToken) provenant du descripteur de message de commande.

Identificateur : MQBACF_EVENT_ACCOUNTING_TOKEN.

Type de données : MQCFBS.

Longueur maximale : MQ_ACCOUNTING_TOKEN_LENGTH.

Renvoyé: Uniquement si EventOrigin est MQEVO_MSG.

Données EventIdentity

Description : Pour les commandes reçues sous forme de message (MQEVO_MSG), données d'identité d'application (donnéesApplIdentity) provenant du descripteur de message de commande.

Identificateur : MQCACF_EVENT_APPL_IDENTITY.

Type de données : MQCFST.

Longueur maximale : MQ_APPL_IDENTITY_DATA_LENGTH.

Renvoyé: Uniquement si EventOrigin est MQEVO_MSG.

EventApplType

Description : Pour les commandes reçues sous forme de message (MQEVO_MSG), type d'application (PutApplType) à partir du descripteur de message du message de commande.

Identificateur : MQIACF_EVENT_APPL_TYPE.

Type de données : MQCFIN.

Renvoyé: Uniquement si EventOrigin est MQEVO_MSG.

EventApplName

Description : Pour les commandes reçues sous forme de message (MQEVO_MSG), nom de l'application (nomPutAppl) à partir du descripteur de message du message de commande.

Identificateur : MQCACF_EVENT_APPL_NAME.

Type de données : MQCFST.

Longueur maximale : MQ_APPL_NAME_LENGTH.

Renvoyé: Uniquement si EventOrigin est MQEVO_MSG.

EventApplOrigin

Description : Pour les commandes reçues sous forme de message (MQEVO_MSG), les données d'origine de l'application (donnéesApplOrigin) provenant du descripteur de message de commande.

Identificateur : MQCACF_EVENT_APPL_ORIGIN.

Type de données : MQCFST.

Longueur maximale : MQ_APPL_ORIGIN_DATA_LENGTH.

Renvoyé: Uniquement si EventOrigin est MQEVO_MSG.

Commande

Description : Code de la commande.

Identificateur : MQIACF_COMMAND.

Type de données : MQCFIN.

Valeurs : **Valeur numérique MQCMD_INQUIRE_PROT_POLICY 205**
Valeur numérique 206 de MQCMD_CREATE_PROT_POLICY
Valeur numérique 207 de MQCMD_DELETE_PROT_POLICY
Valeur numérique 208 de MQCMD_CHANGE_PROT_POLICY
Ils sont définis dans IBM MQ 8.0 `cmqcfc.h`

Renvoyé: Toujours.

CommandData contient des éléments PCF qui composent la commande PCF.

Format de message d'événement de configuration pour AMS

Les événements de configuration sont des messages PCF au format Advanced Message Security standard.

Les valeurs possibles pour le descripteur de message MQMD se trouvent dans Message d'événement MQMD (descripteur de message).

Voici les valeurs MQMD sélectionnées:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

La mémoire tampon de messages se compose de la structure MQCFH et de la structure de paramètres qui la suit. Les valeurs MQCFH possibles se trouvent dans Message d'événement MQCFH (en-tête PCF).

Voici les valeurs MQCFH sélectionnées:

```
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}
```

Les paramètres suivants de MQCFH sont:

EventUserID

Description : ID utilisateur qui a émis la commande ou l'appel qui a généré l'événement. (Il s'agit du même ID utilisateur que celui utilisé pour vérifier les droits d'émission de la commande ou de l'appel ; pour les commandes reçues d'une file d'attente, il s'agit également de l'ID utilisateur (UserIdentifier) du MD du message de commande).

Identificateur : **MQCACF_EVENT_USER_ID**

Type de données : MQCFST.

Longueur maximale : MQ_USER_ID_LENGTH.

Renvoyé: Toujours.

SecurityId

Description : Valeur de MQMD.AccountingToken dans le cas d'un message du serveur de commandes ou Windows SID pour la commande locale.

Identificateur : **ID_SÉCURITÉ_ÉVÉNEMENT_MQBACF**

Type de données : MQCBS.

Longueur maximale : MQ_SECURITY_ID_LENGTH.

Renvoyé: Toujours.

EventOrigin

Description : Origine de l'action à l'origine de l'événement.

Identificateur : **MQIACF_EVENT_ORIGIN**

Type de données : MQCFIN.

Valeurs : **MQEVO_CONSOLE**
Ligne de commande de la console.

MSG MQEVO_MQ
Message de commande du plug-in IBM MQ Explorer.

Renvoyé: Toujours.

EventQMgr

Description : Gestionnaire de files d'attente dans lequel la commande ou l'appel a été entré. (Le gestionnaire de files d'attente dans lequel la commande est exécutée et qui génère l'événement se trouve dans le descripteur du message d'événement).

Identificateur : **MQCACF_EVENT_Q_MGR**

Type de données : MQCFST

Longueur maximale : MQ_Q_MGR_NAME_LENGTH

Renvoyé: Toujours.

ObjectType

Description : Type d'objet.

Identificateur : **MQIACF_OBJECT_TYPE**

Type de données : MQCFIN

Valeur : **POLITIQUE MQOT_PROT_DE**
Règle de protection Advanced Message Security . **1019** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Renvoyé: Toujours.

PolicyName

Description : Nom de la règle Advanced Message Security .

Identificateur : **MQCA_POLICY_NAME**.

Type de données : MQCFST.

Valeur : **2112** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Longueur maximale : MQ_OBJECT_NAME_LENGTH.

Renvoyé: Toujours.

PolicyVersion

Description : Version de la règle Advanced Message Security .

Identificateur : **MQIA_POLICY_VERSION**

Type de données : MQCFIN

Valeur **238** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Renvoyé: Toujours

TolerateFlag

Description : Indicateur de tolérance de la règle Advanced Message Security .

Identificateur : **MQIA_TOLERATE_NON protégé**

Type de données : MQCFIN

Valeur **235** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Renvoyé: Toujours.

SignatureAlgorithm

Description : Algorithme de signature de règle Advanced Message Security .

Identificateur : **Algorithme de signature (MQIA_SIGNATURE_ALGORITHM)**

Type de données : MQCFIN

Valeur : **236** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Renvoyé: Chaque fois qu'un algorithme de signature est défini dans la règle Advanced Message Security

EncryptionAlgorithm

Description : Algorithme de chiffrement de la règle Advanced Message Security .

Identificateur : **Algorithme de chiffrement MQIA_ENCRYPTION_ALGORITHM**

Type de données : MQCFIN

Valeur : **237** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Renvoyé: Chaque fois qu'un algorithme de chiffrement est défini dans la règle IBM MQ

SignerDNs

Description : Sujet DistinguishedName des signataires autorisés.

Identificateur : **MQCA_SIGNER_DN**

Type de données : MQCFSL

Valeur : **2113** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Longueur maximale : Nom distinctif de signataire le plus long dans la règle, mais pas plus long que MQ_DISTINGUISHED_NAME_LENGTH

Renvoyé: Chaque fois qu'il est défini dans la règle IBM MQ .

RecipientDNs

Description : Sujet DistinguishedName des signataires autorisés.

Identificateur : **MQCA_RECIPIENT_DN**

Type de données : MQCFSL

Valeur : **2114** -Valeur numérique définie dans IBM MQ 8.0 ou dans le fichier `cmqc.h` .

Longueur maximale : Nom distinctif du destinataire le plus long dans la règle, mais pas MQ_DISTINGUISHED_NAME_LENGTH.

Renvoyé: Chaque fois qu'il est défini dans la règle IBM MQ .

Remarques

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Consultez votre représentant IBM local pour obtenir des informations sur les produits et services actuellement disponibles dans votre région. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout produit, programme ou service fonctionnellement équivalent qui ne porte pas atteinte à un droit de propriété intellectuelle IBM peut être utilisé à la place. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour obtenir des informations sur les licences relatives aux informations sur deux octets (DBCS), contactez le service de la propriété intellectuelle IBM de votre pays ou envoyez vos demandes de renseignements, par écrit, à :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Le paragraphe suivant ne s’applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et/ou programmes décrits dans ce document.

Les références à des sites Web non IBM sont fournies uniquement à titre d'information et n'impliquent en aucune façon une adhésion de ces sites Web. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
Coordinateur d'interopérabilité logicielle, département 49XA
3605 Autoroute 52 N

Rochester, MN 55901
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans le présent document et tous les éléments sous disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Contrat sur les produits et services IBM, aux Conditions Internationales d'Utilisation de Logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Licence sur les droits d'auteur :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Documentation sur l'interface de programmation

Les informations d'interface de programmation, si elles sont fournies, sont destinées à vous aider à créer un logiciel d'application à utiliser avec ce programme.

Ce manuel contient des informations sur les interfaces de programmation prévues qui permettent au client d'écrire des programmes pour obtenir les services d' IBM MQ.

Toutefois, lesdites informations peuvent également contenir des données de diagnostic, de modification et d'optimisation. Ces données vous permettent de déboguer votre application.

Important : N'utilisez pas ces informations de diagnostic, de modification et d'optimisation en tant qu'interface de programmation car elles sont susceptibles d'être modifiées.

Marques

IBM, le logo IBM , ibm.com, sont des marques d' IBM Corporation dans de nombreux pays. La liste actualisée de toutes les marques d' IBM est disponible sur la page Web "Copyright and trademark

information"www.ibm.com/legal/copytrade.shtml. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.

Microsoft et Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Ce produit inclut des logiciels développés par le projet Eclipse (https://www.eclipse.org/).

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.

Référence :

(1P) P/N: